NKAP BOLTING SYSTEM

TECHNICAL SPECIFICATION

GROUP 01

Team member	Student Number	Email	Contact Number
Jerome Amenigy (Team Leader)	u19131918	u19131918@tuks.co.za	0617993245
Divya Bagratee (Project Manager)	u19078065	u19078065@tuks.co.za	0833680673
S'nethemba Xulu	u19239158	u19239158@tuks.co.za	0659374090
Chardé Devine Vries	u19072912	u19072912@tuks.co.za	0767890267
Sonali Badrinath	u19146851	u19146851@tuks.co.za	0713623778

DELIVERABLE 3 & 4

THIS DOCUMENT CONTAINS THE TECHNICAL SPECIFICATION OF THE SYSTEM WE INTEND TO DEVELOP FOR OUR CLIENT. IN THIS DELIVERABLE WE WILL BE DETAILING THE SYSTEM ARCHITECTURE, SYSTEM DATABASE DESIGN THROUGH THE ERD AND SIZE ESTIMATION OF DATABSASE; AS WELL AS SYSTEM INTERFACE DESIGN USING INPUT AND OUTPUT DESIGN AND ENTITY RELATIONSHIP MODELLING.

Table of Contents

INTRODUCTION

This document contains our technical specification for the NKAP Bolting System. We will look at the system architecture design which includes a Full set of Use Case Diagrams and Design Use Case Documentation of each Use Case. We will also look at Object-Oriented Design and System Database Design which will include the Entity Relationship model and the size estimation of the proposed database. We will then look at the System Interface Design which includes inputs, outputs as well as other interfaces used for our system and lastly Package Design Specifications which includes test specifications, hardware and software requirements, network/web layout specifications, and validations.

SYSTEM ARCHITECTURE DESIGN

1. Use Cases

INTRODUCTION

This section contains the use cases in the form of a full set of use case diagrams as well as Design Use Case Documentation of each Use Case.

1.1. USE CASE DIAGRAMS

2. **Use Case Diagram Key**		
Element	**Description**	**Example**
System	Outer Rectangle Element with the system name, 0. NKAP Bolting system and black outline.	
Sub-System	Inner rectangle element with black outline and Sub-System name.	
Use Case	Use Case element with black outline and green fill.	

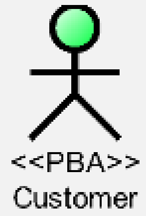
Actor	Use Case actor element with green fill and name of actor and actor type below.	
Association	Use Case relationship line with ‘initiates’ tag.	
Uses Association	Use case dependency relationship with ‘uses’ tag.	

Figure 1 Use Case Diagram Key

1.1.1 Login Subsystem

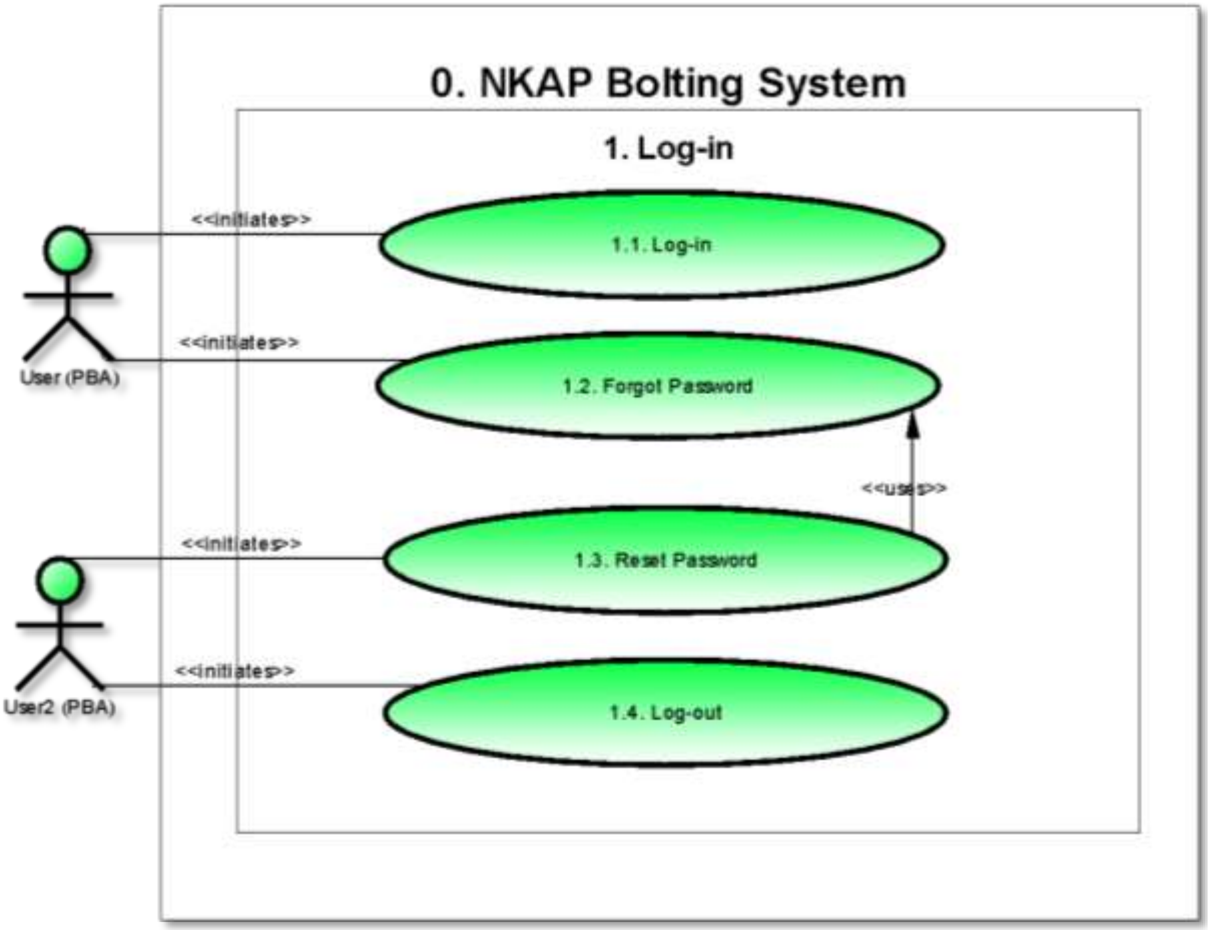
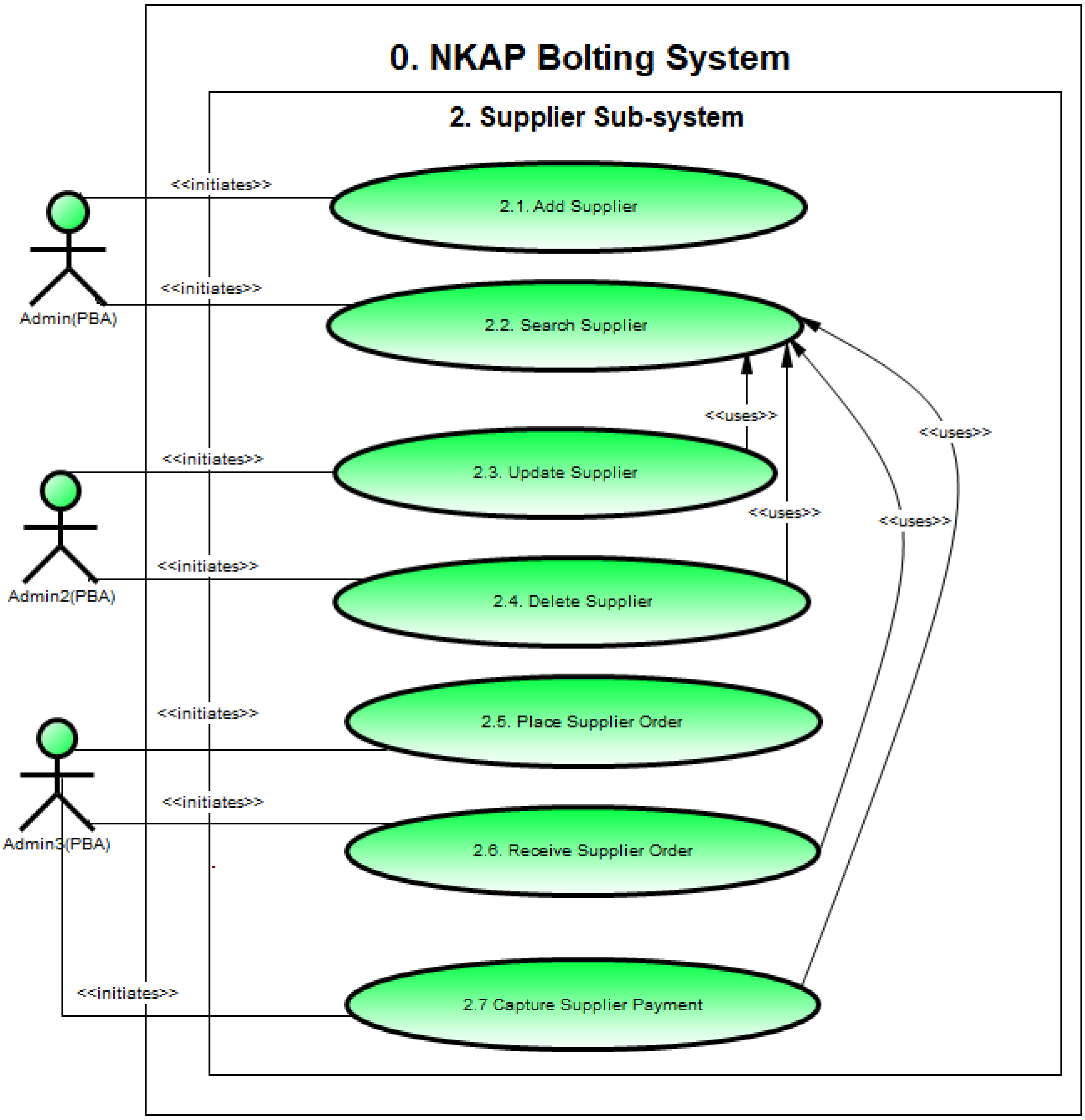


Figure 1 Login Use Case Diagram



1.1.2 Supplier Subsystem

Figure 2 Supplier Use Case Diagram

1.1.3 User Subsystem

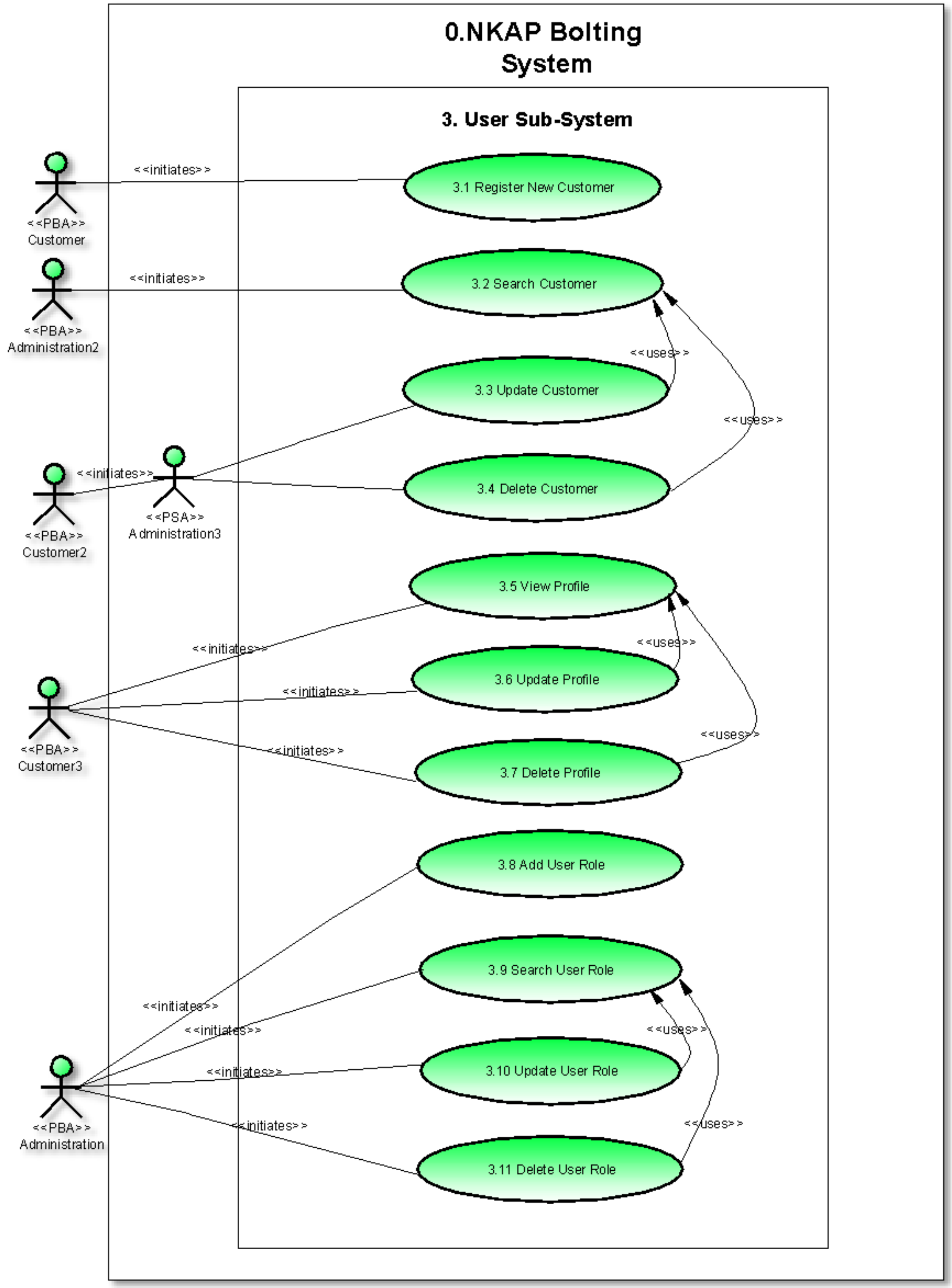


Figure 3 User Use Case Diagram

1.1.4 Employee Delivery Shifts Subsystem

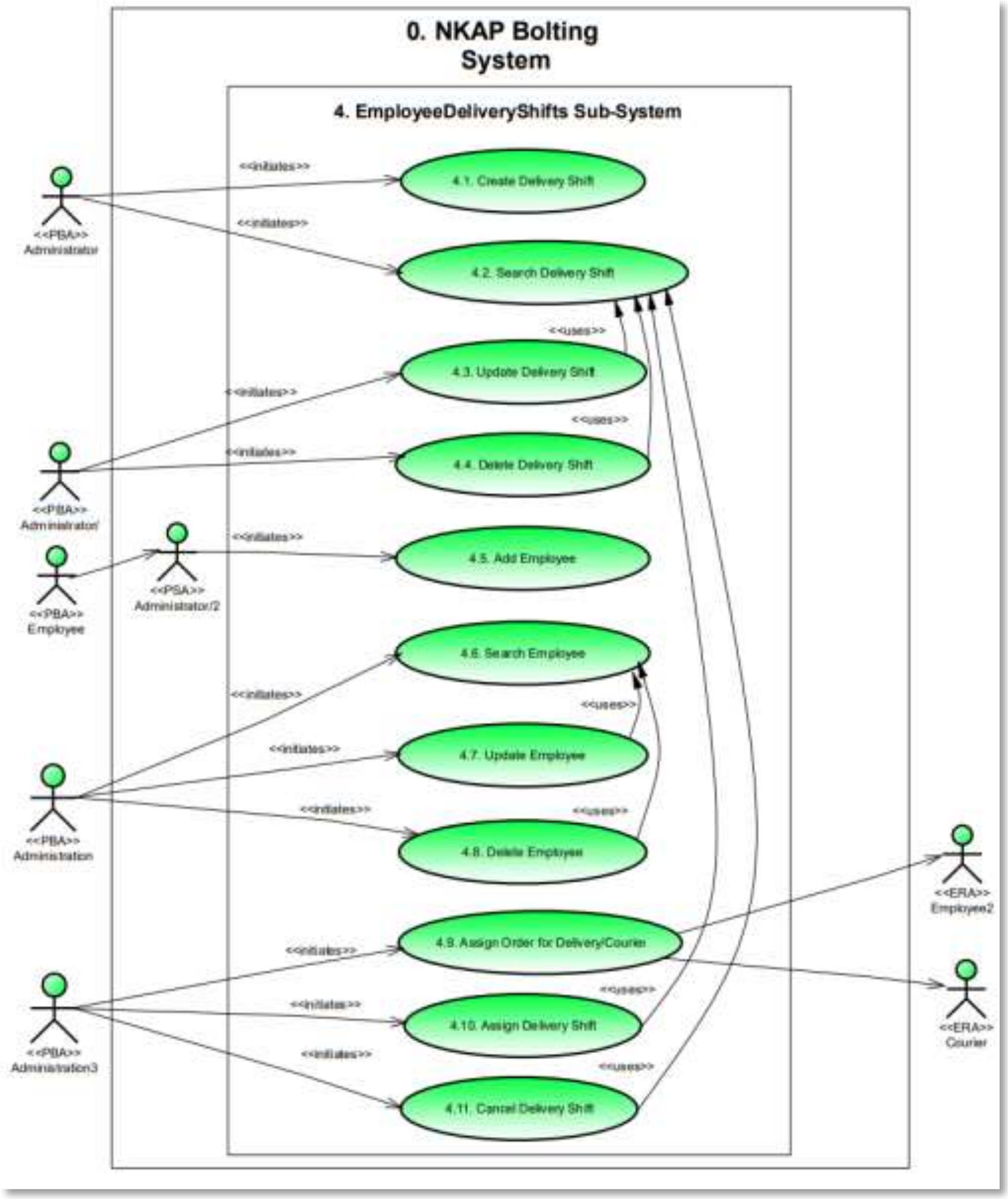


Figure 4 Employee Delivery Shifts Use Case Diagram

1.1.5 Administration Back Office Subsystem

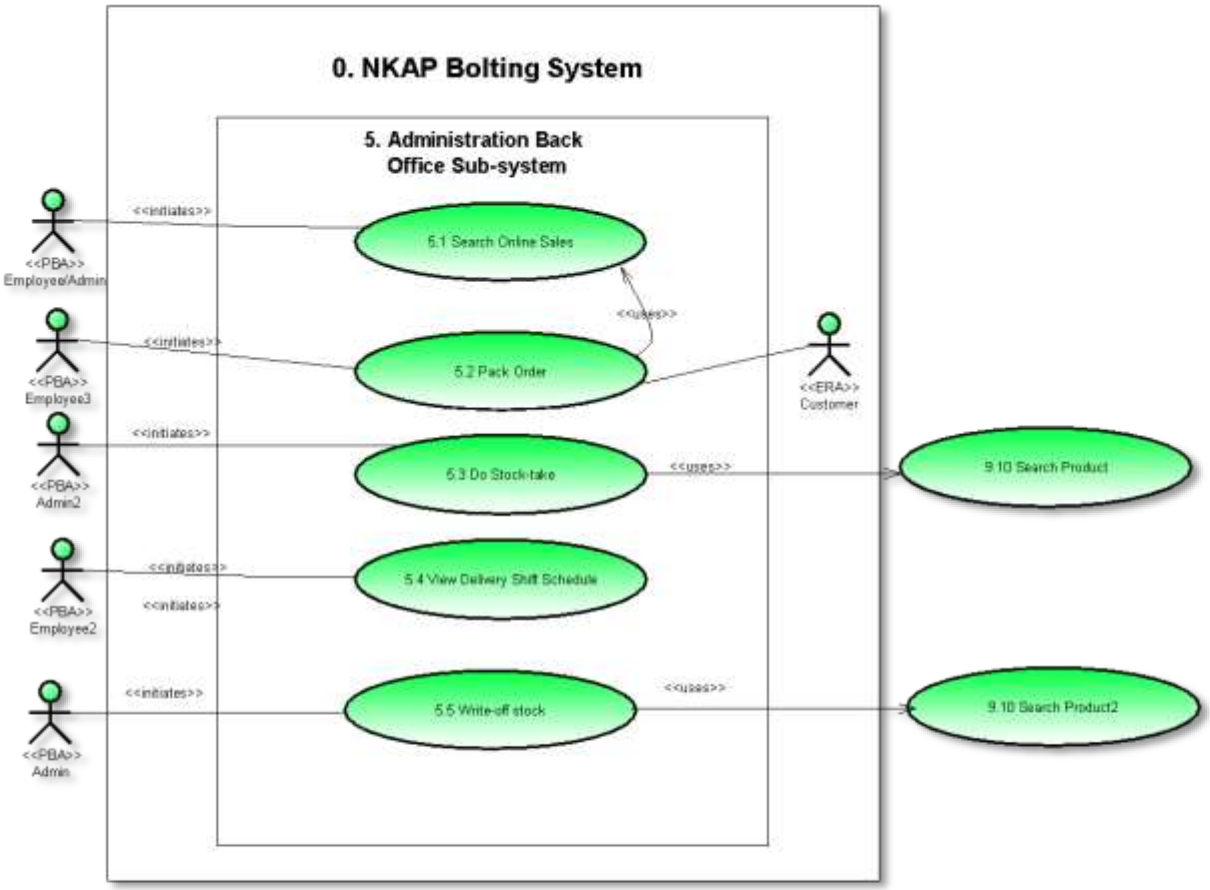


Figure 5 Administration Back Office Use Case Diagram

1.1.6 Customer Subsystem

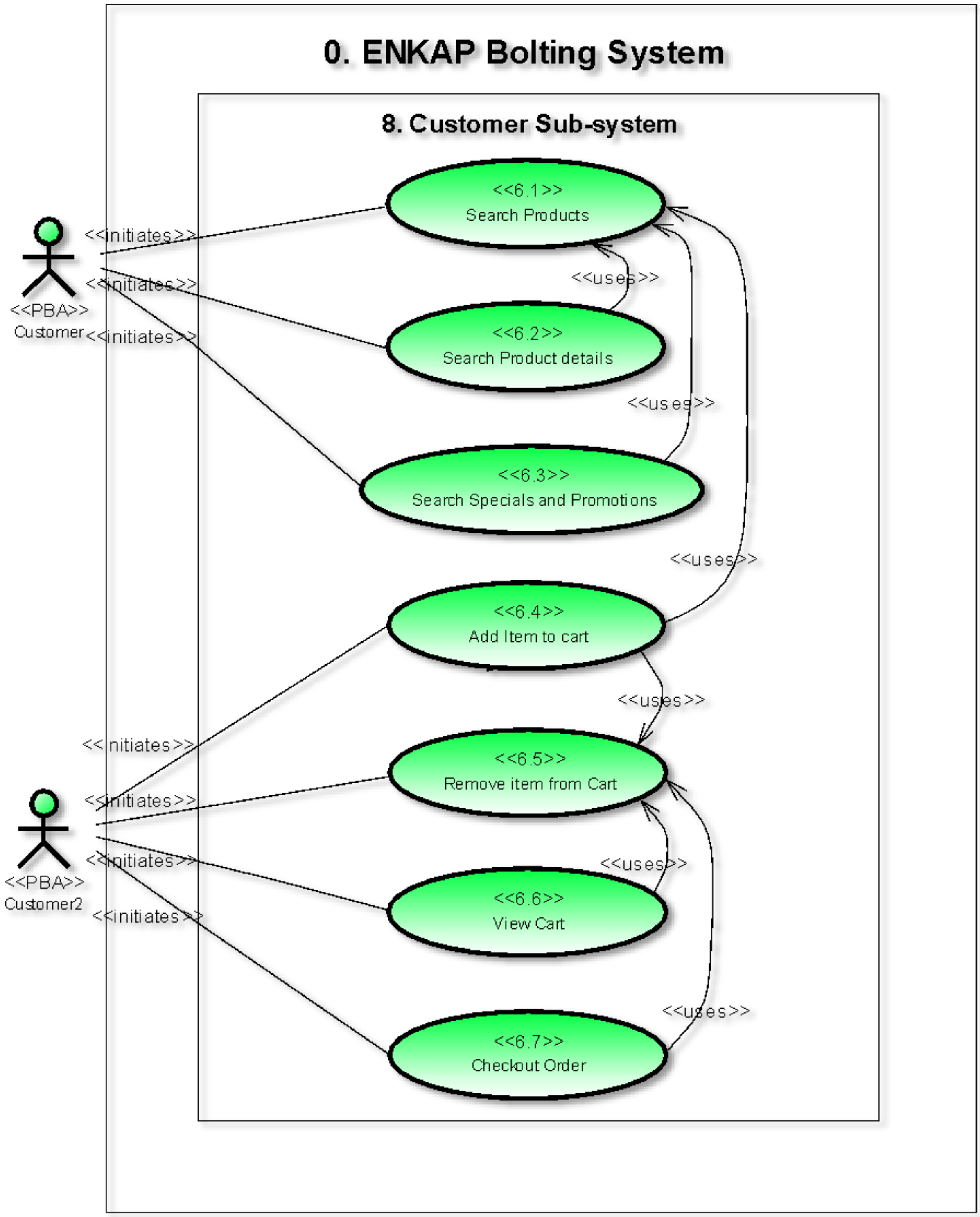


Figure 6 Customer Use Case Diagram

1.1.7 Reports Subsystem

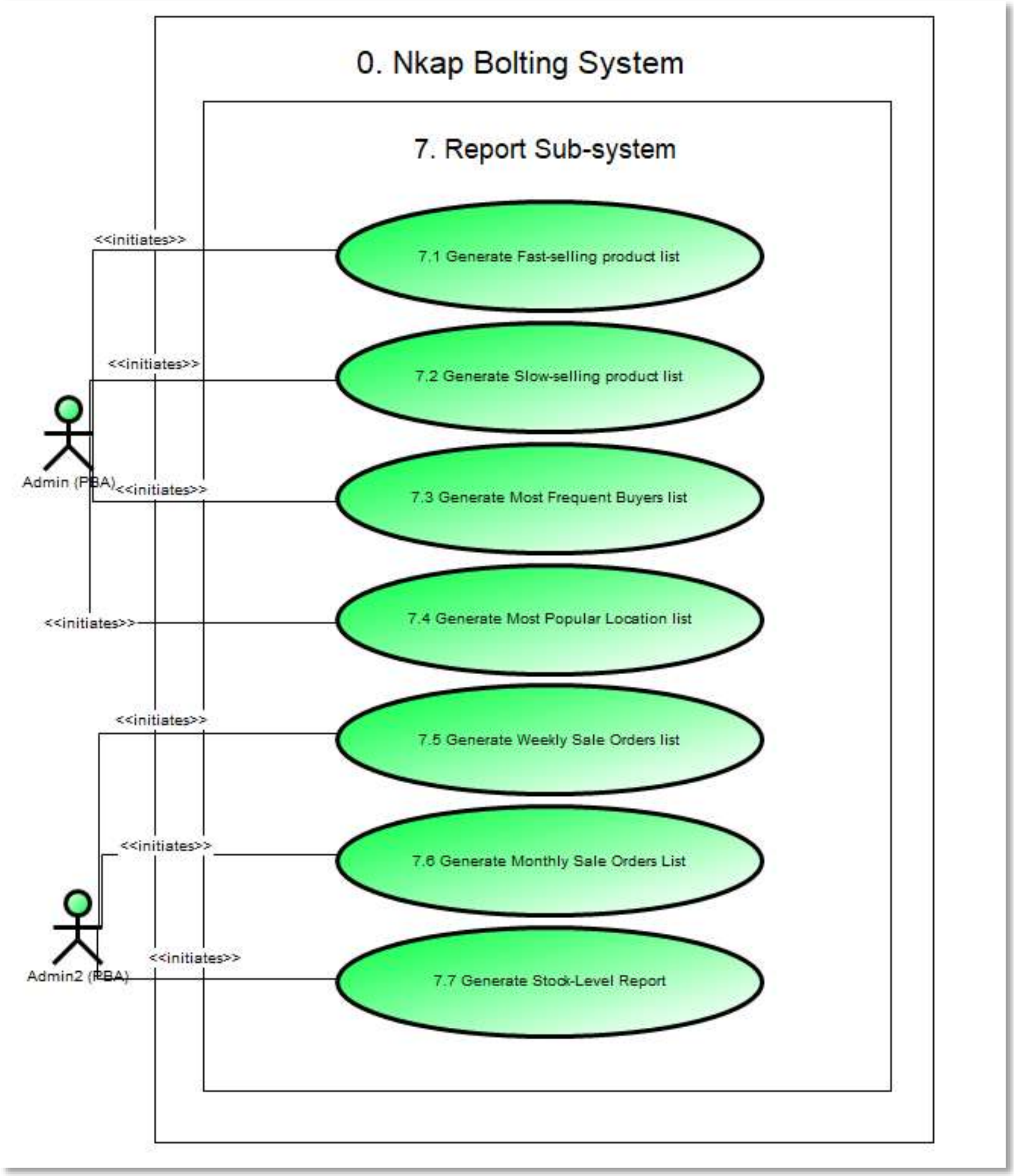


Figure 7 Reports Use Case Diagram

1.1.8 Delivery Subsystem

UPDATED DELIVERY SUBSYTEM

(Paste Use case diagram above the caption)

Figure 8 Delivery Use Case Diagram

1.1.9 Products Subsystem

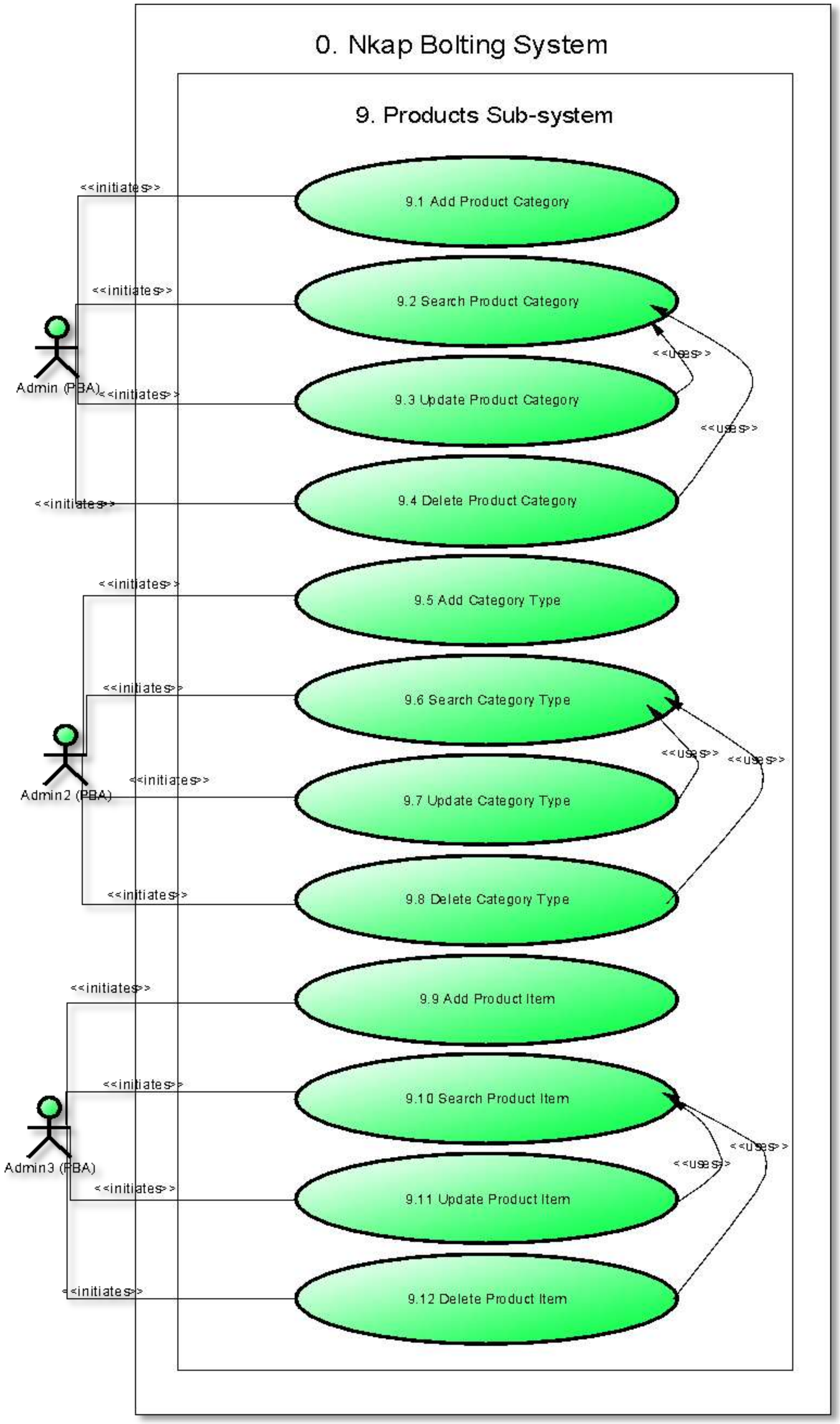


Figure 9 Products Use Case Diagram

1.1.10 Specials Subsystem

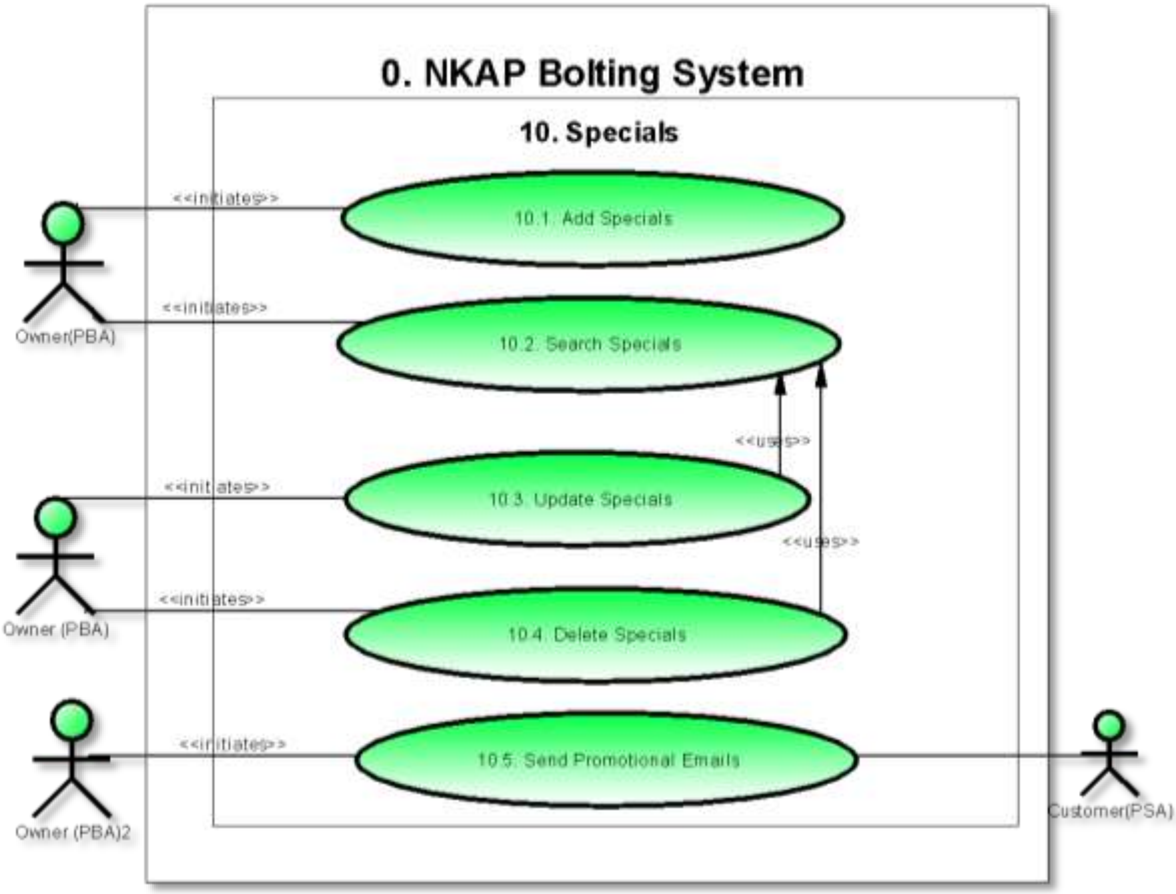


Figure 10 Specials Use Case Diagram

2.1. DESIGN USE CASE DOCUMENTATION

2.1.1. LOGIN SUBSYSTEM

<table>
<tr><th colspan="5">NKAP BOLTING SYSTEM</th></tr>
<tr><td>DATE:</td><td colspan="4">03-06-2021</td></tr>
<tr><td>VERSION:</td><td colspan="4">1.0</td></tr>
<tr><td>AUTHOR(S)</td><td colspan="4">S’nethemba Xulu</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Log-in</td><td colspan="2">USE CASE TYPE</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">1.1</td><td colspan="2" rowspan="3">Business Requirements: ☐
System Analysis: ☐
System Design: ☑</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">High</td></tr>
<tr><td>SOURCE:</td><td colspan="2">NKAP Bolting System Requirements</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR:</td><td colspan="4">User (Customer, Admin)</td></tr>
<tr><td>PRIMARY THE SYSTEM ACTOR:</td><td colspan="4">None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="4">None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="4">None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="4">The use case describes the event where the user wants to log in to the system. The use case begins when the user enters their Log-in details (username and password). The system will verify the details using the <u>User</u> table. The use case ends when the user is logged in to the system. The system will allow access to functionalities based on their access level defined by their UserRole_ID.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="4">• The user should already be registered on to the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="4">The user wishes to log-in to the system.</td></tr>
<tr><td rowspan="3">TYPICAL COURSE OF EVENTS:</td><td rowspan="2">ACTOR ACTION:</td><td colspan="3">SYSTEM RESPONSE:</td></tr>
<tr><td>Manual Action</td><td colspan="2">Automated Action</td></tr>
<tr><td>Step 1: The user wants to log in to the system.</td><td></td><td colspan="2">Step 2: The system will display the log-in screen with the following HTML elements:
<table><tr><th>Input name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Login</td><td>Heading</td><td>This serves as the</td></tr></table></td></tr>
</table>

<table>
<tr><td></td><td></td><td></td><td>
<table>
<tr><td></td><td></td><td>header of the page.</td></tr>
<tr><td>Username</td><td>Label</td><td>None</td></tr>
<tr><td>Username</td><td>Textbox</td><td>None</td></tr>
<tr><td>Password</td><td>Label</td><td>None</td></tr>
<tr><td>Password</td><td>Textbox</td><td>None</td></tr>
<tr><td>Forgot Password</td><td>Link</td><td>Invokes Use Case 1.2 Forgot Password.</td></tr>
<tr><td>Log-in</td><td>Button</td><td>None</td></tr>
<tr><td>Register</td><td>Button</td><td>Invokes Use Case 3.1 Register New Customer</td></tr>
</table>

The system will prompt the user to enter their Log-in details.</td></tr>
<tr><td></td><td>Step 3: The user enters their log-in details.</td><td></td><td></td></tr>
<tr><td></td><td>Step 4: The user clicks on the log-in button</td><td></td><td>Step 5: The system captures and validates the user log-in details by reading the Users, Username and Password from the <u>User</u> table. [Alt]</td></tr>
<tr><td></td><td></td><td></td><td>Step 6: A SQL read query is used to search for the login details entered in the User table:
Username
Password</td></tr>
<tr><td></td><td></td><td></td><td>Step 7: The system uses a hashing algorithm on the entered password and compares the entered password to the password in the User table. [Alt]</td></tr>
<tr><td></td><td></td><td></td><td>Step 8: The system will add a record in the AuditTrail entity using the SQL Add. The system will add the:
Username
Date
Time</td></tr>
</table>

<table>
<tr><td rowspan="2">ALTERNATE COURSES:</td><td>[Alt Step 5]: A validation error occurs because the user entered information that is incorrect. The system will notify the user that their login details are incorrect.

The system will display a modal with the following controls, to inform the user of the error:

<table>
<tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Validation Error</td><td>Label</td><td>This serves as a header for the modal.</td></tr>
<tr><td>The entered login credentials are incorrect. Please try again.</td><td>Label</td><td>Tells the user that the details they entered are incorrect.</td></tr>
<tr><td>OK</td><td>Button</td><td>None</td></tr>
</table>

Return to step 3.</td></tr>
<tr><td>[Alt Step 7]: The password entered doesn’t match with the password in the User table. The system will notify the user that their password is incorrect.

The system will display a modal with the following controls, to inform the user of the error:

<table>
<tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Validation Error</td><td>Label</td><td>This serves as a header for the modal.</td></tr>
<tr><td>The entered password is incorrect. Please try again.</td><td>Label</td><td>Tells the user that the password they entered is incorrect.</td></tr>
<tr><td>OK</td><td>Button</td><td>None</td></tr>
</table>

Return to step 3</td></tr>
<tr><td>CONCLUSION:</td><td>The system logs the user in to the system and the relevant functionalities are made available to them.</td></tr>
<tr><td>POST-CONDITION:</td><td>• The user will be able to access the system and its relevant functionalities.
• The User table is successfully updated.</td></tr>
<tr><td>BUSINESS RULES:</td><td>The users should have been registered on to the system before they can access the system.</td></tr>
<tr><td>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:</td><td>None</td></tr>
</table>

ASSUMPTIONS:	The users have registered before.
OPEN ISSUES:	None

NKAP BOLTING SYSTEM			
DATE:	03-06-2021		
VERSION:	**1.0**		
AUTHOR(S)	S'nethemba Xulu		
USE CASE NAME:	**Forgot Password**	**USE CASE TYPE**	
USE CASE ID:	**1.2**	**Business Requirements:**	☐
PRIORITY:	High	**System Analysis:**	☐
SOURCE:	NKAP Bolting System Requirements	**System Design:**	☑
PRIMARY BUSINESS ACTOR:	User (Customer)		
PRIMARY THE SYSTEM ACTOR:	None		
OTHER PARTICIPATING ACTORS:	None		
OTHER INTERESTED STAKEHOLDERS:	None		
DESCRIPTION:	This use case describes the event where the user has forgotten their login details and they wish to create new details. This use case begins when the user clicks on the forgot password link on the login screen. The use case ends when the system opens the reset password screen for the user.		
PRE-CONDITION:	• The users login details are incorrect.		
TRIGGER:	The user clicking on the forgot password button.		
TYPICAL COURSE OF EVENTS:	**ACTOR ACTION:**	**SYSTEM RESPONSE:**	
		Manual Action	Automated Action

<table>
<tr><td></td><td>Step 1: The user clicks on the “Forgot password” link.</td><td></td><td>Step 2: The system displays the “Forgot Password” page with the following controls:
<table>
<tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Forgot Password</td><td>Label</td><td>This serves as a header for the modal.</td></tr>
<tr><td>Email Address</td><td>Label</td><td>Prompts the user to enter their email address.</td></tr>
<tr><td>Email Address</td><td>Textbox</td><td>None</td></tr>
<tr><td>Submit</td><td>Button</td><td>Allows the user to submit the Reset password request.</td></tr>
<tr><td>Cancel</td><td>Button</td><td>None</td></tr>
</table></td></tr>
<tr><td></td><td>Step 3: The user enters their email address.</td><td></td><td></td></tr>
<tr><td></td><td>Step 4: The user clicks on the “Submit” button. [Alt]</td><td></td><td>Step 5: The system validates the email address that has been captured in the Customer table as follows: [Alt]
<table>
<tr><th>Input Name</th><th>Attribute in table</th><th>Validation Requirements</th></tr>
<tr><td>Email Address</td><td>Customer_EmailAddress</td><td>Required, Max 100 characters, must contain “@” and a domain extension (e.g. “.com”)</td></tr>
</table></td></tr>
</table>

			Step 6: The system uses a SQL Read Query to search the **Customer** table for a customer having the entered email address. **[Alt]**
			Step 7: The system generates an email with the reset password link and sends it to the customers Customer_EmailAddress.
			Step 8: The system notifies the user that a link has been sent to their email address, by a pop-up with the following controls: **Input Name** / **Control Type** / **Notes** **Outcome** / Label / This is a header for the pop-up **A password reset link has been sent to your email.** / Label / None **Ok** / Button / None
ALTERNATE COURSES:	**[Alt Step 4]:** The user clicks on the "Cancel" button. Terminate the use case.		
	[Alt Step 5]: The email address captured is invalid. The system notifies the user with the following controls: **Input Name** / **Control Type** / **Notes** **Invalid Email** / Label / This is a header for the pop-up **Invalid email address. Please try again.** / Label / None **Ok** / Button / None		
	[Alt Step 6]: The captured email address does not exist in the Customer table. The system will notify the user using a dialog box with the following controls:		

Step 8 pop-up controls:

Input Name	Control Type	Notes
Outcome	Label	This is a header for the pop-up
A password reset link has been sent to your email.	Label	None
Ok	Button	None

Alt Step 5 controls:

Input Name	Control Type	Notes
Invalid Email	Label	This is a header for the pop-up
Invalid email address. Please try again.	Label	None
Ok	Button	None

<table>
<tr><td></td><td>

Input Name	Control Type	Notes
Email Error	Label	This is a header for the pop-up
The entered email address does not exist. Please enter another email address.	Label	None
Ok	Button	None

</td></tr>
<tr><td>CONCLUSION:</td><td>The use case ends when the user has been told that a link has been sent to their email.</td></tr>
<tr><td>POST-CONDITION:</td><td>• The email has been sent to the user.</td></tr>
<tr><td>BUSINESS RULES:</td><td>1. None</td></tr>
<tr><td>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:</td><td>None</td></tr>
<tr><td>ASSUMPTIONS:</td><td>None</td></tr>
<tr><td>OPEN ISSUES:</td><td>None</td></tr>
</table>

<table>
<tr><th colspan="3">NKAP BOLTING SYSTEM</th></tr>
<tr><td>DATE:</td><td colspan="2">03-06-2021</td></tr>
<tr><td>VERSION:</td><td colspan="2">1.0</td></tr>
<tr><td>AUTHOR(S)</td><td colspan="2">S’nethemba Xulu</td></tr>
<tr><td>USE CASE NAME:</td><td>Reset Password</td><td>USE CASE TYPE</td></tr>
<tr><td>USE CASE ID:</td><td>1.3</td><td rowspan="3">Business Requirements: ☐
System Analysis: ☐
System Design: ☑</td></tr>
<tr><td>PRIORITY:</td><td>High</td></tr>
<tr><td>SOURCE:</td><td>NKAP Bolting System Requirements</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR:</td><td colspan="2">User (Customer, Admin)</td></tr>
<tr><td>PRIMARY THE SYSTEM ACTOR:</td><td colspan="2">None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="2">None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="2">None</td></tr>
</table>

<table>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case describes the event where the user changes their password. The user will be navigated to a “New Password” page. They will enter their new password and click on the submit button. The system will then update the password of the user in the User table.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The user should have forgotten their password.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The user requests to reset their password.</td></tr>
<tr><td rowspan="4">TYPICAL COURSE OF EVENTS:</td><td rowspan="2">ACTOR ACTION:</td><td colspan="2">SYSTEM RESPONSE:</td></tr>
<tr><td>Manual Action</td><td>Automated Action</td></tr>
<tr><td>Step 1: The user navigates to the “Reset Password” page by clicking on the link they received in their email.</td><td></td><td>Step 2: The system displays the “Reset Password” page with the following controls:
<table>
<tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Reset Password</td><td>Label</td><td>Prompts the user to enter their new password</td></tr>
<tr><td>New Password</td><td>Textbox</td><td>None</td></tr>
<tr><td>Confirm Password</td><td>Label</td><td>Prompts the user to confirm their new password.</td></tr>
<tr><td>Confirm Password</td><td>Textbox</td><td>None</td></tr>
<tr><td>Submit</td><td>Button</td><td>Submits the new password.</td></tr>
<tr><td>Cancel</td><td>Button</td><td>Cancels the whole use case.</td></tr>
</table></td></tr>
<tr><td>Step 3: The user enters the password information.</td><td></td><td></td></tr>
</table>

<table>
<tr><td rowspan="2"></td><td>Step 4: The user clicks on the “Submit” button. [Alt]</td><td></td><td>Step 5: The system validates that the information that has been captured matches the validation requirements of the User table [Alt]:
<table>
<tr><th>Input Name</th><th>Attribute in Table</th><th>Validation Requirements</th></tr>
<tr><td>Password</td><td>User_Password</td><td>Max 12 characters</td></tr>
<tr><td>Confirm Password</td><td>User_Password</td><td>Must match the above password.</td></tr>
</table></td></tr>
<tr><td></td><td></td><td>Step 6: The system saves the updated password in the User table:
• User_Password</td></tr>
<tr><td rowspan="2">ALTERNATE

COURSES:</td><td colspan="3">[Alt Step 4]: The user clicks on the “Cancel” button. Terminate the use case.</td></tr>
<tr><td colspan="3">[Alt Step 5]: There is a validation error on the password that the user entered. The system will show a pop-up with the following details:
<table>
<tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Password does not match with confirm password</td><td>Label</td><td>None</td></tr>
<tr><td>Ok</td><td>Button</td><td>None</td></tr>
</table></td></tr>
<tr><td>CONCLUSION:</td><td colspan="3">The user’s password will be updated in the database.</td></tr>
<tr><td>POST-CONDITION:</td><td colspan="3">• The user will be redirected to the log-in screen.</td></tr>
<tr><td>BUSINESS RULES:</td><td colspan="3">1. None</td></tr>
<tr><td>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:</td><td colspan="3">None</td></tr>
<tr><td>ASSUMPTIONS:</td><td colspan="3">The user forgot their password.</td></tr>
<tr><td>OPEN ISSUES:</td><td colspan="3">None</td></tr>
</table>

<table>
<tr><th colspan="4">NKAP BOLTING SYSTEM</th></tr>
<tr><td>DATE:</td><td colspan="3">03-06-2021</td></tr>
<tr><td>VERSION:</td><td colspan="3">1.0</td></tr>
<tr><td>AUTHOR(S)</td><td colspan="3">S’nethemba Xulu</td></tr>
<tr><td>USE CASE NAME:</td><td>Logout</td><td colspan="2">USE CASE TYPE</td></tr>
<tr><td>USE CASE ID:</td><td>1.4</td><td>Business Requirements:</td><td>☐</td></tr>
<tr><td>PRIORITY:</td><td>High</td><td>System Analysis:</td><td>☐</td></tr>
<tr><td>SOURCE:</td><td>NKAP Bolting System Requirements</td><td>System Design:</td><td>☑</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR:</td><td colspan="3">User (Customer, Admin)</td></tr>
<tr><td>PRIMARY THE SYSTEM ACTOR:</td><td colspan="3">None</td></tr>
</table>

<table>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">The use case describes the event where the user wants to log out of the system. The use case begins when the user clicks on the logout option. The system will confirm with the user if they can confirm that they want to logout of the system. The use case ends when the user is successfully logged out.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">The user should be logged in to the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The user clicks on the logout item in the navbar.</td></tr>
<tr><td rowspan="4">TYPICAL COURSE OF EVENTS:</td><td rowspan="2">ACTOR ACTION:</td><td colspan="2">SYSTEM RESPONSE:</td></tr>
<tr><td>Manual Action</td><td>Automated Action</td></tr>
<tr><td>Step 1: The user wants to log out of the system.</td><td></td><td></td></tr>
<tr><td>Step 2: The user clicks on the logout item in the navbar.</td><td></td><td>Step 3: The system will prompt the user to confirm their logout request, using a pop-up with the following controls:
<table>
<tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Outcome</td><td>Label</td><td>This serves as a header for the pop-up</td></tr>
<tr><td>Are you sure you want to logout?</td><td>Label</td><td>Confirms if the user really wants to logout of the system.</td></tr>
<tr><td>Yes</td><td>Button</td><td>User will be logged out.</td></tr>
<tr><td>Cancel</td><td>Button</td><td>User will remain on the same page.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td rowspan="3"></td><td></td><td></td><td></td></tr>
<tr><td>Step 4: The user clicks on the “Yes” button, which confirms the logging out of the user. [Alt]</td><td></td><td>Step 5: The system ends the AuditTrail and an SQL delete query is used to remove the AuditTrail from the User table.

<table>
<tr><th>Details to be removed</th><th>Attribute in table</th><th>Details</th></tr>
<tr><td>AuditTrail_ID</td><td>AuditTrail_ID</td><td>Token removed at the end of the AuditTrail.</td></tr>
</table></td></tr>
<tr><td></td><td></td><td>Step 6: The system will log the user out.</td></tr>
<tr><td>ALTERNATE COURSES:</td><td colspan="3">[ALT Step 4]: The user clicks on the cancel button, which cancels the logout. Terminate use case.</td></tr>
<tr><td>CONCLUSION:</td><td colspan="3">The user is logged out of the system.</td></tr>
<tr><td>POST-CONDITION:</td><td colspan="3">• The user is logged out of the system.
• The User table is successfully updated.</td></tr>
<tr><td>BUSINESS RULES:</td><td colspan="3">1. AuditTrail_ID is removed from the local storage.</td></tr>
<tr><td>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:</td><td colspan="3">None</td></tr>
<tr><td>ASSUMPTIONS:</td><td colspan="3">The user is already logged in to the system.</td></tr>
<tr><td>OPEN ISSUES:</td><td colspan="3">None</td></tr>
</table>

2.1.2. SUPPLIER SUBSYTEM

NKAP BOLTING SYSTEM		
DATE:	04-06-2021	
VERSION:	**1.0**	
AUTHOR(S)	S'nethemba Xulu	
Use Case Name:	**Add Supplier**	**Use case type**
Use Case ID:	**2.1**	**Business Requirements:** ☐ **System Analysis:** ☐ **System Design:** ☑
Priority:	High	
Source:	NKAP Bolting System Requirements	
Primary Business Actor:	Administration	
Primary The system Actor:	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case describes the event where the admin wants to add a new supplier on to the system. The use case begins when the admin requests to add the supplier. The admin will enter all the supplier details and the information	

<table>
<tr><td></td><td colspan="3">will be captured in the Supplier table. The use case ends when the supplier has been successfully added.</td></tr>
<tr><td>Pre-condition:</td><td colspan="3">The supplier does not exist on the system.
The admin is on the Supplier screen.</td></tr>
<tr><td>Trigger:</td><td colspan="3">The admin requesting to add a new supplier.</td></tr>
<tr><td rowspan="3">Typical Course

of Events:</td><td rowspan="2">Actor Action:</td><td colspan="2">system Response:</td></tr>
<tr><td>Manual Action</td><td>Automated Action</td></tr>
<tr><td>Step 1: The admin requests to add a new supplier.</td><td></td><td></td></tr>
<tr><td></td><td>Step 2: The admin will click on the “Add Supplier” button in the Supplier Screen.</td><td></td><td>Step 3: The system responds by loading the “Add Supplier” modal with the following controls:

<table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Add Supplier</td><td>Heading</td><td>None</td></tr><tr><td>Supplier Name:</td><td>Label</td><td>This is the label for the name of the supplier.</td></tr><tr><td>Name</td><td>Textbox</td><td>None</td></tr><tr><td>Supplier Contact Number:</td><td>Label</td><td>This is the label for the contact number of</td></tr></table></td></tr>
</table>

					the supplier
			Contact Number	Textbox	This control should only allow for numeric inputs.
			Email Address :	Label	This is the label for the email address of the supplier
			Email	Textbox	None
			Supplier Address Line 1:	Label	This is the label for the address line 1 of the supplier
			Address Line	Textbox	None
			Supplier Address Line 2:	Label	This is the label for the address line 2 of the supplier
			Address Line	Textbox	None

Supplier Address Line 3:	Label	This is the label for the address line 3 of the supplier
Address Line	Textbox	None
Supplier City/Town:	Label	This is the label for the city/town of the supplier
City	Textbox	None
Supplier Postal code:	Label	This is the label for the postal code of the supplier
Postal Code	Textbox	None
Supplier Type:	Label	This is the label for the supplier type of the supplier
Type	Dropdown	This will be populated with the information retrieved from the

			<table><tr><td></td><td></td><td>**SupplierType** table</td></tr><tr><td>**Add**</td><td>Button</td><td>None</td></tr><tr><td>**Cancel**</td><td>Button</td><td>This button is used when the admin decides to terminate the process of adding a new supplier.</td></tr></table>
	Step 4: The admin enters the details in the respective places: • Supplier Name • Contact Number • Email Address • AddressLine1 • AddressLine2 • City/Town		

<table>
<tr><td></td><td>Postal Code
Supplier Type</td><td></td><td></td></tr>
<tr><td></td><td>Step 5: The admin clicks on the “Add” button [Alt]</td><td></td><td>Step 6: The system captures and validates the information entered by the admin against the Supplier table [Alt]:
<table>
<tr><th>Input Name</th><th>Attribute in table</th><th>Validation requirements</th></tr>
<tr><td>Name:</td><td>Supplier_Name</td><td>Required, maximum of 100 characters.</td></tr>
<tr><td>Contact Number</td><td>Supplier_ ContactNumber</td><td>Required, maximum of 10 characters</td></tr>
<tr><td>Email</td><td>Supplier_ EmailAddress</td><td>Required, maximum of 100 characters</td></tr>
<tr><td>Address Line 1</td><td>Supplier_ AddressLine1</td><td>Required, maximum of 100 characters</td></tr>
<tr><td>Address Line 2</td><td>Supplier_ AddressLine2</td><td>Required, maximum of 100 characters</td></tr>
</table></td></tr>
</table>

Address Line 3	Supplier_ AddressLine3	Required, maximum of 100 characters
City	Supplier_ City_Town	Required, maximum of 100 characters
Postal Code	Supplier_ PostalCode	Required, maximum of 4 numeric characters

Step 7: The system uses a SQL insert query to save the captured information in the **Supplier** table **[Alt]**:

Information to be saved:	Attribute in table	Details
Supplier_ID	Supplier_ID	Generated by the system by reading the last Supplier_ID from the **Supplier** table

		and adding it by one.
Name	Supplier_Name	None
Contact Number	Supplier_ ContactNumber	None
Email	Supplier_ EmailAddress	None
Address Line 1	Supplier_ AddressLine1	None
Address Line 2	Supplier_ AddressLine2	None
Address Line 3	Supplier_ AddressLine3	None
City	Supplier_ City_Town	None
Postal Code	Supplier_ PostalCode	None

Alternate Courses:	**[Alt Step 5]**: The admin clicks on the “Cancel” button which terminates this use case. Return to step 4

	[Alt Step 6]: The details entered were not in the right format, so the validation failed. The system will display to the admin where the validation errors occurred by showing a red outline where the validation failed.
	[Alt Step 7]: The new supplier could not be added to the system. The system will display a modal informing the admin that the saving of the supplier information was unsuccessful.

Input Name	Control Type	Notes
Error	Label	This serves as a header for the modal.
There was an error while saving the new supplier. Please try again later.	Label	This is to show that the saving of the supplier information was unsuccessful.
OK	Button	None

Conclusion:	This use case concludes when the supplier has been successfully added on the system.
Post-condition:	• The supplier has been added to the **Supplier** table.
Business Rules:	Only an admin can add a new supplier to the system.
Implementation Constraints and	None

Specifications:	
Assumptions:	None
Open Issues:	None

NKAP BOLTING SYSTEM		
DATE:	04-06-2021	
VERSION:	**1.0**	
AUTHOR(S)	S'nethemba Xulu	
Use Case Name:	**Search Supplier**	**Use case type**
Use Case ID:	**2.2**	**Business Requirements:** ☐
Priority:	High	**System Analysis:** ☐
Source:	NKAP Bolting System Requirements	**System Design:** ☑
Primary Business Actor:	Admin	
Primary The system Actor:	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	

<table>
<tr><td>Description:</td><td colspan="3">This use case describes the event where the admin wishes to search for a supplier. The admin will enter the search parameters like the suppler name. The system will search for the supplier based on the parameters the admin entered. This use case concludes when the supplier is shown the results of the search query.</td></tr>
<tr><td>Pre-condition:</td><td colspan="3">The admin should be logged on to the system.
The supplier should be already exist in the system.</td></tr>
<tr><td>Trigger:</td><td colspan="3">The admin wishes to search for a supplier.</td></tr>
<tr><td rowspan="4">Typical Course of Events:</td><td rowspan="2">Actor Action:</td><td colspan="2">System Response:</td></tr>
<tr><td>Manual Action</td><td>Automated Action</td></tr>
<tr><td>Step 1: The admin would like to search for a supplier.</td><td></td><td></td></tr>
<tr><td>Step 2: The admin inputs the search criteria and clicks on the</td><td></td><td>Step 3: The system captures and validates the information against the Supplier table [Alt]:</td></tr>
</table>

	search icon.		
			Step 4: A SQL read query is used to read the following attributes from the **Supplier** table based on the search criteria: **Supplier** table: ◦ Supplier_ID(PK) ◦ Supplier_Name ◦ Supplier_ContactNumber ◦ Supplier_EmailAddress ◦ Supplier_Location ◦ Supplier_AddressLine1 ◦ Supplier_AddressLine2 ◦ Supplier_City_Town ◦ Supplier_PostalCode ◦ Supplier_Balance **SupplierType** table: • SupplierType_ID(PK) • Description
			Step 5: The system will show the search results received from step 6 in a table.
Alternate courses:	**[Alt Step 5]:** The validation failed. The admin is prompted to enter a new search query.		

Conclusion:	The use case concludes when the admin receives the results of the searched criteria.
Post-condition:	• The admin receives the results of the suppliers with the same searched criteria
Business Rules:	**1.** None
Implementation Constraints and Specifications:	None
Assumptions:	None
Open Issues:	None

NKAP BOLTING SYSTEM		
DATE:	04-06-2021	
VERSION:	**1.0**	
AUTHOR(S)	S'nethemba Xulu	
Use Case Name:	**Update Supplier**	**Use case type**
Use Case ID:	**2.3**	**Business Requirements:** ☐
Priority:	High	**System Analysis:** ☐
Source:	NKAP Bolting System Requirements	**System Design:** ☑
Primary Business Actor:	Admin	
Primary The system Actor:	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case describes the event in which an admin wants to update the details of a supplier. The system will display the previous information that was added before and then the admin will add the new information. This use case ends when the information has been updated successfully.	

<table>
<tr><td>Pre-condition:</td><td colspan="3">The admin should be logged on to the system.</td></tr>
<tr><td>Trigger:</td><td colspan="3">The admin wishes update the details of the supplier.</td></tr>
<tr><td rowspan="4">Typical Course

of Events:</td><td rowspan="2">Actor Action:</td><td colspan="2">System Response:</td></tr>
<tr><td>Manual Action</td><td>Automated Action</td></tr>
<tr><td>Step 1: The admin requests to update the details of a supplier.</td><td></td><td>Step 2: The system invokes use case 2.2 Search Supplier.</td></tr>
<tr><td>Step 3: Admin clicks on the “Update” button.</td><td></td><td>Step 4: The system responds by loading the “Update Supplier” modal with the following controls:
<table>
<tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Update Supplier</td><td>Heading</td><td>None</td></tr>
<tr><td>Supplier Name:</td><td>Label</td><td>This is the label for the name of the supplier.</td></tr>
<tr><td>Name</td><td>Textbox</td><td>None</td></tr>
<tr><td>Supplier Contact Number:</td><td>Label</td><td>This is the label for the contact</td></tr>
</table></td></tr>
</table>

		number of the supplier
Contact Number	Textbox	This control should only allow for numeric inputs.
Email Address:	Label	This is the label for the email address of the supplier
Email	Textbox	None
Supplier Address Line 1:	Label	This is the label for the address line 1 of the supplier
Address Line	Textbox	None
Supplier Address Line 2:	Label	This is the label for the address line 2 of the supplier
Address Line	Textbox	None
Supplier Address Line 3:	Label	This is the label for the address line 3 of the supplier
Address Line	Textbox	None

Supplier City/Town :	Label	This is the label for the city/town of the supplier
City	Textbox	None
Supplier Postal code:	Label	This is the label for the postal code of the supplier
Postal Code	Textbox	None
Supplier Type:	Label	This is the label for the supplier type of the supplier
Type	Dropdown	This will be populated with the information retrieved from the **SupplierType** table
Update	Button	None
Cancel	Button	This button is used when the admin decides to terminate the process of adding a new supplier.

Step 5: The admin enters the new information and clicks on the "**Update**" button. **[Alt]**

Step 6: The system captures and validates the information entered by the admin against the Supplier table **[Alt]:**

Input Name	**Attribute in table**	**Validation requirements**
Name:	Supplier_Name	Required, maximum of 100 characters.
Contact Number	Supplier_ContactNumber	Required, maximum of 10 characters
Email	Supplier_EmailAddress	Required, maximum of 100 characters
Address Line 1	Supplier_AddressLine1	Required, maximum of 100 characters
Address Line 2	Supplier_AddressLine2	Required, maximum of 100 characters

Address Line 3	Supplier_AddressLine3	Required, maximum of 100 characters
City	Supplier_City_Town	Required, maximum of 100 characters
Postal Code	Supplier_PostalCode	Required, maximum of 4 numeric characters

Step 7: The system uses a SQL insert query to save the captured information in the **Supplier** table **[Alt]**:

Information to be saved:	**Attribute in table**	**Details**
Supplier_ID	Supplier_ID	None.
Name	Supplier_Name	None
Contact Number	Supplier_ContactNumber	None
Email	Supplier_EmailAddress	None

Address Line 1	Supplier_AddressLine1	None
Address Line 2	Supplier_AddressLine2	None
Address Line 3	Supplier_AddressLine3	None
City	Supplier_City_Town	None
Postal Code	Supplier_PostalCode	None

Alternate Courses:

[Alt Step 5]: The admin clicks on the "**Cancel**" button which terminates the this use case.

Return to **step 1.**

[Alt Step 6]: The details entered were not in the right format, so the validation failed. The system will display to the admin where the validation errors occurred by showing a red outline where the validation failed.

[Alt Step 7]: The updated supplier information could not be added to the system. The system will display a modal informing the admin that the saving of the supplier information was unsuccessful.

Input Name	Control Type	Notes
Error	Label	This serves as a header for the modal.
There was an error while saving the new supplier information.	Label	This is to show that the saving of the supplier

<table>
<tr><td></td><td><table>
<tr><td>Please try again later.</td><td></td><td>information was unsuccessful.</td></tr>
<tr><td>OK</td><td>Button</td><td>None</td></tr>
</table></td></tr>
<tr><td>Conclusion:</td><td>This use case concludes when the supplier information has been successfully updated on the system.</td></tr>
<tr><td>Post-condition:</td><td>• The supplier has been updated in the Supplier table.</td></tr>
<tr><td>Business Rules:</td><td>1. Only an admin can update the information of the supplier in the system.</td></tr>
<tr><td>Implementation Constraints and Specifications:</td><td>None</td></tr>
<tr><td>Assumptions:</td><td>None</td></tr>
<tr><td>Open Issues:</td><td>None</td></tr>
</table>

<table>
<tr><th colspan="3">NKAP BOLTING SYSTEM</th></tr>
<tr><td>DATE:</td><td colspan="2">04-06-2021</td></tr>
<tr><td>VERSION:</td><td colspan="2">1.0</td></tr>
<tr><td>AUTHOR(S)</td><td colspan="2">S’nethemba Xulu</td></tr>
<tr><td>Use Case Name:</td><td>Delete Supplier</td><td>Use case type</td></tr>
<tr><td>Use Case ID:</td><td>2.4</td><td rowspan="3">Business Requirements: ☐
System Analysis: ☐
System Design: ☑</td></tr>
<tr><td>Priority:</td><td>High</td></tr>
<tr><td>Source:</td><td>NKAP Bolting System Requirements</td></tr>
<tr><td>Primary Business Actor:</td><td colspan="2">Admin</td></tr>
<tr><td>Primary The system Actor:</td><td colspan="2">None</td></tr>
<tr><td>Other Participating Actors:</td><td colspan="2">None</td></tr>
<tr><td>Other Interested Stakeholders:</td><td colspan="2">None</td></tr>
<tr><td>Description:</td><td colspan="2">This use case describes the process where the admin wants to delete the supplier from the system. This use case begins when the Administrator requests to delete a supplier. This use case concludes when the admin gets notified that the supplier has been successfully removed from the system.</td></tr>
</table>

<table>
<tr><td>Pre-condition:</td><td colspan="3">The admin should be logged in to the system.
The supplier should already exist on the systems database.</td></tr>
<tr><td>Trigger:</td><td colspan="3">The admin requesting to delete a supplier.</td></tr>
<tr><td rowspan="4">Typical Course of Events:</td><td rowspan="2">Actor Action:</td><td colspan="2">system Response:</td></tr>
<tr><td>Manual Action</td><td>Automated Action</td></tr>
<tr><td>Step 1: The admin request to delete a supplier from the system.</td><td></td><td>Step 2: The system invokes Use case 2.2 Search Supplier.</td></tr>
<tr><td>Step 3: The admin clicks on the “Delete” button on the Supplier screen.</td><td></td><td>Step 4: The system displays a modal, with the following controls:
<table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Confirm deletion</td><td>Heading</td><td>This serves as a header for the modal.</td></tr><tr><td>Are you sure you want to delete</td><td>Label</td><td>None</td></tr></table></td></tr>
</table>

<table>
<tr><td></td><td></td><td></td><td>

this supplier?		
Yes	Button	This is to confirm that the admin wants to delete a supplier.
No	Button	This is selected when the admin does not want to delete a supplier.

</td></tr>
<tr><td></td><td>Step 5: The admin clicks on the “Yes” button to confirm the deletion of the supplier. [Alt]</td><td></td><td>

Step 6: The system uses a SQL delete query to remove the supplier information from the **Supplier** table **[Alt]:**

Information to be deleted:	Attribute in table	Details
Supplier_ID	Supplier_ID	Generated by the system by reading the last Supplier_ID from

</td></tr>
</table>

		the **Supplier** table and adding it by one.
Name	Supplier_Name	None
Contact Number	Supplier_Contact Number	None
Email	Supplier_EmailAddress	None
Address Line 1	Supplier_Address Line1	None
Address Line 2	Supplier_Address Line2	None
Address Line 3	Supplier_Address Line3	None
City	Supplier_City_Town	None
Postal Code	Supplier_PostalCode	None

Step 7: The system removes the deleted supplier from the table in the Supplier screen.

Alternate Courses:

[Alt Step 5]: The admin clicks on the “No” button.

Terminate this use case.

[Alt Step 6]: The supplier could not be deleted from the system. The system will display a modal informing the admin that the deletion of the supplier was unsuccessful.

Input Name	Control Type	Notes
Error	Label	This serves as a header for the modal.
There was an error while deleting the supplier. Please try again later.	Label	This is to show that the saving of the supplier information was unsuccessful.
OK	Button	None

Conclusion:	This use case ends when the supplier has been successfully deleted from the system.
Post-condition:	• The details of the supplier have been removed from the **Supplier** table.
Business Rules:	**1.** Only the admin can remove a supplier.
Implementation Constraints and Specifications:	None
Assumptions:	None
Open Issues:	**None**

DATE:	07-06-2021	
VERSION:	**1.0**	
AUTHOR(S)	S'nethemba Xulu, Jerome Amenigy	
Use Case Name:	**Place Supplier Order**	**Use case type**
Use Case ID:	**2.5**	**Business Requirements:** ☐
Priority:	High	**System Analysis:** ☐
Source:	NKAP Bolting System Requirements	**System Design:** ☑
Primary Business Actor:	Admin	
Primary The system Actor:	None	
Other Participating Actors:	None	
Other Interested Stakeholders:	None	
Description:	This use case describes the event where the admin navigates to the "**Place Supplier Order**" screen in order to generate a list of what the business wants to order from the supplier. The admin will select all the items they wish to purchase and the quantity. This list will be saved on to the system in the **SupplierOrder** and **SupplierOrderLine** tables.	
Pre-condition:	• The admin is logged on to the system.	
Trigger:	The admin requesting to place an order to the supplier.	
Typical Course		**system Response:**

<table>
<tr><th>of Events:</th><th>Actor Action:</th><th>Manual Action</th><th>Automated Action</th></tr>
<tr><td></td><td>Step 1: The admin requests to place a supplier order.</td><td></td><td></td></tr>
<tr><td></td><td>Step 2: The admin clicks on the “Place Supplier Order” option under the “Supplier” tab.</td><td></td><td>Step 3: The system displays the “Place Supplier Order” screen with the following controls:
<table>
<tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Place Supplier Order</td><td>Heading</td><td>This serves as a header for the screen.</td></tr>
<tr><td>Supplier</td><td>Label</td><td>None</td></tr>
<tr><td>Supplier</td><td>Dropdown</td><td>This is for the admin to choose the supplier that the order is going to be placed to.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr>
<td></td>
<td></td>
<td></td>
<td>
<table>
<tr><td>Category</td><td>Label</td><td>None</td></tr>
<tr><td>Category</td><td>Dropdown</td><td>This is for the admin to choose the category.</td></tr>
<tr><td>Submit</td><td>Button</td><td>When this button is clicked a table gets displayed of all the product items in that category.</td></tr>
</table>
</td>
</tr>
<tr>
<td></td>
<td>Step 4: The admin selects the Supplier and Category.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Step 5: The admin clicks on the</td>
<td></td>
<td>Step 6: The system responds by showing the list of the product items in that category as a list.</td>
</tr>
</table>

<table>
<tr>
<td></td>
<td>submit button.</td>
<td></td>
<td>The table will have the following headings:
<table>
<tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Select</td><td>Label</td><td>Under the select heading there will be check boxes on each row.</td></tr>
<tr><td>Product Item</td><td>Label</td><td>Under the product item heading there will be the names of each product item on each row.</td></tr>
<tr><td>Price</td><td>Label</td><td>Under the price heading there will be the prices of each product</td></tr>
</table>
</td>
</tr>
</table>

<table>
<tr>
<td></td>
<td></td>
<td></td>
<td>
<table>
<tr><td></td><td></td><td>item on each row.</td></tr>
<tr><td>Quantity</td><td>Label</td><td>The admin will select how many units of that product item they want to order.</td></tr>
</table>
At the end of the screen there will be the following buttons:
<table>
<tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Place Order</td><td>Button</td><td>None</td></tr>
<tr><td>Cancel</td><td>Button</td><td>None</td></tr>
</table>
</td>
</tr>
<tr>
<td></td>
<td>Step 7: The admin selects the product items they</td>
<td></td>
<td></td>
</tr>
</table>

	want and adds the quantity.		
	Step 8: The admin clicks on the "Place Order" button. **[Alt]**		**Step 9**: The system will capture the product items chosen and the quantity amounts of each item from the **ProductItems** table: • ProductItem_Name • ProductItem_CostPrice
			Step 10: The system will save the information in the **SupplierOrderLine** and **SupplierOrder** tables.
Alternate Courses:	**[Alt Step 8]:** The admin clicks on the "Cancel" button. Terminate the use case.		
Conclusion:	The supplier order has been placed on the system.		
Post-condition:	• The supplier order has been saved on to the system.		
Business Rules:	1. Only an admin can place a supplier order. 2. The Supplier must confirm that they have available stock quantities before the supplier order is captured on the system		
Implementation Constraints and Specifications:	None		
Assumptions:	None		
Open Issues:	None		

<table>
<tr><th colspan="3">NKAP BOLTING SYSTEM</th></tr>
<tr><td>DATE:</td><td colspan="2">04-06-2021</td></tr>
<tr><td>VERSION:</td><td colspan="2">1.0</td></tr>
<tr><td>AUTHOR(S)</td><td colspan="2">S’nethemba Xulu, Jerome Amenigy</td></tr>
<tr><td>Use Case Name:</td><td>Receive Supplier Order</td><td>Use case type</td></tr>
<tr><td>Use Case ID:</td><td>2.6</td><td rowspan="3">Business Requirements: ☐
System Analysis: ☐
System Design: ☑</td></tr>
<tr><td>Priority:</td><td>High</td></tr>
<tr><td>Source:</td><td>NKAP Bolting System Requirements</td></tr>
<tr><td>Primary Business Actor:</td><td colspan="2">Admin</td></tr>
<tr><td>Primary The system Actor:</td><td colspan="2">None</td></tr>
<tr><td>Other Participating Actors:</td><td colspan="2">None</td></tr>
<tr><td>Other Interested Stakeholders:</td><td colspan="2">None</td></tr>
<tr><td>Description:</td><td colspan="2">This use case describes the event where the admin receives the order from the supplier. The admin will check if the order is correct by comparing it with the invoice. Then they will capture the order on to the system. The use case ends when the product item quantity on hand is updated.</td></tr>
</table>

Pre-condition:	• The order should have been sent already.		
Trigger:	The admin receiving the order from the Supplier.		
Typical Course of Events:	**Actor Action:**	**system Response:**	
		Manual Action	Automated Action
	Step 1: The admin checks that the Supplier invoice corresponds with the items that the Supplier has delivered to the admin.		
	Step 2: The admin accepts the inventory order.		
	Step 3: The admin requests to receive		**Step 4**: The system invokes Use Case 2.2 **Search Supplier**.

<table>
<tr><td></td><td>the supplier order.</td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td>Step 5: The system responds by loading the search criteria results with the following controls:

<table>
<tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Supplier</td><td>Heading</td><td>None</td></tr>
<tr><td>Search:</td><td>Label</td><td>This is the label to prompt the user to enter a search query.</td></tr>
<tr><td>Search</td><td>Textbox</td><td>None</td></tr>
<tr><td>Search icon</td><td>Icon button</td><td>None</td></tr>
<tr><td>Supplier Table</td><td>Table</td><td>This is to show the Suppliers that are saved in the database in a table.</td></tr>
<tr><td>Supplier Name</td><td>Column</td><td>This is the column for the supplier names.</td></tr>
</table>
</td></tr>
</table>

			Supplier Contact Number:	Column	This is the column for the contact number of the suppliers	
			Supplier Email Address	Column	This is the column for the email address of the suppliers	
			Supplier Type	Column	This is the column for the type of supplier.	
			Supplier Balance	Column	This is the column for the balance still owed to the supplier.	
			Options	Icon	This is to show the update and delete button for each row.	
			Update	Button	None	
			Delete	Button	None	
			Capture Payment	Button	None	

			<table><tr><td>**Receive Supplier Order**</td><td>Button</td><td>None</td></tr></table>
	Step 6: The admin clicks on the "**Receive Supplier Order**" option.		**Step 7:** The system shows the "**Receive Supplier Order**" screen with the following controls:

Control Name	Control Type	Notes
Receive Supplier Order	Heading	None
Supplier Order Table	Table	None
Supplier Order ID	Column	None
Order Date Placed	Column	None
Order Date Received	Column	None
Order Date Selector	Date Selector	None
Invoice Number	Column	None
Invoice Number Textbox	Textbox	None

Invoice Date	Column	None	
Invoice Date Selector	Date Selector	None	
Invoice Total	Column	None	
Invoice Total Textbox	Textbox	None	
Items	Column	None	
Add Button	Button	None	
Save Button	Button	None	

	Step 8: The admin selects the Supplier Order he wants		
	Step 9: The admin enters the relevant order and date		

<table>
<tr><td></td><td>information</td><td></td><td></td></tr>
<tr><td></td><td>Step 10: The admin clicks on the “Add” button in the “Items” column</td><td></td><td>Step 11: The system displays a modal with the following controls:
<table>
<tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Add Invoice Items</td><td>Label</td><td>None</td></tr>
<tr><td>Product Item Name</td><td>Label</td><td>None</td></tr>
<tr><td>Name</td><td>Textbox</td><td>None</td></tr>
<tr><td>Product Item Quantity</td><td>Label</td><td>None</td></tr>
<tr><td>Quantity</td><td>Textbox</td><td>None</td></tr>
<tr><td>Product Item Cost</td><td>Label</td><td>None</td></tr>
<tr><td>Cost</td><td>Textbox</td><td>None</td></tr>
<tr><td>Add Button</td><td>Button</td><td>None</td></tr>
<tr><td>Cancel</td><td>Button</td><td>None</td></tr>
</table></td></tr>
<tr><td></td><td>Step 12: The admin enters the name ,quantity and cost of the</td><td></td><td></td></tr>
</table>

	items on the invoice		
	Step 13: The admin clicks on the add button		**Step 14:** The system saves and updates the information entered in the: **SupplierInvoiceLine Table:**
	Step 15: The admin clicks the “Save Button”		**Step 16:** The system saves and updates the information entered in the **Supplier** Table:

SupplierInvoiceLine Table:

Information to be updated:	Attribute in table	Details
Supplier InvoiceLine	**Supplier InvoiceLine**	None
Quantity_ Received	**Quantity_ Received**	None
SupplierItem_ Name	**SupplierItem_ Name**	None
LineItem_Cost	**LineItem_Cost**	None
SupplierInvoice _ID	**SupplierInvoice_ID**	None

Supplier Table:

Information to be updated:	Attribute in table	Details

Supplier_Balance	**Supplier_Balance**	None

SupplierInvoice Table:

Information to be updated:	Attribute in table	Details
Supplier Invoice_ID	**Supplier Invoice_ID**	None
Supplier Invoice_Date	**Supplier Invoice_Date**	None
Supplier Invoice_Total	**Supplier Invoice_Total**	None

Step 17: The system updates the product item quantity on hand in the **ProductItem** Table:

Information to be updated:	Attribute in table	Details
Quantity on hand	**Quantity_on_hand**	None

Alternate Courses:	**[Alt Step 2]:** The invoice doesn't correspond with the items delivered. Terminate Use Case.
	[Alt Step 12] The admin clicks on the cancel button and is taken back to the Receive Supplier order screen.
Conclusion:	This use case concludes when the information has been updated in the relevant tables.
Post-condition:	• The new items from the Supplier have been captured on to the system successfully.
Business Rules:	1. None.
Implementation Constraints and Specifications:	None
Assumptions:	None
Open Issues:	None

NKAP BOLTING SYSTEM			
DATE:	04-06-2021		
VERSION:	**1.0**		
AUTHOR(S)	S'nethemba Xulu, Jerome Amenigy		
Use Case Name:	**Capture Supplier Payment**	**Use case type**	
Use Case ID:	**2.7**	**Business Requirements:**	☐

<table>
<tr><td>Priority:</td><td colspan="2">High</td><td rowspan="2">System Analysis: ☐

System Design: ☑</td></tr>
<tr><td>Source:</td><td colspan="2">NKAP Bolting System Requirements</td></tr>
<tr><td>Primary Business Actor:</td><td colspan="3">Admin</td></tr>
<tr><td>Primary The system Actor:</td><td colspan="3">None</td></tr>
<tr><td>Other Participating Actors:</td><td colspan="3">None</td></tr>
<tr><td>Other Interested Stakeholders:</td><td colspan="3">None</td></tr>
<tr><td>Description:</td><td colspan="3">This use case describes the event where</td></tr>
<tr><td>Pre-condition:</td><td colspan="3">• This use case describes the event when the admin wants to capture a payment made to the Supplier. The admin will capture the payment and the information will be added to the Supplier table and the Supplier Payment table.</td></tr>
<tr><td>Trigger:</td><td colspan="3">The admin requests to capture a payment made to the Supplier.</td></tr>
<tr><td rowspan="3">Typical Course

of Events:</td><td rowspan="2">Actor Action:</td><td colspan="2">system Response:</td></tr>
<tr><td>Manual Action</td><td>Automated Action</td></tr>
<tr><td>Step 1: The admin requests to capture a payment</td><td></td><td>Step 2: The system invokes Use Case 2.2 Search Supplier.</td></tr>
</table>

<table>
<tr><td></td><td>made to the supplier.</td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td>Step 3: The system responds by loading the search criteria results with the following controls:
<table>
<tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Supplier</td><td>Heading</td><td>None</td></tr>
<tr><td>Search:</td><td>Label</td><td>This is the label to prompt the user to enter a search query.</td></tr>
<tr><td>Search</td><td>Textbox</td><td>None</td></tr>
<tr><td>Search icon</td><td>Icon button</td><td>None</td></tr>
<tr><td>Supplier Table</td><td>Table</td><td>This is to show the Suppliers that are saved in the database in a table.</td></tr>
<tr><td>Supplier Name</td><td>Column</td><td>This is the column for the supplier names.</td></tr>
</table>
</td></tr>
</table>

Supplier Contact Number:	Column	This is the column for the contact number of the suppliers
Supplier Email Address	Column	This is the column for the email address of the suppliers
Supplier Type	Column	This is the column for the type of supplier.
Supplier Balance	Column	This is the column for the balance still owed to the supplier.
Options	Icon	This is to show the update and delete button for each row.
Update	Button	None
Delete	Button	None
Receive Supplier Order	Button	None

<table>
<tr>
<td></td>
<td></td>
<td></td>
<td>
<table>
<tr><td>Capture Payment</td><td>Button</td><td>None</td></tr>
</table>
</td>
</tr>
<tr>
<td></td>
<td>Step 4: The admin will click on the “Capture Payment” button.</td>
<td></td>
<td>
Step 5: The system will show the “Capture Payment” screen with the following controls:
<table>
<tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Capture Payment</td><td>Heading</td><td>None</td></tr>
<tr><td>Supplier Name:</td><td>Label</td><td>This is the label for the name of the supplier.</td></tr>
<tr><td>Name</td><td>Textbox</td><td>This texbox is already filled with the information of the current supplier. (inactive textbox)</td></tr>
<tr><td>Supplier Contact Number:</td><td>Label</td><td>This is the label for the contact number of the supplier</td></tr>
</table>
</td>
</tr>
</table>

			Contact Number	Textbox	This texbox is already filled with the information of the current supplier. (inactive textbox)
			Email Address:	Label	This is the label for the email address of the supplier
			Email	Textbox	This texbox is already filled with the information of the current supplier. (inactive textbox)
			Supplier Address Line 1:	Label	This is the label for the address line 1 of the supplier
			Address Line	Textbox	This texbox is already

					filled with the information of the current supplier. (inactive textbox)
			Supplier Address Line 2:	Label	This is the label for the address line 2 of the supplier
			Address Line	Textbox	This texbox is already filled with the information of the current supplier. (inactive textbox)
			Supplier Address Line 3:	Label	This is the label for the address line 3 of the supplier
			Address Line	Textbox	This texbox is already filled with the

					information of the current supplier. (inactive textbox)
			Supplier City/Town:	Label	This is the label for the city/town of the supplier
			City	Textbox	This texbox is already filled with the information of the current supplier. (inactive textbox)
			Supplier Postal code:	Label	This is the label for the postal code of the supplier
			Postal Code	Textbox	This texbox is already filled with the information of the current supplier.

					(inactive textbox)
			Supplier Type:	Label	This is the label for the supplier type of the supplier
			Type	Dropdown	This dropdown is already filled with the information of the current supplier. (inactive drop down)
			Capture Payment	Label	None
			Capture payment	Textbox	None
			Add	Button	None
			Cancel	Button	This button is used when the admin decides to terminate the process of adding a

<table>
<tr><td></td><td></td><td></td><td><table><tr><td></td><td></td><td>new supplier.</td></tr></table></td></tr>
<tr><td></td><td>Step 6: The admin will add the amount they have paid to the Supplier.</td><td></td><td>Step 7: The system will capture and validate the information. [Alt]</td></tr>
<tr><td></td><td></td><td></td><td>Step 8: The system will take the full amount of the Supplier_Balance from the Supplier table and subtract it with the amount added.</td></tr>
<tr><td></td><td></td><td></td><td>Step 9: The system adds the information to the Supplier Table:
<table><tr><th>Information to be added:</th><th>Attribute in table</th><th>Details</th></tr><tr><td>Supplier_ Balance</td><td>Supplier_ Balance</td><td>None</td></tr></table>
SupplierPayment table:
<table><tr><th>Informatio n to be added:</th><th>Attribute in table</th><th>Detail s</th></tr></table></td></tr>
</table>

<table>
<tr><td></td><td></td><td></td><td><table>
<tr><td>Supplier Payment_ID</td><td>Supplier Payment_ID</td><td>None</td></tr>
<tr><td>Supplier_ Date</td><td>Supplier_ Date</td><td>None</td></tr>
<tr><td>Supplier Payment_ Amount</td><td>Supplier Payment_ Amount</td><td>None</td></tr>
<tr><td>Supplier_ID</td><td>Supplier_ID</td><td>None</td></tr>
</table></td></tr>
<tr><td>Alternate COURSES:</td><td colspan="3">[Alt Step 7]: Validation failed. Outline the area where the validation occurred.</td></tr>
<tr><td>Conclusion:</td><td colspan="3">The use case concludes when the system has notified the admin that the information has been captured successfully.</td></tr>
<tr><td>Post-condition:</td><td colspan="3">• The new payment to the supplier has been added in the relevant tables.</td></tr>
<tr><td>Business Rules:</td><td colspan="3">1. None</td></tr>
<tr><td>Implementation Constraints and Specifications:</td><td colspan="3">None</td></tr>
<tr><td>Assumptions:</td><td colspan="3">None</td></tr>
<tr><td>Open Issues:</td><td colspan="3">None</td></tr>
</table>

2.1.3. USER SUBSYSTEM

<table>
<tr><th colspan="5">NKAP BOLTING SYSTEM</th></tr>
<tr><td>DATE:</td><td colspan="4">03-06-21</td></tr>
<tr><td>VERSION:</td><td colspan="4">1.0</td></tr>
<tr><td>AUTHOR(S):</td><td colspan="4">Divya Bagratee</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Register New Customer</td><td colspan="2">USE CASE TYPE</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">3.1</td><td colspan="2" rowspan="3">Business Requirements: ☐
System Analysis: ☐
System Design: ☑</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">High</td></tr>
<tr><td>SOURCE:</td><td colspan="2">NKAP Bolting System Requirements</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR:</td><td colspan="4">Customer</td></tr>
<tr><td>PRIMARY THE SYSTEM ACTOR:</td><td colspan="4">None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="4">None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="4">None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="4">This use case describes the event where a Customer wants to registers on the system. The system will prompt the new Customer to provide their details by displaying the Register New Customer Screen. The system will validate the registration information. Once the system has successfully validated the Information, it will be captured and stored in the <u>User Entity</u> and the <u>Customer Entity</u> respectively. The use case ends when the Customer has successfully registered on the system.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="4">• The customer attempting to register must not already exist on the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="4">A new customer attempts to register on the system.</td></tr>
<tr><td rowspan="3">TYPICAL COURSE OF EVENTS:</td><td rowspan="2">ACTOR ACTION:</td><td colspan="3">SYSTEM RESPONSE:</td></tr>
<tr><td colspan="2">Manual Action</td><td>Automated Action</td></tr>
<tr><td>Step 1: A new customer wants to register</td><td colspan="2"></td><td></td></tr>
</table>

<table>
<tr><td></td><td>on the system.</td><td></td><td></td></tr>
<tr><td></td><td>Step 2: The customer clicks the User Icon on the navigation bar.</td><td></td><td>Step 3: The system displays the login dropdown menu item</td></tr>
<tr><td></td><td>Step 4: The customer clicks the “login” dropdown menu item and thereafter clicks the “Register” button on the login screen.</td><td></td><td>Step 5: The system displays the register new Customer Modal with the following html controls and enables the input controls:
<table>
<tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Register New Customer</td><td>Modal Heading</td><td>None</td></tr>
<tr><td>Go Back</td><td>Button</td><td>Enables the Customer to return to the previous screen</td></tr>
<tr><td>Title</td><td>Label</td><td>Hidden. Displayed when the textbox is selected.</td></tr>
<tr><td>Title</td><td>Dropdown</td><td>Enables customer to select their title</td></tr>
<tr><td>Name</td><td>Label</td><td>Hidden. Displayed when the dropdown is selected.</td></tr>
<tr><td>Name</td><td>Textbox</td><td>Enables customer to enter their name</td></tr>
<tr><td>Surname</td><td>Label</td><td>Hidden. Displayed when the textbox is selected.</td></tr>
<tr><td>Surname</td><td>Textbox</td><td>Enables customer to enter their Surname</td></tr>
<tr><td>Cellphone Number</td><td>Label</td><td>Hidden. Displayed when the textbox is selected.</td></tr>
<tr><td>Cellphone Number</td><td>Textbox</td><td>Enables customer to enter their Cellphone Number</td></tr>
</table></td></tr>
</table>

Email Address	Label	Hidden. Displayed when the textbox is selected.
Email Address	Textbox	Enables customer to enter their Email Address
VAT Registration Number	Label	Label for VAT Registration Number
VAT Registration Number	Textbox	Enables customer to enter their VAT Registration Number
Business Name	Label	Hidden. Displayed when the textbox is selected.
Business Name	Textbox	Enables customer to enter their Business Name
Next	Button	Enables Customer to proceed with registration.

Step 6: The customer inputs the requested details and clicks on the “Next” button

Step 7: The system validates the captured customer information according to the validation requirements of the **Customer Entity**:

Input Name	Attribute in Entity	Validation Requirements
Title	Title_ID	• Required
Name	Customer_Name	• Required • Max 30 characters • Alphabet only
Surname	Customer_Surname	• Required • Max 30 characters • Alphabet only
Cellphone Number	Customer_CellPhoneNumber	• Required • Max 10 characters

		• Numeric characters only
Email Address	Customer_ EmailAddress	• Required • Max 100 characters • Must contain “@” and domain extension
VAT Registration Number	Customer_ VATReg	• Max 10 characters • Numeric characters only
Business Name	Customer_ BusinessName	• Max 50 characters

Step 8: The system displays the register new Customer Modal with the following html controls and enables the input controls:

Input Name	Control Type	Notes
User Details	Modal Heading	None
Go Back	Button	Enables customer to return to previous screen.
Username	Label	Hidden. Displayed when the textbox is selected.
Username	Textbox	Enables customer to enter their Username
Password	Label	Hidden. Displayed when the textbox is selected.
Password	Textbox	Enables customer to enter their Password
Confirm Password	Label	Hidden. Displayed when the textbox is selected.

<table>
<tr>
<td></td>
<td></td>
<td></td>
<td>
<table>
<tr><td>Confirm Password</td><td>Textbox</td><td>Enables customer to enter their Confirmed Password</td></tr>
<tr><td>Register</td><td>Button</td><td>Button when clicked will proceed with the registration process</td></tr>
</table>
</td>
</tr>
<tr>
<td></td>
<td>Step 9: The customer inputs the requested details and clicks on the “Register” button.</td>
<td></td>
<td>
Step 10: The system validates the captured information according to the validation requirements of the User Entity:
<table>
<tr><th>Input Name</th><th>Attribute in Entity</th><th>Validation Requirements</th></tr>
<tr><td>Username</td><td>User_Username</td><td>Required
Max 50 characters
Alphanumeric characters only</td></tr>
<tr><td>Password</td><td>User_Password</td><td>Required
Max 50 characters
Alphabet characters only</td></tr>
</table>
</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>
Step 11: The system displays a Confirm Details modal:
<table>
<tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Confirm Details</td><td>Modal Heading</td><td>None</td></tr>
<tr><td>Is your captured information correct?</td><td>Label</td><td>Label to confirm the captured information is correct</td></tr>
<tr><td>Cancel</td><td>Button</td><td>Button clicked when the customer does not confirm their information.</td></tr>
<tr><td>Confirm</td><td>Button</td><td>Button clicked when the customer does confirm their information.</td></tr>
</table>
</td>
</tr>
</table>

<table>
<tr>
<td></td>
<td>Step 12: The customer confirms their details</td>
<td></td>
<td>
Step 13: The system uses a SQL insert query to save the new customer information in the <u>Customer Table</u>:
<table>
<tr><th>Details to be Saved</th><th>Attribute in Table</th><th>Notes</th></tr>
<tr><td>Customer ID</td><td>Customer_ID</td><td>The system generates a new Customer_ID by retrieving the last Customer_ID in the Customer table and incrementing it by one.</td></tr>
<tr><td>Title</td><td>Title_ID</td><td>None</td></tr>
<tr><td>Name</td><td>Customer_Name</td><td>None</td></tr>
<tr><td>Surname</td><td>Customer_ Surname</td><td>None</td></tr>
<tr><td>Cellphone Number</td><td>Customer_ CellphoneNumber</td><td>None</td></tr>
<tr><td>Email Address</td><td>Customer_ EmailAddress</td><td>None</td></tr>
<tr><td>VAT Registration Number</td><td>Customer_ VATReg</td><td>None</td></tr>
<tr><td>Business Name</td><td>Customer_ BusinessName</td><td>None</td></tr>
</table>
And the <u>User Table</u>:
<table>
<tr><th>Details to be saved</th><th>Attribute in table</th><th>Notes</th></tr>
<tr><td>User ID</td><td>User_ID</td><td>The system generates a new User_ID by retrieving the last User_ID in the User table and incrementing it by one.</td></tr>
</table>
</td>
</tr>
</table>

Username	User_ Username	Label to confirm the captured information is correct
Password	User_ Password	Button clicked when the customer does not confirm their information.
UserRole ID	UserRole_ID	The system sets the UserRole_ID to "customer"

Step 14: The system saves the new customer details in the relevant entities and displays a success modal:

Input Name	Control Type	Notes
Success	Modal Heading	None
You have successfully registered	Label	Label to indicate the customer has registered successfully
Ok	Button	None

ALTERNATE COURSES:

ALT STEP 7 and 10: The system fails to validate the inputted information and displays a Validation Error Modal.

Input Name	Control Type	Notes
Validation Error	Modal Heading	None
Entered details are in the incorrect format	Label	Label to indicate that the inputted details do not fulfil the validation requirements
Ok	Button	None

ALT STEP 12: The customer does not confirm their details and clicks the "cancel" button. **Terminate Use Case.**

<table>
<tr><td></td><td></td></tr>
<tr><td>CONCLUSION:</td><td>ALT STEP 14: The system fails to create the new customer and save the details in the User entity and Customer entity and displays an error modal. Terminate Use Case.
<table>
<tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Error</td><td>Modal Heading</td><td>None</td></tr>
<tr><td>An unexpected error occurred when saving the customer.</td><td>Label</td><td>Label to indicate that the customer has failed to register.</td></tr>
<tr><td>Ok</td><td>Button</td><td>None</td></tr>
</table></td></tr>
<tr><td>POST-CONDITION:</td><td>• The Customer is able to log in to their account on the NKAP Bolting System</td></tr>
<tr><td>BUSINESS RULES:</td><td>1. The Customer cannot already exist on the system. (They cannot register twice).</td></tr>
<tr><td>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:</td><td>None</td></tr>
<tr><td>ASSUMPTIONS:</td><td>None</td></tr>
<tr><td>OPEN ISSUES:</td><td>None</td></tr>
</table>

<table>
<tr><th colspan="3">NKAP BOLTING SYSTEM</th></tr>
<tr><td>DATE:</td><td colspan="2">03-06-21</td></tr>
<tr><td>VERSION:</td><td colspan="2">1.0</td></tr>
<tr><td>AUTHOR(S):</td><td colspan="2">Divya Bagratee</td></tr>
<tr><td>USE CASE NAME:</td><td>3.2</td><td>USE CASE TYPE</td></tr>
<tr><td>USE CASE ID:</td><td>Search Customer</td><td>Business Requirements: ☐</td></tr>
<tr><td>PRIORITY:</td><td>High</td><td>System Analysis: ☐</td></tr>
<tr><td>SOURCE:</td><td>NKAP Bolting System Requirements</td><td>System Design: ☑</td></tr>
</table>

<table>
<tr><td>PRIMARY BUSINESS ACTOR:</td><td colspan="3">Admin</td></tr>
<tr><td>PRIMARY THE SYSTEM ACTOR:</td><td colspan="3">None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">The Use Case describes the event where an Admin wants to search for a Customer on the system. The Admin will select the “User” menu option on the navigation bar and click the “Search Customer” dropdown menu item. The system will display the Search Customer Screen and request the search criteria from the admin. The admin will enter the search criteria and the system will search for a matching record in the <u>Customer Entity</u>. The system will display the customer which matched the search criterion entered by the admin in a datagrid view. The admin will select the customer they want to view and the use case ends when the system successfully displays the searched customer’s information.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">The customer must exist on the system</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">An admin wants to search for a customer on the system.</td></tr>
<tr><td rowspan="4">TYPICAL COURSE OF EVENTS:</td><td rowspan="2">ACTOR ACTION:</td><td colspan="2">SYSTEM RESPONSE:</td></tr>
<tr><td>Manual Action</td><td>Automated Action</td></tr>
<tr><td>Step 1: An admin wants to search for a customer on the system.</td><td></td><td></td></tr>
<tr><td>Step 2: The admin selects the User menu item in the navigation bar and thereafter selects the “Search Customer” drop down menu option.</td><td></td><td>Step 3: The system displays the Search Customer Modal with the following html components and enables the input controls:
<table>
<tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Search Customer</td><td>Modal Heading</td><td>None</td></tr>
<tr><td>Search Criteria</td><td>Textbox</td><td>Enables admin to enter the search criteria</td></tr>
<tr><td>Search</td><td>Button</td><td>None</td></tr>
</table></td></tr>
</table>

<table>
<tr><td></td><td>Step 4: The admin enters the search criteria and clicks the “search” button.</td><td></td><td>Step 5: The system searches for a matching record in the <u>Customer Entity</u> using a SQL read query.</td></tr>
<tr><td></td><td></td><td></td><td>Step 6: The system displays the record(s) matching the search criteria provided in a datagrid view with the following components:
<table>
<tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Customer</td><td>Modal Heading</td><td>None</td></tr>
<tr><td>Customer Name</td><td>Table Column1</td><td>Column in the datagrid view.</td></tr>
<tr><td>Business Name</td><td>Table Column 2</td><td>Column in the datagrid view.</td></tr>
<tr><td>Action</td><td>Table Column 3</td><td>Column in the datagrid view.</td></tr>
<tr><td>Vertical Elipsis & Pencil</td><td>Buttons</td><td>Buttons when clicked display the options to update and delete the customer.</td></tr>
</table></td></tr>
<tr><td></td><td>Step 7: The admin selects the customer they want to view by clicking the Customer Name.</td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td>Step 8: The system displays the “View Customer” Modal with the following html components:
<table>
<tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>View Customer</td><td>Modal Headin g</td><td>None</td></tr>
</table></td></tr>
</table>

Title	Label	Label for title
Name	Label	Label for Name
Surname	Label	Label for Surname
Cellphone Number	Label	Label for Cellphone Number
Email Address	Label	Label for Email Address
VAT Registration Number	Label	Label for VAT Registration Number
Business Name	Label	Label for Business Name
Go Back	Button	Allows Admin to return to previous screen when clicked

ALTERNATE COURSES:

ALT STEP 6: No Users matching the criteria entered by the admin is found and the system displays an error modal. **Terminate Use Case.**

Input Name	Control Type	Notes
Error	Modal Heading	None
No customers matching the search criteria was found.	Label	Label displayed when no matching records were found in the database.
Ok	Button	None

CONCLUSION:	The use case concludes when the system successfully displayed the searched Customer's Information.
POST-CONDITION:	• The Admin can view the searched Customer's Information
BUSINESS RULES:	1. Only an Admin can search for a Customer

IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	None
ASSUMPTIONS:	None
OPEN ISSUES:	None

<table>
<tr><td>DATE:</td><td colspan="3">03-06-21</td></tr>
<tr><td>VERSION:</td><td colspan="3">1.0</td></tr>
<tr><td>AUTHOR(S)</td><td colspan="3">Divya Bagratee</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">3.3</td><td>USE CASE TYPE</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">Update Customer</td><td rowspan="3">Business Requirements: ☐
System Analysis: ☐
System Design: ☑</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">High</td></tr>
<tr><td>SOURCE:</td><td colspan="2">NKAP Bolting System Requirements</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR:</td><td colspan="3">Admin</td></tr>
<tr><td>PRIMARY THE SYSTEM ACTOR:</td><td colspan="3">None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">The Use Case describes the event where a Customer calls in to request their account information to be updated. The admin will select the User Option on the navigation bar and click the “Search Customer” dropdown menu item. The system will enable the admin to edit the customer information retrieved from the <u>Customer Entity.</u> The system will validate and confirm the changes. The use case ends when the system has successfully updated the Customer’s information.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The customer must exist on the system</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">A customer calls in to request their account information to be updated.</td></tr>
<tr><td rowspan="3">TYPICAL COURSE OF EVENTS:</td><td rowspan="2">ACTOR ACTION:</td><td colspan="2">SYSTEM RESPONSE:</td></tr>
<tr><td>Manual Action</td><td>Automated Action</td></tr>
<tr><td>Step 1: A customer calls in to update their information</td><td>Step 2: The admin requests the Customer’s Name.</td><td></td></tr>
<tr><td></td><td>Step 3: The customer provides</td><td>Step 4: The admin searches for the</td><td>Step 5: The system invokes Use Case 3.2 Search Customer.</td></tr>
</table>

<table>
<tr><td></td><td>their name.</td><td>Customer Profile.</td><td></td></tr>
<tr><td></td><td></td><td>Step 6: The admin clicks the “Update” Button</td><td>Step 7: The system displays the Update Customer Modal with the following html components and enables the input controls:
<table>
<tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Update Customer</td><td>Modal Heading</td><td>None</td></tr>
<tr><td>Title</td><td>Label</td><td>Label for title</td></tr>
<tr><td>Title</td><td>Textbox</td><td>Title_ID is retrieved from the <u>Client Entity</u></td></tr>
<tr><td>Name</td><td>Label</td><td>Label for Name</td></tr>
<tr><td>Name</td><td>Textbox</td><td>Customer_Name is retrieved from the <u>Client Entity</u></td></tr>
<tr><td>Surname</td><td>Label</td><td>Label for Surname</td></tr>
<tr><td>Surname</td><td>Textbox</td><td>Customer_Surname is retrieved from the <u>Client Entity</u></td></tr>
<tr><td>Cellphone Number</td><td>Label</td><td>Label for Cellphone Number</td></tr>
<tr><td>Cellphone Number</td><td>Textbox</td><td>Customer_ CellphoneNumber is retrieved from the <u>Client Entity</u></td></tr>
<tr><td>Email Address</td><td>Label</td><td>Label for Email Address</td></tr>
<tr><td>Email Address</td><td>Textbox</td><td>Customer_ EmailAddress is retrieved from the <u>Client Entity</u></td></tr>
<tr><td>VAT Registration Number</td><td>Label</td><td>Label for VAT Registration Number</td></tr>
<tr><td>VAT Registration Number</td><td>Textbox</td><td>Customer_VATReg is retrieved from the <u>Client Entity</u></td></tr>
<tr><td>Business Name</td><td>Label</td><td>Label for Business Name</td></tr>
<tr><td>Business Name</td><td>Textbox</td><td>Customer_ BusinessName</td></tr>
</table>
</td></tr>
</table>

<table>
<tr>
<td></td>
<td></td>
<td></td>
<td>
<table>
<tr><td></td><td></td><td>is retrieved from the <u>Client Entity</u></td></tr>
<tr><td>Update</td><td>Button</td><td>Button which enables the customer to update the their information when clicked</td></tr>
<tr><td>Cancel</td><td>Button</td><td>None</td></tr>
<tr><td></td><td></td><td></td></tr>
</table>
</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Step 8: The admin requests the Customer Information from the customer.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Step 9: The Customer Provides their information.</td>
<td>Step 10: The admin updates the information and clicks the “Update” button.</td>
<td>Step 11: The system validates the customer information according to the validation requirements of the <u>Customer Entity</u>:
<table>
<tr><th>Input Name</th><th>Attribute in Entity</th><th>Validation Requirements</th></tr>
<tr><td>Title</td><td>Title_ID</td><td>Required</td></tr>
<tr><td>Name</td><td>Customer_Name</td><td>Required
Max 30 characters
Alphabet only</td></tr>
<tr><td>Surname</td><td>Customer_Surname</td><td>Required
Max 30 characters
Alphabet only</td></tr>
<tr><td>Cellphone Number</td><td>Customer_CellPhoneNumber</td><td>Required
Max 10 characters
Numeric characters only</td></tr>
<tr><td>Email Address</td><td>Customer_</td><td>Required</td></tr>
</table>
</td>
</tr>
</table>

<table>
<tr>
<td></td>
<td></td>
<td></td>
<td>
<table>
<tr><td></td><td>EmailAddress</td><td>Max 100 characters
Must contain “@” and domain extension</td></tr>
<tr><td>VAT Registration Number</td><td>Customer_VATReg</td><td>Max 10 characters
Numeric characters only</td></tr>
<tr><td>Business Name</td><td>Customer_BusinessName</td><td>Max 50 characters</td></tr>
</table>
</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>
Step 12: The system requests the admin to confirm the changes and displays a confirmation modal:
<table>
<tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Confirm Update</td><td>Modal Heading</td><td>None</td></tr>
<tr><td>Are you sure you want to update the customer information?</td><td>Label</td><td>Label to confirm the update of the customer’s information</td></tr>
<tr><td>Cancel</td><td>Button</td><td>Button clicked when the customer/admin does not confirm the update.</td></tr>
<tr><td>Confirm</td><td>Button</td><td>Button clicked when the customer/admin does confirm the update.</td></tr>
</table>
</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Step 13: The admin requests confirmation for the update from the customer.</td>
<td></td>
</tr>
</table>

<table>
<tr>
<td></td>
<td>Step 14: The Customer confirms the update.</td>
<td>Step 15: The Admin clicks the “Confirm” Button.</td>
<td>Step 16: The system uses a SQL update query to save the updated information in the <u>Customer Entity</u> as follows:
<table>
<tr><th>Details to be Updated</th><th>Attribute in Table</th><th>Notes</th></tr>
<tr><td>Customer ID</td><td>Customer_ID</td><td>Cannot be Updated (stays the same)</td></tr>
<tr><td>Title</td><td>Title_ID</td><td>Can be updated</td></tr>
<tr><td>Name</td><td>Customer_Name</td><td>Can be updated</td></tr>
<tr><td>Surname</td><td>Customer_ Surname</td><td>Can be updated</td></tr>
<tr><td>Cellphone Number</td><td>Customer_ CellphoneNumber</td><td>Can be updated</td></tr>
<tr><td>Email Address</td><td>Customer_ EmailAddress</td><td>Can be updated</td></tr>
<tr><td>VAT Registration Number</td><td>Customer_ VATReg</td><td>Can be updated</td></tr>
<tr><td>Business Name</td><td>Customer_ BusinessName</td><td>Can be updated</td></tr>
</table>
</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Step 17: The system successfully updated the customer’s information and displays a success modal:
<table>
<tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Success</td><td>Modal Heading</td><td>None</td></tr>
<tr><td>Customer information was successfully updated.</td><td>Label</td><td>Label to indicate the customer’s information was successfully updated.</td></tr>
<tr><td>Ok</td><td>Button</td><td>None</td></tr>
</table>
</td>
</tr>
</table>

<table>
<tr><td></td><td><table><tr><td></td><td>Step 18: The admin informs the customer that their information was successfully updated.</td><td></td></tr></table></td></tr>
<tr><td rowspan="3">ALTERNATE COURSE:</td><td>ALT STEP 14: The Customer does not confirm the update. Terminate Use Case.</td></tr>
<tr><td>ALT STEP 11: The system fails to validate the updated information and displays a Validation Error Modal.
<table>
<tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Validation Error</td><td>Modal Heading</td><td>None</td></tr>
<tr><td>The information entered is in the incorrect format. Please try again.</td><td>Label</td><td>None.</td></tr>
<tr><td>Ok</td><td>Button</td><td>None</td></tr>
</table></td></tr>
<tr><td>ALT STEP 17: The system fails to update the customer information and displays an error modal.
<table>
<tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Error</td><td>Modal Heading</td><td>None</td></tr>
<tr><td>There was an error when trying to update the customer information.</td><td>Label</td><td>Label to indicate the system failed to update the customer’s information.</td></tr>
<tr><td>Ok</td><td>Button</td><td>None</td></tr>
</table></td></tr>
<tr><td>CONCLUSION:</td><td>The use case concludes when the system successfully updates Customer’s Information</td></tr>
<tr><td>POST-CONDITION:</td><td>• The Customer’s Information has been updated</td></tr>
<tr><td>BUSINESS RULES:</td><td>1. The Customer must be logged in to update their own details.
2. The admin must be logged in to update a customer’s details.</td></tr>
</table>

IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	None
ASSUMPTIONS:	None
OPEN ISSUES:	None

NKAP BOLTING SYSTEM			
DATE:	**03-06-21**		
VERSION:	**1.0**		
AUTHOR(S)	**Divya Bagratee**		
USE CASE NAME:	**3.4**	**USE CASE TYPE**	
USE CASE ID:	**Delete Customer**	**Business Requirements:**	☐
PRIORITY:	High	**System Analysis:**	☐
SOURCE:	NKAP Bolting System Requirements	**System Design:**	☑
PRIMARY BUSINESS ACTOR:	Admin		
PRIMARY THE SYSTEM ACTOR:	None		
OTHER PARTICIPATING ACTORS:	None		
OTHER INTERESTED STAKEHOLDERS:	None		
DESCRIPTION:	The Use Case describes the event where a customer calls in to request their account to be deleted. The Admin will select the User menu option on the navigation bar and click the “Search Customer” dropdown menu item. The system will retrieve the Customer Information **Customer Entity.** The system will confirm the deletion.		

<table>
<tr><td></td><td colspan="3">The use case ends when the system has successfully deleted the Customer’s Account</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">The customer must exist on the system
The admin must be logged in</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">A customer calls in to request account deletion</td></tr>
<tr><td rowspan="3">TYPICAL COURSE OF EVENTS:</td><td rowspan="2">ACTOR ACTION:</td><td colspan="2">SYSTEM RESPONSE:</td></tr>
<tr><td>Manual Action</td><td>Automated Action</td></tr>
<tr><td>Step 1: A customer requests the admin to delete their account.</td><td>Step 2: The admin requests the Customer’s Name</td><td></td></tr>
<tr><td></td><td>Step 3: The customer provides their name</td><td>Step 4: The admin searches for the Customer Profile.</td><td>Step 5: The system invokes Use Case 3.2 Search Customer.</td></tr>
<tr><td></td><td></td><td>Step 6: The admin clicks the “Delete” Button.</td><td>Step 7: The system requests confirmation for the deletion and displays a confirmation modal:
<table>
<tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Confirm Deletion</td><td>Modal Heading</td><td>None</td></tr>
<tr><td>Are you sure you want to delete this customer?</td><td>Label</td><td>Label to request confirmation of deletion.</td></tr>
<tr><td>Confirm</td><td>Button</td><td>Button when clicked will proceed with the deletion.</td></tr>
<tr><td>Cancel</td><td>Button</td><td>Button when clicked will cancel the deletion.</td></tr>
</table></td></tr>
<tr><td></td><td></td><td>Step 8: The admin requests confirmation for the Deletion</td><td></td></tr>
</table>

<table>
<tr><td></td><td></td><td>from the Customer.</td><td></td></tr>
<tr><td></td><td>Step 9: The Customer confirms the deletion.</td><td>Step 10: The Admin clicks the “Confirm” Button.</td><td>Step 11: The system makes use of a SQL Delete query to delete the customer information from the <u>Customer Entity:</u>
<table>
<tr><th>Attribute in Entity</th></tr>
<tr><td>Title_ID</td></tr>
<tr><td>Customer_Name</td></tr>
<tr><td>Customer_Surname</td></tr>
<tr><td>Customer_CellPhoneNumber</td></tr>
<tr><td>Customer_EmailAddress</td></tr>
<tr><td>Customer_VATReg</td></tr>
<tr><td>Customer_BusinessName</td></tr>
</table>
And the <u>User Entity</u>:
<table>
<tr><th>Attribute in table</th></tr>
<tr><td>User_ID</td></tr>
<tr><td>User_Username</td></tr>
<tr><td>User_Password</td></tr>
</table></td></tr>
<tr><td></td><td></td><td></td><td>Step 12: The system displays a success modal to inform the admin that the customer was successfully deleted.
<table>
<tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Success</td><td>Modal Heading</td><td>None</td></tr>
<tr><td>The customer was successfully deleted.</td><td>Label</td><td>Label to indicate the system successfully deleted the customer.</td></tr>
<tr><td>Ok</td><td>Button</td><td>None</td></tr>
</table></td></tr>
<tr><td></td><td></td><td>Step 13: The admin informs the customer that their account was successfully deleted</td><td></td></tr>
</table>

ALTERNATE COURSES:	ALT STEP 9: The customer does not confirm the deletion. Terminate Use Case. ALT STEP 12: The system fails to delete the customer and displays an error modal. **Input Name** / **Control Type** / **Notes**: **Error** / Modal Heading / None **There was an error when trying to delete the customer.** / Label / Label to indicate the system failed to delete the customer. **Ok** / Button / None
CONCLUSION:	The use case concludes when the system successfully deletes the Customer's Account.
POST-CONDITION:	• The Customer's Information has been deleted.
BUSINESS RULES:	1. The admin must be logged in to delete a customer from the system.
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	None
ASSUMPTIONS:	None
OPEN ISSUES:	None

NKAP BOLTING SYSTEM			
DATE:	**03-06-21**		
VERSION:	**1.0**		
AUTHOR(S)	**Divya Bagratee**		
USE CASE NAME:	**View Profile**	**USE CASE TYPE**	
USE CASE ID:	**3.5**	**Business Requirements:**	☐
PRIORITY:	High	**System Analysis:**	☐
SOURCE:	NKAP Bolting System Requirements	**System Design:**	☑

<table>
<tr><td>PRIMARY BUSINESS ACTOR:</td><td colspan="3">Customer</td></tr>
<tr><td>PRIMARY THE SYSTEM ACTOR:</td><td colspan="3">None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case describes the event where a customer wants to view their profile. The customer will click on the “User” icon on the navigation bar and clicks the “View Profile” drop down menu item. The system will display the customer information from the <u>Customer Entity</u>. The use case ends when the system has successfully displayed the customer’s profile.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The customer must be logged in
• The customer must have an account on the system</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">A customer wants to view their profile.</td></tr>
<tr><td rowspan="5">TYPICAL COURSE OF EVENTS:</td><td rowspan="2">ACTOR ACTION:</td><td colspan="2">SYSTEM RESPONSE:</td></tr>
<tr><td>Manual Action</td><td>Automated Action</td></tr>
<tr><td>Step 1: A customer wants to view their profile.</td><td></td><td></td></tr>
<tr><td>Step 2: The customer clicks on the “User” icon on the navigation bar and clicks the “View Profile” drop down menu item.</td><td></td><td>Step 3: The system makes use of a SQL Read query to retrieve the customer information from the <u>Customer Entity.</u></td></tr>
<tr><td></td><td></td><td>Step 4: The system displays the Profile Modal with the following html components :
<table>
<tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>View Profile</td><td>Modal Heading</td><td>None</td></tr>
<tr><td>Title</td><td>Label</td><td>Label for title</td></tr>
<tr><td>Name</td><td>Label</td><td>Label for Name</td></tr>
</table></td></tr>
</table>

<table>
<tr>
<td></td>
<td>
<table>
<tr><td>Surname</td><td>Label</td><td>Label for Surname</td></tr>
<tr><td>Email Address</td><td>Label</td><td>Label for Email Address</td></tr>
<tr><td>Contact Number</td><td>Label</td><td>Label for Contact Number</td></tr>
<tr><td>VAT Registration Number</td><td>Label</td><td>Label for VAT Registration Number</td></tr>
<tr><td>Business Name</td><td>Label</td><td>Label for Business Name</td></tr>
<tr><td>Update</td><td>Button</td><td>Button which enables the customer to update their information when clicked</td></tr>
<tr><td>Delete</td><td>Button</td><td>Button which enables customer to delete their account when clicked.</td></tr>
</table>
</td>
</tr>
<tr>
<td>ALTERNATE COURSES:</td>
<td>
ALT STEP 4: The system fails to retrieve and display the Customer’s profile information and displays an error modal:
<table>
<tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Error</td><td>Modal Heading</td><td>None</td></tr>
<tr><td>There was an unexpected error when trying to display Customer Profile.</td><td>Label</td><td>Label to indicate the system failed to display the Customer Profile,</td></tr>
<tr><td>Ok</td><td>Button</td><td>None</td></tr>
</table>
</td>
</tr>
<tr><td>CONCLUSION:</td><td>The use case concludes when the system has successfully displayed the Customer’s profile.</td></tr>
<tr><td>POST-CONDITION:</td><td>• The customer can view their profile.</td></tr>
<tr><td>BUSINESS RULES:</td><td>1. The customer must be logged in to view their profile.</td></tr>
</table>

IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	None
ASSUMPTIONS:	None
OPEN ISSUES:	None

<table>
<tr><th colspan="5">NKAP BOLTING SYSTEM</th></tr>
<tr><td>DATE:</td><td colspan="4">03-06-21</td></tr>
<tr><td>VERSION:</td><td colspan="4">1.0</td></tr>
<tr><td>AUTHOR(S)</td><td colspan="4">Divya Bagratee</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Update Profile</td><td colspan="2">USE CASE TYPE</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">3.6</td><td rowspan="3" colspan="2">Business Requirements: ☐
System Analysis: ☐
System Design: ☑</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">High</td></tr>
<tr><td>SOURCE:</td><td colspan="2">NKAP Bolting System Requirements</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR:</td><td colspan="4">Customer</td></tr>
<tr><td>PRIMARY THE SYSTEM ACTOR:</td><td colspan="4">None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="4">None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="4">None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="4">This use case describes the event where a customer wants to update their profile. The customer will select the User Icon on the navigation bar and click the View Profile Drop-Down Menu Item. The system will enable the customer to update the customer information retrieved from the <u>Customer Entity.</u> The system will validate and confirm the changes. The use case ends when the system has successfully updated the Customer’s information</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="4">• The customer must be logged in.</td></tr>
<tr><td>TRIGGER:</td><td colspan="4">A customer wants to update their profile.</td></tr>
<tr><td rowspan="4">TYPICAL COURSE OF EVENTS:</td><td rowspan="2">ACTOR ACTION:</td><td colspan="3">SYSTEM RESPONSE:</td></tr>
<tr><td>Manual Action</td><td colspan="2">Automated Action</td></tr>
<tr><td>Step 1: A customer wants to update their profile information.</td><td></td><td colspan="2"></td></tr>
<tr><td>Step 2: The customer selects the User Icon on the navigation bar and</td><td></td><td colspan="2">Step 3: The system invokes Use case 3.5 View Profile.</td></tr>
</table>

<table>
<tr><td></td><td>clicks the “View Profile” Drop-Down Menu Item</td><td></td><td></td></tr>
<tr><td></td><td>Step 4: The customer clicks on the “Update” Button.</td><td></td><td>Step 5: The system displays the Update Profile Modal with the following html components and enables the input controls:
<table>
<tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Update Profile</td><td>Modal Heading</td><td>None</td></tr>
<tr><td>Title</td><td>Label</td><td>Label for title</td></tr>
<tr><td>Title</td><td>Textbox</td><td>Title_ID is retrieved from the <u>Customer Entity</u></td></tr>
<tr><td>Name</td><td>Label</td><td>Label for Name</td></tr>
<tr><td>Name</td><td>Textbox</td><td>Customer_Name is retrieved from the <u>Customer Entity</u></td></tr>
<tr><td>Surname</td><td>Label</td><td>Label for Surname</td></tr>
<tr><td>Surname</td><td>Textbox</td><td>Customer_Surname is retrieved from the <u>Customer Entity</u></td></tr>
<tr><td>Cellphone Number</td><td>Label</td><td>Label for Cellphone Number</td></tr>
<tr><td>Cellphone Number</td><td>Textbox</td><td>Customer_ CellphoneNumber is retrieved from the <u>Customer Entity</u></td></tr>
<tr><td>Email Address</td><td>Label</td><td>Label for Email Address</td></tr>
<tr><td>Email Address</td><td>Textbox</td><td>Customer_ EmailAddress is retrieved from the <u>Customer Entity</u></td></tr>
<tr><td>VAT Registration Number</td><td>Label</td><td>Label for VAT Registration Number</td></tr>
<tr><td>VAT Registration Number</td><td>Textbox</td><td>Customer_VATReg is retrieved from the <u>Customer Entity</u></td></tr>
<tr><td>Business Name</td><td>Label</td><td>Label for Business Name</td></tr>
</table>
</td></tr>
</table>

			Business Name | Textbox | Customer_ BusinessName is retrieved from the **Customer Entity**
			Update | Button | Button which enables the customer to update the their information when clicked
			Cancel | Button | None
Step 6: The customer updates their information and clicks the "Update" Button.			**Step 7:** The system validates the updated information according to the validation requirements of the **Customer Entity**:

Input Name	Attribute in Entity	Validation Requirements
Title	Title_ID	• Required
Name	Customer_ Name	• Required • Max 30 characters • Alphabet only
Surname	Customer_ Surname	• Required • Max 30 characters • Alphabet only
Cellphone Number	Customer_ CellPhoneNumber	• Required • Max 10 characters • Numeric characters only
Email Address	Customer_ EmailAddress	• Required • Max 100 characters • Must contain "@" and domain extension
VAT Registration Number	Customer_ VATReg	• Max 10 characters

		• Numeric characters only
Business Name	Customer_ BusinessName	• Max 50 characters

Step 8: The system requests the customer to confirm the update and displays a Confirmation modal:

Input Name	Control Type	Notes
Confirm Update	Modal Heading	None
Are you sure you want to update Profile?	Label	Label to confirm the update of the customer's information
Cancel	Button	Button clicked when the customer does not confirm the update.
Confirm	Button	Button clicked when the customer does confirm the update.

Step 9: The Customer clicks the "Confirm" button.

Step 10: The system uses a SQL update query to save the updated information in the <u>**Customer Entity**</u> as follows:

Details to be Updated	Attribute in Table	Notes
Customer ID	Customer_ID	Cannot be Updated (stays the same)
Title	Title_ID	Can be updated
Name	Customer_Name	Can be updated
Surname	Customer_ Surname	Can be updated
Cellphone Number	Customer_ CellphoneNumber	Can be updated
Email Address	Customer_ EmailAddress	Can be updated
VAT Registration Number	Customer_ VATReg	Can be updated

Input Name	Control Type	Notes
Business Name	Customer_ BusinessName	Can be updated

Step 11: The system successfully updates the Customer's profile and displays a success modal:

Input Name	Control Type	Notes
Success	Modal Heading	None
Customer Profile was successfully updated.	Label	Label to indicate the customer's profile was successfully updated.
Ok	Button	None

ALTERNATE COURSES:

ALT-STEP 7: The system fails to validate the updated information and displays a Validation Error Modal:

Input Name	Control Type	Notes
Validation Error	Modal Heading	None
The information entered is in the incorrect format. Please try again.	Label	None.
Ok	Button	None

ALT-STEP 9: The Customer does not confirm the update and clicks the "Cancel Button". Terminate Use Case.

ALT-STEP 11 : The system fails to update the Customer Profile and displays an error modal:

Input Name	Control Type	Notes
Error	Modal Heading	None
There was an unexpected error when trying to update the customer Profile.	Label	Label to indicate the system failed to update the customer's Profile.
Ok	Button	None

CONCLUSION:	The use case concludes when the customer profile has been successfully updated.
POST-CONDITION:	• The customer Profile is updated.
BUSINESS RULES:	1. The customer must be logged in to update their profile
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	None
ASSUMPTIONS:	None
OPEN ISSUES:	None

2.1.4. EMPLOYEE SUBSYSTEM

NKAP BOLTING SYSTEM			
DATE:	13-06-2021		
VERSION:	**1.0**		
AUTHOR(S)	Sonali Badrinath		
USE CASE NAME:	**Create Delivery Shift**	**USE CASE TYPE**	
USE CASE ID:	**4.1.**	**Business Requirements:**	☐
PRIORITY:	High	**System Analysis:**	☐
SOURCE:	NKAP Bolting System Requirements	**System Design:**	☑
PRIMARY BUSINESS ACTOR:	None		
PRIMARY THE SYSTEM ACTOR:	None		
OTHER PARTICIPATING ACTORS:	None		
OTHER INTERESTED STAKEHOLDERS:	None		
DESCRIPTION:	This use case describes the event where the admin wants to add a new delivery shift on to the system. The use case begins when the admin requests to add the delivery shift. The admin will enter all the delivery		

<table>
<tr><th colspan="5">NKAP BOLTING SYSTEM</th></tr>
<tr><td>DATE:</td><td colspan="4">03-06-21</td></tr>
<tr><td>VERSION:</td><td colspan="4">1.0</td></tr>
<tr><td>AUTHOR(S)</td><td colspan="4">Divya Bagratee</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Delete Profile</td><td colspan="2">USE CASE TYPE</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">3.7</td><td colspan="2">Business Requirements: ☐</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">High</td><td colspan="2">System Analysis: ☐</td></tr>
<tr><td>SOURCE:</td><td colspan="2">NKAP Bolting System Requirements</td><td colspan="2">System Design: ☑</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR:</td><td colspan="4">Customer</td></tr>
<tr><td>PRIMARY THE SYSTEM ACTOR:</td><td colspan="4">None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="4">None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="4">None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="4">This use case describes the event where a customer wants to delete their account. The Customer will select the User Icon on the navigation bar and click the “View Profile” dropdown menu item. The system will retrieve the Customer Information <u>Customer Entity.</u> The system will confirm the deletion. The use case ends when the system has successfully deleted the Customer’s Account.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="4">• The customer must be logged in.</td></tr>
<tr><td>TRIGGER:</td><td colspan="4">A customer wants to delete their profile.</td></tr>
<tr><td rowspan="5">TYPICAL COURSE OF EVENTS:</td><td rowspan="2">ACTOR ACTION:</td><td colspan="3">SYSTEM RESPONSE:</td></tr>
<tr><td>Manual Action</td><td colspan="2">Automated Action</td></tr>
<tr><td>Step 1: A customer wants to delete their account.</td><td></td><td colspan="2"></td></tr>
<tr><td>Step 2: The customer will select the User Icon on the navigation bar and click the “View Profile” dropdown menu item.</td><td></td><td colspan="2">Step 3: The system will invoke use case 3.5 View Profile.</td></tr>
<tr><td>Step 4: The customer will click the “Delete Button”.</td><td></td><td colspan="2">Step 5: The system will request confirmation of the deletion and display a Confirmation modal:</td></tr>
</table>

<table>
<tr>
<td></td>
<td></td>
<td></td>
<td>
<table>
<tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Confirmation</td><td>Modal Heading</td><td>None</td></tr>
<tr><td>Are you sure you want to delete your account?</td><td>Label</td><td>Label to request confirmation of deletion.</td></tr>
<tr><td>Confirm</td><td>Button</td><td>Button when clicked will proceed with the deletion.</td></tr>
<tr><td>Cancel</td><td>Button</td><td>Button when clicked will cancel the deletion.</td></tr>
</table>
</td>
</tr>
<tr>
<td></td>
<td>Step 6: The customer clicks the “confirm” button.</td>
<td></td>
<td>
Step 7: The system makes use of a SQL Delete query to delete the customer information from the <u>Customer Entity:</u>
<table>
<tr><th>Attribute in Entity</th></tr>
<tr><td>Title_ID</td></tr>
<tr><td>Customer_Name</td></tr>
<tr><td>Customer_Surname</td></tr>
<tr><td>Customer_CellPhoneNumber</td></tr>
<tr><td>Customer_EmailAddress</td></tr>
<tr><td>Customer_VATReg</td></tr>
<tr><td>Customer_BusinessName</td></tr>
</table>
And the <u>User Entity</u>:
<table>
<tr><th>Attribute in table</th></tr>
<tr><td>User_ID</td></tr>
<tr><td>User_Username</td></tr>
<tr><td>User_Password</td></tr>
</table>
</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>
Step 8: The system displays a success modal to inform the admin that the customer was successfully deleted.
<table>
<tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr>
</table>
</td>
</tr>
</table>

Success	Modal Heading	None
Customer Profile was successfully deleted.	Label	Label to indicate the system successfully deleted the customer.
Ok	Button	None

ALTERNATE COURSES:	ALT-STEP 6: The customer does not confirm the deletion and clicks the “Cancel” button. **Terminate Use Case.**
	ALT-STEP 8: The system fails to delete the Customer’s account and displays an Error Modal:

Input Name	Control Type	Notes
Error	Modal Heading	None
There was an unexpected error when trying to delete the customer account.	Label	Label to indicate the system failed to delete the customer’s account.
Ok	Button	None

CONCLUSION:	The use case concludes when the system has successfully deleted the Customer” Account.
POST-CONDITION:	• The customer no longer exists on the system.
BUSINESS RULES:	1. The customer must be logged in to delete their account.
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	None
ASSUMPTIONS:	None
OPEN ISSUES:	None

	shift details and the information will be captured in the **Shift** table. The use case ends when the delivery shift has been successfully added.		
PRE-CONDITION:	• The admin should be logged onto the system.		
TRIGGER:	The admin requesting to add a new delivery shift.		
TYPICAL COURSE OF EVENTS:	**ACTOR ACTION:**	**SYSTEM RESPONSE:**	
		Manual Action	Automated Action

<table>
<tr><th colspan="5">NKAP BOLTING SYSTEM</th></tr>
<tr><td>DATE:</td><td colspan="4">03-06-21</td></tr>
<tr><td>VERSION:</td><td colspan="4">1.0</td></tr>
<tr><td>AUTHOR(S)</td><td colspan="4">Divya Bagratee</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Add User Role</td><td colspan="2">USE CASE TYPE</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">3.8</td><td>Business Requirements:</td><td>☐</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">High</td><td>System Analysis:</td><td>☐</td></tr>
<tr><td>SOURCE:</td><td colspan="2">NKAP Bolting System Requirements</td><td>System Design:</td><td>☑</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR:</td><td colspan="4">Admin</td></tr>
<tr><td>PRIMARY THE SYSTEM ACTOR:</td><td colspan="4">None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="4">None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="4">None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="4">This use case describes the event where an Admin wants to add a User Role On the system. The admin will select the User Menu item on the navigation bar and click the “Add User Role” dropdown menu item. The system will display the Add User Role Screen and prompt the admin to provide the New User Role Information. The system will validate and save the User Role in the <u>UserRole Entity.</u> The use case ends when the system successfully adds a User Role to the system.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="4">The Admin must be logged in to the system</td></tr>
<tr><td>TRIGGER:</td><td colspan="4">An admin wants to add a User Role to the system.</td></tr>
<tr><td rowspan="4">TYPICAL COURSE OF EVENTS:</td><td rowspan="2">ACTOR ACTION:</td><td colspan="3">SYSTEM RESPONSE:</td></tr>
<tr><td>Manual Action</td><td colspan="2">Automated Action</td></tr>
<tr><td>Step 1: An admin wants to add a User Role on the system.</td><td></td><td colspan="2"></td></tr>
<tr><td>Step 2: The admin will select the User Menu</td><td></td><td colspan="2">Step 3: The system will display the Add User Role Modal with the following html controls and will enable the input controls:<table><tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr></table></td></tr>
</table>

item on the navigation bar and click the “Add User Role” dropdown menu item.

Add User Role	Heading	None
User Role Name	Label	Label for User Role Name
User Role Name	Textbox	Enables the admin to enter the User Role Name.
User Role Description	Label	Label for User Role Description
User Role Description	Textbox	Enables admin to enter the User Role description.
Add	Button	Button when clicked will proceed with the add user role process.
Cancel	Button	Button when clicked will terminate the add user role process.

Step 4: The admin provides the User Role details and clicks the “Add” button.

Step 5: The system validates the User Role details according to the validation requirements of the **User Role Entity:**

Input Name	Attribute in Entity	Validation Requirements
User Role Name	UserRole_Name	• Required • Max 50 characters • Alphabet only
User Role Description	UserRole_Descrip	• Required • Max 200 characters • Alphabet only

Step 6: The system requests confirmation of the user role details and displays a confirmation modal:

Input Name	Control Type	Notes
Confirmation	Modal Heading	None

Step 1: The admin

Are the User Role details correct?	Label	Label to request confirmation of User Role details.
Confirm	Button	Button when clicked will proceed with adding the User Role.
Cancel	Button	Button when clicked will cancel the adding of the User Role.

Step 7: The admin clicks the “confirm” button.

Step 8: The system makes use of a SQL Insert query to save the User Role in the **UserRole Entity** as follows:

Details to be Saved	Attribute in Table	Notes
User Role ID	UserRole_ID	The system generates a new UserRole_ID by retrieving the last UserRole_ID in the UserRole table and incrementing it by one.
User Role Description	UserRole_Desc	None
UserRole Name	UserRole_Name	None

Step 9: The system successfully adds the User Role to the system and displays a success modal:

Input Name	Control Type	Notes
Success	Modal Heading	None
The User Role was successfully added.	Label	Label to indicate the system successfully added the User Role.
Ok	Button	None

ALTERNATE COURSES:

ALT STEP 5: The system fails to validate the User Role details and displays a validation error modal.

Input Name	Control Type	Notes

	Validation Error	Modal Heading	None
	The information entered is in the incorrect format. Please try again.	Label	None.
	Ok	Button	None
	ALT STEP 7: The Admin clicks the "cancel" button. **Terminate Use Case.**		
	ALT STEP 9: The system fails to save the User Role details and displays an error modal:		
	Input Name	**Control Type**	**Notes**
	Error	Modal Heading	None
	There was an unexpected error when trying to add the User Role.	Label	Label to indicate the system failed to Add the User Role.
	Ok	Button	None
CONCLUSION:	The Use Case concludes when the system has successfully added the User Role.		
POST-CONDITION:	• The UserRole has been added to the UserRole Entity.		
BUSINESS RULES:	1. Only an admin can add a User Role 2. The admin must be logged in to add a User Role.		
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	None		
ASSUMPTIONS:	None		
OPEN ISSUES:	None		

	requests to add a new delivery shift.		
	Step 2: The admin will click on the "**Create Delivery Shift**" menu		**Step 3:** The system responds by loading the "**Create Delivery Shift**" modal with the following controls:

NKAP BOLTING SYSTEM			
DATE:	**03-06-21**		
VERSION:	**1.0**		
AUTHOR(S)	**Divya Bagratee**		
USE CASE NAME:	**Search User Role**	**USE CASE TYPE**	
USE CASE ID:	**3.9**	**Business Requirements:**	☐
PRIORITY:	High	**System Analysis:**	☐
SOURCE:	NKAP Bolting System Requirements	**System Design:**	☑
PRIMARY BUSINESS ACTOR:	Admin		
PRIMARY THE SYSTEM ACTOR:	None		
OTHER PARTICIPATING ACTORS:	None		
OTHER INTERESTED STAKEHOLDERS:	None		
DESCRIPTION:	This use case describes the event where an Admin wants to search for a User Role. The admin will select the User Menu item on the navigation bar and click the “Search User Role” dropdown menu item. The system will display the Search User Role Screen and prompt the admin to provide the Search Criteria. The system will retrieve and display the User Role matching the search criteria from the **UserRole Entity.** The use case ends when the system has successfully displayed the User Role.		
PRE-CONDITION:	• The User Role must exist on the system		
TRIGGER:	An admin wants to search for a User Role.		

TYPICAL COURSE OF EVENTS:

ACTOR ACTION:	SYSTEM RESPONSE: Manual Action	SYSTEM RESPONSE: Automated Action
Step 1: An admin wants to search for a user role.		
Step 2: The admin will select the User Menu item on the navigation bar and click the “Search User Role” dropdown menu item.		**Step 3:** The system will display the search user role screen with the following html components and enables the input controls:

Input Name	Control Type	Notes
Search User Role	Heading	None

	item on the navigation bar.		Control Name / Control Type / Notes

<table>
<tr><td></td><td></td><td></td><td>
<table>
<tr><td>User Role Name</td><td>Textbox</td><td>Enables the admin to enter the User Role Name.</td></tr>
<tr><td>Search</td><td>Button</td><td>Button when clicked will search for a matching record in the <u>UserRole Table.</u></td></tr>
</table>
</td></tr>
<tr><td></td><td>Step 4: The admin provides the search criteria and clicks the “Search” Button.</td><td></td><td>Step 5: The system searches for a matching record in the <u>UserRole Entity</u> by making use of a SQL read query.</td></tr>
<tr><td></td><td></td><td></td><td>Step 6: The system displays the record(s) matching the search criteria provided in a datagrid view with the following components:
<table>
<tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Search User Role</td><td>Modal Heading</td><td>None</td></tr>
<tr><td>User Role</td><td>Table Column1</td><td>Column in the datagrid view</td></tr>
<tr><td>Action</td><td>Table Column 2</td><td>Column in the datagrid view</td></tr>
<tr><td>Vertical Elipses & Pencil</td><td>Buttons</td><td>Buttons found in each row of the datagrid view, when clicked allows the admin to update or delete the user role.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr>
<td></td>
<td>Step 7: The Admin selects the User Role they wish to view by clicking on it.</td>
<td></td>
<td>Step 8: The system displays the “View User Role” modal with the following html components:
<table>
<tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>View User Role</td><td>Modal Heading</td><td>None</td></tr>
<tr><td>Go Back</td><td>Button</td><td>When clicked returns to the previous screen.</td></tr>
<tr><td>User Role</td><td>Label</td><td>Label for the UserRole_Name retrieved from the <u>UserRole Entity.</u></td></tr>
<tr><td>User Role Description</td><td>Label</td><td>Label for the UserRole_Desc retrieved from the <u>UserRole Entity.</u></td></tr>
</table>
</td>
</tr>
<tr>
<td>ALTERNATE COURSES:</td>
<td colspan="3">ALT STEP 6: No Users matching the criteria entered by the admin is found and the system displays an error modal. Terminate Use Case.
<table>
<tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Error</td><td>Modal Heading</td><td>None</td></tr>
<tr><td>No User Role’s matching the search criteria was found.</td><td>Label</td><td>Label displayed when no matching records were found in the database.</td></tr>
<tr><td>Ok</td><td>Button</td><td>None</td></tr>
</table>
</td>
</tr>
<tr><td>CONCLUSION:</td><td colspan="3">The use case concludes when the system successfully displays the User Role.</td></tr>
<tr><td>POST-CONDITION:</td><td colspan="3">• The admin is able to view the User Role</td></tr>
<tr><td>BUSINESS RULES:</td><td colspan="3">1. Only an admin can search for a User Role.
2. The User Role must exist on the system.</td></tr>
<tr><td>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:</td><td colspan="3">None</td></tr>
<tr><td>ASSUMPTIONS:</td><td colspan="3">None</td></tr>
<tr><td>OPEN ISSUES:</td><td colspan="3">None</td></tr>
</table>

<table>
<tr><th colspan="5">NKAP BOLTING SYSTEM</th></tr>
<tr><td>DATE:</td><td colspan="4">03-06-21</td></tr>
<tr><td>VERSION:</td><td colspan="4">1.0</td></tr>
<tr><td>AUTHOR(S)</td><td colspan="4">Divya Bagratee</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Update User Role</td><td colspan="2">USE CASE TYPE</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">3.10</td><td colspan="2" rowspan="3">Business Requirements: ☐
System Analysis: ☐
System Design: ☑</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">High</td></tr>
<tr><td>SOURCE:</td><td colspan="2">NKAP Bolting System Requirements</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR:</td><td colspan="4">Admin</td></tr>
<tr><td>PRIMARY THE SYSTEM ACTOR:</td><td colspan="4">None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="4">None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="4">None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="4">This use case describes the event where an Admin wants to update a User Role. The admin will select the User Menu item on the navigation bar and click the “Search User Role” dropdown menu item. The system will display the Update User Role Screen and prompt the admin to provide the Updated details. The use case ends when the system has successfully updated the User Role.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="4">• The admin must be logged in to the system.
• The User Role must exist on the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="4">An admin wants to update a User Role</td></tr>
<tr><td rowspan="4">TYPICAL COURSE OF EVENTS:</td><td rowspan="2">ACTOR ACTION:</td><td colspan="3">SYSTEM RESPONSE:</td></tr>
<tr><td colspan="2">Manual Action</td><td>Automated Action</td></tr>
<tr><td>Step 1: The Admin wants to update a User Role</td><td colspan="2"></td><td></td></tr>
<tr><td>Step 2: The admin will select the User Menu item on</td><td colspan="2"></td><td>Step 3: The system will invoke Use Case 3.9 Search User Role.</td></tr>
</table>

<table>
<tr><td></td><td>the navigation bar and click the “Search User Role” dropdown menu item.</td><td></td><td></td></tr>
<tr><td></td><td>Step 4: The admin will click the “Update” button.</td><td></td><td>Step 5: The system will display the Update User Role Modal with the following html components and enable the input controls:
<table>
<tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Update User Role</td><td>Modal Heading</td><td>None</td></tr>
<tr><td>Name</td><td>Label</td><td>Hidden. Displayed when textbox is selected.</td></tr>
<tr><td>Name</td><td>Textbox</td><td>Enables the admin to update the User Role name.</td></tr>
<tr><td>Description</td><td>Label</td><td>Hidden. Displayed when textbox is selected.</td></tr>
<tr><td>Description</td><td>Textbox</td><td>Enables the admin to update the User Role description.</td></tr>
<tr><td>Update</td><td>Button</td><td>Button when clicked proceeds with the update process.</td></tr>
<tr><td>Cancel</td><td>Button</td><td>Button when clicked terminates the Update process.</td></tr>
</table></td></tr>
<tr><td></td><td>Step 6: The admin provides the updated details and clicks the “Update” button.</td><td></td><td>Step 7: The system validates the User Role information according to the Validation requirements of the <u>UserRole Entity:</u>
<table>
<tr><th>Input Name</th><th>Attribute in Entity</th><th>Validation Requirements</th></tr>
<tr><td>User Role Name</td><td>UserRole_Name</td><td>Required
Max 50 characters
Alphabet only</td></tr>
</table></td></tr>
</table>

User Role Description	UserRole_Descrip	• Required • Max 200 characters • Alphabet only

Step 8: The system requests confirmation of the user role details and displays a confirmation modal:

Input Name	Control Type	Notes
Confirm update	Modal Heading	None
Are you sure you want to update the User Role details.	Label	Label to request confirmation of User Role details.
Confirm	Button	Button when clicked will proceed with updating of the User Role.
Cancel	Button	Button when clicked will cancel the Updating of the User Role.

Step 9: The admin clicks the “Confirm” Button.

Step 10: The system makes use of a SQL Update query to Update the User Role in the **UserRole Entity** as follows:

Details to be Updated	Attribute in Table	Notes
User Role ID	UserRole_ID	Cannot be Updated (Stays the same)
User Role Description	UserRole_Desc	Can be updated
User Role Name	UserRole_Name	Can be updated

Step 11: The system successfully updates the User Role and displays a success modal:

Input Name	Control Type	Notes

			Success	Modal Heading	None
			The User Role was successfully Updated.	Label	Label to indicate the system successfully updated the User Role.
			Ok	Button	None

ALTERNATE COURSES:

ALT-STEP 7: The system fails to Validate the Updated details and displays a Validation Error Modal:

Input Name	Control Type	Notes
Validation Error	Modal Heading	None
The information entered is in the incorrect format. Please try again.	Label	None.
Ok	Button	None

ALT-STEP 9: The admin does not confirm the update and clicks the "Cancel" button. Terminate Use Case.

ALT-STEP 11: The system fails to update the User Role and displays an error modal:

Input Name	Control Type	Notes
Error	Modal Heading	None
There was an error when trying to update the User Role.	Label	Label to indicate the system failed to update the User Role.
Ok	Button	None

CONCLUSION:	The use case concludes when the system has successfully updated the User Role.
POST-CONDITION:	• The User Role has been updated.
BUSINESS RULES:	1. Only an admin can Update a User Role. 2. The User Role must exist on the system.
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	None
ASSUMPTIONS:	None
OPEN ISSUES:	None

<table>
<tr><th colspan="4">NKAP BOLTING SYSTEM</th></tr>
<tr><td>DATE:</td><td colspan="3">03-06-21</td></tr>
<tr><td>VERSION:</td><td colspan="3">1.0</td></tr>
<tr><td>AUTHOR(S)</td><td colspan="3">Divya Bagratee</td></tr>
<tr><td>USE CASE NAME:</td><td>Delete User Role</td><td colspan="2">USE CASE TYPE</td></tr>
<tr><td>USE CASE ID:</td><td>3.11</td><td colspan="2" rowspan="3">Business Requirements: ☐
System Analysis: ☐
System Design: ☑</td></tr>
<tr><td>PRIORITY:</td><td>High</td></tr>
<tr><td>SOURCE:</td><td>NKAP Bolting System Requirements</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR:</td><td colspan="3">Admin</td></tr>
<tr><td>PRIMARY THE SYSTEM ACTOR:</td><td colspan="3">None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case describes the event where an Admin wants to delete a User Role. The admin will select the User Menu item on the navigation bar and click the “Search User Role” dropdown menu item. The system will display the Delete User Role Screen. The use case ends when the system has successfully updated the User Role.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The admin must be logged in to the system.
• The User Role must exist on the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">An admin wants to delete a User Role</td></tr>
<tr><td rowspan="5">TYPICAL COURSE OF EVENTS:</td><td rowspan="2">ACTOR ACTION:</td><td colspan="2">SYSTEM RESPONSE:</td></tr>
<tr><td>Manual Action</td><td>Automated Action</td></tr>
<tr><td>Step 1: The Admin wants to delete a User Role</td><td></td><td></td></tr>
<tr><td>Step 2: The admin will select the User Menu item on the navigation bar and click the “Search User Role”</td><td></td><td>Step 3: The system will invoke Use Case 3.9 Search User Role.</td></tr>
<tr><td>Step 4: The admin will click the “Delete” button.</td><td></td><td>Step 5: The system requests confirmation of the deletion and displays a confirmation modal:</td></tr>
</table>

<table>
<tr>
<td></td>
<td></td>
<td></td>
<td>
<table>
<tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Confirmation</td><td>Modal Heading</td><td>None</td></tr>
<tr><td>Are you sure you want to delete the User Role?</td><td>Label</td><td>Label to request confirmation of deletion of of User Role.</td></tr>
<tr><td>Confirm</td><td>Button</td><td>Button when clicked will proceed with deletion of the User Role.</td></tr>
<tr><td>Cancel</td><td>Button</td><td>Button when clicked will cancel the deletion of the User Role.</td></tr>
</table>
</td>
</tr>
<tr>
<td></td>
<td>Step 6: The admin clicks the “Confirm” Button.</td>
<td></td>
<td>Step 7: The system makes use of a SQL Delete Query to Delete the User Role in the <u>UserRole Entity</u>:
<table>
<tr><th>Attribute in Entity</th></tr>
<tr><td>UserRole_ID</td></tr>
<tr><td>UserRole_Name</td></tr>
<tr><td>UserRole_Description</td></tr>
</table>
</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Step 8: : The system displays a success modal to inform the admin that the customer was successfully deleted.
<table>
<tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Success</td><td>Modal Heading</td><td>None</td></tr>
<tr><td>The User Role was successfully deleted.</td><td>Label</td><td>Label to indicate the system successfully deleted the User Role.</td></tr>
<tr><td>Ok</td><td>Button</td><td>None</td></tr>
</table>
</td>
</tr>
</table>

ALTERNATE COURSES:	**ALT STEP 6**: The admin clicks the “cancel” button. **Terminate Use Case.**
	ALT STEP 8: The system fails to delete the User Role and displays an error modal.

Input Name	Control Type	Notes
Error	Modal Heading	None
There was an error when trying to delete the User Role.	Label	Label to indicate the system failed to delete the User Role.
Ok	Button	None

CONCLUSION:	The use case concludes when the system has successfully deleted the User Role.
POST-CONDITION:	• The User Role no longer exists on the system.
BUSINESS RULES:	1. Only an admin can Delete a User Role.
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	None
ASSUMPTIONS:	None
OPEN ISSUES:	None

Create Delivery Shift	Heading	None
Start Time:	Placeholder Text	This is the Placeholder Text for the start time of the delivery shift
Start Time:	Dropdown	None
End Time:	Placeholder Text	This is the Placeholder Text for the end time

			<table><tr><td></td><td></td><td>of the delivery shift</td></tr><tr><td>End Time:</td><td>Dropdown</td><td>None</td></tr><tr><td>Date:</td><td>Placeholder Text</td><td>This is the Placeholder Text for the date of the delivery shift</td></tr><tr><td>Date:</td><td>Calendar</td><td>None</td></tr><tr><td>Add</td><td>Button</td><td>This button is used when the admin decides to proceed with the addition of the delivery shift.</td></tr><tr><td>Cancel</td><td>Button</td><td>This button is used when the admin decides to terminate the process of adding a new delivery shift.</td></tr></table>
	Step 4: The admin enters the details in the respective places: • Start Time • End Time • Date		
	Step 5: The admin clicks on the "**Add**" button **[Alt]**		**Step 6:** The system displays a modal for the admin to confirm the creation of the delivery shift: <table><tr><td>Input Name</td><td>Control Type</td><td>Notes</td></tr></table>

<table>
<tr>
<td></td>
<td></td>
<td></td>
<td>
<table>
<tr><td>Confirm to Proceed</td><td>Placeholder Text</td><td>This serves as a header for the modal.</td></tr>
<tr><td>Would you like to proceed with the creation of a new delivery shift?</td><td>Placeholder Text</td><td>This is to allow the admin to finalize the decision to submit the creation of the delivery shift with the details provided.</td></tr>
<tr><td>Yes</td><td>Button</td><td>Proceeds to the next step (Step 7)</td></tr>
<tr><td>No</td><td>Button</td><td>Allows user to enter delivery shift details again.</td></tr>
</table>
</td>
</tr>
<tr>
<td></td>
<td>Step 7: The admin selects the “Yes” button. [Alt]</td>
<td></td>
<td>Step 8: The system validates that the information entered by the admin does not already exist as a delivery shift.
<table>
<tr><th>Input Name</th><th>Attribute in table</th><th>Notes</th></tr>
<tr><td>Start Time</td><td>StartTime attribute in Time table</td><td rowspan="2">Retrieve the Time_ID if the StartTime and EndTime are atrributes associated with a Time_ID. If there is a Time_ID associated with both times, then retrieve the Shift_ID</td></tr>
<tr><td>End Time</td><td>EndTime attribute in Time table</td></tr>
</table>
</td>
</tr>
</table>

		that contains the specific Time_ID from the **Shift** table.
Date	DayOfTheWeek	Retrieve the Date_ID and retrieve the Shift_ID that contains the specific Date_ID from the **Shift** table.

The Shift_ID associated with the matched Time_ID and the Shift_ID associated with the matched Date_ID are not the same. Therefore, the creation for the requested delivery shift does not already exist on the system.

Step 9: The system uses a SQL insert query to save the captured information by performing the following **[Alt]**:

Time table**:**

Information to be saved:	Attribute in table	Details
Time ID	Time_ID	Generated by the system by reading the last Time_ID from the Time table and adding it by one.

The system saves the following information from the inputted times to be associated with the generated Time_ID.

Input Name	Attribute in table	Validation requirements
Start Time	StartTime	Required
End Time	EndTime	Required

The **Time_ID** is then stored as an attribute associated with the generated **Shift_ID** in the **Shift** table.

Date table**:**

Information to be saved:	Attribute in table	Details
Date ID	Date_ID	Generated by the system by reading the last Date_ID from the Date table and adding it by one.

The system saves the following information from the inputted date to be associated with the generated Date_ID.

Input Name	Attribute in table	Validation requirements
Date	DayOfTheWeek	Required

The **Date_ID** is then stored as an attribute associated with the generated **Shift_ID** in the **Shift** table.

Shift table:

Information to be saved:	Attribute in table	Details

Shift ID	Shift_ID	Generated by the system by reading the last Shift_ID from the **Shift** table and adding it by one.
Date ID	Date_ID	Retrieved from the **Date** table.
Time ID	Time_ID	Retrieved from the **Time** table.

Step 10: The system saves the new delivery shift details in the relevant entities and displays a success modal:

Input Name	Control Type	Notes
Success	Modal Heading	None
Delivery Shift was successfully created.	Placeholder Text	Placeholder Text to indicate the delivery shift information is saved.
Ok	Button	None

ALTERNATE COURSES:

[Alt Step 5]: The admin clicks on the “Cancel” button which terminates this use case.

[Alt Step 7]: The admin clicks on the “No” button which makes the admin **Return to step 4.**

[Alt Step 8]: The Shift_ID associated with the matched Time_ID and the Shift_ID associated with the matched Date_ID are the same. Therefore, the creation for the requested delivery shift already exists on the system.

Input Name	Control Type	Notes

VALIDATION UNSUCCESSFUL	Placeholder Text	This serves as a header for the modal.
The information provided in the date and time already exists in a delivery shift.	Placeholder Text	This is to show that the delivery shift already exists with the parameters the admin entered.
BACK	Button	This button allows the admin the return to the ‘**Create Delivery Shift**’ screen to change the entered information.

After the admin clicks on the “**Back**” button the admin will **Return to step 4.**

[Alt Step 10]: The new delivery shift could not be added to the system. The system will display a modal informing the admin that the saving of the delivery shift information was unsuccessful.

Input Name	Control Type	Notes
Unsuccessful	Placeholder Text	This serves as a header for the modal.
Failed to create delivery shift. Please try again later.	Placeholder Text	This is to show that the saving of the delivery shift information was unsuccessful.
OK	Button	None

CONCLUSION:	This use case concludes when the delivery shift has been successfully added on the system.
POST-CONDITION:	• The delivery shift has been added to the **Shift** table.
BUSINESS RULES:	Only an admin can add a new delivery shift to the system
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	None
ASSUMPTIONS:	None

OPEN ISSUES:	None

<table>
<tr><th colspan="3">NKAP BOLTING SYSTEM</th></tr>
<tr><td>DATE:</td><td colspan="2">13-06-2021</td></tr>
<tr><td>VERSION:</td><td colspan="2">1.0</td></tr>
<tr><td>AUTHOR(S)</td><td colspan="2">Sonali Badrinath</td></tr>
<tr><td>USE CASE NAME:</td><td>Search Delivery Shift</td><td>USE CASE TYPE</td></tr>
<tr><td>USE CASE ID:</td><td>4.2</td><td rowspan="3">Business Requirements: ☐
System Analysis: ☐
System Design: ☑</td></tr>
<tr><td>PRIORITY:</td><td>High</td></tr>
<tr><td>SOURCE:</td><td>NKAP Bolting System Requirements</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR:</td><td colspan="2">Admin</td></tr>
<tr><td>PRIMARY THE SYSTEM ACTOR:</td><td colspan="2">None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="2">None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="2">None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="2">This use case describes the event where the admin wishes to search for a delivery shift. The admin will enter the date. The system will search for the delivery shift based on the date the admin entered. This</td></tr>
</table>

<table>
<tr><td></td><td colspan="3">use case concludes when the delivery shift is shown the results of the search query.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">The admin should be logged on to the system.
The delivery shift should be already exist in the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The admin wishes to search for a delivery shift.</td></tr>
<tr><td rowspan="4">TYPICAL COURSE OF EVENTS:</td><td rowspan="2">ACTOR ACTION:</td><td colspan="2">SYSTEM RESPONSE:</td></tr>
<tr><td>Manual Action</td><td>Automated Action</td></tr>
<tr><td>Step 1: The admin would like to search for a delivery shift.</td><td></td><td></td></tr>
<tr><td>Step 2: The admin clicks on the “Search Delivery Shift” screen.</td><td></td><td>Step 3: The system responds by loading the “Search Delivery Shift” screen with the following controls:
<table>
<tr><td>Control Name</td><td>Control Type</td><td>Notes</td></tr>
<tr><td>Search Delivery Shift</td><td>Heading</td><td>None</td></tr>
<tr><td>Search Date:</td><td>Placeholder Text</td><td>This is the Placeholder Text to prompt the user to select the delivery date of the delivery shift they want to search.</td></tr>
<tr><td>Search Date</td><td>Textbox</td><td>None</td></tr>
<tr><td>Search icon</td><td>Icon button</td><td>None</td></tr>
</table></td></tr>
</table>

Delivery Shift Table	Table	This is to show the **Delivery Shifts** that are saved in the database in a table.
Date	Column	This is the column for the **Delivery Shift** date.
Time	Column	This is the column for the delivery shift start time and end time.
Assign To	Column	This is the column is to allow the admin to assign a delivery shift to an employee. (**Use Case 4.10. Assign Delivery Shift**)
Options	Icon	This is to show a drop down for the options to update and delete information on each row.
Update	Button	Present in Options dropdown. Used to update the information of a delivery shift such as the date, start time and end time.
Delete	Button	Present in Options dropdown. Used to delete shift details such as date, start time, end time and employee information if it was assigned.

			<table><tr><td>Cancel</td><td>Button</td><td>Present in Options dropdown. Used to cancel a shift that was assigned to an employee.</td></tr></table>
	Step 4: The admin inputs the search criteria and clicks on the search icon.		**Step 5:** The system validates the information against the **Date** table **[Alt]**: <table><tr><th>Input Name</th><th>Attribute Name</th><th>Validation Requirements</th></tr><tr><td>Search Date</td><td>DayOfTheWeek</td><td>Required, Date format must be entered as shown in the placeholder text.</td></tr></table>
			Step 6: A SQL read query is used to read the following attributes based on the search criteria: The system will use the DayOfTheWeek associated with the Date_ID attribute from the **Date** table to retrieve all shifts that contain that specific Date_ID from the **Shift** table: **Date_ID** to retrieve and display the associated **DayOfTheWeek** attribute from the **Date** Table in the Date column. **Time_ID** from the **Time** table to retrieve and display the StartTime attribute in the Start Time column and EndTime attribute in the End Time column. Furthermore, the system will use the identified Shift_ID that match the search criteria to check and display the names of the employees that have been assigned to the delivery shifts with the requested date. The system will match the employee names by:

<table>
<tr><td></td><td></td><td></td><td>

- Match the Shift_ID of the shift that was earlier determined to qualify as a search result to the EmployeeShift_ID in the EmployeeShift table.
- The system will identify the Employee_ID associated with that EmployeeShift_ID in the EmployeeShift table.
- The system will retrieve the Employee_Name attribute from the Employee table that is associate with the Employee_ID.
- The Employee_Name will be displayed in the Assign To Column with the option to Assign another employee.

</td></tr>
<tr><td></td><td></td><td></td><td>Step 7: The system will show the search results received from step 6 in a form of a table. [Alt]</td></tr>
<tr><td>ALTERNATE</td><td colspan="3">

Step 5: The details entered were not in the right format, so the validation failed or there are not any existing data on the requested information. The system will display a validation unsuccessful modal to the admin.

Input Name	Control Type	Notes
Error	Placeholder Text	This serves as a header for the modal.
No matches found. Please check the format of the entered information.	Placeholder Text	This is to show that there were no matches found.
OK	Button	None

</td></tr>
<tr><td>CONCLUSION:</td><td colspan="3">The use case concludes when the admin receives the results of the searched criteria.</td></tr>
<tr><td>POST-CONDITION:</td><td colspan="3">

- The admin receives the results of the delivery shifts with the same searched criteria

</td></tr>
<tr><td>BUSINESS RULES:</td><td colspan="3">1. None</td></tr>
<tr><td>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:</td><td colspan="3">None</td></tr>
</table>

ASSUMPTIONS:	None
OPEN ISSUES:	None

<table>
<tr><th colspan="4">NKAP BOLTING SYSTEM</th></tr>
<tr><td>DATE:</td><td colspan="3">13-06-2021</td></tr>
<tr><td>VERSION:</td><td colspan="3">1.0</td></tr>
<tr><td>AUTHOR(S)</td><td colspan="3">Sonali Badrinath</td></tr>
<tr><td>USE CASE NAME:</td><td>Update Delivery Shift</td><td colspan="2">USE CASE TYPE</td></tr>
<tr><td>USE CASE ID:</td><td>4.3.</td><td colspan="2" rowspan="3">Business Requirements: ☐
System Analysis: ☐
System Design: ☑</td></tr>
<tr><td>PRIORITY:</td><td>3</td></tr>
<tr><td>SOURCE:</td><td>NKAP Bolting System Requirements</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR:</td><td colspan="3">Admin</td></tr>
<tr><td>PRIMARY THE SYSTEM ACTOR:</td><td colspan="3">None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">Employee</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case describes the event in which an admin wants to update the details of a delivery shift. This use case ends when the information has been updated successfully.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The admin should be logged on to the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The admin wishes update the details of the delivery shift.</td></tr>
<tr><td rowspan="2">TYPICAL COURSE OF EVENTS:</td><td rowspan="2">Actor Action:</td><td colspan="2">Actor Action:</td></tr>
<tr><td>Manual Action</td><td>Automated Action</td></tr>
</table>

<table>
<tr><td></td><td>Step 1: The admin requests to update the details of a delivery shift by selecting the “Update” option in the Option column dropdown of the line of information of which they want to update.</td><td></td><td>Step 2: The system invokes use case 4.2. Search Delivery Shift.</td></tr>
<tr><td></td><td></td><td></td><td>Step 3: The system responds by loading the “Update Delivery Shift” modal with the following controls:
<table>
<tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Update Delivery Shift</td><td>Heading</td><td>None</td></tr>
<tr><td>Start Time:</td><td>Placeholder Text</td><td>This is the Placeholder Text for the start time of the delivery shift</td></tr>
<tr><td>Start Time:</td><td>Dropdown</td><td>None</td></tr>
<tr><td>End Time:</td><td>Placeholder Text</td><td>This is the Placeholder Text for the end time of the delivery shift</td></tr>
<tr><td>End Time:</td><td>Dropdown</td><td>None</td></tr>
</table></td></tr>
</table>

			<table><tr><td>Date:</td><td>Placeholder Text</td><td>This is the Placeholder Text for the date of the delivery shift</td></tr><tr><td>Date:</td><td>Calendar</td><td>None</td></tr><tr><td>Done</td><td>Button</td><td>This button is used for when the admin is done updating the information and wants to proceed to update the new information on the system.</td></tr><tr><td>Cancel</td><td>Button</td><td>This button is used when the admin decides to terminate the process of adding a new delivery shift.</td></tr></table>
	Step 4: The admin enters the new information and clicks on the "**Done**" button. **[Alt]**		**Step 5**: The system validates that the information entered by the admin does not already exist as a delivery shift. <table><tr><th>Input Name</th><th>Attribute in table</th><th>Notes</th></tr><tr><td>Start Time</td><td>StartTime attribute in Time table</td><td rowspan="2">Retrieve the Time_ID if the StartTime and EndTime are atrributes associated with a Time_ID. If there is a Time_ID associated with both times, then retrieve</td></tr><tr><td>End Time</td><td>EndTime attribute in Time table</td></tr></table>

		the Shift_ID that contains the specific Time_ID from the **Shift** table.
Date	DayOfTheWeek	Retrieve the Date_ID and retrieve the Shift_ID that contains the specific Date_ID from the **Shift** table.

The Shift_ID associated with the matched Time_ID and the Shift_ID associated with the matched Date_ID are not the same. Therefore, the creation for the requested delivery shift does not already exist on the system.

Step 6: The system uses a SQL insert query to update the captured information in the following **[Alt]**:

Time table**:**

Input Name	Attribute in table	Validation requirements
Start Time	StartTime	Required
End Time	EndTime	Required

Date table**:**

Input Name	Attribute in table	Validation requirements
Date	DayOfTheWeek	Required

Step 7: The system saves the new delivery shift details in the relevant entities and displays a success modal:

Input Name	Control Type	Notes
Success	Modal Heading	None
Delivery Shift information was successfully updated.	Placeholder Text	Placeholder Text to indicate the employee information is saved.
Ok	Button	None

ALTERNATE COURSES:

[Alt Step 4]: The admin clicks on the "**Cancel**" button which terminates this use case.

Return to **step 1.**

[Alt Step 5]: The Shift_ID associated with the matched Time_ID and the Shift_ID associated with the matched Date_ID are the same. Therefore, the update for the requested delivery shift already exists on the system.

Input Name	Control Type	Notes
VALIDATION UNSUCCESSFUL	Placeholder Text	This serves as a header for the modal.
The information provided in the date and time already exists in a delivery shift.	Placeholder Text	This is to show that the delivery shift already exists with the parameters the admin entered.
BACK	Button	This button allows the admin the return to the '**Update Delivery Shift**' screen to change the entered information.

<table>
<tr><td></td><td>After the admin clicks on the “Back” button the admin will Return to step 4.</td></tr>
<tr><td></td><td>[Alt Step 7]: The updated delivery shift information could not be added to the system. The system will display a modal informing the admin that the saving of the delivery shift information was unsuccessful.
<table>
<tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Error</td><td>Placeholder Text</td><td>This serves as a header for the modal.</td></tr>
<tr><td>Failed to update delivery shift information. Please try again later.</td><td>Placeholder Text</td><td>This is to show that the saving of the delivery shift information was unsuccessful.</td></tr>
<tr><td>OK</td><td>Button</td><td>None</td></tr>
</table></td></tr>
<tr><td>CONCLUSION:</td><td>This use case concludes when the delivery shift information has been successfully updated on the system.</td></tr>
<tr><td>POST-CONDITION:</td><td>The delivery shift has been updated in the Delivery Shift table.</td></tr>
<tr><td>BUSINESS RULES:</td><td>1. Only an admin can update the information of the delivery shift in the system.</td></tr>
<tr><td>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:</td><td>None</td></tr>
<tr><td>ASSUMPTIONS:</td><td>None</td></tr>
<tr><td>OPEN ISSUES:</td><td>None</td></tr>
</table>

<table>
<tr><th colspan="5">NKAP BOLTING SYSTEM</th></tr>
<tr><td>DATE:</td><td colspan="4">13-06-2021</td></tr>
<tr><td>VERSION:</td><td colspan="4">1.0</td></tr>
<tr><td>AUTHOR(S)</td><td colspan="4">Sonali Badrinath</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Delete Delivery Shift</td><td colspan="2">USE CASE TYPE</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">4.4.</td><td colspan="2" rowspan="3">Business Requirements: ☐
System Analysis: ☐
System Design: ☑</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">High</td></tr>
<tr><td>SOURCE:</td><td colspan="2">NKAP Bolting System Requirements</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR:</td><td colspan="4">Admin</td></tr>
<tr><td>PRIMARY THE SYSTEM ACTOR:</td><td colspan="4">None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="4">None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="4">None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="4">This use case describes the process where the admin wants to delete the delivery shift from the system. This use case begins when the Administrator requests to delete a delivery shift. This use case concludes when the admin gets notified that the delivery shift has been successfully removed from the system.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="4">• The admin should be logged in to the system.
• The delivery shift should already exist on the systems database.</td></tr>
<tr><td>TRIGGER:</td><td colspan="4">The admin requesting to delete a delivery shift.</td></tr>
<tr><td rowspan="3">TYPICAL COURSE OF EVENTS:</td><td rowspan="2">ACTOR ACTION:</td><td colspan="3">SYSTEM RESPONSE:</td></tr>
<tr><td>Manual Action</td><td colspan="2">Automated Action</td></tr>
<tr><td>Step 1: The admin request to delete a</td><td></td><td colspan="2">Step 2: The system invokes Use case 4.2. Search Delivery Shift.</td></tr>
</table>

<table>
<tr><td></td><td>delivery shift from the system.</td><td></td><td></td></tr>
<tr><td></td><td>Step 3: The admin requests to update the details of an employee by selecting the “Delete” option in the Option column dropdown of the line of information of which they want to delete.</td><td></td><td>Step 4: The system displays a modal, with the following controls:
<table>
<tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Confirm to Proceed</td><td>Placeholder Text</td><td>This serves as a header for the modal.</td></tr>
<tr><td>Would you like to proceed with the deletion of the selected delivery shift?</td><td>Placeholder Text</td><td>This is to allow the admin to finalize the decision to submit the deletion of the delivery shift.</td></tr>
<tr><td>Yes</td><td>Button</td><td>Proceeds to the next step (Step 6)</td></tr>
<tr><td>No</td><td>Button</td><td>Allows user to return to the Search Delivery Shift screen.</td></tr>
</table></td></tr>
<tr><td></td><td>Step 5: The admin clicks on the “Yes” button to confirm the deletion of the delivery shift. [Alt]</td><td></td><td>Step 6: The system uses a SQL delete query to remove the delivery shift information from the [Alt]:

From the Shift table:
<table>
<tr><th>Information to be deleted:</th><th>Attribute in table</th><th>Details</th></tr>
<tr><td>Shift ID</td><td>Shift_ID</td><td>This will automatically delete the:</td></tr>
</table></td></tr>
</table>

<table>
<tr><td></td><td></td><td></td><td><table><tr><td></td><td></td><td>- Date_ID and the DayOfTheWeek associated with the ID in the Date table.
- Time_ID and the StartTime, EndTime atrributes associated with the ID in the Time table.
- If the selected shift was assigned the EmployeeShift_ID associated with the Shift_ID will be deleted.</td></tr></table></td></tr>
<tr><td></td><td></td><td></td><td>Step 7: The system removes the deleted delivery shift from the table in the Search Delivery Shift screen.</td></tr>
<tr><td></td><td></td><td></td><td>Step 8: The system displays a success modal:<table><tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Success</td><td>Modal Heading</td><td>None</td></tr><tr><td>Delivery Shift information was successfully deleted.</td><td>Placeholder Text</td><td>Placeholder Text to indicate the delivery shift information is deleted.</td></tr><tr><td>Ok</td><td>Button</td><td>None</td></tr></table></td></tr>
<tr><td rowspan="2">ALTERNATE COURSES:</td><td colspan="3">[Alt Step 5]: The admin clicks on the “No” button.

Terminate this use case.</td></tr>
<tr><td colspan="3">[Alt Step 6]: The delivery shift could not be deleted from the system. The system will display a modal informing the admin that the deletion of the delivery shift was unsuccessful.</td></tr>
</table>

<table>
<tr><td></td><td>
<table>
<tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>UNSUCCESSFUL</td><td>Placeholder Text</td><td>This serves as a header for the modal.</td></tr>
<tr><td>Failed to delete delivery shift. Please try again later.</td><td>Placeholder Text</td><td>This is to show that the deletion of the delivery shift information was unsuccessful.</td></tr>
<tr><td>OK</td><td>Button</td><td>None</td></tr>
</table>
</td></tr>
<tr><td>CONCLUSION:</td><td>This use case ends when the delivery shift has been successfully deleted from the system.</td></tr>
<tr><td>POST-CONDITION:</td><td>• The details of the delivery shift have been removed from the Delivery Shift table.</td></tr>
<tr><td>BUSINESS RULES:</td><td>1. Only the admin can remove a delivery shift.</td></tr>
<tr><td>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:</td><td>None</td></tr>
<tr><td>ASSUMPTIONS:</td><td>None</td></tr>
<tr><td>OPEN ISSUES:</td><td>None</td></tr>
</table>

<table>
<tr><th colspan="5">NKAP BOLTING SYSTEM</th></tr>
<tr><td>DATE:</td><td colspan="4">23-06-21</td></tr>
<tr><td>VERSION:</td><td colspan="4">1.0</td></tr>
<tr><td>AUTHOR(S)</td><td colspan="4">Sonali Badrinath</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Add Employee</td><td colspan="2">USE CASE TYPE</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">4.5.</td><td colspan="2" rowspan="3">Business Requirements: ☐
System Analysis: ☐
System Design: ☑</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">High</td></tr>
<tr><td>SOURCE:</td><td colspan="2">NKAP Bolting System Requirements</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR:</td><td colspan="4">Administrator</td></tr>
<tr><td>PRIMARY THE SYSTEM ACTOR:</td><td colspan="4">Employee</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="4">None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="4">None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="4">This use case describes the event where an admin wants to add an employee on the system. The system will prompt the admin to provide the new employee information by displaying the Add Employee Screen. The system will validate the information. Once the system has successfully validated the Information, it will be captured and stored in the <u>User Entity</u> and the <u>Employee Entity</u> respectively. The use case ends when the employee has successfully added on the system.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="4">• The admin should be logged onto the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="4">The admin requesting to add a new employee.</td></tr>
<tr><td rowspan="3">TYPICAL COURSE OF EVENTS:</td><td rowspan="2">ACTOR ACTION:</td><td colspan="3">SYSTEM RESPONSE:</td></tr>
<tr><td>Manual Action</td><td colspan="2">Automated Action</td></tr>
<tr><td>Step 1: The admin requests to</td><td></td><td colspan="2"></td></tr>
</table>

<table>
<tr><td></td><td>add a new employee.</td><td></td><td></td></tr>
<tr><td></td><td>Step 2: The admin will click on the “Employee” menu item on the navigation bar.</td><td></td><td>Step 3: The system displays the Employee dropdown menu item</td></tr>
<tr><td></td><td>Step 4: The customer clicks the “Add Employee” dropdown menu item.</td><td></td><td>Step 5: The system responds by loading the “Add Employee” modal with the following controls:

<table>
<tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Add Employee</td><td>Heading</td><td>None</td></tr>
<tr><td>Title:</td><td>Textbox</td><td>This is the Placeholder Text for the title of the employee.</td></tr>
<tr><td>Title:</td><td>Dropdown</td><td>Populated with different titles to select.</td></tr>
<tr><td>Name:</td><td>Placeholder Text</td><td>This is the Placeholder Text for the name of the employee.</td></tr>
<tr><td>Name:</td><td>Textbox</td><td>None</td></tr>
<tr><td>Surname:</td><td>Placeholder Text</td><td>This is the Placeholder Text for the surname of the employee.</td></tr>
</table>
</td></tr>
</table>

			Surname:	Textbox	None
			Cell Number:	Placeholder Text	This is the Placeholder Text for the cell number of the employee
			Cell Number:	Textbox	None
			Address:	Placeholder Text	This is the Placeholder Text for the address of the employee.
			Address:	Textbox	None
			Date of Birth:	Placeholder Text	This is the Placeholder Text for the date of birth of the employee.
			DOB:	Calendar	None
			ID Number:	Placeholder Text	This is the Placeholder Text for the ID Number of the employee
			ID Number:	Textbox	None
			Next	Button	This button is used when the admin decides to proceed with the addition of the employee.

				User Role	Placeholder Text	This is the Placeholder Text for the User Role to be selected
				User Role	Dropdown (disabled)	Populated with all existing user roles from the database.
				Username	Placeholder Text	This is the Placeholder Text for the Username of the employee
				Username	Textbox (disabled)	None
				Password	Placeholder Text	This is the Placeholder Text for the Password of the employee account.
				Password	Textbox (disabled)	None
				Confirm Password	Placeholder Text	This is the Placeholder Text for the Password Confirmation.
				Confirm Password	Textbox (disabled)	None

		Add Employee	Button (disabled)	This button is used when the admin has to create the new login details for the employee.
		Cancel	Button	This button is used when the admin decides to terminate the process of adding a new employee.

Step 6: The employee communicates their details to the admin.	**Step 7:** The admin enters the details in the respective places: • Title • Name • Surname • Cell Number • Address • Date of Birth • ID Number The admin clicks on the "Next" button	**Step 8**: The system validates the captured employee information according to the validation requirements of the **Employee Entity**:

Input Name	Attribute in table	Validation requirements
Employee_ID	Employee_ID	Generated by the system by reading the last Employee_ID from the **Employee** table and adding it by one.
Title	Title_ID	Required
Name	Employee_Name	Required, maximum

		of 200 characters
Surname	Employee_Surname	Required, 10 digits
Cell Number:	Employe_Number	Required, maximum of 200 characters
Address:	Employee_Address	Required, maximum of 200 characters
Date of Birth:	Employee_DOB	Required, Date
ID Number:	Employee_IDNumber	Required, 13 digits

Step 9: The system enables and disables the following controls:

Control Name	Control Type	Notes
Add Employee	Heading	None
Title:	Textbox	This is the Placeholder Text for the title of the employee.
Title:	Dropdown (disable)	Populated with different titles to select.
Name:	Placeholder Text	This is the Placeholder Text for the name of the employee.

Name:	Textbox (disable)	None
Surname:	Placeholder Text	This is the Placeholder Text for the surname of the employee
Surname:	Textbox (disable)	None
Cell Number:	Placeholder Text	This is the Placeholder Text for the cell number of the employee
Cell Number:	Textbox (disable)	None
Address:	Placeholder Text	This is the Placeholder Text for the address of the employee.
Address:	Textbox (disable)	None
Date of Birth:	Placeholder Text	This is the Placeholder Text for the date of birth of the employee.
DOB:	Calendar (disable)	None
ID Number:	Placeholder Text	This is the Placeholder Text for the ID Number of the employee

			ID Number:	Textbox (disable)	None
			Next	Button (disable)	This button is used when the admin decides to proceed with the addition of the employee.
			User Role	Placeholder Text	This is the Placeholder Text for the User Role to be selected
			User Role	Dropdown (enabled)	Populated with all existing user roles from the database.
			Username	Placeholder Text	This is the Placeholder Text for the Username of the employee
			Username	Textbox (enabled)	None
			Password	Placeholder Text	This is the Placeholder Text for the Password of the employee account.
			Password	Textbox (enabled)	None

<table>
<tr>
<td></td>
<td></td>
<td></td>
<td>
<table>
<tr><td>Confirm Password</td><td>Placeholder Text</td><td>This is the Placeholder Text for the Password Confirmation.</td></tr>
<tr><td>Confirm Password</td><td>Textbox (enabled)</td><td>None</td></tr>
<tr><td>Add Employee</td><td>Button (enabled)</td><td>This button is used when the admin has to create the new login details for the employee.</td></tr>
<tr><td>Cancel</td><td>Button</td><td>This button is used when the admin decides to terminate the process of adding a new employee.</td></tr>
</table>
</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Step 10: The admin enters the temporary login details for the employee by providing the following details:

Username
Password
Confirm

</td>
<td>Step 11: The system validates the captured information according to the validation requirements of the User Entity:
<table>
<tr><th>Input Name</th><th>Attribute in Entity</th><th>Validation Requirements</th></tr>
<tr><td>Username</td><td>User_Username</td><td>Required
Max 50 characters
Alphanumeric characters only</td></tr>
<tr><td>Password</td><td>User_Password</td><td>Required</td></tr>
</table>
</td>
</tr>
</table>

<table>
<tr>
<td></td>
<td></td>
<td>
Password
User Role
The admin clicks on the “Add Employee” button.
</td>
<td>
<table>
<tr><td></td><td></td><td>Max 50 characters
Alphabet characters only</td></tr>
</table>
</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>
Step 12: The system displays a modal for the admin to confirm the addition of the employee information:
<table>
<tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Confirm to Proceed</td><td>Placeholder Text</td><td>This serves as a header for the modal.</td></tr>
<tr><td>Would you like to proceed with the addition of a new employee?</td><td>Placeholder Text</td><td>This is to allow the admin to finalize the decision to submit the addition of the employee with the details provided.</td></tr>
<tr><td>Yes</td><td>Button</td><td>Proceeds to the next step (Step 14)</td></tr>
<tr><td>No</td><td>Button</td><td>Allows user to enter employee details again.</td></tr>
</table>
</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Step 13: The admin selects the “Yes” button. [Alt]</td>
<td>Step 14: The system uses a SQL insert query to save the new employee information in the <u>Employee Table:</u></td>
</tr>
</table>

And the User Table:

Input Name	Attribute in table	Validation requirements
Employee_ID	Employee_ID	Generated by the system by reading the last Employee_ID from the **Employee** table and adding it by one.
Title	Title_ID	Required
Name	Employee_Name	Required, maximum of 200 characters
Surname	Employee_Surname	Required, 10 digits
Cell Number:	Employe_Number	Required, maximum of 200 characters
Address:	Employee_Address	Required, maximum of 200 characters
Date of Birth:	Employee_DOB	Required, Date
ID Number:	Employee_IDNumber	Required, 13 digits
Details to be saved	Attribute in table	Notes

User ID	User_ID	The system generates a new User_ID by retrieving the last User_ID in the User table and incrementing it by one.
Username	User_ Username	None
Password	User_ Password	None
UserRole ID	UserRole_ID	None

Step 15: The system saves the new employee details in the relevant entities and displays a success modal:

Input Name	Control Type	Notes
Success	Modal Heading	None
Employee information was successfully created.	Placeholder Text	Placeholder Text to indicate the employee information is saved.
Ok	Button	None

ALTERNATE COURSES:

[Alt Step 13]: The admin clicks on the “No” button which makes the admin **Return to step 7.**

ALT STEP 8 and 11: The system fails to validate the inputted information and displays a Validation Unsuccessful Modal.

Input Name	Control Type	Notes

<table>
<tr><td></td><td>
<table>
<tr><td>VALIDATION UNSUCCESSFUL</td><td>Placeholder Text</td><td>This serves as a header for the modal.</td></tr>
<tr><td>The information provided in the given parameters is not in the correct format.</td><td>Placeholder Text</td><td>This is to show that the employee information was incorrectly entered.</td></tr>
<tr><td>BACK</td><td>Button</td><td>This button allows the admin the return to the ‘Add Employee’ screen to change the entered information.</td></tr>
</table>
</td></tr>
<tr><td></td><td>
[Alt Step 15]: The new employee could not be added to the system. The system will display a modal informing the admin that the saving of the employee information was unsuccessful.
<table>
<tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Unsuccessful</td><td>Placeholder Text</td><td>This serves as a header for the modal.</td></tr>
<tr><td>Failed to create employee information. Please try again later.</td><td>Placeholder Text</td><td>This is to show that the saving of the employee information was unsuccessful.</td></tr>
<tr><td>OK</td><td>Button</td><td>None</td></tr>
</table>
</td></tr>
<tr><td></td><td>This use case concludes when the employee has been successfully added on the system.</td></tr>
<tr><td>CONCLUSION:</td><td>The employee has been added to the Employee table and <u>User</u> table.</td></tr>
<tr><td>POST-CONDITION:</td><td>Only an admin can add a new employee to the system
• The employee must reset their password after the use case.</td></tr>
<tr><td>BUSINESS RULES:</td><td>None</td></tr>
<tr><td>IMPLEMENTATION CONSTRAINTS</td><td>None</td></tr>
</table>

AND SPECIFICATIONS:	
ASSUMPTIONS:	None
OPEN ISSUES:	

NKAP BOLTING SYSTEM			
DATE:	13-06-2021		
VERSION:	**1.0**		
AUTHOR(S)	Sonali Badrinath		
USE CASE NAME:	**Search Employee**	USE CASE TYPE	
USE CASE ID:	**4.6.**	**Business Requirements:**	□
PRIORITY:	High	**System Analysis:**	□

SOURCE:	NKAP Bolting System Requirements	**System Design:** ☑
PRIMARY BUSINESS ACTOR:	Admin	
PRIMARY THE SYSTEM ACTOR:	None	
OTHER PARTICIPATING ACTORS:	None	
OTHER INTERESTED STAKEHOLDERS:	None	
DESCRIPTION:	This use case describes the event where the admin wishes to search for an employee. The admin will enter the search parameters like the employee's name and surname. The system will search for the employee based on the parameters the admin entered. This use case concludes when the employee is shown the results of the search query.	
PRE-CONDITION:	• The admin should be logged on to the system. • The employee should be already exist in the system.	
TRIGGER:	The admin wishes to search for a employee.	

TYPICAL COURSE OF EVENTS:	**ACTOR ACTION:**	**SYSTEM RESPONSE:**	
		Manual Action	Automated Action
	Step 1: The admin would like to search for an employee.		
	Step 2: The admin clicks on the "**Search Employee**" screen.		**Step 3**: The system responds by loading the "**Search Employee**" screen with the following controls: Control Name / Control Type / Notes

			Search Employee	Heading	None
			Name:	Placeholder Text	This is the Placeholder Text to prompt the user to enter a search query.
			Search Name	Textbox	None
			Surname:	Placeholder Text	This is the Placeholder Text to prompt the user to enter a search query.
			Search Surname	Textbox	None
			Search Icon	Icon	Used to give consent to perform the search.
			Employee Table	Table	This is to show the **Employees** that are saved in the database in a table.
			ID Number	Column	This is the column for the Employee ID Number
			Title	Column	This is the column for the employee title.
			Fullname	Column	This is the column for the employee name and surname.
			Date of Birth	Column	This is the column for the employee date of birth

<table>
<tr>
<td></td>
<td></td>
<td></td>
<td>
<table>
<tr><td>Address</td><td>Column</td><td>This is the column for the employee address</td></tr>
<tr><td>Number</td><td>Column</td><td>This is the column for the employee number</td></tr>
<tr><td>Options</td><td>Icon</td><td>This is to show a drop down for the options to update and delete information on each row.</td></tr>
<tr><td>Update</td><td>Button</td><td>Present in Options dropdown. Used to update the information of an employee.</td></tr>
<tr><td>Delete</td><td>Button</td><td>Present in Options dropdown. Used to delete shift details.</td></tr>
</table>
</td>
</tr>
<tr>
<td></td>
<td>Step 4: The admin inputs the search criteria and clicks on the search button.</td>
<td></td>
<td>
Step 5: The system captures and validates the information against the Employee table [Alt]:
<table>
<tr><th>Input Name</th><th>Attribute Name</th><th>Validation Requirments</th></tr>
<tr><td>Search Name:</td><td>Employee_Name</td><td>Maximum of 50 characters</td></tr>
<tr><td>Search Surname:</td><td>Employee_Surname</td><td>Maximum of 50 characters</td></tr>
</table>
</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>
Step 6: A SQL read query is used to read the following attributes from the Employee table based on the search criteria:

Employee table:
Employee_ID(PK)
</td>
</tr>
</table>

<table>
<tr><td></td><td></td><td></td><td>Employee_IDNumber in the ID Number column
Title_ID attribute information displayed in the Title column
Employee_Name attribute information displayed in the Name column
Employee_Surname attribute information displayed in the Surname column
Employee_Number attribute information displayed in the Number column
Employee_Address attribute information displayed in the Address column
Employee_DOB attribute information displayed in the DOB column</td></tr>
<tr><td></td><td></td><td></td><td>Step 7: The system will show the search results received from step 6 in a form of a table. [Alt]</td></tr>
<tr><td>ALTERNATE</td><td colspan="3">Step 5: The details entered were not in the right format, so the validation failed or there are not any existing data on the requested information. The system will display a validation unsuccessful modal to the admin.
<table><tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Error</td><td>Placeholder Text</td><td>This serves as a header for the modal.</td></tr><tr><td>No matches found. Please check the format of the entered information.</td><td>Placeholder Text</td><td>This is to show that there were no matches found.</td></tr><tr><td>OK</td><td>Button</td><td>None</td></tr></table></td></tr>
<tr><td>CONCLUSION:</td><td colspan="3">The use case concludes when the admin receives the results of the searched criteria.</td></tr>
<tr><td>POST-CONDITION:</td><td colspan="3">The admin receives the results of the employees with the same searched criteria</td></tr>
<tr><td>BUSINESS RULES:</td><td colspan="3">1. None</td></tr>
</table>

IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	None
ASSUMPTIONS:	None
OPEN ISSUES:	None

<table>
<tr><th colspan="4">NKAP BOLTING SYSTEM</th></tr>
<tr><td>DATE:</td><td colspan="3">13-06-2021</td></tr>
<tr><td>VERSION:</td><td colspan="3">1.0</td></tr>
<tr><td>AUTHOR(S)</td><td colspan="3">Sonali Badrinath</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Update Employee</td><td>USE CASE TYPE</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">4.7.</td><td>Business Requirements: ☐</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">3</td><td>System Analysis: ☐</td></tr>
<tr><td>SOURCE:</td><td colspan="2">NKAP Bolting System Requirements</td><td>System Design: ☑</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR:</td><td colspan="3">Admin</td></tr>
<tr><td>PRIMARY THE SYSTEM ACTOR:</td><td colspan="3">None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">Employee</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case describes the event in which an admin wants to update the details of an employee. The system will display the previous information that was added before and then the admin will add the new information. This use case ends when the information has been updated successfully.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The admin should be logged on to the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The admin wishes update the details of the employee.</td></tr>
<tr><td rowspan="3">TYPICAL COURSE OF EVENTS:</td><td rowspan="2">Actor Action:</td><td colspan="2">Actor Action:</td></tr>
<tr><td>Manual Action</td><td>Automated Action</td></tr>
<tr><td>Step 1: The admin requests to update the details of an employee by selecting</td><td></td><td>Step 2: The system invokes use case 4.6. Search Employee.</td></tr>
</table>

	the "**Update**" option in the Option column dropdown of the line of information of which they want to update.		
			Step 3: The system responds by loading the "Update Employee" modal with the following controls:

Control Name	Control Type	Notes
Update Employee	Heading	None
Title:	Textbox	This is the Placeholder Text for the title of the employee.
Title:	Dropdown	Populated with different titles to select.
Name:	Placeholder Text	This is the Placeholder Text for the start time of the delivery shift
Name:	Textbox	None
Surname:	Placeholder Text	This is the Placeholder Text for the end time of

		the delivery shift
Surname:	Textbox	None
Name:	Placeholder Text	This is the Placeholder Text for the start time of the delivery shift
Name:	Textbox	None
Cell Number:	Placeholder Text	This is the Placeholder Text for the cell number of the employee
Cell Number:	Textbox	None
Address:	Placeholder Text	This is the Placeholder Text for the address of the employee.
Address:	Textbox	None
Date of Birth:	Placeholder Text	This is the Placeholder Text for the date of birth of the employee.
DOB:	Calendar	None
ID Number:	Placeholder Text	This is the Placeholder Text for the ID Number of the employee

			ID Number: | Textbox | None

ID Number:	Textbox	None
Done	Button	This button is used for when the admin is done updating the information and wants to proceed to update the new information on the system.
Cancel	Button	This button is used when the admin decides to terminate the process of adding a new employee.

Step 4: The admin enters the new information and clicks on the **"Done"** button. **[Alt]**

Step 5: The system captures and validates the information entered by the admin against the Employee table **[Alt]:**

Input Name	Attribute in table	Validation requirements
Title	Title_ID	Required
Name	Employee_Name	Required, maximum of 200 characters
Surname	Employee_Surname	Required, 10 digits

Cell Number :	Employe_Number	Required, maximum of 200 characters
Address :	Employee_Address	Required, maximum of 200 characters
Date of Birth:	Employee_DOB	Required, Date
ID Number:	Employee_IDNumber	Required, 13 digits

Step 6: The system uses a SQL insert query to update the captured information in the **Employee** table **[Alt]**:

Information to be saved:	Attribute in table	Details
Title	Title_ID	Match the Title_ID of the selected Title_Desc
Name	Employee_Name	None

			Surname	Employee_Surname	None
			Cell Number:	Employe_Number	None
			Address:	Employee_Address	None
			Date of Birth:	Employee_DOB	None
			ID Numb er:	Employee_IDNumbe r	None

			Step 7: The system saves the new employee details in the relevant entities and displays a success modal: Input Name / Control Type / Notes: Success / Modal Heading / None Employee information was successfully updated. / Placeholder Text / Placeholder Text to indicate the employee information is saved. Ok / Button / None
ALTERNATE COURSES:	**[Alt Step 4]**: The admin clicks on the "**Cancel**" button which terminates this use case. Return to **step 1.**		
	[Alt Step 5]: The details entered were not in the right format, so the validation failed. The system will display a validation unsuccessful modal to the admin.		

Input Name	Control Type	Notes
Success	Modal Heading	None
Employee information was successfully updated.	Placeholder Text	Placeholder Text to indicate the employee information is saved.
Ok	Button	None

Input Name	Control Type	Notes
VALIDATION UNSUCCESSFUL	Placeholder Text	This serves as a header for the modal.
The information provided in the given parameters is not in the correct format.	Placeholder Text	This is to show that the employee information was incorrectly entered.
BACK	Button	This button allows the admin the return to the '**Update Employee**' screen to edit the entered information in the correct format.

After the admin clicks on the "**Back**" button the system will show the admin where the validation errors occurred by showing a red outline where the validation failed.

[Alt Step 6]: The updated employee information could not be added to the system. The system will display a modal informing the admin that the saving of the employee information was unsuccessful.

Input Name	Control Type	Notes
UNSUCCESSFUL	Placeholder Text	This serves as a header for the modal.
Failed to update employee information. Please try again later.	Placeholder Text	This is to show that the saving of the employee information was unsuccessful.
OK	Button	None

CONCLUSION:	This use case concludes when the employee information has been successfully updated on the system.
POST-CONDITION:	• The employee has been updated in the **Employee** table.
BUSINESS RULES:	**1.** Only an admin can update the information of the employee in the system.
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	None
ASSUMPTIONS:	None
OPEN ISSUES:	None

NKAP BOLTING SYSTEM		
DATE:	13-06-2021	
VERSION:	**1.0**	
AUTHOR(S)	Sonali Badrinath	
USE CASE NAME:	**Delete Employee**	**USE CASE TYPE**
USE CASE ID:	**4.8.**	**Business Requirements:** ☐

PRIORITY:	High	**System Analysis:** ☐		
SOURCE:	NKAP Bolting System Requirements	**System Design:** ☑		
PRIMARY BUSINESS ACTOR:	Admin			
PRIMARY THE SYSTEM ACTOR:	None			
OTHER PARTICIPATING ACTORS:	None			
OTHER INTERESTED STAKEHOLDERS:	None			
DESCRIPTION:	This use case describes the process where the admin wants to delete the employee from the system. This use case begins when the Administrator requests to delete a employee. This use case concludes when the admin gets notified that the employee has been successfully removed from the system.			
PRE-CONDITION:	• The admin should be logged in to the system. • The employee should already exist on the systems database.			
TRIGGER:	The admin requesting to delete an employee.			
TYPICAL COURSE OF EVENTS:	**ACTOR ACTION:**	**SYSTEM RESPONSE:**		
		Manual Action	Automated Action	
	Step 1: The admin request to delete a employee from the system.		**Step 2**: The system invokes **Use case 4.6. Search Employee**.	
	Step 3: The admin requests to update the details of an		**Step 4**: The system displays a modal, with the following controls:	

Input Name	Control Type	Notes

<table>
<tr>
<td></td>
<td>employee by selecting the “Delete” option in the Option column dropdown of the line of information of which they want to delete.</td>
<td></td>
<td>
<table>
<tr><td>Confirm to Proceed</td><td>Placeholder Text</td><td>This serves as a header for the modal.</td></tr>
<tr><td>Would you like to proceed with the deletion of the selected employee?</td><td>Placeholder Text</td><td>This is to allow the admin to finalize the decision to submit the deletion of the employee.</td></tr>
<tr><td>Yes</td><td>Button</td><td>Proceeds to the next step (Step 6)</td></tr>
<tr><td>No</td><td>Button</td><td>Allows user to return to the Search Employee screen.</td></tr>
</table>
</td>
</tr>
<tr>
<td></td>
<td>Step 5: The admin clicks on the “Yes” button to confirm the deletion of the employee. [Alt]</td>
<td></td>
<td>Step 6: The system uses a SQL delete query to remove the employee information from the Employee table [Alt]:
<table>
<tr><th>Information to be deleted:</th><th>Attribute in table</th><th>Details</th></tr>
<tr><td>Employee ID</td><td>Employee_ID</td><td>This will automatically delete all associated attributes such as the Title_Desc, Employee_Name, Employee_Surname, Employee_CellNumber, Employee_Address, Employee_DOB and Employee_IDNumber of the selected employee.</td></tr>
</table>
</td>
</tr>
</table>

<table>
<tr><td></td><td></td><td></td><td>Step 7: The system removes the deleted employee from the table in the Search Employee screen.</td></tr>
<tr><td></td><td></td><td></td><td>Step 8: The system displays a success modal:
<table>
<tr><td>Input Name</td><td>Control Type</td><td>Notes</td></tr>
<tr><td>Success</td><td>Modal Heading</td><td>None</td></tr>
<tr><td>Employee information was successfully deleted.</td><td>Placeholder Text</td><td>Placeholder Text to indicate the employee information is deleted.</td></tr>
<tr><td>Ok</td><td>Button</td><td>None</td></tr>
</table></td></tr>
<tr><td rowspan="2">ALTERNATE COURSES:</td><td colspan="3">[Alt Step 5]: The admin clicks on the “No” button.

Terminate this use case.</td></tr>
<tr><td colspan="3">[Alt Step 8]: The employee could not be deleted from the system. The system will display a modal informing the admin that the deletion of the employee was unsuccessful.
<table>
<tr><td>Input Name</td><td>Control Type</td><td>Notes</td></tr>
<tr><td>UNSUCCESSFUL</td><td>Placeholder Text</td><td>This serves as a header for the modal.</td></tr>
<tr><td>Failed to delete employee. Please try again later.</td><td>Placeholder Text</td><td>This is to show that the deletion of the employee information was unsuccessful.</td></tr>
<tr><td>OK</td><td>Button</td><td>None</td></tr>
</table></td></tr>
<tr><td>CONCLUSION:</td><td colspan="3">This use case ends when the employee has been successfully deleted from the system.</td></tr>
<tr><td>POST-CONDITION:</td><td colspan="3">• The details of the employee have been removed from the Employee table.</td></tr>
<tr><td>BUSINESS RULES:</td><td colspan="3">1. Only the admin can remove a employee.</td></tr>
<tr><td>IMPLEMENTATION CONSTRAINTS</td><td colspan="3">None</td></tr>
</table>

AND SPECIFICATIONS:	
ASSUMPTIONS:	None
OPEN ISSUES:	None

<table>
<tr><th colspan="4">NKAP BOLTING SYSTEM</th></tr>
<tr><td>DATE:</td><td colspan="3">13-06-2021</td></tr>
<tr><td>VERSION:</td><td colspan="3">1.0</td></tr>
<tr><td>AUTHOR(S)</td><td colspan="3">Sonali Badrinath</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Assign Order for Delivery/Courier</td><td>USE CASE TYPE</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">4.9.</td><td rowspan="3">Business Requirements: ☐
System Analysis: ☐
System Design: ☑</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">High</td></tr>
<tr><td>SOURCE:</td><td colspan="2">NKAP Bolting System Requirements</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR:</td><td colspan="3">Administrator</td></tr>
<tr><td>PRIMARY THE SYSTEM ACTOR:</td><td colspan="3">None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">● Customer
● Employee
● Courier</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case describes the event where the admin wants to assign an order to be delivered to the customer. The use case begins when the admin requests to assign an order for a delivery or courier depending on the distance. Details of the assigned delivery will be stored in the Delivery table. The use case ends when the been successfully assigned.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The admin must be logged in.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The admin wants to assign an order to be delivered to the customer.</td></tr>
<tr><td rowspan="3">TYPICAL COURSE OF EVENTS:</td><td rowspan="2">ACTOR ACTION:</td><td colspan="2">SYSTEM RESPONSE:</td></tr>
<tr><td>Manual Action</td><td>Automated Action</td></tr>
<tr><td>Step 1: The admin wants to assign an order</td><td></td><td></td></tr>
</table>

	to be delivered to the customer.		
	Step 2: The admin will click on the "**Assign Order Delivery**" sub menu item of the **'Admin'** on the navigation bar.		**Step 3:** The system responds by loading the following: Screen with the following controls:

Control Name	Control Type	Notes
Assign Order Delivery	Heading	None
Unscheduled Deliveries	Table	None
Sale ID	Column	Only sales that have: • Have a Collection_Status of 'Available' • Have a Sale_Receive Type of 'Delivery' • Have a Sale_Assign attribute that is 'False'
Customer Name	Column	None
Address	Column	None
Distance	Column	None
Delivery/Courier	Column	This each column cell contains both Delivery and Courier buttons in each row.
Delivery	Button	This button is used when the admin wants to assign an

					order for a local delivery.
			Courier	Button	This button is used when the admin wants to assign an order for courier.

Step 4: Based on the distance in the Distance column the admin determines that the order must be delivered via courier. The admin clicks on the **“Courier" button** that is in the row of which the admin wants to assign the order, that is in the **“Delivery/Courier”** column of the **“Unscheduled Deliveries”** table **[ALT]**

Step 5: The system responds by loading the “**Courier**” modal with the following controls:

Control Name	Control Type	Notes
Courier	Heading	None
Courier Name	Placeholder Text	This is the Placeholder Text for the searching the courier's name of which to assign the order
Search Courier Name	Textbox	None
Courier Name:	Placeholder Text	This is the Placeholder Text for the name of the courier
Search Icon	Icon Button	None
Instruction Text	Label	This informs the user as to how to assign the order. The label Text will read “Click on email link to compose courier request”
Sale ID	Label	None
Sale ID	Textbox (disabled)	None

				Courier Delivery	Table	This is to show the search results of Couriers that are saved in the database.	
				Name	Column	This is the column for the Courier name	
				Type	Column	This is the column for the courier type	
				Number	Column	This is the column for the number of the courier	
				Email	Column	This is the column for the email of the courier. The email links are functional.	
				Done	Button	This button is clicked when the admin is done assigning the order.	
				Cancel	Button	This button is used when the admin decides to terminate the process of assigning an order to a courier.	

<table>
<tr><td></td><td>Step 6: Admin wants to search for the courier. [ALT]</td><td></td><td>Step 7: The system invokes Use Case 8.4. Search Courier. [ALT]</td></tr>
<tr><td></td><td>Step 8: The admin clicks on the email address of the courier they want to assign the order to. [ALT]</td><td></td><td>Step 9: The system directs the admin to the web browser whereby the admin can compose the email for the order and the email address of the courier will be automatically filled in the “To:” section. [ALT]</td></tr>
<tr><td></td><td>Step 10: The admin will enter the delivery information for the order in the email to the courier. [ALT]</td><td></td><td></td></tr>
<tr><td></td><td>Step 11: Once, done the admin sends the email. The admin navigates back to the “Courier” modal. [ALT]</td><td></td><td></td></tr>
<tr><td></td><td>Step 12: The admin clicks on ‘Done’ button. [ALT]</td><td></td><td>Step 13: The system displays a popup message for the admin to confirm that the courier has been assigned an order:
<table>
<tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Attention</td><td>Placeholder Text</td><td>This serves as a header for the modal.</td></tr>
<tr><td>Clicking on the Done button will update the order information</td><td>Placeholder Text</td><td>This is to remind the admin that by clicking the Done button that the</td></tr>
</table></td></tr>
</table>

to be assigned.		order will be saved on the system as being assigned.
Ok	Button	Proceeds to the next step (**Step 15**)
Back	Button	Return the admin to the Courier Screen and allow the user to assign an order to a courier or use the Cancel button to terminate the courier assignment process.

Step 14: The admin clicks on the OK button. **[ALT]**

Step 15: The system uses a SQL insert query to save the captured information in the: **[Alt]**

Delivery table

Information to be saved:	Attribute in table	Details
Delivery ID	Delivery_ID	Generated by the system by reading the last Delivery_ID from the **Delivery** table and adding it by one.
Courier ID	Courier_ID	Use the

		Courier_ID associated with the selected Courier_Email. Store the Courier_ID with the Delivery _ID that was generated.

Sale table

Information to be saved:	Attribute in table	Details
Sale_Assign=True	Sale_Assign	Use the Sale_ID to match the Sale_Assign attribute and update information to true. This will mean that the sale is recorded as assigned
Courier Email	Courier_Email	None

Step 16: The system saves the new delivery assignment details and displays a success modal: **[ALT]**

Input Name	Control Type	Notes
Success	Modal Heading	None
Saving of assigned order information has been successful.	Placeholder Text	Placeholder Text to indicate the order has been updated to be assigned

					on the system.
			Ok	Button	None

ALTERNATE COURSES:

[Alt Step 4(delivery option)]: Based on the distance in the Distance column the admin determines that the order must be delivered locally. The admin clicks on the **"Delivery" button** that is in the row of which the admin wants to assign the order, that is in the **"Delivery/Courier"** column of the **"Unscheduled Deliveries"** table.

[Alt Step 5 (delivery option)]: The system responds by loading the "**Delivery**" modal with the following controls:

Control Name	Control Type	Notes
Delivery	Heading	None
Select Employee	Placeholder Text	This is the Placeholder Text to allows selection of an employee that will be assigned to the order.
Employee	Dropdown	Populated selecting all the Employee_Name data from the Employee table.
Done	Button	This button is clicked when the admin is done assigning the order.
Cancel	Button	This button is used when the admin decides to terminate the process of assigning an order to an employee. Returns user to

		“Assign Order For Delivery” screen	

[Alt Step 6 (courier option)]: The admin selects the “Cancel” button. **Return to Step 3.**

[ALT Step 6a (delivery option)]: The admin selects the employee from the Employee dropdown to assign the delivery of the order. Then clicks on the ‘Done’ button. **Proceed to ALT Step 7(delivery option)**

[ALT Step 6b (delivery option)]: The admin clicks on the ‘Cancel’ button. **Return to Step 5.**

[ALT Step 7(delivery option)]: The system displays a modal for the admin to confirm the assigning of the order to an employee:

Input Name	Control Type	Notes
Confirm to Proceed	Placeholder Text	This serves as a header for the modal.
Would you like to proceed with the assigning of this order to this employee?	Placeholder Text	This is to allow the admin to finalize the decision to submit the assigning of the order to an employee.
Yes	Button	Proceeds to the save information (**ALT Step 8(delivery option)**)
No	Button	Allows user to change the selected employee name again.

[ALT Step 8a (delivery option)]: The admin selects the “Yes” button. **Proceed to ALT Step 9(delivery option)**

[ALT Step 8b (delivery option)]: The admin selects the “No” button. **Return to Step 6.**

[ALT Step 9 (delivery option)]: The system uses a SQL insert query to save the captured information in the:

Delivery table

Information to be saved:	Attribute in table	Details
Delivery ID	Delivery_ID	Generated by the system by reading the last Delivery_ID from the **Delivery** table and adding it by one.

Sale table

Information to be saved:	Attribute in table	Details
Sale_Assign=True	Sale_Assign	Use the Sale_ID to match the Sale_Assign attribute and update information to true. This will mean that the sale is recorded as assigned

Employee table

Input Name	Attribute in table	Details
Employee Name	Employee_Name	Identify the Employee_ID of the selected Employee_Name.

EmployeeShift table

Information to be saved:	Attribute in table	Details
Delivery ID	Delivery_ID	The system will look for

			EmployeeShift_ID that has the selected Employee_ID to check if the selected employee has a delivery shift. The system will check if the delivery shift date is on the date of which the employee is assigned to do the delivery. The system will check that the Shift_ID in the EmployeeShift table has a Date_ID that has a DayOfTheWeek attribute that is from the current day and onwards. The system saves the Delivery_ID with the Employee_ID.	

[ALT Step 10a (delivery option)]: The system saves the new delivery assignment details and displays a success modal:

Input Name	Control Type	Notes
Success	Modal Heading	None
Saving of assigned order information has been successful.	Placeholder Text	Placeholder Text to indicate the order has been updated to be assigned to the employee on the system.
Ok	Button	None

[ALT Step 10b (delivery option)]: The order could not be assigned for delivery on the system. The system will display a modal:

Input Name	Control Type	Notes
Error	Placeholder Text	This serves as a header for the modal.
There was an error while saving the assigning information. Please try again later.	Placeholder Text	This is to show that the saving of the assignment information was unsuccessful.
OK	Button	None

[ALT Step 10c (delivery option)]: The system could not find an assigned delivery shift to the selected employee that is on the current date or a date onwards from the current date. The employee cannot be assigned to the delivery of the order. The system will display the following modal:

Input Name	Control Type	Notes
Error	Placeholder Text	This serves as a header for the modal.
Employee cannot be assigned to deliver order as they are not assigned to an appropriate shift.	Placeholder Text	This is to show that the order cannot be assigned to deliver the order as they are not assigned to a shift within the required date parameter.
OK	Button	None

[Alt Step 16(courier option)]: The order could not be assigned for delivery on the system. The system will display a modal:

Input Name	Control Type	Notes
Error	Placeholder Text	This serves as a header for the modal.

<table>
<tr><td></td><td><table><tr><td>There was an error while saving the assigning information. Please try again later.</td><td>Placeholder Text</td><td>This is to show that the saving of the assignment information was unsuccessful.</td></tr><tr><td>OK</td><td>Button</td><td>None</td></tr></table></td></tr>
<tr><td>CONCLUSION:</td><td>This use case concludes when the employee has been successfully added on the system.</td></tr>
<tr><td>POST-CONDITION:</td><td>The employee has been added to the Employee table.</td></tr>
<tr><td>BUSINESS RULES:</td><td>Only an admin can add a new employee to the system</td></tr>
<tr><td>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:</td><td>None</td></tr>
<tr><td>ASSUMPTIONS:</td><td>None</td></tr>
<tr><td>OPEN ISSUES:</td><td>None</td></tr>
</table>

<table>
<tr><th colspan="3">NKAP BOLTING SYSTEM</th></tr>
<tr><td>DATE:</td><td colspan="2">13-06-2021</td></tr>
<tr><td>VERSION:</td><td colspan="2">1.0</td></tr>
<tr><td>AUTHOR(S)</td><td colspan="2">Sonali Badrinath</td></tr>
<tr><td>USE CASE NAME:</td><td>Assign Delivery Shift</td><td>USE CASE TYPE</td></tr>
<tr><td>USE CASE ID:</td><td>4.10.</td><td rowspan="3">Business Requirements: ☐
System Analysis: ☐
System Design: ☑</td></tr>
<tr><td>PRIORITY:</td><td>High</td></tr>
<tr><td>SOURCE:</td><td>NKAP Bolting System Requirements</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR:</td><td colspan="2">Admin</td></tr>
<tr><td>PRIMARY THE SYSTEM ACTOR:</td><td colspan="2">None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="2">None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS :</td><td colspan="2">Employee</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="2">An Admin will be able to view the specific shift and assign it to an employee. The system will update the shift to be assigned to the employee on the shift schedule.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="2">• The admin should be on the Search Delivery Shift screen</td></tr>
<tr><td>TRIGGER:</td><td colspan="2">The admin requesting to assign a delivery shift.</td></tr>
<tr><td></td><td></td><td>SYSTEM RESPONSE:</td></tr>
</table>

		Manual Action	Automated Action
	Step 1: The admin wants to assign a delivery shift to an employee.		**Step 2**: The system invokes use case **4.2. Search Delivery Shift.**
	Step 3: The admin will click on the "Assign" button present in the Assign To column of the Delivery Table on the Search Delivery Shift Screen.		**Step 4:** The system responds by loading the "**Assign Delivery Shift**" modal with the following controls:
	Step 5: The admin selects the		

Control Name	Control Type	Notes
Assign Shift	Heading	None
Select Employee	Placeholder Text	None
Employee Name	Dropdown	None
Proceed	Button	This button is used for when the admin is done selecting the employee name of which to assign the shift.
Cancel	Button	This button is used when the admin decides to terminate the process of adding a new employee.

<table>
<tr><td></td><td>employee name.</td><td></td><td></td></tr>
<tr><td></td><td>Step 6: The admin clicks on the “Proceed” button [Alt]</td><td></td><td>Step 7: The system displays a modal, with the following controls:
<table>
<tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Confirm to Proceed</td><td>Placeholder Text</td><td>This serves as a header for the modal.</td></tr>
<tr><td>Would you like to proceed with assigning this employee to the selected delivery shift?</td><td>Placeholder Text</td><td>This is to allow the admin to finalize the decision to submit the assigning of the delivery shift.</td></tr>
<tr><td>Yes</td><td>Button</td><td>Proceeds to the next step (Step 8)</td></tr>
<tr><td>No</td><td>Button</td><td>Allows user to return to the Assign Shift screen.</td></tr>
</table></td></tr>
<tr><td></td><td>Step 8: The admin clicks on the “Yes” button to confirm the assigning of the delivery shift. [Alt]</td><td></td><td>Step 9: The system uses a SQL insert query to save the captured information in the EmployeeShift table [Alt]:
<table>
<tr><th>Information to be saved:</th><th>Attribute in table</th><th>Details</th></tr>
<tr><td>EmployeeShift_ID</td><td>EmployeeShift_ID</td><td>Generated by the system by reading the last EmployeeShift</td></tr>
</table></td></tr>
</table>

			_ID from the **EmployeeShift** table and adding it by one.
		Employee ID	Employee_ID	The system will retrieve the Employee_ID associated with the Employee_Name that was selected and save it as an attribute in the EmployeeShift table with the chosen Shift's ID.

Step 10: The system displays a success modal:

Input Name	Control Type	Notes
Success	Modal Heading	None
Delivery shift was successfully assigned.	Placeholder Text	Placeholder Text to indicate the delivery shift is assigned to the employee and saved on the system.
Ok	Button	None

ALTERNATE COURSES:	**[Alt Step 6]**: The admin clicks on the "Cancel" button which terminates this use case. **Return to step 2**
	[Alt Step 8]: The admin clicks on No button. **Return to Step 4.**
	[Alt Step 10]: Failed to assign employee to delivery shift. The system will display the following unsuccessful modal.

<table>
<tr><td></td><td>
<table>
<tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>UNSUCCESSFUL</td><td>Placeholder Text</td><td>This serves as a header for the modal.</td></tr>
<tr><td>Failed to assign delivery shift to employee. Please try again later.</td><td>Placeholder Text</td><td>This is to show that the assignment of the delivery shift information was unsuccessful.</td></tr>
<tr><td>OK</td><td>Button</td><td>None</td></tr>
</table>
</td></tr>
<tr><td>CONCLUSION:</td><td>This use case concludes when the employee has been successfully added on the system.</td></tr>
<tr><td>POST-CONDITION:</td><td>The employee has been added to the Employee table.</td></tr>
<tr><td>BUSINESS RULES:</td><td>Only an admin can add a new employee to the system</td></tr>
<tr><td>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:</td><td>None</td></tr>
<tr><td>ASSUMPTIONS:</td><td>None</td></tr>
<tr><td>OPEN ISSUES:</td><td>None</td></tr>
</table>

<table>
<tr><th colspan="5">NKAP BOLTING SYSTEM</th></tr>
<tr><td>DATE:</td><td colspan="4">13-06-2021</td></tr>
<tr><td>VERSION:</td><td colspan="4">1.0</td></tr>
<tr><td>AUTHOR(S)</td><td colspan="4">Sonali Badrinath</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Cancel Delivery Shift</td><td colspan="2">USE CASE TYPE</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">4.11.</td><td colspan="2" rowspan="3">Business Requirements: ☐
System Analysis: ☐
System Design: ☑</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">High</td></tr>
<tr><td>SOURCE:</td><td colspan="2">NKAP Bolting System Requirements</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR:</td><td colspan="4">Administrator</td></tr>
<tr><td>PRIMARY THE SYSTEM ACTOR:</td><td colspan="4">None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="4">None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="4">Employee</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="4">An Admin will be able to view the specific shift and cancel it. The system will update the assigned shift information using the EmployeeShift table.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="4">• The admin should be on the Search Delivery Shift screen</td></tr>
<tr><td>TRIGGER:</td><td colspan="4">The admin requesting to cancel an assigned delivery shift.</td></tr>
<tr><td rowspan="3">TYPICAL COURSE OF EVENTS:</td><td rowspan="2">ACTOR ACTION:</td><td colspan="3">SYSTEM RESPONSE:</td></tr>
<tr><td>Manual Action</td><td colspan="2">Automated Action</td></tr>
<tr><td>Step 1: The admin wants to cancel an assigned delivery shift.</td><td></td><td colspan="2">Step 2: The system invokes Use case 4.2. Search Delivery Shift.</td></tr>
<tr><td></td><td>Step 3: The admin requests to cancel the assigned</td><td></td><td colspan="2">Step 4: The system will display a confirmation popup to confirm the cancellation.</td></tr>
</table>

<table>
<tr>
<td></td>
<td>shift of an employee by selecting the “Cancel” option in the Assign To column that in the line of the delivery shift information of which they want to cancel.</td>
<td></td>
<td>
<table>
<tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Confirm to Proceed</td><td>Placeholder Text</td><td>This serves as a header for the modal.</td></tr>
<tr><td>Would you like to proceed with the cancellation of the selected assigned delivery shift?</td><td>Placeholder Text</td><td>This is to allow the admin to finalize the decision to submit the cancel the delivery shift.</td></tr>
<tr><td>Yes</td><td>Button</td><td>Proceeds to the next step (Step 6)</td></tr>
<tr><td>No</td><td>Button</td><td>Allows user to return to the Search Delivery Shift screen.</td></tr>
</table>
</td>
</tr>
<tr>
<td></td>
<td>Step 5: The admin clicks on the “Yes” button to confirm the cancellation of the delivery shift. [Alt]</td>
<td></td>
<td>
Step 6: The system uses a SQL delete query to remove the employee information from the EmployeeShift table [Alt]:
<table>
<tr><th>Information to

be deleted:</th><th>Attribute in table</th><th>Details</th></tr>
<tr><td>Employee ID</td><td>Employee_ID</td><td>The Employee_ID present in the EmployeeShift table is removed.</td></tr>
</table>
</td>
</tr>
</table>

<table>
<tr><td></td><td></td><td></td><td>Step 7: The system displays:
<table><tr><td>Input Name</td><td>Control Type</td><td>Notes</td></tr><tr><td>Success</td><td>Modal Heading</td><td>None</td></tr><tr><td>Cancellation of the assigned shift was successful.</td><td>Placeholder Text</td><td>Placeholder Text to indicate the cancellation is done on the system.</td></tr><tr><td>Ok</td><td>Button</td><td>None</td></tr></table></td></tr>
<tr><td></td><td></td><td></td><td>Step 8: The system will remove the employee name present in the Assign to column of the Delivery Shift table. [Alt]</td></tr>
<tr><td>ALTERNATE COURSES:</td><td colspan="3">[Alt Step 5]: The admin clicks on the “Cancel” button which terminates this use case.

Return to step 4</td></tr>
<tr><td></td><td colspan="3">[Alt Step 7]: The cancellation could not be performed. The system will display a modal informing the admin. The admin will Return to Step 3
<table><tr><td>Input Name</td><td>Control Type</td><td>Notes</td></tr><tr><td>Cannot be Cancelled</td><td>Placeholder Text</td><td>This serves as a header for the modal.</td></tr><tr><td>This delivery shift cannot be cancelled as it is not yet assigned to an employee. Please try again later.</td><td>Placeholder Text</td><td>This is to show that the admin that the delivery shift is not assigned and therefore cannot be cancelled.</td></tr><tr><td>Yes</td><td>Button</td><td>None</td></tr><tr><td>No</td><td>Button</td><td>None</td></tr></table></td></tr>
<tr><td>CONCLUSION:</td><td colspan="3">This use case concludes when the employee has been successfully added on the system.</td></tr>
</table>

POST-CONDITION:	• The employee no longer exists in the **EmployeeShift** table.
BUSINESS RULES:	Only an admin can add a new employee to the system
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	None
ASSUMPTIONS:	None
OPEN ISSUES:	None

<table>
<tr><th colspan="4">NKAP BOLTING SYSTEM</th></tr>
<tr><td>DATE:</td><td colspan="3">13-06-2021</td></tr>
<tr><td>VERSION:</td><td colspan="3">1.0</td></tr>
<tr><td>AUTHOR(S)</td><td colspan="3">Sonali Badrinath</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Cancel Delivery Shift</td><td>USE CASE TYPE</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">4.11.</td><td rowspan="3">Business Requirements: ☐
System Analysis: ☐
System Design: ☑</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">High</td></tr>
<tr><td>SOURCE:</td><td colspan="2">NKAP Bolting System Requirements</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR:</td><td colspan="3">Administrator</td></tr>
<tr><td>PRIMARY THE SYSTEM ACTOR:</td><td colspan="3">None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">Employee</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">An Admin will be able to view the specific shift and cancel it. The system will update the assigned shift information using the EmployeeShift table.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The admin should be on the Search Delivery Shift screen</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The admin requesting to cancel an assigned delivery shift.</td></tr>
<tr><td rowspan="3">TYPICAL COURSE OF EVENTS:</td><td rowspan="2">ACTOR ACTION:</td><td colspan="2">SYSTEM RESPONSE:</td></tr>
<tr><td>Manual Action</td><td>Automated Action</td></tr>
<tr><td>Step 1: The admin wants to cancel an assigned delivery shift.</td><td></td><td>Step 2: The system invokes Use case 4.2. Search Delivery Shift.</td></tr>
</table>

<table>
<tr>
<td></td>
<td>Step 3: The admin requests to cancel the assigned shift of an employee by selecting the “Cancel” option in the Assign To column that in the line of the delivery shift information of which they want to cancel.</td>
<td></td>
<td>Step 4: The system will display a confirmation popup to confirm the cancellation.
<table>
<tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Confirm to Proceed</td><td>Label</td><td>This serves as a header for the modal.</td></tr>
<tr><td>Would you like to proceed with the cancellation of the selected assigned delivery shift?</td><td>Label</td><td>This is to allow the admin to finalize the decision to submit the cancel the delivery shift.</td></tr>
<tr><td>Yes</td><td>Button</td><td>Proceeds to the next step (Step 6)</td></tr>
<tr><td>No</td><td>Button</td><td>Allows user to return to the Search Delivery Shift screen.</td></tr>
</table>
</td>
</tr>
<tr>
<td></td>
<td>Step 5: The admin clicks on the “Yes” button to confirm the cancellation of the delivery shift. [Alt]</td>
<td></td>
<td>Step 6: The system uses a SQL delete query to remove the employee information from the EmployeeShift table [Alt]:
<table>
<tr><th>Information to be deleted:</th><th>Attribute in table</th><th>Details</th></tr>
<tr><td>Employee ID</td><td>Employee_ID</td><td>The Employee_ID present in the EmployeeShi</td></tr>
</table>
</td>
</tr>
</table>

			ft table is removed.
			Step 7: The system displays: Input Name / Control Type / Notes Success / Modal Heading / None Cancellation of the assigned shift was successful. / Label / Label to indicate the cancellation is done on the system. Ok / Button / None
			Step 8: The system will remove the employee name present in the Assign to column of the Delivery Shift table. **[Alt]**
ALTERNATE COURSES:	**[Alt Step 5]**: The admin clicks on the "Cancel" button which terminates this use case. Return to step 4		
	[Alt Step 7]: The cancellation could not be performed. The system will display a modal informing the admin. The admin will **Return to Step 3**		

Input Name	Control Type	Notes
Cannot be Cancelled	Label	This serves as a header for the modal.
This delivery shift cannot be cancelled as it is not yet assigned to an employee. Please try again later.	Label	This is to show that the admin that the delivery shift is not assigned and therefore cannot be cancelled.
Yes	Button	None
No	Button	None

CONCLUSION:	This use case concludes when the employee has been successfully added on the system.

POST-CONDITION:	• The employee no longer exists in the **EmployeeShift** table.
BUSINESS RULES:	Only an admin can add a new employee to the system
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	None
ASSUMPTIONS:	None
OPEN ISSUES:	None

2.1.5. ADMINISTRATION BACK-OFFICE SUBSYSTEM

<table>
<tr><td colspan="4">NKAP BOLTING SYSTEM</td></tr>
<tr><td>DATE:</td><td colspan="3">05-06-21</td></tr>
<tr><td>VERSION:</td><td colspan="3">1.0</td></tr>
<tr><td>AUTHOR(S)</td><td colspan="3">Divya Bagratee</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Search Online Sales</td><td>USE CASE TYPE</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">5.1</td><td>Business Requirements: ☐</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">High</td><td>System Analysis: ☐</td></tr>
<tr><td>SOURCE:</td><td colspan="2">NKAP Bolting System Requirements</td><td>System Design: ☑</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR:</td><td colspan="3">Employee/Admin</td></tr>
<tr><td>PRIMARY THE SYSTEM ACTOR:</td><td colspan="3">None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case describes the event where an employee would like to view a Sale. The employee/admin will select the Administration Back Office Menu item on the navigation bar and click the “Search Online Sales” dropdown menu item. The system will display the Search Online Sale screen and request the search criteria from the employee/admin. The system will search for a matching record in the database and display it to the employee/admin in a datagrid view. The use case ends when the system successfully displays the Sale to the admin/employee.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The Sale must exist on the system.
• The employee/admin must be logged in</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">An Employee/Admin wants to search for a Sale Order</td></tr>
<tr><td rowspan="3">TYPICAL COURSE OF EVENTS:</td><td rowspan="2">ACTOR ACTION:</td><td colspan="2">SYSTEM RESPONSE:</td></tr>
<tr><td>Manual Action</td><td>Automated Action</td></tr>
<tr><td>Step 1: An employee/admin wants to search for an online Sale.</td><td></td><td></td></tr>
<tr><td></td><td>Step 2: The employee/admin will select the Administration Back Office Menu item on the navigation bar and click the “Search Online</td><td></td><td>Step 3: The system will display the Search Online Sale screen with the following html components and enables the input controls.<table><tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr></table></td></tr>
</table>

	Sales” dropdown menu item.		

Search Online Sales	Heading	None
Filter	Label	Label for filter
Filter	dropdown	Enables the employee/admin to select the filter they want to search for an online sale by
Select Date	Label	Displayed if the employee/admin chooses the search by date filter option.
Calendar	Calendar	Enables the employee/admin to choose the date. Displayed if the search by date filter option was chosen.
Select Month	Label	Displayed if the employee admin chooses the search by month filter option.
Month	Dropdown	Enables the employee/admin to choose the month. Displayed if the search by month filter option was chosen.
Select Order Status	Label	Displayed if the employee/admin chooses the search by order status filter option.
Order Status	Dropdown	Enables the employee/admin to choose the Order Status. Displayed if the search by order status filter option was chosen.
Search	Button	Button when clicked will search for a matching record in the database.

<table>
<tr><td></td><td></td><td></td><td><table><tr><td>Cancel</td><td>Button</td><td>Button when clicked will terminate the search process.</td></tr></table></td></tr>
<tr><td></td><td>Step 4: The employee/admin selects the filter option and click “Search” Button.</td><td></td><td>Step 5: The system makes use of a SQL read query to retrieve the matching Sale records from the <u>Sale Entity</u>.</td></tr>
<tr><td></td><td></td><td></td><td>Step 6: The system will display the Online Sales in a datagrid view with the following html components:<table>
<tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Online Sales</td><td>Heading</td><td>None</td></tr>
<tr><td>Sale Number</td><td>Table Column 1</td><td>Column in the data grid view</td></tr>
<tr><td>Sale Date</td><td>Table Column 2</td><td>Column in the data grid view</td></tr>
<tr><td>Order Status</td><td>Table Column 3</td><td>Column in the data grid view</td></tr>
<tr><td>Actions</td><td>Table Column 4</td><td>Column in the data grid view</td></tr>
<tr><td>View</td><td>Button</td><td>Button when clicked will view the full details of a specific sale.</td></tr>
<tr><td>Back</td><td>Button</td><td>None</td></tr>
</table></td></tr>
<tr><td></td><td>Step 7: The admin selects the sale they want to view by clicking the “View” button.</td><td></td><td>Step 8: The system displays the View Sale Modal with the following html components:<table>
<tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>View Sale</td><td>Heading</td><td>None</td></tr>
<tr><td>Sale Number</td><td>Label</td><td>Label for Sale Number</td></tr>
<tr><td>Sale Number</td><td>Text</td><td>Sale_ID retrieved from <u>Sale Entity</u></td></tr>
<tr><td>Sale Date</td><td>Label</td><td>Label for Sale Date</td></tr>
<tr><td>Sale Date</td><td>Text</td><td>SaleOrder_Date retrieved from <u>Sale Entity</u></td></tr>
</table></td></tr>
</table>

Order Status	Label	Label for Order Status
Order Status	Text	OrderStatus_ID retrieved from **Sale Entity**
Customer	Label	Label for Customer
Customer	Text	Customer_ID retrieved from **Sale Entity**
Payment Type	Label	Label for Payment Type
Payment Type	Text	PaymentType_ID retrieved from **Sale Entity**
Payment Date	Label	Label for Payment Date
Payment Date	Text	Payment_Date retrieved from **Sale Entity**
Payment Amount	Label	Label for Payment Amount
Payment Amount	Text	Payment_Amount retrieved from **Sale Entity**
Recieval Type	Label	Label for Recieval Type
Recieval Type	Text	SaleOrder_ RecievalType from **Sale Entity**
Items	Label	Label for Items Table
Items	Column 1 in table	ProductItem_Name from the **ProductItem Entity**
Quantity	Column 2 in table	SaleLine_Quantity from the **SaleLine Entity**
Back	Button	Button when clicked will return to the Sale Screen.
Pack Order	Button	Button when clicked allows the order Status to be updated.

ALTERNATE COURSES:	ALT-STEP 6: The system fails to find matching records in the database and displays an error modal: **Input Name** / **Control Type** / **Notes** **Error** / Modal Heading / None **No sales matching the search criteria was found.** / Label / Label to indicate the system failed to find a matching Sale record in the database. **Ok** / Button / None
	ALT-STEP 8: The system fails to display the sale and displays an error modal: **Input Name** / **Control Type** / **Notes** **Error** / Modal Heading / None **There was an unexpected error when trying to display the sales.** / Label / Label to indicate the system failed to display the Sales. **Ok** / Button / None
CONCLUSION:	The use case concludes when the system has successfully displayed the sales.
POST-CONDITION:	• The employee/admin is able to view the sales.
BUSINESS RULES:	1. Only Employees and Admins can search for a Sale.
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	None
ASSUMPTIONS:	None
OPEN ISSUES:	None

<table>
<tr><td colspan="5">NKAP BOLTING SYSTEM</td></tr>
<tr><td>DATE:</td><td colspan="4">05-06-21</td></tr>
<tr><td>VERSION:</td><td colspan="4">1.0</td></tr>
<tr><td>AUTHOR(S)</td><td colspan="4">Divya Bagratee</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Pack Order</td><td colspan="2">USE CASE TYPE</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">5.2</td><td colspan="2" rowspan="3">Business Requirements: ☐
System Analysis: ☐
System Design: ☑</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">High</td></tr>
<tr><td>SOURCE:</td><td colspan="2">NKAP Bolting System Requirements</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR:</td><td colspan="4">Employee</td></tr>
<tr><td>PRIMARY THE SYSTEM ACTOR:</td><td colspan="4">None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="4">None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="4">None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="4">This use case describes the event where an employee searches for orders which require packing and proceeds to pack the order. Once the employee has packed the order, they will indicate so on the system and the system will update the order status of the order appropriately in the Sale Entity. The system will notify the customer that their order is ready for collection.The use case concludes when the Order has been packed and the status has been updated accordingly.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="4">• The Employee must be logged in to the system.
• The sale must exist on the system and have an order status of “Needs Packing”.</td></tr>
<tr><td>TRIGGER:</td><td colspan="4">An employee wants to pack an order.</td></tr>
<tr><td rowspan="5">TYPICAL COURSE OF EVENTS:</td><td rowspan="2">ACTOR ACTION:</td><td colspan="3">SYSTEM RESPONSE:</td></tr>
<tr><td>Manual Action</td><td colspan="2">Automated Action</td></tr>
<tr><td>Step 1: An employee wants to pack an order.</td><td></td><td colspan="2"></td></tr>
<tr><td>Step 2: The employee will select the Administration Back Office Menu item on the navigation bar and click the “Pack Order” dropdown menu item.</td><td></td><td colspan="2">Step 3: The system invokes Use Case 5.1 Search Online Sales.</td></tr>
<tr><td>Step 4: The employee clicks the “Pack Order” button.</td><td></td><td colspan="2">Step 5: The system displays the Pack Order screen with the following html components and enables the input controls:</td></tr>
</table>

<table>
<tr><td></td><td></td><td></td><td>
<table>
<tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Pack Order</td><td>Heading</td><td>None</td></tr>
<tr><td>Has this order been packed?</td><td>Label</td><td>Label to prompt confirmation that the order has been packed.</td></tr>
<tr><td>Has this order been packed?</td><td>Checkbox</td><td>Checked by employee to confirm order is packed</td></tr>
<tr><td>For Delivery</td><td>Label</td><td>Label to indicate order is for delivery.</td></tr>
<tr><td>For Delivery</td><td>Checkbox</td><td>Checked by employee if order is for delivery</td></tr>
<tr><td>For Collection</td><td>Label</td><td>Label to indicate order is for Collection</td></tr>
<tr><td>For Collection</td><td>Checkbox</td><td>Checked by employee if order is for collection.</td></tr>
<tr><td>Confirm</td><td>Button</td><td>Button when clicked will update the order status appropriately.</td></tr>
<tr><td>Cancel</td><td>Button</td><td>Button when clicked will terminate the Use Case</td></tr>
</table>
</td></tr>
<tr><td></td><td>Step 6: The employee confirms the order has been packed and that the order is for collection and clicks the “Confirm” button.</td><td></td><td>
Step 7: The system updates Order Status to “Ready for Collection” by making use of a SQL Update query:
<table>
<tr><th>Details to be Updated</th><th>Attribute in Entity</th><th>Notes</th></tr>
<tr><td>Order Status</td><td>OrderStatus_ID in the <u>Sale Entity</u></td><td>None</td></tr>
</table>
</td></tr>
</table>

			Step 8: The system notifies the customer via SMS that their order is ready for collection by retrieving the • Customer_CellPhoneNumber from the <u>Customer Entity.</u>
			Step 9: The system successfully completes the Pack Order Process and displays a success Modal:

Input Name	Control Type	Notes
Success	Modal Heading	None
Order has been Packed.	Label	Label to indicate the Order was packed.
Ok	Button	None

ALTERNATE	ALT-STEP 6: The admin clicks the “Cancel” Button. Terminate Use Case.
COURSES:	ALT STEP 7: The order is for delivery. The system updates the Order Status to “Ready for Delivery” by making use of a SQL Update query. Proceed to Step 9.

Details to be Updated	Attribute in Entity	Notes
Order Status	OrderStatus_ID in the **<u>Sale Entity</u>**	None

ALT-STEP 9: The system fails to update the Order Status and displays an error modal:

Input Name	Control Type	Notes
Error	Modal Heading	None
There was an unexpected error when trying to complete the pack order process.	Label	Label to indicate the system failed to complete the pack order process.

	Ok | Button | None
CONCLUSION:	The use case concludes when the Order has been packed and the status has been updated accordingly
POST-CONDITION:	• The Order Status has been updated
BUSINESS RULES:	1. Once an order has been packed and the status has been updated, it cannot be edited.
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	None
ASSUMPTIONS:	None
OPEN ISSUES:	None

<table>
<tr><td colspan="4">NKAP BOLTING SYSTEM</td></tr>
<tr><td>DATE:</td><td colspan="3">05-06-21</td></tr>
<tr><td>VERSION:</td><td colspan="3">1.0</td></tr>
<tr><td>AUTHOR(S)</td><td colspan="3">Divya Bagratee</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Do Stock-Take</td><td>USE CASE TYPE</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">5.3</td><td>Business Requirements: ☐</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">High</td><td>System Analysis: ☐</td></tr>
<tr><td>SOURCE:</td><td colspan="2">NKAP Bolting System Requirements</td><td>System Design: ☑</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR:</td><td colspan="3">Admin</td></tr>
<tr><td>PRIMARY THE SYSTEM ACTOR:</td><td colspan="3">None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This Use Case Describes the Event where an Admin conducts a stock take for an item. The admin will select the Administration Back Office Menu item on the navigation bar and click the “Do Stock-Take” dropdown menu item. The admin will record the stock take date in the StockTake Entity and the quantity of the stock in the ProductItem_StockTake Entity. The system will then adjust the quantity on hand for the item in the ProductItem Entity. The Use Case Ends when the system has successfully completed the stock take.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The Admin must know the value of stock on hand for an item
• The Admin must be logged in</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">An Admin wants to conduct a stock-take.</td></tr>
<tr><td rowspan="4">TYPICAL COURSE OF EVENTS:</td><td rowspan="2">ACTOR ACTION:</td><td colspan="2">SYSTEM RESPONSE:</td></tr>
<tr><td>Manual Action</td><td>Automated Action</td></tr>
<tr><td>Step 1: An admin wants to do a Stock-Take</td><td></td><td></td></tr>
<tr><td>Step 2: The admin will select the Administration Back Office Menu item on the navigation bar and click the “Do Stock-Take” dropdown menu item.</td><td></td><td>Step 3: The system invokes Use Case 9.10 Search Product</td></tr>
</table>

<table>
<tr>
<td></td>
<td>Step 4: The admin clicks the “Do Stock-Take Button”</td>
<td></td>
<td>Step 5: The system displays the Do-stock Take Modal with the following html components and enables the input controls:
<table>
<tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Stock-Take</td><td>Modal Heading</td><td>None</td></tr>
<tr><td>Date</td><td>Label</td><td>Label for Stock Take Date</td></tr>
<tr><td>Stock Take Date</td><td>Date Picker</td><td>Enables the admin to select the date of the stock take.</td></tr>
<tr><td>Quantity</td><td>Textbox</td><td>Enables the admin to enter the stock take quantity</td></tr>
<tr><td>Save</td><td>Button</td><td>Button when clicked proceeds with the stock-take</td></tr>
<tr><td>Cancel</td><td>Button</td><td>Button when clicked terminates the stock-take process</td></tr>
</table>
</td>
</tr>
<tr>
<td></td>
<td>Step 6: The admin provides the stock-take details and clicks the “Save” button.</td>
<td></td>
<td>Step 7: The system validates the stock take details according to the validation requirements of the relevant entities:
<table>
<tr><th>Input Name</th><th>Attribute in Entity</th><th>Validation Requirements</th></tr>
<tr><td>Stock Take Date</td><td>StockTake_Date in <u>StockTake Entity</u></td><td>Required
Must be in Date Format</td></tr>
<tr><td>Stock Take Quantity</td><td>StockTake_Quantity in <u>ProductItem_StockTake</u></td><td>Required
Integers only</td></tr>
</table>
</td>
</tr>
</table>

<table>
<tr>
<td></td>
<td></td>
<td></td>
<td>Step 8: The system requests confirmation for the stock take from the admin and displays a confirmation modal.
<table>
<tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Confirmation</td><td>Modal Heading</td><td>None</td></tr>
<tr><td>Do you want to proceed with the stock take?</td><td>Label</td><td>Label to prompt confirmation for the stock take</td></tr>
<tr><td>Confirm</td><td>Button</td><td>Button when clicked will proceed with the stock take</td></tr>
<tr><td>Cancel</td><td>Button</td><td>Button when clicked will terminate the stock take process.</td></tr>
</table>
</td>
</tr>
<tr>
<td></td>
<td>Step 9: The admin clicks the “Confirm” button.</td>
<td></td>
<td>Step 10: The system s makes use of a SQL insert query to save the stock take details in the relevant entities:
<table>
<tr><th>Details to be saved</th><th>Attribute in Entity</th><th>Notes</th></tr>
<tr><td>Stock Take ID</td><td>StockTake_ID in <u>StockTake Entity</u></td><td>The system will generate a unique StockTake_ID by retrieving the previous StockTake_ID and incrementing it by on1.</td></tr>
<tr><td>Stock Take Date</td><td>StockTake_Date in <u>StockTake Entity</u></td><td>None</td></tr>
<tr><td>Stock Take Quantity</td><td>StockTake_Quantity in <u>ProductItem_</u> <u>StockTake</u></td><td>None</td></tr>
</table>
</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Step 11: The system will update the</td>
</tr>
</table>

			• Quantity_on_hand in the ProductItem Entity for the specific product.
			Step 12: The system successfully completes the stock take and displays a success modal: **Input Name** / **Control Type** / **Notes** **Success** / Modal Heading / None **Stock Take was completed successfully.** / Label / Label to indicate the system successfully completed the stock take. **Ok** / Button / None
ALTERNATE	ALT-STEP 9: The Admin clicks the “cancel” button. Terminate Use Case.		
	ALT-STEP 7: The system fails to validate the stock take details and displays a Validation Error Modal:		

Input Name	Control Type	Notes
Validation Error	Modal Heading	None
The information entered is in the incorrect format. Please try again.	Label	None.
Ok	Button	None

COURSES:

ALT-STEP 12: The system fails to complete the stock-take and displays an error modal:

Input Name	Control Type	Notes
Error	Modal Heading	None
There was an unexpected error when trying to do the stock-take,	Label	Label to indicate the system failed to do the stock-take.
Ok	Button	None

CONCLUSION:	The Use Case Concludes when the system has successfully completed the stock take.
POST-CONDITION:	• The quantity on hand for a product is accurate.

BUSINESS RULES:	1. Only an admin can do a stock take 2. The admin must be logged in.
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	None
ASSUMPTIONS:	None
OPEN ISSUES:	None

NKAP BOLTING SYSTEM				
DATE:	06-06-21			
VERSION:	1.0			
AUTHOR(S)	Divya Bagratee			
USE CASE NAME:	View Delivery Shift Schedule	USE CASE TYPE		
USE CASE ID:	5.4	Business Requirements:	☐	
PRIORITY:	High	System Analysis:	☐	
SOURCE:	NKAP Bolting System Requirements	System Design:	☑	
PRIMARY BUSINESS ACTOR:	Employee			
PRIMARY THE SYSTEM ACTOR:	None			
OTHER PARTICIPATING ACTORS:	None			
OTHER INTERESTED STAKEHOLDERS:	None			
DESCRIPTION:	This use case describes the event where employee views the Shift Schedule. The Employee will select the Administration Back Office Menu item on the navigation bar and click the “View Shift Schedule” dropdown menu item. The system will retrieve the shift schedule information from the <u>Shift Entity</u> and <u>EmployeeShift Entity.</u> The use case ends when the system has successfully displayed the Shift Schedule to the employee.			
PRE-CONDITION:	• The employee must be logged in			
TRIGGER:	An Employee wants to view the shift schedule			
TYPICAL COURSE OF EVENTS:	ACTOR ACTION:	SYSTEM RESPONSE:		
		Manual Action	Automated Action	
	Step 1: An employee wants to view the Delivery shift schedule.			

<table>
<tr><td></td><td>Step 2: The employee will select the Administration Back Office Menu item on the navigation bar and click the “View Delivery Shift Schedule” dropdown menu item.</td><td></td><td>Step 3: The system makes use of a SQL Read query to retrieve the Delivery Shift Schedule information from the <u>Shift Entity</u> and <u>EmployeeShift Entity.</u></td></tr>
<tr><td></td><td></td><td></td><td>Step 4: The system displays the Shift Schedule screen with the following html components:
<table>
<tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Schedule</td><td>Heading</td><td>None</td></tr>
<tr><td>Employee</td><td>Column 1 in table</td><td>Employee_ID retrieved from the <u>Employee_Shift Entity.</u>
Shows the Employee Names.</td></tr>
<tr><td>Monday</td><td>Column 2 in table</td><td>Date_ID retrieved from the <u>Shift Entity.</u>
Shows the day of the week.</td></tr>
<tr><td>Tuesday</td><td>Column 3 in table</td><td>Date_ID retrieved from the <u>Shift Entity.</u>
Shows the day of the week.</td></tr>
<tr><td>Wednesday</td><td>Column 4 in table</td><td>Date_ID retrieved from the <u>Shift Entity.</u>
Shows the day of the week.</td></tr>
<tr><td>Thursday</td><td>Column 5 in table</td><td>Date_ID retrieved from the <u>Shift Entity.</u>
Shows the day of the week.</td></tr>
<tr><td>Friday</td><td>Column 6 in table</td><td>Date_ID retrieved from the <u>Shift Entity.</u>
Shows the day of the week.</td></tr>
</table></td></tr>
</table>

			Column Rows	Column Rows	Time_ID retrieved from the **Shift Entity.** Shows the time slot the employee works on a specific day
ALTERNATE COURSES:	ALT-STEP 4: The system fails to display the shift schedule and displays an error modal:				

Input Name	Control Type	Notes
Error	Modal Heading	None
There was an unexpected error when trying to display the Delivery Shift Schedule.	Label	Label to indicate the system failed to display the shift schedule.
Ok	Button	None

CONCLUSION:	The Use Case Concludes when the system has successfully displayed the shift schedule to the employee
POST-CONDITION:	• The employee knows when their next shift is
BUSINESS RULES:	1. Only an Employee and admin can view the shift schedule
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	None
ASSUMPTIONS:	None
OPEN ISSUES:	None

NKAP BOLTING SYSTEM			
DATE:	06-06-21		
VERSION:	1.0		
AUTHOR(S)	Divya Bagratee		
USE CASE NAME:	Write-off Stock	USE CASE TYPE	
USE CASE ID:	5.5	Business Requirements:	☐
PRIORITY:	High	System Analysis:	☐
SOURCE:	NKAP Bolting System Requirements	System Design:	☑
PRIMARY BUSINESS ACTOR:	Employee		

<table>
<tr><td>PRIMARY THE SYSTEM ACTOR:</td><td colspan="3">None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case describes the event where the Admin wants to write off stock. The admin will select the Administration Back Office Menu item on the navigation bar and click the “Write-Off Stock” dropdown menu item. The system will write-off stock according to the write off information provided by the admin and update the relevant entities.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">The admin must be logged in to the system</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">An admin wants to write off stock.</td></tr>
<tr><td rowspan="5">TYPICAL COURSE OF EVENTS:</td><td rowspan="2">ACTOR ACTION:</td><td colspan="2">SYSTEM RESPONSE:</td></tr>
<tr><td>Manual Action</td><td>Automated Action</td></tr>
<tr><td>Step 1: An admin wants to write off stock.</td><td></td><td></td></tr>
<tr><td>Step 2: The admin will select the Administration Back Office Menu item on the navigation bar and click the “Write-Off Stock” dropdown menu item.</td><td></td><td>Step 3: The system will invoke use case 9.10 Search Product</td></tr>
<tr><td>Step 4: The admin will click the “write-off stock” button.</td><td></td><td>Step 5: The system will display the Write Off Stock screen with the following html components:
<table>
<tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Write-Off Stock</td><td>Heading</td><td>None</td></tr>
<tr><td>Date</td><td>Label</td><td>Label for Write-Off Date</td></tr>
<tr><td>Date</td><td>Date Picker</td><td>Enables admin to select write-off date</td></tr>
<tr><td>Quantity</td><td>Textbox</td><td>Enables admin to enter write-off quantity</td></tr>
<tr><td>Reason</td><td>Textbox</td><td>Enables admin to enter write-off reason</td></tr>
</table></td></tr>
</table>

			<table><tr><td>**Save**</td><td>Button</td><td>Button when clicked proceeds with write-off process</td></tr><tr><td>**Cancel**</td><td>Button</td><td>Button when clicked terminates write-off process</td></tr></table>
	Step 6: The admin provides the write-off information and clicks the "save" button.		Step 7: The system validates the write off information according to the validation requirements of the relevant entities:<table><tr><th>Input Name</th><th>Attribute in Entity</th><th>Validation Requirements</th></tr><tr><td>**Write-off quantity**</td><td>WriteOff_Quantity in **ProductItem_ WriteOffStock Entity**</td><td>• Required • Integers only</td></tr><tr><td>**Write-off Reason**</td><td>WriteOff_Reason in **ProductItem_ WriteOffStock Entity**</td><td>• Required • Alphabet only</td></tr><tr><td>**Write-off Date**</td><td>WriteOff_Date. in **Write-Off Entity**</td><td>• Required • Must be in Date Format</td></tr></table>
			Step 8: The system prompts the admin to confirm the write-off and displays a confirmation modal:<table><tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>**Confirmation**</td><td>Modal Heading</td><td>None</td></tr><tr><td>**Do you want to proceed with the Write-off?**</td><td>Label</td><td>Label to prompt confirmation for the stock take</td></tr><tr><td>**Confirm**</td><td>Button</td><td>Button when clicked will proceed with the write-off</td></tr><tr><td>**Cancel**</td><td>Button</td><td>Button when clicked will terminate the write-off process.</td></tr></table>

<table>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td>Step 9: The admin will click the “Confirm” button.</td><td></td><td>Step 10: The system makes use of a SQL insert query to save the write-off details in the relevant entities:
<table>
<tr><th>Details to be Saved</th><th>Attribute in Entity</th><th>None</th></tr>
<tr><td>Write-Off ID</td><td>WriteOff_ID in <u>Write-Off Entity</u></td><td>The system will generate a unique Write-off_ID by retrieving the previous Write-Off_ID and incrementing it by one.</td></tr>
<tr><td>Write-off quantity</td><td>WriteOff_Quantity in <u>ProductItem_ WriteOffStock Entity</u></td><td>None</td></tr>
<tr><td>Write-off Reason</td><td>WriteOff_Reason in <u>ProductItem_ WriteOffStock Entity</u></td><td>None</td></tr>
<tr><td>Write-off Date</td><td>WriteOff_Date. in <u>Write-Off Entity</u></td><td>None</td></tr>
</table></td></tr>
<tr><td></td><td></td><td></td><td>Step 11: The system decreases the Quantity_On_Hand in the ProductItem Entity by the write-off quantity specified by the admin.</td></tr>
<tr><td></td><td></td><td></td><td>Step 12: The system successfully completes the Write-off and displays a success modal:
<table>
<tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Success</td><td>Modal Heading</td><td>None</td></tr>
<tr><td>Write-Off was completed successfully.</td><td>Label</td><td>Label to indicate the system successfully completed the write-off.</td></tr>
</table></td></tr>
</table>

<table>
<tr><td></td><td><table><tr><td></td><td></td><td>Ok</td><td>Button</td><td>None</td></tr></table></td></tr>
<tr><td rowspan="3">ALTERNATE
COURSES:</td><td>ALT-STEP 9: The admin clicks the “cancel” button. Terminate Use Case.</td></tr>
<tr><td>ALT-STEP 7: The system fails to validate the write-off details and displays a validation error modal:
<table>
<tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Validation Error</td><td>Modal Heading</td><td>None</td></tr>
<tr><td>The information entered is in the incorrect format. Please try again.</td><td>Label</td><td>None.</td></tr>
<tr><td>Ok</td><td>Button</td><td>None</td></tr>
</table></td></tr>
<tr><td>ALT-STEP 12: The system fails to complete the write-off and displays an error modal:
<table>
<tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Error</td><td>Modal Heading</td><td>None</td></tr>
<tr><td>There was an unexpected error when trying to do the write-off,</td><td>Label</td><td>Label to indicate the system failed to do the write-off.</td></tr>
<tr><td>Ok</td><td>Button</td><td>None</td></tr>
</table></td></tr>
<tr><td>CONCLUSION:</td><td>There is record of the stock being written-off and the stock is no longer available for sale.</td></tr>
<tr><td>POST-CONDITION:</td><td>• The stock is written-off and is not available for sale anymore.</td></tr>
<tr><td>BUSINESS RULES:</td><td>1. Only an Admin can write-off stock.</td></tr>
<tr><td>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:</td><td>None</td></tr>
<tr><td>ASSUMPTIONS:</td><td>None</td></tr>
<tr><td>OPEN ISSUES:</td><td>None</td></tr>
</table>

2.1.6. CUSTOMER SUBSYSTEM

NKAP Bolting System

Author (s): Chardé Devine Vries **Date:06/06/2021**

Version:

___1.0___

USE CASE NAME:	**View Products**	**USE CASE TYPE**	
USE CASE ID:	**6.1**	**Business Requirements:**	☐
PRIORITY:	High	**System Analysis:**	☐
SOURCE:	NKAP Bolting system	**System Design:**	☑
PRIMARY BUSINESS ACTOR:	Customer		
PRIMARY THE SYSTEM ACTOR:	None		
OTHER PARTICIPATING ACTORS:	None		
OTHER INTERESTED STAKEHOLDERS:	None		
DESCRIPTION:	This use case describes the event where a customer would like to browse all available products. The use case starts when the customer clicks on the products tab, they can invoke the "view product details" use cases in order to select a product. The system will retrieve information from the "ProductItem" entity. The system displays all the relevant information. The customer is able to view and browse through all available product.		
PRE-CONDITION:	• The customer must have an account and be logged into the system		
TRIGGER:	A user wishes to browse the products on the system		

TYPICAL COURSE OF EVENTS:		**SYSTEM RESPONSE:**	
	ACTOR ACTION:	**MANUAL ACTION**	**AUTOMATED ACTION**
	Step 1: The customer request to browse through the products by category.		

<table>
<tr>
<td></td>
<td>Step 2: The customer clicks on the ‘Products’ tab on the menu-strip and selects a product category from the menu-items:
Anchors
Bolts
Clamps
Fasteners
Nuts
Pins
Rivets
Screws
Shackles
Thread Rods
Turnbuckle
Wall Fixers
Washers</td>
<td></td>
<td>Step 3: The system loads the Products screen with the following elements(showing the selected product category):
<table>
<tr><th>Element Name</th><th>Element Type</th><th>Notes:</th></tr>
<tr><td>Products</td><td>Modal Heading</td><td>To indicate what the screen is for</td></tr>
<tr><td>lblSearch</td><td>Label</td><td>This is the label to prompt the user to enter a search query.</td></tr>
<tr><td>txtSearch</td><td>Textbox</td><td>To enter the search criteria</td></tr>
<tr><td>btnSearch icon</td><td>Icon button</td><td>This will submit the search criteria</td></tr>
<tr><td>cardProductItems</td><td>Card</td><td>To show each product item. Each card contains:
ProductItem_Name
ProductItem_Image
Retrieved from the <u>Productitem entity</u></td></tr>
<tr><td colspan="3">The following elements are in cardProductItems:</td></tr>
<tr><td>imgProductItem</td><td>Img</td><td>To display the product item image</td></tr>
<tr><td>lblProductItemName</td><td>Label</td><td>To display the name of the Product Item</td></tr>
<tr><td>lblProductItem_Cost</td><td>Label</td><td>To display the cost of the product item</td></tr>
</table></td>
</tr>
</table>

<table>
<tr><td></td><td></td><td></td><td><table><tr><td>btnAddCartItem</td><td>Button</td><td>To add an item to the cart.</td></tr></table>
[ALT]</td></tr>
<tr><td>ALTERNATE COURSES:</td><td colspan="3">ALT Step 3: There system fails to load the Products view:
<table><tr><th>Element Name</th><th>Element Type</th><th>Notes</th></tr><tr><td>Error</td><td>Label</td><td>This serves as a header for the modal.</td></tr><tr><td>There was an error loading the Products page.

Please try again later.</td><td>Label</td><td>This is to show that loading the Products view was information was unsuccessful.</td></tr><tr><td>OK</td><td>Button</td><td>None</td></tr></table>Use Case Terminated</td></tr>
<tr><td>CONCLUSION:</td><td colspan="3">The use case concludes when the relevant product details are displayed successfully.</td></tr>
<tr><td>POST-CONDITION:</td><td colspan="3">The customer can see the Products screen</td></tr>
<tr><td>BUSINESS RULES:</td><td colspan="3">None</td></tr>
<tr><td>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:</td><td colspan="3">None</td></tr>
<tr><td>ASSUMPTIONS:</td><td colspan="3">None</td></tr>
<tr><td>OPEN ISSUES:</td><td colspan="3">None</td></tr>
</table>

NKAP Bolting System

Author (s): **Chardé Devine Vries** **Date:06/06/2021**

Version:

___1.0___

Use Case Name:	**Search Products Details**	**USE CASE TYPE**	
Use Case ID:	**6.2**	**Business Requirements:**	☐
Priority:	High	**System Analysis:**	☐
Source:	NKAP Bolting system	**System Design:**	☑
Primary Business Actor:	Customer		
Primary The system Actor:	None		
Other Participating Actors:	None		
Other Interested Stakeholders:	None		
Description:	This use case describes the process where the customer searches for a specific Product. Once the customer enters the search criteria the system will retrieve the matching results from the **ProductItem entity** the use case ends once the retrieved results have been displayed to the admin.		
Pre-condition:	• The customer must have an account and be logged into the system		
Trigger:	A user wishes to browse the products on the system		

Typical Course of Events:	**Actor Action:**	**System Response:**	
		MANUAL ACTION	**AUTOMATED ACTION**
	Step 1: The customer requests to search for a product.		**Step 2:** The system invokes Use Case 6.1 Search Products.
	Step 3: The customer enters the product search criteria into the search textbox.		

<table>
<tr><td></td><td>Step 4: The customer clicks on the search icon.</td><td></td><td>Step 5: The system captures and validates the information against the <u>Productitem table</u>
<table><tr><th>Input Name</th><th>Attribute Type</th><th>Validation Requirements</th></tr><tr><td>Name:</td><td>ProductItem_Name</td><td>Maximum of 50 characters</td></tr><tr><td>ID:</td><td>ProductItem_ID</td><td>Characters must be integers</td></tr></table>ALT</td></tr>
<tr><td></td><td></td><td></td><td>Step 6: The system uses a read query to read the matching entries from the Product Category entity. The system retrieves:

<u>Productitem</u> table:
● ProductItem _ID
● ProductItem_Name</td></tr>
<tr><td></td><td></td><td></td><td>Step 7: The system displays the retrieved search results.</td></tr>
</table>

<table>
<tr>
<td></td>
<td>Step 8: The customer clicks on the product they were searching for.</td>
<td></td>
<td>Step 9: The system loads the Product Details view with the following elements(showing the selected product category):
<table>
<tr><th>Element Name</th><th>Element Type</th><th>Notes:</th></tr>
<tr><td>Product Details</td><td>Modal Header</td><td>To indicate what the view is for</td></tr>
<tr><td colspan="3">The following elements are Retrieved from the <u>ProductItem entity</u> , <u>CategoryType entity, ProductCategory entity</u></td></tr>
<tr><td>lblProductCategoryandTypeDesc</td><td>Label</td><td>To display the description of the Product Category
CategoryType_Desc
• ProductCategory_Desc</td></tr>
<tr><td>lblProductItemName</td><td>Label</td><td>To display the name of the Product Item
○ ProductItem_Name</td></tr>
<tr><td>lblProductItemDesc</td><td>Label</td><td>To display the description of the product item
○ ProductItem_Desc</td></tr>
<tr><td>imgProductItem</td><td>Img</td><td>To display the product item image</td></tr>
</table>
</td>
</tr>
</table>

		○ ProductItem_Image
lblProductItem_Cost	Label	To display the cost of the product item ○ ProductItem_Cost
btnShowLess	Button	To display the less of the description information of the product item
btnAddCartItem	Button	To add an item to the cart.

[ALT]

ALTERNATE

ALT Step 5: The system fails to capture and validate the information and displays the error message.

Element Name	Element Type	Notes
Error	Label	This serves as a header for the modal.
Please ensure that the details you have entered are valid	Notification/Pop-up	To inform the user that the details entered are invalid.
OK	Button	None

Use Case Terminated

<table>
<tr><td>COURSES:</td><td>ALT Step 9: There system fails to retrieve the product items:
<table>
<tr><th>Element Name</th><th>Element Type</th><th>Notes</th></tr>
<tr><td>Error</td><td>Label</td><td>This serves as a header for the modal.</td></tr>
<tr><td>There was an error loading the Products page.

Please try again later.</td><td>Label</td><td>This is to show that the saving of the product category information was unsuccessful.</td></tr>
<tr><td>OK</td><td>Button</td><td>None</td></tr>
</table>
Use Case Terminated</td></tr>
<tr><td>CONCLUSION:</td><td>The use case concludes when the relevant product details are displayed successfully.</td></tr>
<tr><td>POST-CONDITION:</td><td>User sees Products detail view</td></tr>
<tr><td>BUSINESS RULES:</td><td>None</td></tr>
<tr><td>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:</td><td>None</td></tr>
<tr><td>ASSUMPTIONS:</td><td>None</td></tr>
<tr><td>OPEN ISSUES:</td><td>None</td></tr>
</table>

NKAP Bolting System

Author (s): **Chardé Devine Vries** **Date:06/06/2021**

Version: **___1.0___**

Use Case Name:	**Search Specials and Promotions**	**USE CASE TYPE**		
Use Case ID:	**6.3**	**Business Requirements:**	☐	
Priority:	High	**System Analysis:**	☐	
Source:	NKAP Bolting system	**System Design:**	☑	
Primary Business Actor:	Customer			
Primary The system Actor:	None			
Other Participating Actors:	None			
Other Interested Stakeholders:	None			
Description:	This use case describes the event where a customer would like to browse all available specials and promotions. The use case starts when the customer clicks on the **Promotions** tab, they can invoke the “view product details” use cases in order to view a selected product. The system will retrieve information from the **ProductItem entity** and **Product_Special entity** The system displays all the relevant information. The customer is able to view and browse through all available product.			
Pre-condition:	• The customer must have an account and be logged into the system			
Trigger:	A user wishes to browse the products with specials and promotions on the system			
Typical Course of Events:	**Actor Action:**	**System Response:**		
		MANUAL ACTION	**AUTOMATED ACTION**	
	Step 1: The customer request to browse through the specials and promotions.			

<table>
<tr>
<td></td>
<td>Step 2: The customer clicks on the ‘Promotions’ menu item on the menu-strip.</td>
<td></td>
<td>Step 3: The system loads the Promotions screen with the following elements(showing the selected product category):
<table>
<tr><th>Element Name</th><th>Element Type</th><th>Notes:</th></tr>
<tr><td>Specials and Promotions</td><td>Header</td><td>To indicate what the screen is for</td></tr>
<tr><td>lblSearch</td><td>Label</td><td>This is the label to prompt the user to enter a search query.</td></tr>
<tr><td>txtSearch</td><td>Textbox</td><td>To enter the search criteria</td></tr>
<tr><td>btnSearch icon</td><td>Icon button</td><td>This will submit the search criteria</td></tr>
<tr><td>cardProductItems</td><td>Card</td><td>To show each product item.
Each card contains:
• ProductItem_Name
• ProductItem_Image
• Special_Price
• Special_EndDate
Retrieved from the:
• <u>ProductItem entity</u>
• <u>Product_Special entity</u>
• <u>Special entity</u></td></tr>
<tr><td colspan="3">The following elements are in cardProductItems:</td></tr>
<tr><td>btnAddCartItem</td><td>Button</td><td>To add an item to the cart.</td></tr>
</table>
</td>
</tr>
</table>

<table>
<tr><td></td><td></td><td></td><td>
<table>
<tr><td>lblProductItemName</td><td>Label</td><td>To display the name of the Product Item</td></tr>
<tr><td>imgProductItem</td><td>Image</td><td>To display the product item image</td></tr>
<tr><td>lblSpecialPrice</td><td>Label</td><td>To display the price of the Product Item</td></tr>
<tr><td>lblSpecialEnds</td><td>Label</td><td>To display the end date of the Special price</td></tr>
</table>
[ALT]
</td></tr>
<tr><td></td><td>Step 4: The customer enters the product search criteria into the search textbox.</td><td></td><td></td></tr>
<tr><td></td><td>Step 5: The customer clicks on the search icon.</td><td></td><td>
Step 6: The system captures and validates the information against the <u>Productitem table</u>
<table>
<tr><th>Input Name</th><th>Attribute Type</th><th>Validati Requiren</th></tr>
<tr><td>Name:</td><td>ProductItem_Name</td><td>Maximu characte</td></tr>
<tr><td>ID:</td><td>ProductItem_ID</td><td>Characte be intege</td></tr>
</table>
ALT
</td></tr>
<tr><td></td><td></td><td></td><td>
Step 7: The system uses a read query to read the matching entries from the Product Category entity. The system retrieves:

<u>Productitem</u> table:

ProductItem _ID
ProductItem_Name

</td></tr>
</table>

			Step 8: The system displays the retrieved search results.
	Step 9: The admin clicks on the product special they were searching for.		**Step 10:** The system loads the **SpecialProduct Details** view with the following elements(showing the selected product category):

Element Name	Element Type	Notes:
Special Product Details	Header	To indicate what the view is for
The following elements are Retrieved from the **ProductItem entity** ,**Product_Special, Discount entity** and **Special entity**		
imgProductItem	Img	To display the product item image ○ ProductItem_Image
lblProductCategoryDesc	Label	To display the description of the Product Category ○ CategoryType_Desc ○ ProductCategory_Desc
lblProductItemName	Label	To display the name of the Product Item ○ ProductItem_Name

lblSpecialPrice	Label	To display the price of the Product Item ○ Special_Price
lblProductItemDesc	Label	To display the description of the product item ○ ProductItem_Desc
lblSpecialEnds	Label	To display the end date of the Special price ○ Special_EndDate
btnShowLess	Button	To display the less of the description information of the product item
btnAddCartItem	Button	To add an item to the cart.

[ALT]

ALTERNATE

ALT Step 3: The system fails to capture and validate the information and displays the error message.

Element Name	Element Type	Notes
Error	Label	This serves as a header for the modal.
Please ensure that the details you have entered are valid	Notification/Pop-up	To inform the user that the details entered are invalid.

Courses:	<table><tr><td>**OK**</td><td>Button</td><td>None</td></tr></table>Use Case Terminated
	ALT Step 10: There system fails load Promotions page:<table><tr><th>Element Name</th><th>Element Type</th><th>Notes</th></tr><tr><td>**Error**</td><td>Label</td><td>This serves as a header for the modal.</td></tr><tr><td>**There was an error loading the Promotions page.** **Please try again later.**</td><td>Label</td><td>This is to show that the saving of the product category information was unsuccessful.</td></tr><tr><td>**OK**</td><td>Button</td><td>None</td></tr></table>Use Case Terminated
Conclusion:	The use case concludes when the relevant product details are displayed successfully.
Post-condition:	User sees Promotions view
Business Rules:	None
Implementation Constraints and Specifications:	None
Assumptions:	None
Open Issues:	None

NKAP Bolting System

Author (s): **Chardé Devine Vries** **Date:06/06/2021**

Version: **1.0**

USE CASE NAME:	**Add item to cart**	**USE CASE TYPE**
USE CASE ID:	**6.4**	**Business Requirements:** ☐
PRIORITY:	High	**System Analysis:** ☐
SOURCE:	NKAP Bolting system	**System Design:** ☑
PRIMARY BUSINESS ACTOR:	Customer	
PRIMARY THE SYSTEM ACTOR:	None	
OTHER PARTICIPATING ACTORS:	None	
OTHER INTERESTED STAKEHOLDERS:	None	
DESCRIPTION:	This use case describes the process where the customer searches for a specific Product. Once the customer enters the search criteria the system will retrieve the matching results from the **ProductItem entity** the use case ends once the retrieved results have been displayed to the admin.	
PRE-CONDITION:	• The customer must have an account and be logged into the system	
TRIGGER:	A user wishes to browse the products on the system	

TYPICAL COURSE OF EVENTS:	**ACTOR ACTION:**	**SYSTEM RESPONSE:**	
		MANUAL ACTION	**AUTOMATED ACTION**
	Step 1: The customer requests to add a product item to cart.		
	Step 2: The customer clicks the 'add to cart' button on either the Product view, Product Details view or Promotions view		**Step 3:** The system confirm that the customer would like to add the item to cart by displaying a message box:

Element name	**Element type**	**Notes:**

<table>
<tr>
<td></td>
<td></td>
<td></td>
<td>
<table>
<tr><td>Confirm Add to cart</td><td>Header</td><td>To indicate what the view is for</td></tr>
<tr><td>lblConfirmMessage</td><td>Label</td><td>Text asking if they would like to add the item to cart</td></tr>
<tr><td>btnAdd</td><td>Button</td><td>To proceed and add item to cart.</td></tr>
<tr><td>btnCancel</td><td>Button</td><td>To cancel the process and add item to cart.</td></tr>
</table>
</td>
</tr>
<tr>
<td></td>
<td>Step 4: The customer clicks the Add button

ALT</td>
<td></td>
<td>Step 5: The system uses a SQL insert to add the Product Item to the cart the <u>CartLine table</u> and generates a new cart if one does not exist yet.
<table>
<tr><th>Element Name</th><th>Element Type</th><th>Notes</th></tr>
<tr><td>CartLineID</td><td>CartLineID</td><td>The system generates a new CartLine_ID by retrieving the last CartLine_ID in the CartLine table and incrementing it by one.</td></tr>
<tr><td>ProductItem_ID</td><td>ProductItem_ID</td><td>The system captures and adds the</td></tr>
</table>
</td>
</tr>
</table>

		ProductItem ID
CartLine Quantity	CartLine Quantity	The system updates the CartLine Quantity in the CartLine table by incrementing it by one.

ALT

Step 6: The system displays a successfully added to cart message box

Element Name	Element Type	Notes
Added to cart	Header	To indicate what the view is for
Added to cart	Label	To inform the user that the item was successfully added to cart
btnOkay	Button	To proceed

Step 7: The system updates the cart icon in the page header. Using the CartLine_Quantity attribute in the **CartLine entity** to show the amount of items in the cart.

ALTERNATE	**ALT Step 4:** The user clicks the cancel button.

COURSES:	Use Case Terminated
	ALT Step 6: There system fails to add the product items to cart:
	Use Case Terminated
CONCLUSION:	The use case concludes when the relevant product details are displayed successfully.
POST-CONDITION:	User sees Products detail view
BUSINESS RULES:	None
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	None
ASSUMPTIONS:	None
OPEN ISSUES:	None

Element Name	Element Type	Notes
Error	Label	This serves as a header for the modal.
There was an error while added the product to cart. Please try again.	Label	This is to show that adding the item to cart was unsuccessful.
OK	Button	None

NKAP Bolting System

Author (s): **<u>Chardé Devine Vries</u>** **Date:<u>06/06/2021</u>**

Version:

<u>1.0</u>

<table>
<tr><td>Use Case Name:</td><td colspan="2">View Cart</td><td>USE CASE TYPE</td></tr>
<tr><td>Use Case ID:</td><td colspan="2">6.5</td><td rowspan="3">Business Requirements: ☐
System Analysis: ☐
System Design: ☑</td></tr>
<tr><td>Priority:</td><td colspan="2">High</td></tr>
<tr><td>Source:</td><td colspan="2">NKAP Bolting system</td></tr>
<tr><td>Primary Business Actor:</td><td colspan="3">Customer</td></tr>
<tr><td>Primary The system Actor:</td><td colspan="3">None</td></tr>
<tr><td>Other Participating Actors:</td><td colspan="3">None</td></tr>
<tr><td>Other Interested Stakeholders:</td><td colspan="3">None</td></tr>
<tr><td>Description:</td><td colspan="3">This use case describes the event where a customer would like to see all the items in their cart. The use case starts when the customer clicks on the Cart icon in the page header. The system will retrieve information from the <u>CartLine entity</u>.The system displays all the relevant information. The customer is able to view all cart items, remove items or checkout.</td></tr>
<tr><td>Pre-condition:</td><td colspan="3">• The customer must have an account and be logged into the system</td></tr>
<tr><td>Trigger:</td><td colspan="3">A user wishes to view items in cart.</td></tr>
<tr><td rowspan="2">Typical Course of Events:</td><td rowspan="2">Actor Action:</td><td colspan="2">System Response:</td></tr>
<tr><td>MANUAL ACTION</td><td>AUTOMATED ACTION</td></tr>
<tr><td></td><td>Step 1: The customer request to see the cart.</td><td></td><td></td></tr>
<tr><td></td><td>Step 2: The customer clicks on the ‘Cart’ icon on the menu-strip.</td><td></td><td>Step 3: The system loads the Cart view with the following elements<table><tr><td>Element Name</td><td>Element Type</td><td>Notes:</td></tr></table></td></tr>
</table>

Cart	Header	To indicate what the screen is for
tblCart	Table	To show each product item in cart using a SQL Insert Each row contains: • ProductItem_Name • ProductItem_Image • Special_Price or ProductItem_Cost • Special_EndDate (if Special_Price is shown) Retrieved from the: • <u>**ProductItem entity**</u> • <u>**Product_Special entity**</u> • <u>**Special entity**</u>
Subtotal	Label	To display subtotal label
lblSubtotal	Label	To display subtotal
VAT	Label	To display VAT label
lblVAT	Label	To display VAT total
Total	Label	To display total label

lblTotal	Label	To display total price
Checkout	Button	To go to checkout screen
Back	Button	
The following elements are displayed in *tblCart* for each ietm*:*		
imgProductItem	*Image*	*To display the product item image*
lblProductItemName	*Label*	*To display the name of the Product Item*
lblQuantity	*Label*	*To display the name of the Product Item*
nudQuantity	*Numeric Up Down*	*To select the quantity of the Product Item*
SpecialPrice	*Label*	*To display the label price of the Product Item*
lblSpecialPrice	*Label*	*To display the price of the Product Item*
ProductItem_Cost	*Label*	*To display the label cost of the product item*

lblProductItem_Cost	*Label*	*To display the cost of the product item*
btnRemoveCartItem	*Button*	*To remove an item from the cart.*

[ALT]

ALTERNATE COURSES:	**ALT Step 3:** There system fails to retrieve the cart items:

Element Name	Element Type	Notes
Error	Label	This serves as a header for the modal.
There was an error loading the cart page. **Please try again later.**	Label	This is to show that there was an error while loading the cart.
OK	Button	None

Use Case Terminated

CONCLUSION:	The use case concludes when the relevant cart is displayed successfully.
POST-CONDITION:	User sees Cart view
BUSINESS RULES:	None
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	None
ASSUMPTIONS:	None
OPEN ISSUES:	None

Version: ___1.0___

<table>
<tr><td>Use Case Name:</td><td>Remove Cart item</td><td colspan="2">USE CASE TYPE</td></tr>
<tr><td>Use Case ID:</td><td>6.6</td><td colspan="2" rowspan="3">Business Requirements: ☐
System Analysis: ☐
System Design: ☑</td></tr>
<tr><td>Priority:</td><td>High</td></tr>
<tr><td>Source:</td><td>NKAP Bolting system</td></tr>
<tr><td>Primary Business Actor:</td><td colspan="3">Customer</td></tr>
<tr><td>Primary The system Actor:</td><td colspan="3">None</td></tr>
<tr><td>Other Participating Actors:</td><td colspan="3">None</td></tr>
<tr><td>Other Interested Stakeholders:</td><td colspan="3">None</td></tr>
<tr><td>Description:</td><td colspan="3">This use case describes the event where a customer would like remove an items from the cart. The use case starts when the customer clicks on the Remove icon found in the cart view. The system will remove information from the <u>CartLine entity</u>. The system displays all the relevant information. The use case concludes when the customer sees the successfully removed from cart message box.</td></tr>
<tr><td>Pre-condition:</td><td colspan="3">• The customer must have an account and be logged into the system</td></tr>
<tr><td>Trigger:</td><td colspan="3">A user wishes to remove an item from the cart.</td></tr>
<tr><td rowspan="3">Typical Course of Events:</td><td rowspan="2">Actor Action:</td><td colspan="2">System Response:</td></tr>
<tr><td>MANUAL ACTION</td><td>AUTOMATED ACTION</td></tr>
<tr><td>Step 1: The customer request to remove an item from the cart.</td><td></td><td></td></tr>
<tr><td></td><td>Step 2: The customer clicks on the 'Remove icon found next to the item in the Cart view, cart table.</td><td></td><td>Step 3: The system confirm that the customer would like to remove the item from cart by displaying a message box:

<table><tr><th>Element name</th><th>Element type</th><th>Notes:</th></tr><tr><td>Confirm Remove from cart</td><td>Header</td><td>To indicate what the</td></tr></table></td></tr>
</table>

<table>
<tr>
<td></td>
<td></td>
<td></td>
<td>
<table>
<tr><td></td><td></td><td>view is for</td></tr>
<tr><td>lblConfirmMessage</td><td>Label</td><td>Text asking if they would like to remove the item from cart</td></tr>
<tr><td>btnRemove</td><td>Button</td><td>To proceed and remove item to cart.</td></tr>
<tr><td>btnCancel</td><td>Button</td><td>To cancel the process and remove item from cart.</td></tr>
</table>
</td>
</tr>
<tr>
<td></td>
<td>Step 4: The customer clicks the Remove button

ALT</td>
<td></td>
<td>Step 5: The system uses a SQL Delete to remove the Product Item from the cart in the <u>CartLine table</u>
<table>
<tr><th>Element Name</th><th>Element Type</th><th>Notes</th></tr>
<tr><td>ProductItem_ID</td><td>ProductItem_ID</td><td>The system removes the ProductItem ID</td></tr>
</table>
</td>
</tr>
</table>

CartLine Quantity	CartLine Quantity	The system updates the CartLine Quantity in the CartLine table by decrementing it by one.

ALT

Step 6: The system displays a successfully removed from cart message box

Element Name	Element Type	Notes
Removed from cart	Header	To indicate what the view is for
Removed from cart	Label	To inform the user that the item was successfully removed from cart
btnOkay	Button	To proceed

			Step 7: The system updates the cart icon in the page header. Using the CartLine_Quantity attribute in the **CartLine entity** to show the amount of items in the cart.
ALTERNATE COURSES:	**ALT Step 4:** The user clicks the cancel button. Use Case Terminated		
	ALT Step 5: There system fails to remove the product item from cart:		

Element Name	Element Type	Notes
Error	Label	This serves as a header for the modal.
There was an error while removing the product from cart. Please try again.	Label	This is to show that removing the item from cart was unsuccessful.
OK	Button	None

Use Case Terminated

CONCLUSION:	The use case concludes when the removed cart item message box is displayed successfully.
POST-CONDITION:	Cart is updated.
BUSINESS RULES:	None
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	None
ASSUMPTIONS:	None
OPEN ISSUES:	None

NKAP Bolting System

Author (s): **Chardé Devine Vries** **Date:06/06/2021**

Version:

___1.0___

Use Case Name:	**Checkout Order**	**USE CASE TYPE**	
Use Case ID:	**6.7**	**Business Requirements:**	☐
Priority:	High	**System Analysis:**	☐
Source:	NKAP Bolting system	**System Design:**	☑
Primary Business Actor:	Customer		
Primary The system Actor:	None		
Other Participating Actors:	Virtual Card Service		
Other Interested Stakeholders:	None		
Description:	This use case describes the event where a customer would like to checkout all the items in their cart. The system checks if the customer is logged in. The system asks if they would like to pick up the order or if it should be delivered, which is followed by a delivery form that the customer has to fill in if they want it to be delivered. The system displays a modal with the order total. The customer requests to pay through the Virtual Card Service (PayPal). The transaction takes place on their system and once complete we save the transaction details on our system. The use case ends when the system displays a Payment successful popup.		
Pre-condition:	• The customer should have product items in their cart.		
Trigger:	A customer clicks the Checkout button		
Typical Course of Events:	**Actor Action:**	**System Response:**	
		MANUAL ACTION	**AUTOMATED ACTION**
	Step 1: The customer request to go to checkout.		
	Step 2: The customer clicks the 'Checkout' button located in the Cart view.		**Step 3:** The system uses a SQL read in the Audit Trail entity to see if the customer is logged in.

<table>
<tr><td></td><td></td><td></td><td>
<table>
<tr><th>Element Name</th><th>Element Type</th><th>Notes</th></tr>
<tr><td>AuditTrail Desc</td><td>AuditTrail Desc</td><td>The system reads.</td></tr>
</table>
ALT</td></tr>
<tr><td></td><td></td><td></td><td>Step 4: The system prompts the customer to select between delivery and collection.
<table>
<tr><th>Element Name</th><th>Element Type</th><th>Notes</th></tr>
<tr><td>Checkout - Delivery or Collect</td><td>Label</td><td>This serves as a header for the modal.</td></tr>
<tr><td>Delivery</td><td>Radio Button</td><td>None</td></tr>
<tr><td>Collection</td><td>Radio Button</td><td>None</td></tr>
<tr><td>Proceed</td><td>Button</td><td>If Delivery proceed to delivery form, if not proceed to step 9.</td></tr>
<tr><td>Cancel</td><td>Button</td><td>None</td></tr>
</table>
[ALT]</td></tr>
<tr><td></td><td>Step 5: The customer selects the delivery</td><td></td><td>Step 6: The system displays delivery form.</td></tr>
</table>

<table>
<tr>
<td></td>
<td>option and clicks the proceed button.

ALT</td>
<td></td>
<td>
<table>
<tr><th>Element Name</th><th>Element Type</th><th>Notes</th></tr>
<tr><td>Delivery form</td><td>Form</td><td>Saved to the:
Address entity
Province Entity
City Entity
Suburb Entity</td></tr>
<tr><td colspan="3">The following elements are displayed in the delivery form:</td></tr>
<tr><td>lblAddressLine1</td><td>Label</td><td>To display the Address_Line1 label</td></tr>
<tr><td>txtAddress_Line1</td><td>Textbox</td><td>To enter the Address Line</td></tr>
<tr><td>lblAddressLine2</td><td>Label</td><td>To display the Address_Line2 label</td></tr>
<tr><td>txtAddress_Line2</td><td>Textbox</td><td>To enter the Address Line</td></tr>
<tr><td>lblAddressLine3</td><td>Label</td><td>To display the Address_Line3 label</td></tr>
<tr><td>txtAddress_Line3</td><td>Textbox</td><td>To enter the Address Line</td></tr>
<tr><td>lblPostalCode</td><td>Label</td><td>To display the Postal code label</td></tr>
</table>
</td>
</tr>
</table>

<table>
<tr>
<td></td>
<td></td>
<td></td>
<td>
<table>
<tr><td>txtPostalCode</td><td>Textbox</td><td>To enter the Postal Code</td></tr>
<tr><td>lblProvince</td><td>Label</td><td>To display the label Province</td></tr>
<tr><td>sltProvince</td><td>Dropdown</td><td>To select the Province</td></tr>
<tr><td>lblCity</td><td>Label</td><td>To display the label City</td></tr>
<tr><td>sltCity</td><td>Dropdown</td><td>To select the City</td></tr>
<tr><td>lblSuburb</td><td>Label</td><td>To display the label Suburb</td></tr>
<tr><td>sltSuburb</td><td>Dropdown</td><td>To select the Suburb</td></tr>
<tr><td>btnProceed</td><td>Button</td><td>Proceed to checkout</td></tr>
</table>
</td>
</tr>
<tr>
<td></td>
<td>Step 7: The customer enters their address information and click the proceed button.</td>
<td></td>
<td>Step 8: The system uses a SQL insert to capture the AddressID using the Delivery table and generates a new deliveryID, the system also uses SQL Update to update the SaleOrder_ReceiveType in the Sale Entity.
<table>
<tr><th>Element Name</th><th>Attribute</th><th>Notes</th></tr>
<tr><td>AddressID</td><td></td><td>the Delivery table</td></tr>
</table>
</td>
</tr>
</table>

			<table><tr><td>DeliveryID</td><td>Delivery_ID</td><td>The system generates a new Delivery_ID by retrieving the last Delivery_ID in the Delivery table and incrementing it by one in Delivery entity.</td></tr><tr><td>SaleOrder_ReceiveType</td><td>SaleOrder_ReceiveType</td><td>The system captures and adds the new SaleOrder_ReceiveType in Sale entity</td></tr></table>
			Step 8: The system validates the information using a SQL read from the • **Delivery entity** • **Address entity** • **Province entity** • **City entity** • **Suburb entity** <table><tr><th>Input Name</th><th>Attribute in Table</th><th>Validation Requirements</th></tr><tr><td>txtAddress_Line1</td><td>Address_Line1</td><td>Max 100 characters</td></tr><tr><td>txtAddress_Line2</td><td>Address_Line2</td><td>Max 100 characters</td></tr></table>

txtAddress_Line3	Address_Line2	Max 100 characters
txtPostalCode	Address Postal Code	Integer
sltProvince	Province_Description	Must be selected
sltCity	City_Description	Must be selected
sltSuburb	Suburb_Description	Must be selected

ALT

Step 9: The system displays a message box with the amount due and prompts the user to use available payment method.

Element Name	Element Type	Notes
lblSubtotal	Label	To display subtotal
lblVAT	Label	To display VAT total
lblTotal	Label	To display total price
Subtotal	Label	To display subtotal label
VAT	Label	To display VAT label

<table>
<tr>
<td></td>
<td></td>
<td></td>
<td>
<table>
<tr><td>Total</td><td>Label</td><td>To display total label</td></tr>
<tr><td>Checkout -Payment Method</td><td>Button</td><td>Virtual Card Services – Bank as option</td></tr>
<tr><td>Exit icon</td><td>Button</td><td>Invoke Use Case 6.1 Search Products</td></tr>
</table>
</td>
</tr>
<tr>
<td></td>
<td>Step 10: The user clicks the ‘Checkout with PayPal’ button.</td>
<td></td>
<td>Step 11: The system loads the payment gateway.</td>
</tr>
<tr>
<td></td>
<td>Step 12: The customer makes a payment through the Virtual card services (PayPal).</td>
<td></td>
<td>Step 13: The system uses a SQL insert to record the transaction in the sale table and generates a new SaleID.
<table>
<tr><th>Element Name</th><th>Attribute</th><th>Notes</th></tr>
<tr><td>SaleID</td><td>SaleID</td><td>The system generates a new SaleID by retrieving the last SaleID D in the Sale table and incremen</td></tr>
</table>
</td>
</tr>
</table>

					ting it by one.
			Customer ID	Customer ID	The system captures and adds the SaleOrder Desc
			Payment Type ID	PaymentType ID	The system captures the Payment Type ID
			OrderStatus ID	OrderStatus ID	The system captures the OrderStatus ID
			SaleOrder Desc	SaleOrder Desc	The system captures and adds the SaleOrder Desc
			SaleOrder Date	SaleOrder Date	The system captures

			and adds the SaleOrder Date using the current Date.
	Payment Date	Payment Date	The system captures and adds the Payment Date using the current Date.
	Payment Amount	Payment Amount	The system captures and adds the Payment Amount by calculating the total of the cart items.

ALT

Step 14: The system displays a Payment successful message box.

Element Name	Element Type	Notes

Payment Success full	Label	To inform the user that the payment was successful.
btnOkay	Button	To proceed.

ALT

ALTERNATE COURSES:

ALT Step 2: The customer is not logged in, the system displays the following:

Element Name	Element Type	Notes
Hi, please sign in or register as a new user.	Label	This serves as a header for the modal.
Sign in	Button	Invoke Use Case 1.1 Login
Register	Button	Invoke Use Case 3.1 Register as new customer
Exit icon	Button	Invoke Use Case 6.1 Search Products

ALT Step 4: The customer selects the cancel button

Terminate Use Case

ALT Step 8: The customer chooses Collection radio button.

The system uses SQL Update query to update SaleOrder_ReceiveType to ‘collection’ in the Sale Entity.

Proceed to step 7.

ALT Step 13: There system fails to update the delivery details:

Element Name	Element Type	Notes
Error	Label	This serves as a header for the modal.

<table>
<tr><td></td><td>
<table>
<tr><td>There was an error while updating the delivery details.

Please try again.</td><td>Label</td><td>This is to show that updating the delivery details was unsuccessful.</td></tr>
<tr><td>OK</td><td>Button</td><td>None</td></tr>
</table>
Use Case Terminated
</td></tr>
<tr><td></td><td>
ALT Step 14: There was an error on the VCS system and the payment failed.
<table>
<tr><th>Element Name</th><th>Element Type</th><th>Notes</th></tr>
<tr><td>Error</td><td>Label</td><td>This serves as a header for the modal.</td></tr>
<tr><td>Payment unsuccessful

Please try again.</td><td>Label</td><td>This is to show that the payment was unsuccessful.</td></tr>
<tr><td>OK</td><td>Button</td><td>Invoke Use Case 6.5</td></tr>
</table>
</td></tr>
<tr><td>CONCLUSION:</td><td>The use case concludes when the system displays a Payment successful message box.</td></tr>
<tr><td>POST-CONDITION:</td><td>Checkout successful – updated SQL etc Sale entity…</td></tr>
<tr><td>BUSINESS RULES:</td><td>Customer needs to be logged in in order to make the payment.</td></tr>
<tr><td>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:</td><td>None</td></tr>
<tr><td>ASSUMPTIONS:</td><td>None</td></tr>
<tr><td>OPEN ISSUES:</td><td>None</td></tr>
</table>

2.1.7. REPORTS SUBSYSTEM

NKAP BOLTING SYSTEM			
DATE:	9/06/2021		
VERSION:	1.0		
AUTHOR(S)	Jerome Amenigy		
USE CASE NAME:	Generate fast-selling product list	USE CASE TYPE	
USE CASE ID:	7.1	Business Requirements: ☐	
PRIORITY:	High	System Analysis: ☐	
SOURCE:	NKAP Bolting System Requirements	System Design: ☑	
PRIMARY BUSINESS ACTOR:	Admin		
PRIMARY THE SYSTEM ACTOR:	None		
OTHER PARTICIPATING ACTORS:	None		
OTHER INTERESTED STAKEHOLDERS:	None		
DESCRIPTION:	This use case describes the event whereby a list is generated based on the fast-selling products by the admin. The admin will first navigate to the products tab and select the "fast-selling product list" report in the report's menu item in the products tab and enter the period for which the list should be generated. The system will then retrieve and display those items which had the most sales in the period the admin has selected.		
PRE-CONDITION:	• The admin is logged in to the system • Sales have occurred on the system		
TRIGGER:	The admin wants to review the fastest selling items available on the system.		
TYPICAL COURSE OF EVENTS:	ACTOR ACTION:	SYSTEM RESPONSE:	
		Manual Action	Automated Action
	Step 1: The requests to generate a report		
	Step 2: The admin clicks on the Reports		**Step 3:** The system displays the "Reports" screen with the following inputs and controls on it:

menu option

Control Name	Control Type	Notes
Reports	Heading	None
Generate Report	Label	None
Report type	Label	A dropdown list containing all the different reports available
Period start date	Date/time selector	The control to select the report start date
Period end date	Date/time selector	The control to select the report end date
Generate	Button	Button used to generate the selected report.
Report Name	Label	A label where the name of the generated report will be displayed.
Download	Button	Button used to download the generated reports
Report type	Dropdown	A dropdown list containing all the different reports available
Please select the	Label	None.

<table>
<tr>
<td></td>
<td></td>
<td></td>
<td>
<table>
<tr><td>report period</td><td></td><td></td></tr>
<tr><td>Report Panel</td><td>Panel</td><td>Area where the generated report will be displayed.</td></tr>
<tr><td>Start Date</td><td>Label</td><td>None.</td></tr>
<tr><td>End Date</td><td>Label</td><td>None.</td></tr>
</table>
</td>
</tr>
<tr>
<td></td>
<td>Step 4: The admin selects the “Fast-selling product list”</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Step 5: The admin selects the required period.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Step 6: The admin clicks the “Generate” button</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Step 7: The system retrieves the required information from the database tables using a SQL Read.
ProductItem:
ProductItem_ID
ProductItem_Name
ProductItem_Desc
CategoryType_ID
Sale:
Sale_ID
SaleOrder_Number
SaleOrder_Date
SaleOrder_Desc
SaleLine
SaleLine_ID
ProductItem_ID</td>
</tr>
</table>

			• Sale_ID • SaleLine_Qunatity
			Step 8: The system compiles and generates the retrieved information into the Fast-selling product list.
			Step 9: The system displays the Fast-selling product list in the report panel
ALTERNATE COURSES			
CONCLUSION:	A list based on the fastest selling items is generated and displayed to the admin		
POST-CONDITION:	• The admin has viewed the generated fast-selling product list		
BUSINESS RULES:	1. The list is based on the sales made through the system		
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	None		
ASSUMPTIONS:	None		
OPEN ISSUES:	None		

NKAP BOLTING SYSTEM		
DATE:	**9/06/2021**	
VERSION:	**1.0**	
AUTHOR(S)	**Jerome Amenigy**	
USE CASE NAME:	**Generate slow-selling product list**	**USE CASE TYPE**
USE CASE ID:	**7.2**	**Business Requirements:** ☐ **System Analysis:** ☐ **System Design:** ☑
PRIORITY:	High	
SOURCE:	NKAP Bolting System Requirements	
PRIMARY BUSINESS ACTOR:	Admin	
PRIMARY THE SYSTEM ACTOR:	None	
OTHER PARTICIPATING ACTORS:	None	
OTHER INTERESTED STAKEHOLDERS:	None	

<table>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case describes the event whereby a list is generated based on the slow-selling products by the admin. The admin will first navigate to the products tab and select the “slow-selling product list” report in the report’s menu item in the products tab and enter the period for which the list should be generated. The system will then retrieve and display those items which had the lowest amount sales in the period the admin has selected.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">Sales have occurred on the system.
The admin is logged in to the system</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The admin wants to review the items which are not being sold very often in comparison to other items.</td></tr>
<tr><td rowspan="4">TYPICAL COURSE OF EVENTS:</td><td rowspan="2">ACTOR ACTION:</td><td colspan="2">SYSTEM RESPONSE:</td></tr>
<tr><td>Manual Action</td><td>Automated Action</td></tr>
<tr><td>Step 1: The requests to generate a report</td><td></td><td></td></tr>
<tr><td>Step 2: The admin clicks on the Reports menu option</td><td></td><td>Step 3: The system displays the “Reports” screen with the following inputs and controls on it:
<table>
<tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Reports</td><td>Heading</td><td>None</td></tr>
<tr><td>Generate Report</td><td>Label</td><td>None</td></tr>
<tr><td>Report type</td><td>Label</td><td>A dropdown list containing all the different reports available</td></tr>
<tr><td>Period start date</td><td>Date/time selector</td><td>The control to select the report start date</td></tr>
<tr><td>Period end date</td><td>Date/time selector</td><td>The control to select the report end date</td></tr>
<tr><td>Generate</td><td>Button</td><td>Button used to generate</td></tr>
</table></td></tr>
</table>

		the selected report.
Report Name	Label	A label where the name of the generated report will be displayed.
Download	Button	Button used to download the generated reports
Report type	Dropdown	A dropdown list containing all the different reports available
Please select the report period	Label	None.
Report Panel	Panel	Area where the generated report will be displayed.
Start Date	Label	None.
End Date	Label	None.

	Step 4: The admin selects the “Slow-selling product list”		
	Step 5: The admin selects the		

	required period.		
	Step 6: The admin clicks the “Generate” button		
			Step 7: The system retrieves the required information from the database tables using a SQL Read. **ProductItem:** • ProductItem_ID • ProductItem_Name • ProductItem_Desc • CategoryType_ID **Sale:** • Sale_ID • SaleOrder_Number • SaleOrder_Date • SaleOrder_Desc **SaleLine** • SaleLine_ID • ProductItem_ID • Sale_ID • SaleLine_ Qunatity
			Step 8: The system compiles and generates the retrieved information into the Slow-selling product list.
			Step 9: The system displays the Slow-selling product list in the report panel
ALTERNATE COURSES			
CONCLUSION:	A list based on the slowest selling items is generated and displayed to the admin		
POST-CONDITION:	• The admin has viewed the generated slow-selling product list		
BUSINESS RULES:	1. The list is based on the sales made through the system		
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	None		
ASSUMPTIONS:	None		
OPEN ISSUES:	None		

<table>
<tr><th colspan="5">NKAP BOLTING SYSTEM</th></tr>
<tr><td>DATE:</td><td colspan="4">9/06/2021</td></tr>
<tr><td>VERSION:</td><td colspan="4">1.0</td></tr>
<tr><td>AUTHOR(S)</td><td colspan="4">Jerome Amenigy</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Generate most frequent buyers report</td><td colspan="2">USE CASE TYPE</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">7.3</td><td colspan="2" rowspan="3">Business Requirements: ☐
System Analysis: ☐
System Design: ☑</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">High</td></tr>
<tr><td>SOURCE:</td><td colspan="2">NKAP Bolting System Requirements</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR:</td><td colspan="4">Admin</td></tr>
<tr><td>PRIMARY THE SYSTEM ACTOR:</td><td colspan="4">None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="4">None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="4">None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="4">This use case describes the event whereby the admin generates a report based on the Customers who have placed the most orders. The system will retrieve information from the Customer and Sale tables and then display a list of the customers who have completed the most purchases in the past month</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="4">• Sales have occurred on the system
• The admin is logged in to the system</td></tr>
<tr><td>TRIGGER:</td><td colspan="4">The admin wants to view who the most frequent customers are buying from the business.</td></tr>
<tr><td rowspan="4">TYPICAL COURSE OF EVENTS:</td><td rowspan="2">ACTOR ACTION:</td><td colspan="3">SYSTEM RESPONSE:</td></tr>
<tr><td>Manual Action</td><td colspan="2">Automated Action</td></tr>
<tr><td>Step 1: The requests to generate a report</td><td></td><td colspan="2"></td></tr>
<tr><td>Step 2: The admin clicks on the Reports menu option</td><td></td><td colspan="2">Step 3: The system displays the “Reports” screen with the following inputs and controls on it:
<table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Reports</td><td>Heading</td><td>None</td></tr></table></td></tr>
</table>

			Generate Report	Label	None
			Report type	Label	A dropdown list containing all the different reports available
			Period start date	Date/time selector	The control to select the report start date
			Period end date	Date/time selector	The control to select the report end date
			Generate	Button	Button used to generate the selected report.
			Report Name	Label	A label where the name of the generated report will be displayed.
			Download	Button	Button used to download the generated reports
			Report type	Dropdown	A dropdown list containing all the different

<table>
<tr>
<td></td>
<td></td>
<td></td>
<td>
<table>
<tr><td></td><td></td><td>reports available</td></tr>
<tr><td>Please select the report period</td><td>Label</td><td>None.</td></tr>
<tr><td>Report Panel</td><td>Panel</td><td>Area where the generated report will be displayed.</td></tr>
<tr><td>Start Date</td><td>Label</td><td>None.</td></tr>
<tr><td>End Date</td><td>Label</td><td>None.</td></tr>
</table>
</td>
</tr>
<tr>
<td></td>
<td>Step 4: The admin selects the “Most frequent buyer report”</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Step 5: The admin clicks the “Generate” button</td>
<td></td>
<td>Step 6:
The system retrieves:

Sale_ID
SaleOrder_Date
SaleOrder_Desc
Delivery_ID
Courier_ID
Payment_Date
Customer_ID

And

Customer_ID
Customer_Name
Customer_Surname
Customer_CellphoneNumber
Customer_EmailAddress
Customer_Businesname

From the <u>Sale</u> and <u>Customer</u> tables using a SQL Read</td>
</tr>
</table>

			Step 7: The system sorts through the retrieved information and filters for the top results by using a count
			Step 8: The system displays the most frequent buyers report
ALTERNATE COURSES			
CONCLUSION:	The list of the most frequent buyers has been generated and displayed.		
POST-CONDITION:	• The admin has viewed the generated report		
BUSINESS RULES:	1. None		
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	None		
ASSUMPTIONS:	None		
OPEN ISSUES:	None		

NKAP BOLTING SYSTEM		
DATE:	**9/06/2021**	
VERSION:	**1.0**	
AUTHOR(S)	**Jerome Amenigy**	
USE CASE NAME:	**Generate most popular location report**	**USE CASE TYPE**
USE CASE ID:	**7.4**	**Business Requirements:** ☐
PRIORITY:	High	**System Analysis:** ☐
SOURCE:	NKAP Bolting System Requirements	**System Design:** ☑
PRIMARY BUSINESS ACTOR:	Admin	
PRIMARY THE SYSTEM ACTOR:	None	
OTHER PARTICIPATING ACTORS:	None	
OTHER INTERESTED STAKEHOLDERS:	None	
DESCRIPTION:	This use case describes the process where the admin generates a list based on the areas in which the most purchases were made through the system.	

<table>
<tr><td>PRE-CONDITION:</td><td colspan="3">• Sales must have occurred over the past month</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The admin wants to view a report based on the locations where customers are ordering the most.</td></tr>
<tr><td rowspan="4">TYPICAL COURSE OF EVENTS:</td><td rowspan="2">ACTOR ACTION:</td><td colspan="2">SYSTEM RESPONSE:</td></tr>
<tr><td>Manual Action</td><td>Automated Action</td></tr>
<tr><td>Step 1: The requests to generate a report</td><td></td><td></td></tr>
<tr><td>Step 2: The admin clicks on the Reports menu option</td><td></td><td>Step 3: The system displays the “Reports” screen with the following inputs and controls on it:
<table>
<tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Reports</td><td>Heading</td><td>None</td></tr>
<tr><td>Generate Report</td><td>Label</td><td>None</td></tr>
<tr><td>Report type</td><td>Label</td><td>A dropdown list containing all the different reports available</td></tr>
<tr><td>Period start date</td><td>Date/time selector</td><td>The control to select the report start date</td></tr>
<tr><td>Period end date</td><td>Date/time selector</td><td>The control to select the report end date</td></tr>
<tr><td>Generate</td><td>Button</td><td>Button used to generate the selected report.</td></tr>
<tr><td>Report Name</td><td>Label</td><td>A label where the name of the</td></tr>
</table></td></tr>
</table>

<table>
<tr>
<td></td>
<td></td>
<td></td>
<td>
<table>
<tr><td></td><td></td><td>generated report will be displayed.</td></tr>
<tr><td>Download</td><td>Button</td><td>Button used to download the generated reports</td></tr>
<tr><td>Report type</td><td>Dropdown</td><td>A dropdown list containing all the different reports available</td></tr>
<tr><td>Please select the report period</td><td>Label</td><td>None.</td></tr>
<tr><td>Report Panel</td><td>Panel</td><td>Area where the generated report will be displayed.</td></tr>
<tr><td>Start Date</td><td>Label</td><td>None.</td></tr>
<tr><td>End Date</td><td>Label</td><td>None.</td></tr>
</table>
</td>
</tr>
<tr>
<td></td>
<td>Step 4: The admin selects the “most popular location report”</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Step 5: The admin clicks the “Generate” button</td>
<td></td>
<td>Step 6: The system retrieves the following information from the <u>Customer</u> table

Customer_ID
Customer_Name
Customer_Surname
Customer_Cellphone
Customer_EmailAddress
Customer_Businesname

,

Sale_ID

</td>
</tr>
</table>

			• SaleOrder_Date • SaleOrder_Desc From the Sale table, • Address_ID • Customer_ID • Province_ID • Address_Line1 • Address_Line2 • Address_Line3 • Address_PostalCode From the Address table, • Province_ID • Province_Description From the Province entity, • City_ID • Province_ID • City_Description From the city entity, • Suburb_ID • City_ID • Suburb_Description From the Suburb table using a SQL Read
			Step 7: The system sorts through the retrieved information and filters for the areas which had the most sales in the past month
			Step 8: The system displays the most popular location report
ALTERNATE			
CONCLUSION:	The most popular location report has been generated and displayed		
POST-CONDITION:	• The admin views the most popular location report		
BUSINESS RULES:	1. None		
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	None		
ASSUMPTIONS:	None		
OPEN ISSUES:	None		

<table>
<tr><th colspan="4">NKAP BOLTING SYSTEM</th></tr>
<tr><td>DATE:</td><td colspan="3">9/06/2021</td></tr>
<tr><td>VERSION:</td><td colspan="3">1.0</td></tr>
<tr><td>AUTHOR(S)</td><td colspan="3">Jerome Amenigy</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Generate Weekly Sale Order List</td><td>USE CASE TYPE</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">7.5</td><td rowspan="3">Business Requirements: ☐
System Analysis: ☐
System Design: ☑</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">High</td></tr>
<tr><td>SOURCE:</td><td colspan="2">NKAP Bolting System Requirements</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR:</td><td colspan="3">Admin</td></tr>
<tr><td>PRIMARY THE SYSTEM ACTOR:</td><td colspan="3">None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case describes the event where the admin generates a report on the sale orders which have occurred over the past week made by customers. The admin will first navigate to the reports tab and then select reports and select the “weekly sale order list” report and then the system will retrieve and display the list based on all the sale orders which have taken place over the past week.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• Sales have occurred on the system in the past week
• The admin is logged in to the system</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The admin wants to view all the sale which have taken place over the past week.</td></tr>
<tr><td rowspan="4">TYPICAL COURSE OF EVENTS:</td><td rowspan="2">ACTOR ACTION:</td><td colspan="2">SYSTEM RESPONSE:</td></tr>
<tr><td>Manual Action</td><td>Automated Action</td></tr>
<tr><td>Step 1: The requests to generate a report</td><td></td><td></td></tr>
<tr><td>Step 2: The admin clicks on the Reports menu option</td><td></td><td>Step 3: The system displays the “Reports” screen with the following inputs and controls on it:
<table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr></table></td></tr>
</table>

			Reports	Heading	None
			Generate Report	Label	None
			Report type	Label	A dropdown list containing all the different reports available
			Period start date	Date/time selector	The control to select the report start date
			Period end date	Date/time selector	The control to select the report end date
			Generate	Button	Button used to generate the selected report.
			Report Name	Label	A label where the name of the generated report will be displayed.
			Download	Button	Button used to download the generated reports
			Report type	Dropdown	A dropdown list containing all the

<table>
<tr><td></td><td></td><td></td><td>
<table>
<tr><td></td><td></td><td>different reports available</td></tr>
<tr><td>Please select the report period</td><td>Label</td><td>None.</td></tr>
<tr><td>Report Panel</td><td>Panel</td><td>Area where the generated report will be displayed.</td></tr>
<tr><td>Start Date</td><td>Label</td><td>None.</td></tr>
<tr><td>End Date</td><td>Label</td><td>None.</td></tr>
</table>
</td></tr>
<tr><td></td><td>Step 4: The admin selects the “Weekly Sale Order List”</td><td></td><td></td></tr>
<tr><td></td><td>Step 5: The admin clicks the “Generate” button</td><td></td><td>Step 6: The system retrieves:
• Sale_ID
• SaleOrder_Date
• SaleOrder_Desc
• Customer_ID
• Payment_Date

And

• Customer_ID
• Customer_Name
• Customer_Surname
• Customer_Cellphone
• Customer_EmailAddress
• Customer_Businesname

From the Sale and Customer tables using a SQL read</td></tr>
<tr><td></td><td></td><td></td><td>Step 7: The system then selects the Sales which have occurred in the past week and puts them in a list with the corresponding customers who made those sale orders also in the list.</td></tr>
<tr><td></td><td></td><td></td><td>Step 8: The system then displays the weekly sale order list to the admin</td></tr>
</table>

ALTERNATE	
CONCLUSION:	The weekly sale order list has been generated and displayed
POST-CONDITION:	• The admin views the generated sale order list
BUSINESS RULES:	1. None
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	None
ASSUMPTIONS:	None
OPEN ISSUES:	None

NKAP BOLTING SYSTEM			
DATE:	**9/06/2021**		
VERSION:	**1.0**		
AUTHOR(S)	**Jerome Amenigy**		
USE CASE NAME:	**Generate Monthly Sale Order List**	**USE CASE TYPE**	
USE CASE ID:	**7.6**	**Business Requirements:**	☐
PRIORITY:	High	**System Analysis:**	☐
SOURCE:	NKAP Bolting System Requirements	**System Design:**	☑
PRIMARY BUSINESS ACTOR:	Admin		
PRIMARY THE SYSTEM ACTOR:	None		
OTHER PARTICIPATING ACTORS:	None		
OTHER INTERESTED STAKEHOLDERS:	None		
DESCRIPTION:	This use case describes the event where the admin generates a report on the sale orders which have occurred over the past month made by customers. The admin will first navigate to the reports tab and then select reports and select the “monthly sale order list” report and then the system will retrieve and display the list based on all the sales which have taken place over the past month		
PRE-CONDITION:	• Sales have occurred over the past month • The admin is logged in to the system		

<table>
<tr><td>TRIGGER:</td><td colspan="3">The admin wants to view all the sales which have taken place over the past month.</td></tr>
<tr><td rowspan="4">TYPICAL COURSE OF EVENTS:</td><td rowspan="2">ACTOR ACTION:</td><td colspan="2">SYSTEM RESPONSE:</td></tr>
<tr><td>Manual Action</td><td>Automated Action</td></tr>
<tr><td>Step 1: The requests to generate a report</td><td></td><td></td></tr>
<tr><td>Step 2: The admin clicks on the Reports menu option</td><td></td><td>Step 3: The system displays the “Reports” screen with the following inputs and controls on it:
<table>
<tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Reports</td><td>Heading</td><td>None</td></tr>
<tr><td>Generate Report</td><td>Label</td><td>None</td></tr>
<tr><td>Report type</td><td>Label</td><td>A dropdown list containing all the different reports available</td></tr>
<tr><td>Period start date</td><td>Date/time selector</td><td>The control to select the report start date</td></tr>
<tr><td>Period end date</td><td>Date/time selector</td><td>The control to select the report end date</td></tr>
<tr><td>Generate</td><td>Button</td><td>Button used to generate the selected report.</td></tr>
<tr><td>Report Name</td><td>Label</td><td>A label where the name of the generated report will</td></tr>
</table>
</td></tr>
</table>

<table>
<tr>
<td></td>
<td></td>
<td></td>
<td>
<table>
<tr><td></td><td></td><td>be displayed.</td></tr>
<tr><td>Download</td><td>Button</td><td>Button used to download the generated reports</td></tr>
<tr><td>Report type</td><td>Dropdown</td><td>A dropdown list containing all the different reports available</td></tr>
<tr><td>Please select the report period</td><td>Label</td><td>None.</td></tr>
<tr><td>Report Panel</td><td>Panel</td><td>Area where the generated report will be displayed.</td></tr>
<tr><td>Start Date</td><td>Label</td><td>None.</td></tr>
<tr><td>End Date</td><td>Label</td><td>None.</td></tr>
</table>
</td>
</tr>
<tr>
<td></td>
<td>Step 4: The admin selects the “Monthly Sale Order List”</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Step 5: The admin clicks the “Generate” button</td>
<td></td>
<td>Step 6: The system retrieves:
• Sale_ID
• SaleOrder_Date
• SaleOrder_Desc
• Customer_ID
• Payment_Date

And

• Customer_ID
• Customer_Name
• Customer_Surname
• Customer_Cellphone</td>
</tr>
</table>

			• Customer_EmailAddress • Customer_Businesname From the Sale and Customer tables using a SQL read
			Step 7: The system then selects the Sales which have occurred in the past month and puts them in a list with the corresponding customers who made those sale orders also in the list.
			Step 8: The system then displays the monthly sale order list to the admin
ALTERNATE			
CONCLUSION:	The weekly sale order list has been generated and displayed		
POST-CONDITION:	• The admin views the generated sale order list		
BUSINESS RULES:	1. None		
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	None		
ASSUMPTIONS:	None		
OPEN ISSUES:	None		

NKAP BOLTING SYSTEM		
DATE:	**9/06/2021**	
VERSION:	**1.0**	
AUTHOR(S)	**Jerome Amenigy**	
USE CASE NAME:	**Generate Stock-Level Report**	**USE CASE TYPE**
USE CASE ID:	**7.7**	**Business Requirements:** ☐
PRIORITY:	High	**System Analysis:** ☐
SOURCE:	NKAP Bolting System Requirements	**System Design:** ☑
PRIMARY BUSINESS ACTOR:	Admin	
PRIMARY THE SYSTEM ACTOR:	None	
OTHER PARTICIPATING ACTORS:	None	

<table>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case describes the event where a stock-level report is generated by the admin. The admin navigates to the reports tab and chooses the generate stock-level report. The system will retrieve information from the ProductItem entity and display it to the admin. The use case ends once the report has been displayed to the admin.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• Products have been added to the system
• The admin is logged in</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The admin wants to view the stock available in the system</td></tr>
<tr><td rowspan="4">TYPICAL COURSE OF EVENTS:</td><td rowspan="2">ACTOR ACTION:</td><td colspan="2">SYSTEM RESPONSE:</td></tr>
<tr><td>Manual Action</td><td>Automated Action</td></tr>
<tr><td>Step 1: The requests to generate a report</td><td></td><td></td></tr>
<tr><td>Step 2: The admin clicks on the Reports menu option</td><td></td><td>Step 3: The system displays the “Reports” screen with the following inputs and controls on it:
<table>
<tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Reports</td><td>Heading</td><td>None</td></tr>
<tr><td>Generate Report</td><td>Label</td><td>None</td></tr>
<tr><td>Report type</td><td>Label</td><td>A dropdown list containing all the different reports available</td></tr>
<tr><td>Period start date</td><td>Date/time selector</td><td>The control to select the report start date</td></tr>
<tr><td>Period end date</td><td>Date/time selector</td><td>The control to select the report end date</td></tr>
</table></td></tr>
</table>

			<table><tr><td>**Generate**</td><td>Button</td><td>Button used to generate the selected report.</td></tr><tr><td>**Report Name**</td><td>Label</td><td>A label where the name of the generated report will be displayed.</td></tr><tr><td>**Download**</td><td>Button</td><td>Button used to download the generated reports</td></tr><tr><td>**Report type**</td><td>Dropdown</td><td>A dropdown list containing all the different reports available</td></tr><tr><td>**Please select the report period**</td><td>Label</td><td>None.</td></tr><tr><td>**Report Panel**</td><td>Panel</td><td>Area where the generated report will be displayed.</td></tr><tr><td>**Start Date**</td><td>Label</td><td>None.</td></tr><tr><td>**End Date**</td><td>Label</td><td>None.</td></tr></table>
	Step 4: The admin selects the “Stock-Level Report”		

	Step 5: The admin clicks the "Generate" button		**Step 6:** The system retrieves • ProductItem_ID • ProductItem_Name • ProductItem_Desc • ProductItem_Image • CategoryType_ID • Quantity on hand , • CategoryType_ID • CategoryType_Desc • ProductCategory_ID And • ProductCategory_ID • ProductCategory_Desc From the ProductItem, CategoryType and ProductCategory tables using a SQL Read.
			Step 7: The system categorizes each ProductItem into its Category type and each category type into its Product Category.
			Step 8: The system then displays the Stock-level report with the necessary columns.
ALTERNATE			
CONCLUSION:	The stock-level report is generated and displayed		
POST-CONDITION:	• The admin views the stock-level report		
BUSINESS RULES:	1. None		
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	None		
ASSUMPTIONS:	None		
OPEN ISSUES:	None		

NKAP BOLTING SYSTEM		
DATE:	**9/06/2021**	
VERSION:	**1.0**	
AUTHOR(S)	**Jerome Amenigy**	
USE CASE NAME:	**Packing Report**	**USE CASE TYPE**

<table>
<tr><td>USE CASE ID:</td><td>7.8</td><td rowspan="3">Business Requirements: ☐
System Analysis: ☐
System Design: ☑</td></tr>
<tr><td>PRIORITY:</td><td>High</td></tr>
<tr><td>SOURCE:</td><td>NKAP Bolting System Requirements</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR:</td><td colspan="2">Admin</td></tr>
<tr><td>PRIMARY THE SYSTEM ACTOR:</td><td colspan="2">None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="2">None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="2">None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="2">This use case describes the event whereby an admin generates a report based on the Sales which still need to be packed. Firstly, the admin requests to generate a packing report on the system. The system will then retrieve all the Sales which have a status of ready to be packed. The system will then populate and display a list with the SaleOrder details.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="2">Sales have taken place on the system
The admin is logged in to the system</td></tr>
<tr><td>TRIGGER:</td><td colspan="2">The admin wants to view all the orders that still needs to be packed</td></tr>
</table>

<table>
<tr><td rowspan="4">TYPICAL COURSE OF EVENTS:</td><td rowspan="2">ACTOR ACTION:</td><td colspan="2">SYSTEM RESPONSE:</td></tr>
<tr><td>Manual Action</td><td>Automated Action</td></tr>
<tr><td>Step 1: The requests to generate a report</td><td></td><td></td></tr>
<tr><td>Step 2: The admin clicks on the Reports menu option</td><td></td><td>Step 3: The system displays the “Reports” screen with the following inputs and controls on it:
<table>
<tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Reports</td><td>Heading</td><td>None</td></tr>
<tr><td>Generate Report</td><td>Label</td><td>None</td></tr>
<tr><td>Report type</td><td>Label</td><td>A dropdown list containing all the different</td></tr>
</table></td></tr>
</table>

		reports available
Period start date	Date/time selector	The control to select the report start date
Period end date	Date/time selector	The control to select the report end date
Generate	Button	Button used to generate the selected report.
Report Name	Label	A label where the name of the generated report will be displayed.
Download	Button	Button used to download the generated reports
Report type	Dropdown	A dropdown list containing all the different reports available
Please select the report period	Label	None.
Report Panel	Panel	Area where the

<table>
<tr><td></td><td></td><td></td><td><table><tr><td></td><td></td><td>generated report will be displayed.</td></tr><tr><td>Start Date</td><td>Label</td><td>None.</td></tr><tr><td>End Date</td><td>Label</td><td>None.</td></tr></table></td></tr>
<tr><td></td><td>Step 4: The admin selects the Packing Report”</td><td></td><td></td></tr>
<tr><td></td><td>Step 5: The admin clicks the “Generate” button</td><td></td><td>Step 6: The system retrieves
Sale_ID
OrderStatus_ID
SaleOrder_Desc
SaleOrder_Date
SaleOrder_Assign
SaleOrder_ReceiveType
Payment_Date
Payment_Amount
And
OrderStatus_ID
OrderStatus_Desc
from the <u>Sale</u> and <u>OrderStatus</u> entities which have the order status of “ready for packing” using a SQL Read.</td></tr>
<tr><td></td><td></td><td></td><td>Step 7: The system sorts the retrieved sales from the first made to the most recent</td></tr>
<tr><td></td><td></td><td></td><td>Step 8: The system displays the list of Sales to the admin as the Packing Report</td></tr>
<tr><td>ALTERNATE</td><td colspan="3"></td></tr>
<tr><td>CONCLUSION:</td><td colspan="3">The Packing report is generated from the Sales which have a status of “ready for packing”</td></tr>
<tr><td>POST-CONDITION:</td><td colspan="3">The system generates and displays the packing report</td></tr>
<tr><td>BUSINESS RULES:</td><td colspan="3">1. The packing report is generated from the Sale Orders which have a status of “ready for packing”</td></tr>
<tr><td>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:</td><td colspan="3">None</td></tr>
</table>

ASSUMPTIONS:	None
OPEN ISSUES:	None

<table>
<tr><th colspan="4">NKAP BOLTING SYSTEM</th></tr>
<tr><td>DATE:</td><td colspan="3">9/06/2021</td></tr>
<tr><td>VERSION:</td><td colspan="3">1.0</td></tr>
<tr><td>AUTHOR(S)</td><td colspan="3">Jerome Amenigy</td></tr>
<tr><td>USE CASE NAME:</td><td>Delivery Report</td><td colspan="2">USE CASE TYPE</td></tr>
<tr><td>USE CASE ID:</td><td>7.9</td><td colspan="2" rowspan="3">Business Requirements: ☐
System Analysis: ☐
System Design: ☑</td></tr>
<tr><td>PRIORITY:</td><td>High</td></tr>
<tr><td>SOURCE:</td><td>NKAP Bolting System Requirements</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR:</td><td colspan="3">Admin</td></tr>
<tr><td>PRIMARY THE SYSTEM ACTOR:</td><td colspan="3">None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case describes the event whereby an admin generates a report based on the Sales which still need to be delivered. The admin will first request to generate a delivery report on the system. The system will then retrieve the Sale Orders with an Order Status of “ready for delivery”</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• Sales have taken place on the system
• The admin is logged in to the system</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The admin wants to view all the sale orders that still needs to be delivered</td></tr>
<tr><td rowspan="4">TYPICAL COURSE OF EVENTS:</td><td rowspan="2">ACTOR ACTION:</td><td colspan="2">SYSTEM RESPONSE:</td></tr>
<tr><td>Manual Action</td><td>Automated Action</td></tr>
<tr><td>Step 1: The requests to generate a report</td><td></td><td></td></tr>
<tr><td>Step 2: The admin clicks on the Reports menu option</td><td></td><td>Step 3: The system displays the “Reports” screen with the following inputs and controls on it:
<table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr></table></td></tr>
</table>

Reports	Heading	None
Generate Report	Label	None
Report type	Label	A dropdown list containing all the different reports available
Period start date	Date/time selector	The control to select the report start date
Period end date	Date/time selector	The control to select the report end date
Generate	Button	Button used to generate the selected report.
Report Name	Label	A label where the name of the generated report will be displayed.
Download	Button	Button used to download the generated reports
Report type	Dropdown	A dropdown list containing all the

<table>
<tr>
<td></td>
<td></td>
<td></td>
<td>
<table>
<tr><td></td><td></td><td>different reports available</td></tr>
<tr><td>Please select the report period</td><td>Label</td><td>None.</td></tr>
<tr><td>Report Panel</td><td>Panel</td><td>Area where the generated report will be displayed.</td></tr>
<tr><td>Start Date</td><td>Label</td><td>None.</td></tr>
<tr><td>End Date</td><td>Label</td><td>None.</td></tr>
</table>
</td>
</tr>
<tr>
<td></td>
<td>Step 4: The admin selects the Delivery Report”</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Step 5: The admin clicks the “Generate” button</td>
<td></td>
<td>Step 6: The system retrieves
• Sale_ID
• OrderStatus_ID
• SaleOrder_Desc
• SaleOrder_Date
• SaleOrder_Assign
• SaleOrder_ReceiveType
• Payment_Date
• Payment_Amount
,
• OrderStatus_ID
• OrderStatus_Desc
,
• Delivery_ID
• Courier_ID
• Delivery_Date
• Address_ID
,
• Address_ID
• Customer_ID
• Province_ID
• Address_Line1
• Address_Line2
• Address_Line3
• Address_PostalCode
,</td>
</tr>
</table>

			• Province_ID • Province_Description , • City_ID • Province_ID • City_Description , • Suburb_ID • City_ID • Suburb_Description And • Customer_ID • Customer_Name from the Sale, OrderStatus, Delivery, Address, Province, City, Suburb and Customer entities which have the order status of "ready for delivery" using a SQL Read.
			Step 7: The system then filters through the retrieved sales and sorts them according to SaleOrder_Date.
			Step 4: The system then displays the list of retrieved Sales to the admin in the correct order as the Delivery Report
ALTERNATE			
CONCLUSION:	The delivery report is generated from the Sale, Delivery and OrderStatus tables and displayed to the admin.		
POST-CONDITION:	• The system has generated and displayed the delivery report		
BUSINESS RULES:	1. The Sales displayed in the report must have an order status of "ready for delivery"		
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	None		
ASSUMPTIONS:	None		
OPEN ISSUES:	None		

2.1.8. DELIVERY SUBSYSTEM

NKAP Bolting System

Author (s): <u>Chardé Devine Vries</u> **Date:<u>06/06/2021</u>**

Version: <u>___1.0___</u>

<table>
<tr><td>USE CASE NAME:</td><td colspan="2">Search delivery details</td><td>USE CASE TYPE</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">8.1</td><td rowspan="3">Business Requirements: ✖
System Analysis: ✖
System Design: 🖫</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">High</td></tr>
<tr><td>SOURCE:</td><td colspan="2">NKAP Bolting system</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR:</td><td colspan="3">Employee</td></tr>
<tr><td>PRIMARY THE SYSTEM ACTOR:</td><td colspan="3">None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case describes the event where an employee views a specific delivery with all its details. The system retrieves information from the <u>Delivery entity</u>, <u>Address entity</u> and <u>Courier entity</u> and displays the list of deliveries to the employee. The employee selects a delivery and the details are displayed. The use case ends when the system has successfully displayed the Delivery to the employee.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The employee must have an account and be logged into the system</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">A employee wants to view deliveries</td></tr>
<tr><td rowspan="4">TYPICAL COURSE OF EVENTS:</td><td rowspan="2">ACTOR ACTION:</td><td colspan="2">SYSTEM RESPONSE:</td></tr>
<tr><td>MANUAL ACTION</td><td>AUTOMATED ACTION</td></tr>
<tr><td>Step 1: The employee request to view deliveries.</td><td></td><td></td></tr>
<tr><td>Step 2: The employee clicks on the ‘Delivery’</td><td></td><td>Step 3: The system loads the Delivery screen with the following elements:</td></tr>
</table>

<table>
<tr><td></td><td>tab from the menu-strip.</td><td></td><td></td><td>
<table>
<tr><th>Element Name</th><th>Element Type</th><th>Notes:</th></tr>
<tr><td>Deliveries</td><td>Header</td><td>To indicate what the screen is for</td></tr>
<tr><td>tblDeliveries</td><td>Table</td><td>To show each delivery.
Retrieved from the:
<u>Delivery entity</u>
<u>Courier entity</u>
<u>Address entity</u>
<u>Province entity</u>
<u>City entity</u>
<u>Suburb entity</u>
<u>OrderStatus</u></td></tr>
<tr><td>Back</td><td>Button</td><td></td></tr>
<tr><td colspan="3">The following elements are displayed in tblDeliveries :</td></tr>
<tr><td>Delivery</td><td>Table Heading</td><td>To display the headings</td></tr>
<tr><td>Delivery ID</td><td>Table column Heading</td><td>Row contains:
Delivery_ID</td></tr>
<tr><td>Tracking Number</td><td>Table column Heading</td><td>Row contains:
Courier_TrackingNumber</td></tr>
<tr><td>Delivery Date</td><td>Table column</td><td>Row contains:</td></tr>
</table>
</td></tr>
</table>

	Heading	• Delivery_Date
Courier	*Table column Heading*	*Row contains:* • Courier_Name • Courier_Number • Courier_Email
Location	*Table column Heading*	*Row contains:* • Address_Line1 • Address_Line2 • Address_Line3 • Address_PostalCode • Province_Description • City_Description • Suburb_Description •
Delivery Status	*Table column Heading*	*Row contains:* • OrderStatus_Desc button
Order Status icon button	*Button*	*To show order status and update when clicked.*

[ALT]

ALTERNATE COURSES:	**ALT Step 3:** The system fails to load delivery page: <table><tr><th>Element Name</th><th>Element Type</th><th>Notes</th></tr><tr><td>**Error**</td><td>Label</td><td>This serves as a header for the modal.</td></tr><tr><td>**There was an error loading the Delivery page.** **Please try again later.**</td><td>Label</td><td>This is to shows that loading the page was unsuccessful.</td></tr><tr><td>**OK**</td><td>Button</td><td>None</td></tr></table>Use Case Terminated
CONCLUSION:	The use case concludes when the delivery view displays.
POST-CONDITION:	The customer can see the Delivery screen
BUSINESS RULES:	None
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	None
ASSUMPTIONS:	None
OPEN ISSUES:	None

NKAP BOLTING SYSTEM			
DATE:	13/06/2021		
VERSION:	**1.0**		
AUTHOR(S)	Sonali Badrinath		
USE CASE NAME:	**Search Courier**	**USE CASE TYPE**	
USE CASE ID:	**8.4.**	**Business Requirements:** ☐	

NKAP BOLTING SYSTEM				
DATE:	13/06/2021			
VERSION:	**1.0**			
AUTHOR(S)	Sonali Badrinath			
USE CASE NAME:	**Add Courier**	**USE CASE TYPE**		
USE CASE ID:	**8.3.**	**Business Requirements:** ☐		
PRIORITY:	Medium	**System Analysis:** ☐		
SOURCE:	NKAP Bolting System Requirements	**System Design:** ☑		
PRIMARY BUSINESS ACTOR:	Administrator			
PRIMARY THE SYSTEM ACTOR:	None			
OTHER PARTICIPATING ACTORS:	None			
OTHER INTERESTED STAKEHOLDERS:	Courier			
DESCRIPTION:	This use case describes the event where the admin wants to add a new courier on to the system. The use case begins when the admin requests to add the courier. The admin will enter all the courier details and the information will be captured in the **Courier** table. The use case ends when the courier has been successfully added.			
PRE-CONDITION:	• The admin should be logged onto the system.			
TRIGGER:	The admin requesting to add a new courier.			
TYPICAL COURSE OF EVENTS:	**ACTOR ACTION:**	**SYSTEM RESPONSE:**		
		Manual Action	Automated Action	
	Step 1: The admin requests to			

PRIORITY:	High	**System Analysis:** ☐
SOURCE:	NKAP Bolting System Requirements	**System Design:** ☑

<table>
<tr><td></td><td>add a new courier.</td><td></td><td></td></tr>
<tr><td></td><td>Step 2: The admin will click on the “Add Courier” menu item on the navigation bar.</td><td></td><td>Step 3: The system responds by loading the “Add Courier” modal with the following controls:
<table>
<tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Add Courier</td><td>Heading</td><td>None</td></tr>
<tr><td>Courier Name:</td><td>Label</td><td>This is the label for the name of the courier</td></tr>
<tr><td>Name:</td><td>Textbox</td><td>None</td></tr>
<tr><td>Courier Type:</td><td>Label</td><td>This is the label for the type of the courier</td></tr>
<tr><td>Type:</td><td>Dropdown</td><td>Populated with different titles to select.</td></tr>
<tr><td>Courier Number:</td><td>Label</td><td>This is the label for the number of the courier</td></tr>
<tr><td>Number:</td><td>Textbox</td><td>None</td></tr>
<tr><td>Courier Email:</td><td>Label</td><td>This is the label for the email of the courier</td></tr>
<tr><td>Email:</td><td>Textbox</td><td>None</td></tr>
<tr><td>Add</td><td>Button</td><td>None</td></tr>
<tr><td>Cancel</td><td>Button</td><td>This button is used when the admin decides to terminate the process of adding a new courier.</td></tr>
</table></td></tr>
<tr><td></td><td>Step 4: The admin enters the details in</td><td></td><td></td></tr>
</table>

<table>
<tr><td></td><td>the respective places:
Name
Type
Number
Email</td><td></td><td></td></tr>
<tr><td></td><td>Step 5: The admin clicks on the “Add” button [Alt]</td><td></td><td>Step 6: The system displays a modal for the admin to confirm the addition of the courier:<table>
<tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Confirm to Proceed</td><td>Label</td><td>This serves as a header for the modal.</td></tr>
<tr><td>Would you like to proceed with the addition of this courier?</td><td>Label</td><td>This is to allow the admin to finalize the decision to submit the addition of the courier with the details provided.</td></tr>
<tr><td>Yes</td><td>Button</td><td>Proceeds to the next step (Step 7)</td></tr>
<tr><td>No</td><td>Button</td><td>Allows user to enter courier details again.</td></tr>
</table></td></tr>
<tr><td></td><td>Step 7: The admin selects the “Yes” button. [Alt]</td><td></td><td>Step 8: The system captures and validates the information entered by the admin against the Courier table [Alt]:<table>
<tr><th>Input Name</th><th>Attribute in table</th><th>Validation requirements</th></tr>
</table></td></tr>
</table>

Courier Name	Courier_Name	Required
Type	CourierType_Desc	Required
Number	Courier_Number	Required, 10 digits
Email	Courier_Email	Required, maximum of 200 characters

Step 9: The system uses a SQL insert query to save the captured information in the **Courier** table **[Alt]**:

Information to be saved:	Attribute in table	Details
Courier ID	Courier_ID	Generated by the system by reading the last Courier_ID from the **Courier** table and adding it by one.
Courier Name	Courier_Name	None
Courier Type	CourierType_ID	Match the CourierType_ID of the selected CourierType_Desc
Courier Number	Courier_Number	None
Courier Email	Courier_Email	None

			Step 10: The system saves the new courier details in the relevant entities and displays a success modal: <table><tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Success</td><td>Modal Heading</td><td>None</td></tr><tr><td>Courier information was successfully created.</td><td>Label</td><td>Label to indicate the courier information is saved.</td></tr><tr><td>Ok</td><td>Button</td><td>None</td></tr></table>
ALTERNATE COURSES:	**[Alt Step 5]**: The admin clicks on the “Cancel” button which terminates this use case. Return to step 4		
	[Alt Step 7]: The admin clicks on the “No” button which makes the admin **Return to step 4.**		
	[Alt Step 8]: The details entered were not in the right format, so the validation failed. The system will display a validation unsuccessful modal to the admin. <table><tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>VALIDATION UNSUCCESSFUL</td><td>Label</td><td>This serves as a header for the modal.</td></tr><tr><td>The information provided in the given parameters is not in the correct format.</td><td>Label</td><td>This is to show that the courier information was incorrectly entered.</td></tr><tr><td>BACK</td><td>Button</td><td>This button allows the admin the return to the ‘Add Courier’ screen to edit the entered information in the correct format.</td></tr></table> After the admin clicks on the “**Back**” button the system will show the admin where the validation errors occurred by showing a red outline where the validation failed.		

<table>
<tr><td></td><td>[Alt Step 10]: The new courier could not be added to the system. The system will display a modal informing the admin that the saving of the courier information was unsuccessful.

<table>
<tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>UNSUCCESSFUL</td><td>Label</td><td>This serves as a header for the modal.</td></tr>
<tr><td>Failed to add new courier information. Please try again later.</td><td>Label</td><td>This is to show that the saving of the courier information was unsuccessful.</td></tr>
<tr><td>OK</td><td>Button</td><td>None</td></tr>
</table></td></tr>
<tr><td>CONCLUSION:</td><td>This use case concludes when the courier has been successfully added on the system.</td></tr>
<tr><td>POST-CONDITION:</td><td>The courier has been added to the Courier table.</td></tr>
<tr><td>BUSINESS RULES:</td><td>Only an admin can add a new courier to the system</td></tr>
<tr><td>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:</td><td>None</td></tr>
<tr><td>ASSUMPTIONS:</td><td>None</td></tr>
<tr><td>OPEN ISSUES:</td><td>None</td></tr>
</table>

PRIMARY BUSINESS ACTOR:	Admin
PRIMARY THE SYSTEM ACTOR:	None
OTHER PARTICIPATING ACTORS:	None

<table>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case describes the event where the admin wishes to search for a courier. The admin will enter the search parameters like the suppler name. The system will search for the courier based on the parameters the admin entered. This use case concludes when the courier is shown the results of the search query.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">The admin should be logged on to the system.
The courier should be already exist in the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The admin wishes to search for a courier.</td></tr>
<tr><td rowspan="4">TYPICAL COURSE OF EVENTS:</td><td rowspan="2">ACTOR ACTION:</td><td colspan="2">SYSTEM RESPONSE:</td></tr>
<tr><td>Manual Action</td><td>Automated Action</td></tr>
<tr><td>Step 1: The admin would like to search for a courier.</td><td></td><td></td></tr>
<tr><td>Step 2: The admin clicks on the “Search Courier” screen.</td><td></td><td>Step 3: The system responds by loading the “Search Courier” screen with the following controls:
<table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Search Courier</td><td>Heading</td><td>None</td></tr><tr><td>Courier Name:</td><td>Label</td><td>This is the label to prompt the user to enter a search query.</td></tr><tr><td>Search Name:</td><td>Textbox</td><td>None</td></tr></table></td></tr>
</table>

Search icon	Icon button	None
Courier Table	Table	This is to show the Couriers that are saved in the database in a table.
Name	Column	This is the column for the Courier name
Type	Column	This is the column for the courier type
Number	Column	This is the column for the number of the courier
Email	Column	This is the column for the emthe courier
Options	Icon	This is to show the update and delete button for each row.
Update	Button	Present in Options dropdown. Used to update the information of a courier.
Delete	Button	Present in Options

<table>
<tr>
<td></td>
<td></td>
<td></td>
<td><table><tr><td></td><td></td><td>dropdown. Used to delete courier details.</td></tr></table></td>
</tr>
<tr>
<td></td>
<td>Step 4: The admin inputs the search criteria and clicks on the search icon.</td>
<td></td>
<td>Step 5: The system captures and validates the information against the Courier table [Alt]:
<table>
<tr><th>Input Name</th><th>Attribute Name</th><th>Validation Requirments</th></tr>
<tr><td>Search Name:</td><td>Courier_Name</td><td>Maximum of 50 characters</td></tr>
</table></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Step 6: A SQL read query is used to read the following attributes from the Courier table based on the search criteria:

Courier table:

Courier_ID(PK)
Courier_Name attribute information displayed in the Name column.
Courier_Type attribute information displayed in the Type column.
Courier_Number attribute information displayed in the Number column.
Courier_Email attribute information displayed in the Email column.
</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Step 7: The system will show the search results received from step 6 in a table. [Alt]</td>
</tr>
<tr>
<td>ALTERNATE</td>
<td colspan="3">Step 5: The details entered were not in the right format, so the validation failed or there are not any existing data on the requested information. The system will display a validation unsuccessful modal to the admin.</td>
</tr>
</table>

	<table><tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Error</td><td>Label</td><td>This serves as a header for the modal.</td></tr><tr><td>No matches found. Please check the format of the entered information.</td><td>Label</td><td>This is to show that there were no matches found.</td></tr><tr><td>OK</td><td>Button</td><td>None</td></tr></table>
CONCLUSION:	The use case concludes when the admin receives the results of the searched criteria.
POST-CONDITION:	• The admin receives the results of the couriers with the same searched criteria
BUSINESS RULES:	**1.** None
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	None
ASSUMPTIONS:	None
OPEN ISSUES:	None

<table>
<tr><td>USE CASE NAME:</td><td>Update Courier</td><td colspan="2">USE CASE TYPE</td></tr>
<tr><td>USE CASE ID:</td><td>8.5.</td><td colspan="2" rowspan="3">Business Requirements: ☐
System Analysis: ☐

System Design: ☑</td></tr>
<tr><td>PRIORITY:</td><td>3</td></tr>
<tr><td>SOURCE:</td><td>NKAP Bolting System Requirements</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR:</td><td colspan="3">Admin</td></tr>
<tr><td>PRIMARY THE SYSTEM ACTOR:</td><td colspan="3">None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case describes the event in which an admin wants to update the details of a courier. The system will display the previous information that was added before and then the admin will add the new information. This use case ends when the information has been updated successfully.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The admin should be logged on to the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The admin wishes to update the details of the courier.</td></tr>
<tr><td rowspan="3">TYPICAL COURSE OF EVENTS:</td><td rowspan="2">Actor Action:</td><td colspan="2">Actor Action:</td></tr>
<tr><td>Manual Action</td><td>Automated Action</td></tr>
<tr><td>Step 1: The admin requests to update the details of an employee by selecting</td><td></td><td>Step 2: The system invokes use case 8.4. Search Courier.</td></tr>
</table>

<table>
<tr><td></td><td>the “Update” option in the Option column dropdown of the line of information of which they want to update.</td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td>Step 3: The system responds by loading the “Update Courier” modal with the following controls:

<table>
<tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Update Courier</td><td>Heading</td><td>None</td></tr>
<tr><td>Courier Name:</td><td>Label</td><td>This is the label for the name of the courier</td></tr>
<tr><td>Name:</td><td>Textbox</td><td>None</td></tr>
<tr><td>Courier Type:</td><td>Label</td><td>This is the label for the type of the courier</td></tr>
<tr><td>Type:</td><td>Dropdown</td><td>Populated with different types to select.</td></tr>
<tr><td>Courier Number:</td><td>Label</td><td>This is the label for the number of the courier</td></tr>
<tr><td>Number:</td><td>Textbox</td><td>None</td></tr>
<tr><td>Courier Email:</td><td>Label</td><td>This is the label for the email of the courier</td></tr>
<tr><td>Email:</td><td>Textbox</td><td>None</td></tr>
</table></td></tr>
</table>

<table>
<tr><td></td><td></td><td></td><td></td><td>
<table>
<tr><td>Done</td><td>Button</td><td>This button is used for when the admin is done updating the information and wants to proceed to update the new information on the system.</td></tr>
<tr><td>Cancel</td><td>Button</td><td>This button is used when the admin decides to terminate the process of adding a new courier.</td></tr>
</table>
</td></tr>
<tr><td></td><td>Step 4: The admin enters the new information and clicks on the “Done”. [Alt]</td><td></td><td></td><td>
Step 5: The system captures and validates the information entered by the admin against the Courier table [Alt]:
<table>
<tr><th>Input Name</th><th>Attribute in table</th><th>Validation requirements</th></tr>
<tr><td>Courier Name</td><td>Courier_Name</td><td>Required</td></tr>
<tr><td>Type</td><td>CourierType_Desc</td><td>Required</td></tr>
<tr><td>Number</td><td>Courier_Number</td><td>Required, 10 digits</td></tr>
<tr><td>Email</td><td>Courier_Email</td><td>Required, maximum of 200 characters</td></tr>
</table>
</td></tr>
<tr><td></td><td></td><td></td><td></td><td>
Step 6: The system uses a SQL insert query to save the captured information in the Courier table [Alt]:
<table>
<tr><th>Information to be saved:</th><th>Attribute in table</th><th>Details</th></tr>
</table>
</td></tr>
</table>

			<table><tr><td>Courier Name</td><td>Courier_Name</td><td>None</td></tr><tr><td>Courier Type</td><td>CourierType_ID</td><td>Match the CourierType_ID of the selected CourierType_Desc</td></tr><tr><td>Courier Number</td><td>Courier_Number</td><td>None</td></tr><tr><td>Courier Email</td><td>Courier_Email</td><td>None</td></tr></table>
			Step 7: The system saves the new courier details in the relevant entities and displays a success modal: <table><tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Success</td><td>Modal Heading</td><td>None</td></tr><tr><td>Courier information was successfully updated.</td><td>Label</td><td>Label to indicate the courier information is saved.</td></tr><tr><td>Ok</td><td>Button</td><td>None</td></tr></table>
ALTERNATE COURSES:	**[Alt Step 4]**: The admin clicks on the "**Cancel**" button which terminates this use case. Return to **step 1.**		
	[Alt Step 5]: The details entered were not in the right format, so the validation failed. The system will display a validation unsuccessful modal to the admin. <table><tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>VALIDATION UNSUCCESSFUL</td><td>Label</td><td>This serves as a header for the modal.</td></tr><tr><td>The information provided in the given parameters is not in the correct format.</td><td>Label</td><td>This is to show that the courier information was incorrectly entered.</td></tr></table>		

BACK	Button	This button allows the admin the return to the ‘**Update Courier**’ screen to edit the entered information in the correct format.

After the admin clicks on the “**Back**” button the system will show the admin where the validation errors occurred by showing a red outline where the validation failed.

[Alt Step 6]: The updated courier information could not be added to the system. The system will display a modal informing the admin that the saving of the courier information was unsuccessful.

Input Name	Control Type	Notes
UNSUCCESSFUL	Label	This serves as a header for the modal.
Failed to update courier information. Please try again later.	Label	This is to show that the saving of the courier information was unsuccessful.
OK	Button	None

CONCLUSION:	This use case concludes when the courier information has been successfully updated on the system.
POST-CONDITION:	• The courier has been updated in the **Courier** table.
BUSINESS RULES:	**1.** Only an admin can update the information of the courier in the system.
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	None
ASSUMPTIONS:	None
OPEN ISSUES:	None

<table>
<tr><th colspan="4">NKAP BOLTING SYSTEM</th></tr>
<tr><td>DATE:</td><td colspan="3">27-08-21</td></tr>
<tr><td>VERSION:</td><td colspan="3">1.0</td></tr>
<tr><td>AUTHOR(S)</td><td colspan="3">Sonali Badrinath</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Collect Order</td><td>USE CASE TYPE</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">8.6.</td><td rowspan="3">Business Requirements: □
System Analysis: □

System Design: ☑</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">High</td></tr>
<tr><td>SOURCE:</td><td colspan="2">NKAP Bolting System Requirements</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR:</td><td colspan="3">Customer</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR:</td><td colspan="3">Employee</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS :</td><td colspan="3">Customer
Owner</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case describes the event where an employee searches for the specific order that the customer has arrived to collect on the system. Once the customer has collected the order, the employee will indicate so on the system and the system will update the order status of the order appropriately in the <u>Sale Entity.</u> The use case concludes when the order status has been updated to collected.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The Employee must be logged in to the system.
• The sale must exist on the system and have an order status of “Ready for Collection”.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The customer wants to collect their order.</td></tr>
<tr><td>TYPICAL</td><td></td><td colspan="2">SYSTEM RESPONSE:</td></tr>
</table>

<table>
<tr><td>COURSE OF EVENTS:</td><td></td><td>Manual Action</td><td>Automated Action</td></tr>
<tr><td></td><td>Step 1: The customer wants to collect their order.</td><td>Step 2: The employee requests the client for their order details</td><td></td></tr>
<tr><td></td><td>Step 3: The customer provides their order details such as their name and sale number.</td><td>Step 4: The employee will select the Admin Menu item on the navigation bar and click the “Online-Sales” dropdown menu item.</td><td></td></tr>
<tr><td></td><td></td><td>Step 5: The system invokes Use Case 5.1 Search Online Sales.</td><td></td></tr>
<tr><td></td><td></td><td>Step 6: The employee clicks the menu icon for the specific order. The employee selects the collected option.</td><td>Step 7: The system updates Order Status to “Collected by making use of a SQL Update query:
<table>
<tr><th>Details to be Updated</th><th>Attribute in Entity</th><th>Notes</th></tr>
<tr><td>Order Sta tus</td><td>OrderStatus_ID in the <u>Sale Entity</u></td><td>None</td></tr>
</table></td></tr>
<tr><td>ALTERNATE</td><td colspan="3">ALT-STEP Step 6: Based on not finding the order that match the details provided by the customer, the employee may Terminate the use case.</td></tr>
<tr><td>CONCLUSION:</td><td colspan="3">The use case concludes when the Order has been collected by the customer and the order status is updated to collected.</td></tr>
<tr><td>POST-CONDITION:</td><td colspan="3">The Order Status has been updated to “Collected”.</td></tr>
</table>

Business Rules:	1. Order status can only be updated to “Collected” if it has an order status of “Ready for Collection”. 2. Once an order status has been updated to “Collected” it cannot be updated.
Implementation Constraints and Specifications:	None
Assumptions:	None
Open Issues:	None

<table>
<tr><td colspan="3">NKAP BOLTING SYSTEM</td></tr>
<tr><td>DATE:</td><td colspan="2">27-08-21</td></tr>
<tr><td>VERSION:</td><td colspan="2">1.0</td></tr>
<tr><td>AUTHOR(S)</td><td colspan="2">Sonali Badrinath</td></tr>
<tr><td>USE CASE NAME:</td><td>Deliver Order</td><td>USE CASE TYPE</td></tr>
<tr><td>USE CASE ID:</td><td>8.7.</td><td rowspan="3">Business Requirements: □
System Analysis: □
System Design: ☑</td></tr>
<tr><td>PRIORITY:</td><td>High</td></tr>
<tr><td>SOURCE:</td><td>NKAP Bolting System Requirements</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR:</td><td colspan="2">Employee</td></tr>
<tr><td>PRIMARY THE SYSTEM ACTOR:</td><td colspan="2">None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="2">None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS :</td><td colspan="2">Customer
Owner</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="2">This use case describes the event where an employee searches for the specific order that has been delivered to a customer. Once the order has been delivered to the customer, the employee will indicate so on the system and the system will update the order status of the order appropriately in the <u>Sale Entity.</u> The use case concludes when the order status has been updated to delivered.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="2">• The Employee must be logged in to the system.
• The sale must exist on the system and have an order status of “Ready for Delivery”.</td></tr>
<tr><td>TRIGGER:</td><td colspan="2">An employee wants to record on the system that the order has been delivered to the customer.</td></tr>
<tr><td>TYPICAL</td><td></td><td>SYSTEM RESPONSE:</td></tr>
</table>

<table>
<tr><td>Course of Events:</td><td></td><td>Manual Action</td><td>Automated Action</td></tr>
<tr><td></td><td>Step 1: The employee to record that he/she has delivered the order to the customer.</td><td>Step 2: The employee will select the Admin Menu item on the navigation bar and click the “Online-Sales” dropdown menu item.</td><td></td></tr>
<tr><td></td><td></td><td>Step 3: The system invokes Use Case 5.1 Search Online Sales.</td><td></td></tr>
<tr><td></td><td></td><td>Step 4: The employee clicks the menu icon for the specific order. The employee selects the delivered option.</td><td>Step 5: The system updates Order Status to “Delivered by making use of a SQL Update query:
<table>
<tr><th>Details to be Updated</th><th>Attribute in Entity</th><th>Notes</th></tr>
<tr><td>Order Status</td><td>OrderStatus_ID in the <u>Sale Entity</u></td><td>None</td></tr>
</table></td></tr>
<tr><td>Alternate</td><td colspan="3">None</td></tr>
<tr><td>Conclusion:</td><td colspan="3">The use case concludes when the Order has been delivered by the employee and the order status is updated to delivered.</td></tr>
<tr><td>Post-condition:</td><td colspan="3">The Order Status has been updated to “Delivered”.</td></tr>
<tr><td>Business Rules:</td><td colspan="3">1. Order status can only be updated to “Delivered” if it has an order status of “Ready for Delivery”.
2. Once an order status has been updated to “Delivered” it cannot be updated.</td></tr>
</table>

Implementation Constraints and Specifications:	None
Assumptions:	None
Open Issues:	None

2.1.9. PRODUCTS SUBSYSTEM

<table>
<tr><th colspan="4">NKAP BOLTING SYSTEM</th></tr>
<tr><td>DATE:</td><td colspan="3">05/06/2021</td></tr>
<tr><td>VERSION:</td><td colspan="3">1.0</td></tr>
<tr><td>AUTHOR(S)</td><td colspan="3">Jerome Amenigy</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Add Product Category</td><td>USE CASE TYPE</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">9.1</td><td rowspan="3">Business Requirements: ☐
System Analysis: ☐
System Design: ☑</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">Medium</td></tr>
<tr><td>SOURCE:</td><td colspan="2">NKAP Bolting System Requirements</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR:</td><td colspan="3">Admin</td></tr>
<tr><td>PRIMARY THE SYSTEM ACTOR:</td><td colspan="3">None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case describes the process whereby an admin adds a new Product Category to the system. The admin will select an option to add a product category to the system and then enter the specified product category’s information and upon saving the addition the system will add it to the <u>ProductCategory</u> Entity</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The admin is logged in to the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The admin wants to add a new product category to the system.</td></tr>
<tr><td rowspan="4">TYPICAL COURSE OF EVENTS:</td><td rowspan="2">ACTOR ACTION:</td><td colspan="2">SYSTEM RESPONSE:</td></tr>
<tr><td>Manual Action</td><td>Automated Action</td></tr>
<tr><td>Step 1: The admin requests to add a new product category on the system</td><td></td><td></td></tr>
<tr><td>Step 2: The admin clicks on the “Add” button in</td><td></td><td>Step 3: The system then displays the Add Product category modal with the following controls:</td></tr>
</table>

<table>
<tr><td></td><td>the Products Screen.</td><td></td><td>
<table>
<tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Add Product Category</td><td>Heading</td><td>None</td></tr>
<tr><td>Product Category Name:</td><td>Label</td><td>A label for the name of the Product Category</td></tr>
<tr><td>Name</td><td>Textbox</td><td>None</td></tr>
<tr><td>Add</td><td>Button</td><td>The button is used to save the product category onto the system</td></tr>
<tr><td>Cancel</td><td>Button</td><td>The button is used to terminate the process of this use case</td></tr>
</table>
</td></tr>
<tr><td></td><td>Step 4: The admin enters the specified product category's details.</td><td></td><td></td></tr>
<tr><td></td><td>Step 5: The admin clicks on the "Add" button</td><td></td><td>Step 6: The system captures the product category's details and validates it against the Product Category Table.
<table>
<tr><th>Input Name</th><th>Attribute in Table</th><th>Validation requirements</th></tr>
<tr><td>Name:</td><td>ProductCategory_Desc</td><td>Maximum character length of 50 characters.</td></tr>
</table>
</td></tr>
<tr><td></td><td></td><td></td><td>Step 7: The system uses an insert query to save the captured information in the ProductCategory Table
<table>
<tr><th>Information to be saved</th><th>Attribute in table</th><th>Details</th></tr>
</table>
</td></tr>
</table>

ProductCategory_ID	ProductCategory_ID	Generated from the system by reading the last Product Category_ID and adding 1
Name	ProductCategory_Desc	None

ALTERNATE COURSES:	**Step 5b**: The user clicks on "cancel" which terminates the use case
	Step 6b: The provided product category details were not in the right format; the system will outline the place where the validation error occurred.
	Step 7b: The system cannot save the new product category; the system then displays a modal telling the admin that the addition was unsuccessful.

Input Name	Control Type	Notes
Error	Label	This serves as a header for the modal.
There was an error while saving the new product category. Please try again later.	Label	This is to show that the saving of the product category information was unsuccessful.
OK	Button	None

CONCLUSION:	This use case concludes once the product category has been successfully added on to the system
POST-CONDITION:	• The product category has been added to the **ProductCategory** Table
BUSINESS RULES:	1. Only the admin can add product categories to the system
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	None
ASSUMPTIONS:	None
OPEN ISSUES:	None

<table>
<tr><th colspan="4">NKAP BOLTING SYSTEM</th></tr>
<tr><td>DATE:</td><td colspan="3">05/06/2021</td></tr>
<tr><td>VERSION:</td><td colspan="3">1.0</td></tr>
<tr><td>AUTHOR(S)</td><td colspan="3">Jerome Amenigy</td></tr>
<tr><td>USE CASE NAME:</td><td>Search Product Category</td><td colspan="2">USE CASE TYPE</td></tr>
<tr><td>USE CASE ID:</td><td>9.2</td><td colspan="2" rowspan="3">Business Requirements: ☐
System Analysis: ☐
System Design: ☑</td></tr>
<tr><td>PRIORITY:</td><td>Medium</td></tr>
<tr><td>SOURCE:</td><td>NKAP Bolting System Requirements</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR:</td><td colspan="3">Admin</td></tr>
<tr><td>PRIMARY THE SYSTEM ACTOR:</td><td colspan="3">None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case describes the process where the admin searches for a specific Product Category. Once the admin enters the search criteria the system will retrieve the matching results from the <u>ProductCategory</u> entity the use case ends once the retrieved results have been displayed to the admin.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">The admin is logged in to the system
The ProductCategory exists on the system</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The admin wants to search for a product category.</td></tr>
<tr><td rowspan="4">TYPICAL COURSE OF EVENTS:</td><td rowspan="2">ACTOR ACTION:</td><td colspan="2">SYSTEM RESPONSE:</td></tr>
<tr><td>Manual Action</td><td>Automated Action</td></tr>
<tr><td>Step 1: The admin requests to search for a product category</td><td></td><td></td></tr>
<tr><td>Step 2: The admin clicks on the “Products” screen.</td><td></td><td>Step 3: The system responds by loading the “Products” screen with the following controls:

<table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr></table></td></tr>
</table>

			<table><tr><td>**Products**</td><td>Heading</td><td>None</td></tr><tr><td>**Search Bar**</td><td>Textbox</td><td>Search bar used to search for category types</td></tr><tr><td>**Product Category**</td><td>Label</td><td>Name of the selected product screen.</td></tr><tr><td>**Product Category Table**</td><td>Table</td><td>This is to show the Product Categories that are saved in the database in a table.</td></tr><tr><td>**Product Category Name**</td><td>Column</td><td>This is the column for the Product Category names.</td></tr><tr><td>**Product Category ID**</td><td>Column</td><td>This is the column for the Product Category IDs.</td></tr><tr><td>**Actions**</td><td>Column</td><td>Column to display update and delete buttons</td></tr><tr><td>**Product Category / Category Type/ Product Item**</td><td>Button Group</td><td>This is the button group whereby the admin can select which level of products he wants to work with</td></tr><tr><td>**Options**</td><td>Button</td><td>This is to show the update and delete button for each row.</td></tr><tr><td>**Add**</td><td>Button</td><td>None</td></tr><tr><td>**Update**</td><td>Button</td><td>None</td></tr><tr><td>**Delete**</td><td>Button</td><td>None</td></tr></table>
	Step 4: The admin selects the “Product Category”		

<table>
<tr><td></td><td>radio button.</td><td></td><td></td></tr>
<tr><td></td><td>Step 5: The admin enters the product category search criteria.</td><td></td><td></td></tr>
<tr><td></td><td>Step 6: The admin clicks on the search icon.</td><td></td><td>Step 7: The system captures and validates the information against the Product Category table.
<table>
<tr><th>Input Name</th><th>Attribute Name</th><th>Validation Requirements</th></tr>
<tr><td>Name:</td><td>ProductCategory_Desc</td><td>Maximum of 50 characters</td></tr>
<tr><td>ID:</td><td>ProductCategory_ID</td><td>Characters must be integers</td></tr>
</table></td></tr>
<tr><td></td><td></td><td></td><td>Step 8: The system uses a read query to read the matching entries from the Product Category entity. The system retrieves:

ProductCategory table:
ProductCategory_ID
ProductCategory_Desc</td></tr>
<tr><td></td><td></td><td></td><td>Step 9: The system displays the retrieved search results in a table.</td></tr>
<tr><td></td><td>Step 10: The admin selects the product category they were searching for.</td><td></td><td></td></tr>
<tr><td>ALTERNATE</td><td colspan="3">Step 7b: The validation fails, and the admin is prompted to enter a new search query.</td></tr>
<tr><td>COURSES:</td><td colspan="3"></td></tr>
<tr><td>CONCLUSION:</td><td colspan="3">The use case concludes once the system displays the search results to the admin and the admin selects the searched product category.</td></tr>
<tr><td>POST-CONDITION:</td><td colspan="3">The admin has received the results of the search</td></tr>
<tr><td>BUSINESS RULES:</td><td colspan="3">1. None</td></tr>
</table>

IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	None
ASSUMPTIONS:	None
OPEN ISSUES:	None

<table>
<tr><th colspan="4">NKAP BOLTING SYSTEM</th></tr>
<tr><td>DATE:</td><td colspan="3">05/06/2021</td></tr>
<tr><td>VERSION:</td><td colspan="3">1.0</td></tr>
<tr><td>AUTHOR(S)</td><td colspan="3">Jerome Amenigy</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Update Product Category</td><td>USE CASE TYPE</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">9.3</td><td>Business Requirements: ☐</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">Medium</td><td>System Analysis: ☐</td></tr>
<tr><td>SOURCE:</td><td colspan="2">NKAP Bolting System Requirements</td><td>System Design: ☑</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR:</td><td colspan="3">Admin</td></tr>
<tr><td>PRIMARY THE SYSTEM ACTOR:</td><td colspan="3">None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case describes the event whereby an already existing Product Category is updated by an admin. Firstly the admin indicates that they want to update a product category, next the system invokes Use Case 9.2 Search category, after the relevant information is displayed the admin chooses to edit the product category. After editing and saving the changes the system updates the <u>ProductCategory</u> entity with the changes made.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">The ProductCategory already exists on the system
The admin is logged in to the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The admin wants to update the Product Category on the system.</td></tr>
<tr><td rowspan="2">TYPICAL COURSE OF EVENTS:</td><td rowspan="2">ACTOR ACTION:</td><td colspan="2">SYSTEM RESPONSE:</td></tr>
<tr><td>Manual Action</td><td>Automated Action</td></tr>
</table>

<table>
<tr><td></td><td>Step 1: The admin requests to update a Product Category.</td><td></td><td>Step 2: The system invokes UC 9.2 “Search Product Category”</td></tr>
<tr><td></td><td>Step 3: The admin clicks on the update button</td><td></td><td>Step 4: The system responds by loading the “Update Product Category” modal with the following controls:
<table>
<tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Update Product Category</td><td>Heading</td><td>None</td></tr>
<tr><td>Product Category Name</td><td>Label</td><td>Label for the name of the product category</td></tr>
<tr><td>Name</td><td>Textbox</td><td>The textbox through which the name for the product category will be provided</td></tr>
<tr><td>Update</td><td>Button</td><td>None</td></tr>
<tr><td>Cancel</td><td>Button</td><td>None</td></tr>
</table></td></tr>
<tr><td></td><td>Step 5: The admin edits and makes all the changes they want to make to the product category.</td><td></td><td></td></tr>
<tr><td></td><td>Step 6: The admin clicks the “Update” button</td><td></td><td>Step 7: The system captures and validates the changes made to the product category entered by the admin against the product category table.
<table>
<tr><th>Input Name</th><th>Attribute in Table</th><th>Validation Requirements</th></tr>
</table></td></tr>
</table>

			Name | ProductCategory_Desc | Maximum of 50 characters
			Step 8: The system updates the ProductCategory table with the changes made.

Information to be updated	Attribute in table	Details
ProductCategory_ID	Product_ID	None
Name	ProductCategory_Desc	None

ALTERNATE COURSES:	**Step 6b:** The admin clicks on the "Cancel" button which terminates the use case
	Step 7b: The validation of the entered information fails; the system will outline the information that does not meet the validation requirements.
	Step 8b: The changes made to the product category could not be saved. The system will display a modal stating the update was unsuccessful

Input Name	Control Type	Notes
Error	Label	This serves as a header for the modal.
There was an error while saving the new product category information. Please try again later.	Label	This is to show that the saving of the product category information was unsuccessful.
OK	Button	None

CONCLUSION:	This use case concludes when the product category information has been successfully updated on the system.
POST-CONDITION:	• The product category is updated in the ProductCategory table
BUSINESS RULES:	1. Only an admin can update Product Categories in the system
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	None
ASSUMPTIONS:	None
OPEN ISSUES:	None

<table>
<tr><th colspan="4">NKAP BOLTING SYSTEM</th></tr>
<tr><td>DATE:</td><td colspan="3">05/06/2021</td></tr>
<tr><td>VERSION:</td><td colspan="3">1.0</td></tr>
<tr><td>AUTHOR(S)</td><td colspan="3">JEROME AMENIGY</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Delete Product Category</td><td>USE CASE TYPE</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">9.4</td><td rowspan="3">Business Requirements: ☐
System Analysis: ☐
System Design: ☑</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">Medium</td></tr>
<tr><td>SOURCE:</td><td colspan="2">NKAP Bolting System Requirements</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR:</td><td colspan="3">Admin</td></tr>
<tr><td>PRIMARY THE SYSTEM ACTOR:</td><td colspan="3">None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case describes the process where the admin deletes a Product Category from the system. Upon indicating that the admin wants to delete a product category the system will invoke use case 9.2, the admin will then proceed to delete the selected product category and the system will remove the selected product category from the <u>ProductCategory</u> entity.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The admin is logged in to the system
• The Product category already exists on the system</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The admin wants to delete a product category</td></tr>
<tr><td rowspan="4">TYPICAL COURSE OF EVENTS:</td><td rowspan="2">ACTOR ACTION:</td><td colspan="2">SYSTEM RESPONSE:</td></tr>
<tr><td>Manual Action</td><td>Automated Action</td></tr>
<tr><td>Step 1: The admin requests to delete a product category</td><td></td><td>Step 2: The system invokes use case 9.2 Search Product Category</td></tr>
<tr><td>Step 3: The admin clicks on the delete button</td><td></td><td>Step 4: The system displays a modal, with the following controls:

<table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr></table></td></tr>
</table>

<table>
<tr><td></td><td></td><td></td><td>
<table>
<tr><td>Confirm deletion</td><td>Heading</td><td>This serves as a header for the modal.</td></tr>
<tr><td>Are you sure you want to delete this product category?</td><td>Label</td><td>None</td></tr>
<tr><td>Yes</td><td>Button</td><td>This is to confirm that the admin wants to delete a Product Category.</td></tr>
<tr><td>No</td><td>Button</td><td>This is selected when the admin does not want to delete a product category.</td></tr>
</table>
</td></tr>
<tr><td></td><td>Step 5: The admin clicks on the “Yes” button to confirm the deletion of the product category.</td><td></td><td>Step 6: The system removes the selected product category from the product category entity.
<table>
<tr><th>Information to be updated</th><th>Attribute in table</th><th>Details</th></tr>
<tr><td>ProductCategory_ID</td><td>Product_ID</td><td>None</td></tr>
<tr><td>Name</td><td>ProductCategory_Desc</td><td>None</td></tr>
</table>
</td></tr>
<tr><td></td><td></td><td></td><td>Step 7: The system removes the deleted product category from the table in the products screen.</td></tr>
<tr><td rowspan="2">ALTERNATE

COURSES:</td><td colspan="3">[Alt Step 5]: The admin clicks on the “No” button and the use case is terminated.</td></tr>
<tr><td colspan="3">[Alt Step 6]: The product category could not be deleted from the system. The system will display a modal informing the admin that the deletion of the product category was unsuccessful.</td></tr>
</table>

	<table><tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>**Error**</td><td>Label</td><td>This serves as a header for the modal.</td></tr><tr><td>**There was an error while deleting the product category. Please try again later.**</td><td>Label</td><td>This is to show that the deletion of the product category information was unsuccessful.</td></tr><tr><td>**OK**</td><td>Button</td><td>None</td></tr></table>
CONCLUSION:	The use case concludes once the selected product category has been successfully deleted from the system.
POST-CONDITION:	• The product category has been removed from the ProductCategory entity.
BUSINESS RULES:	1. Only the admin can delete a product category.
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	None
ASSUMPTIONS:	None
OPEN ISSUES:	None

NKAP BOLTING SYSTEM			
DATE:	**05/06/2021**		
VERSION:	**1.0**		
AUTHOR(S)	**JEROME AMENIGY**		
USE CASE NAME:	**Add Category Type**	**USE CASE TYPE**	
USE CASE ID:	**9.5**	**Business Requirements:**	☐
PRIORITY:	Medium	**System Analysis:**	☐
SOURCE:	NKAP Bolting System Requirements	**System Design:**	☑
PRIMARY BUSINESS ACTOR:	Admin		
PRIMARY THE SYSTEM ACTOR:	None		
OTHER PARTICIPATING ACTORS:	None		

<table>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case describes the process whereby an admin adds a new Category Type to the system. The admin will select an option to add a category type to the system and then enter the specified category type’s information and upon saving the addition the system will add it to the <u>CategoryType</u> Entity.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">The related ProductCategory already exists
The admin is logged in to the system</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The admin wants to add a new category type to the system.</td></tr>
<tr><td rowspan="5">TYPICAL COURSE OF EVENTS:</td><td rowspan="2">ACTOR ACTION:</td><td colspan="2">SYSTEM RESPONSE:</td></tr>
<tr><td>Manual Action</td><td>Automated Action</td></tr>
<tr><td>Step 1: The admin requests to add a new category type on the system</td><td></td><td></td></tr>
<tr><td>Step 2: The admin clicks on the “Add” button in the Products Screen.</td><td></td><td>Step 3: The system retrieves the product categories currently available on the system.
<table>
<tr><th>Information retrieved</th><th>Attribute in table</th><th>Details</th></tr>
<tr><td>ProductCategory_ID</td><td>Product_ID</td><td>None</td></tr>
<tr><td>Name</td><td>ProductCategory_Desc</td><td>None</td></tr>
</table></td></tr>
<tr><td></td><td></td><td>Step 4: The system then displays the Add Product category modal with the following controls:
<table>
<tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Add Category Type</td><td>Heading</td><td>None</td></tr>
<tr><td>Product Categories:</td><td>Dropdown</td><td>A dropdown containing all the existing product categories</td></tr>
</table></td></tr>
</table>

<table>
<tr><td></td><td></td><td></td><td>
<table>
<tr><td>Category TypeName:</td><td>Label</td><td>A label for the name of the Category type</td></tr>
<tr><td>Name</td><td>Textbox</td><td>None</td></tr>
<tr><td>Add</td><td>Button</td><td>The button is used to save the category type onto the system</td></tr>
<tr><td>Cancel</td><td>Button</td><td>The button is used to terminate the process of this use case</td></tr>
</table>
</td></tr>
<tr><td></td><td>Step 5: The admin selects the Product Category that he wants to add the category type to.</td><td></td><td></td></tr>
<tr><td></td><td>Step 6: The admin enters the category type’s details.</td><td></td><td></td></tr>
<tr><td></td><td>Step 7: The admin clicks on the add button</td><td></td><td>Step 8: The system captures the category type’s details and validates it against the CategoryType Table.
<table>
<tr><th>Input Name</th><th>Attribute in Table</th><th>Validation requirements</th></tr>
<tr><td>Name:</td><td>CategoryType _Desc</td><td>Maximum character length of 50 characters.</td></tr>
</table>
</td></tr>
<tr><td></td><td></td><td></td><td>Step 9: The system uses an insert query to save the captured information in the CategoryType Table
<table>
<tr><th>Information to be saved</th><th>Attribute in table</th><th>Details</th></tr>
</table>
</td></tr>
</table>

<table>
<tr><td></td><td><table>
<tr><td>CategoryType _ID</td><td>CategoryType _ID</td><td>Generated from the system by reading the last CategoryType _ID and adding 1</td></tr>
<tr><td>Name</td><td>CategoryType _Desc</td><td>None</td></tr>
</table></td></tr>
<tr><td rowspan="3">ALTERNATE COURSES:</td><td>Step 7b: The user clicks on “cancel” which terminates the use case</td></tr>
<tr><td>Step 8b: The provided category type’s details were not in the right format; the system will outline the place where the validation error occurred.</td></tr>
<tr><td>Step 9b: The system cannot save the new category type; the system then displays a modal telling the admin that the addition was unsuccessful.
<table>
<tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Error</td><td>Label</td><td>This serves as a header for the modal.</td></tr>
<tr><td>There was an error while saving the new category type. Please try again later.</td><td>Label</td><td>This is to show that the saving of the category information type was unsuccessful.</td></tr>
<tr><td>OK</td><td>Button</td><td>None</td></tr>
</table></td></tr>
<tr><td>CONCLUSION:</td><td>This use case concludes once the category type has been successfully added on to the system</td></tr>
<tr><td>POST-CONDITION:</td><td>• The category type has been added to the CategoryType entity.</td></tr>
<tr><td>BUSINESS RULES:</td><td>1. Only the admin can add category types to the system</td></tr>
<tr><td>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:</td><td>None</td></tr>
<tr><td>ASSUMPTIONS:</td><td>None</td></tr>
<tr><td>OPEN ISSUES:</td><td>None</td></tr>
</table>

NKAP BOLTING SYSTEM	
DATE:	**05/06/2021**

<table>
<tr><td>VERSION:</td><td colspan="3">1.0</td></tr>
<tr><td>AUTHOR(S)</td><td colspan="3">JEROME AMENIGY</td></tr>
<tr><td>USE CASE NAME:</td><td>Search Category Type</td><td colspan="2">USE CASE TYPE</td></tr>
<tr><td>USE CASE ID:</td><td>9.6</td><td colspan="2" rowspan="3">Business Requirements: ☐
System Analysis: ☐
System Design: ☑</td></tr>
<tr><td>PRIORITY:</td><td>Medium</td></tr>
<tr><td>SOURCE:</td><td>NKAP Bolting System Requirements</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR:</td><td colspan="3">Admin</td></tr>
<tr><td>PRIMARY THE SYSTEM ACTOR:</td><td colspan="3">None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case describes the process where the admin searches for a specific Category Type. Once the admin enters the search criteria the system will retrieve the matching results from the <u>CategoryType</u> entity the use case ends once the retrieved results have been displayed to the admin.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• Category types already exist on the system
• The admin is logged in to the system</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The admin wants to search for a category type</td></tr>
<tr><td rowspan="4">TYPICAL COURSE OF EVENTS:</td><td rowspan="2">ACTOR ACTION:</td><td colspan="2">SYSTEM RESPONSE:</td></tr>
<tr><td>Manual Action</td><td>Automated Action</td></tr>
<tr><td>Step 1: The admin requests to search for a category type</td><td></td><td></td></tr>
<tr><td>Step 2: The admin clicks on the “Products” screen.</td><td></td><td>Step 3: The system responds by loading the “Products” screen with the following controls:
<table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Products</td><td>Heading</td><td>None</td></tr></table></td></tr>
</table>

<table>
<tr><td></td><td></td><td></td><td>
<table>
<tr><td>Search Bar</td><td>Textbox</td><td>Search bar used to search for product categories</td></tr>
<tr><td>Category Type</td><td>Label</td><td>Name of the selected product screen.</td></tr>
<tr><td>Category Type Table</td><td>Table</td><td>This is to show the Category Types that are saved in the database in a table.</td></tr>
<tr><td>Category Type Name</td><td>Column</td><td>This is the column for the Category type names.</td></tr>
<tr><td>Category Type ID</td><td>Column</td><td>This is the column for the Category type IDs.</td></tr>
<tr><td>Actions Column</td><td>Column</td><td>None</td></tr>
<tr><td>Product Category/ Category Type/ Product Item</td><td>Buton Group</td><td>This is the button group whereby the admin can select which level of products he wants to work with</td></tr>
<tr><td>Options</td><td>Button</td><td>This is to show the update and delete button for each row.</td></tr>
<tr><td>Add</td><td>Button</td><td>None</td></tr>
<tr><td>Update</td><td>Button</td><td>None</td></tr>
<tr><td>Delete</td><td>Button</td><td>None</td></tr>
</table>
</td></tr>
<tr><td></td><td>Step 4: The admin selects the “Category Type” radio button.</td><td></td><td></td></tr>
<tr><td></td><td>Step 5: The admin enters the category type search criteria.</td><td></td><td></td></tr>
</table>

<table>
<tr><td></td><td>Step 6: The admin clicks on the search icon.</td><td></td><td>Step 7: The system captures and validates the information against the Category Type table
.
<table><tr><th>Input Name</th><th>Attribute Name</th><th>Validation Requirements</th></tr><tr><td>Name:</td><td>CategoryType _Desc</td><td>Maximum of 50 characters</td></tr><tr><td>ID:</td><td>CategoryType_ID</td><td>Characters must be integers</td></tr></table></td></tr>
<tr><td></td><td></td><td></td><td>Step 8: The system uses a read query to read the matching entries from the Category type entity. The system retrieves:
CategoryType:
CategoryType_ID
ProductCategory_ID
CategoryType_Desc</td></tr>
<tr><td></td><td></td><td></td><td>Step 9: The system displays the retrieved search results in the table.</td></tr>
<tr><td></td><td>Step 10: The admin selects the category type they were searching for.</td><td></td><td></td></tr>
<tr><td>ALTERNATE COURSES:</td><td colspan="3">Step 7b: The validation fails, and the admin is prompted to enter a new search query.</td></tr>
<tr><td>CONCLUSION:</td><td colspan="3">The use case concludes once the system displays the search results to the admin and the admin selects the searched category type</td></tr>
<tr><td>POST-CONDITION:</td><td colspan="3">The admin receives the results of the search query</td></tr>
<tr><td>BUSINESS RULES:</td><td colspan="3">1. None</td></tr>
<tr><td>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:</td><td colspan="3">None</td></tr>
<tr><td>ASSUMPTIONS:</td><td colspan="3">None</td></tr>
<tr><td>OPEN ISSUES:</td><td colspan="3">None</td></tr>
</table>

NKAP BOLTING SYSTEM	
DATE:	**05/06/2021**
VERSION:	**1.0**

<table>
<tr><td>AUTHOR(S)</td><td colspan="4">JEROME AMENIGY</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Update Category Type</td><td colspan="2">USE CASE TYPE</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">9.7</td><td colspan="2" rowspan="3">Business Requirements: ☐
System Analysis: ☐
System Design: ☑</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">Medium</td></tr>
<tr><td>SOURCE:</td><td colspan="2">NKAP Bolting System Requirements</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR:</td><td colspan="4">Admin</td></tr>
<tr><td>PRIMARY THE SYSTEM ACTOR:</td><td colspan="4">None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="4">None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="4">None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="4">This use case describes the event whereby an already existing Category Type is updated by an admin. Firstly the admin indicates that they want to update a category type, next the system invokes Use Case 9.6 Search category type, after the relevant information is displayed the admin chooses to edit the category type. After editing and saving the changes the system updates the <u>CategoryType</u> entity with the changes made.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="4">Category types already exist on the system
The admin is logged in to the system</td></tr>
<tr><td>TRIGGER:</td><td colspan="4">The admin wants to update the Category Type on the system.</td></tr>
<tr><td rowspan="4">TYPICAL COURSE OF EVENTS:</td><td rowspan="2">ACTOR ACTION:</td><td colspan="3">SYSTEM RESPONSE:</td></tr>
<tr><td>Manual Action</td><td colspan="2">Automated Action</td></tr>
<tr><td>Step 1: The admin requests to update a Category Type.</td><td></td><td colspan="2">Step 2: The system invokes UC 9.6 “Search Category Type”</td></tr>
<tr><td>Step 3: The admin clicks on the update button</td><td></td><td colspan="2">Step 4: The system responds by loading the “Update Category Type” modal with the following controls:
<table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr></table></td></tr>
</table>

<table>
<tr>
<td rowspan="4"></td>
<td></td>
<td></td>
<td>
<table>
<tr><td>Update Category Type</td><td>Heading</td><td>None</td></tr>
<tr><td>Category Type Name</td><td>Label</td><td>Label for the name of the category type</td></tr>
<tr><td>Name</td><td>Textbox</td><td>The textbox through which the name for the category type will be provided</td></tr>
<tr><td>Update</td><td>Button</td><td>None</td></tr>
<tr><td>Cancel</td><td>Button</td><td>None</td></tr>
</table>
</td>
</tr>
<tr>
<td>Step 5: The admin edits the category type accordingly</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Step 6: The admin clicks the “Save” button</td>
<td></td>
<td>Step 7: The system captures and validates the changes made to the category type entered by the admin against the category type table.
<table>
<tr><th>Input Name</th><th>Attribute in Table</th><th>Validation Requirements</th></tr>
<tr><td>Name</td><td>CategoryType_Desc</td><td>Maximum of 50 characters</td></tr>
</table>
</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Step 8: The system updates the CategoryType table with the changes made.
<table>
<tr><th>Information to be updated</th><th>Attribute in table</th><th>Details</th></tr>
<tr><td>CategoryType_ID</td><td>CategoryType_ID</td><td>None</td></tr>
<tr><td>Name</td><td>CategoryType_Desc</td><td>None</td></tr>
</table>
</td>
</tr>
<tr>
<td rowspan="3">ALTERNATE COURSES:</td>
<td colspan="3">Step 6b: The admin clicks on the “Cancel” button which terminates the use case</td>
</tr>
<tr>
<td colspan="3">Step 7b: The validation of the entered information fails; the system will outline the information that does not meet the validation requirements.</td>
</tr>
<tr>
<td colspan="3">Step 8b: The changes made to the category type could not be saved. The system will display a modal stating the update was unsuccessful</td>
</tr>
</table>

Input Name	Control Type	Notes
Error	Label	This serves as a header for the modal.
There was an error while saving the new category type information. Please try again later.	Label	This is to show that the saving of the category type information was unsuccessful.
OK	Button	None

CONCLUSION:	This use case concludes when the category type information has been successfully updated on the system.
POST-CONDITION:	• The category type is updated in the CategoryType table
BUSINESS RULES:	1. Only an admin can update Category Types in the system
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	None
ASSUMPTIONS:	None
OPEN ISSUES:	None

<table>
<tr><th colspan="3">NKAP BOLTING SYSTEM</th></tr>
<tr><td>DATE:</td><td colspan="2">05/06/2021</td></tr>
<tr><td>VERSION:</td><td colspan="2">1.0</td></tr>
<tr><td>AUTHOR(S)</td><td colspan="2">JEROME AMENIGY</td></tr>
<tr><td>USE CASE NAME:</td><td>Delete Category Type</td><td>USE CASE TYPE</td></tr>
<tr><td>USE CASE ID:</td><td>9.8</td><td rowspan="3">Business Requirements: ☐
System Analysis: ☐
System Design: ☑</td></tr>
<tr><td>PRIORITY:</td><td>Medium</td></tr>
<tr><td>SOURCE:</td><td>NKAP Bolting System Requirements</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR:</td><td colspan="2">Admin</td></tr>
<tr><td>PRIMARY THE SYSTEM ACTOR:</td><td colspan="2">None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="2">None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="2">None</td></tr>
</table>

<table>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case describes the process where the admin deletes a Category Type from the system. Upon indicating that the admin wants to delete a category type the system will invoke use case 9.6, the admin will then proceed to delete the selected category type and the system will remove the selected category type from the CategoryType entity.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">Category types already exist on the system
The admin is logged in to the system</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The admin wants to delete a category type</td></tr>
<tr><td rowspan="6">TYPICAL COURSE OF EVENTS:</td><td rowspan="2">ACTOR ACTION:</td><td colspan="2">SYSTEM RESPONSE:</td></tr>
<tr><td>Manual Action</td><td>Automated Action</td></tr>
<tr><td>Step 1: The admin requests to delete a category type</td><td></td><td>Step 2: The system invokes UC 9.6 “Search Category Type”</td></tr>
<tr><td>Step 3: The admin clicks on the delete button</td><td></td><td>Step 4: The system displays a modal, with the following controls:
<table>
<tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Confirm deletion</td><td>Heading</td><td>This serves as a header for the modal.</td></tr>
<tr><td>Are you sure you want to delete this category type?</td><td>Label</td><td>None</td></tr>
<tr><td>Yes</td><td>Button</td><td>This is to confirm that the admin wants to delete a Category Type.</td></tr>
<tr><td>No</td><td>Button</td><td>This is selected when the admin does not want to delete a category type.</td></tr>
</table></td></tr>
<tr><td>Step 5: The admin clicks on the “Yes”</td><td></td><td>Step 6: The system removes the selected category type from the category type entity.</td></tr>
</table>

<table>
<tr><td rowspan="2"></td><td>button to confirm the deletion of the category type.</td><td></td><td>
<table>
<tr><th>Information to be updated</th><th>Attribute in table</th><th>Details</th></tr>
<tr><td>CategoryType_ID</td><td>CategoryType_ID</td><td>None</td></tr>
<tr><td>ProductCategory_ID</td><td>Product_ID</td><td>None</td></tr>
<tr><td>Name</td><td>CategoryType_Desc</td><td>None</td></tr>
</table>
</td></tr>
<tr><td></td><td></td><td>Step 7: The system removes the category type from the table in the products screen.</td></tr>
<tr><td>ALTERNATE</td><td colspan="3">[Alt Step 5]: The admin clicks on the “No” button and the use case is terminated.</td></tr>
<tr><td>COURSES:</td><td colspan="3">[Alt Step 6]: The category type could not be deleted from the system. The system will display a modal informing the admin that the deletion of the category type was unsuccessful.
<table>
<tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Error</td><td>Label</td><td>This serves as a header for the modal.</td></tr>
<tr><td>There was an error while deleting the category type. Please try again later.</td><td>Label</td><td>This is to show that the deletion of the category type information was unsuccessful.</td></tr>
<tr><td>OK</td><td>Button</td><td>None</td></tr>
</table>
</td></tr>
<tr><td>CONCLUSION:</td><td colspan="3">The use case concludes once the selected category type has been successfully deleted from the system.</td></tr>
<tr><td>POST-CONDITION:</td><td colspan="3">• The category type has been removed from the CategoryType entity.</td></tr>
<tr><td>BUSINESS RULES:</td><td colspan="3">1. Only the admin can delete a category type.</td></tr>
<tr><td>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:</td><td colspan="3">None</td></tr>
<tr><td>ASSUMPTIONS:</td><td colspan="3">None</td></tr>
<tr><td>OPEN ISSUES:</td><td colspan="3">None</td></tr>
</table>

NKAP BOLTING SYSTEM	
DATE:	**05/06/2021**

<table>
<tr><td>VERSION:</td><td colspan="3">1.0</td></tr>
<tr><td>AUTHOR(S)</td><td colspan="3">JEROME AMENIGY</td></tr>
<tr><td>USE CASE NAME:</td><td>Add Product Item</td><td colspan="2">USE CASE TYPE</td></tr>
<tr><td>USE CASE ID:</td><td>9.9</td><td colspan="2" rowspan="3">Business Requirements: ☐
System Analysis: ☐
System Design: ☑</td></tr>
<tr><td>PRIORITY:</td><td>Medium</td></tr>
<tr><td>SOURCE:</td><td>NKAP Bolting System Requirements</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR:</td><td colspan="3">Admin</td></tr>
<tr><td>PRIMARY THE SYSTEM ACTOR:</td><td colspan="3">None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case describes the process whereby an admin adds a new Product Item to the system. The admin will select an option to add a product item to the system and then enter the specified product item's information and upon saving the addition the system will add it to the <u>ProductItem</u> Entity.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The admin is logged in to the system.
• The related Category Type already exists</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The admin wants to add a new product category to the system.</td></tr>
<tr><td rowspan="2">TYPICAL COURSE OF EVENTS:</td><td rowspan="2">ACTOR ACTION:</td><td colspan="2">SYSTEM RESPONSE:</td></tr>
<tr><td>Manual Action</td><td>Automated Action</td></tr>
<tr><td></td><td>Step 1: The admin requests to add a new product item on the system</td><td></td><td></td></tr>
<tr><td></td><td>Step 2: The admin selects Product Item radio button</td><td></td><td></td></tr>
<tr><td></td><td>Step 3: The admin clicks on</td><td></td><td>Step 4: The system retrieves the category types currently available on the system.</td></tr>
</table>

	the “Add” button in the Products Screen.		<table><tr><th>Information retrieved</th><th>Attribute in table</th><th>Details</th></tr><tr><td>**CategoryType ID**</td><td>CategoryType_ID</td><td>None</td></tr><tr><td>**ProductCategory ID**</td><td>ProductCategory ID</td><td>None</td></tr><tr><td>**Name**</td><td>CategoryType_Desc</td><td>None</td></tr></table>
			Step 5: The system then displays the Add Product Item modal with the following controls: <table><tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>**Add Product Item**</td><td>Heading</td><td>None</td></tr><tr><td>**Category Type:**</td><td>Dropdown</td><td>A dropdown containing all the existing category types</td></tr><tr><td>**Product Item Name:**</td><td>Label</td><td>Message telling the admin to enter the product item name</td></tr><tr><td>**Name**</td><td>Textbox</td><td>Field to enter the product item name</td></tr><tr><td>**Description**</td><td>Textbox</td><td>Field to enter the product item description</td></tr><tr><td>**Quantity**</td><td>Numeric up down</td><td>Numeric up down to select the product item quantity</td></tr><tr><td>**Image**</td><td>File input</td><td>File input used to select the product item image</td></tr></table>

			<table><tr><td>**Upload Button**</td><td>Button</td><td>Button used to upload the image</td></tr><tr><td>**Cost**</td><td>Textbox</td><td>Field to enter the product item cost</td></tr><tr><td>**Add**</td><td>Button</td><td>The button is used to save the category type onto the system</td></tr><tr><td>**Cancel**</td><td>Button</td><td>The button is used to terminate the process of this use case</td></tr></table>
	Step 6: The admin selects the Category Type that he wants to add the product item to.		
	Step 7: The admin enters the product item's details.		
	Step 8: The admin clicks on the add button		**Step 9:** The system captures the product items details and validates it against the ProductItem Table. <table><tr><th>Input Name</th><th>Attribute in Table</th><th>Validation requirements</th></tr><tr><td>**Name:**</td><td>ProductItem_Name</td><td>Maximum character length of 50 characters.</td></tr><tr><td>**Description**</td><td>ProductItem_Desc</td><td>Maximum character length of</td></tr></table>

		150 characters.
Quantity	Quantity_On_Hand	Entered values must be integers
Image	ProductItem_Image	File Uploaded must be an image
Cost	ProductItem_Cost	Entered values must be decimals

Step 10: The system uses an insert query to save the captured information in the ProductItem Table

Information to be saved	Attribute in table	Details
ProductIte m _ID	ProductIte m_ID	Generated from the system by reading the last ProductItem _ID and adding 1
Name	ProductIte m _Name	None
Description	ProductIte m_Desc	None
Image	ProductIte m_Image	None
Cost	ProductIte m_Cost	None
Quantity	Quantity_ on_Hand	None

ALTERNATE COURSES:	**Step 8b**: The user clicks on “cancel” which terminates the use case
	Step 9b: The provided product item’s details were not in the right format; the system will outline the place where the validation error/s occurred.
	Step 10b: The system cannot save the new product item; the system then displays a modal telling the admin that the addition was unsuccessful.

	<table><tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>**Error**</td><td>Label</td><td>This serves as a header for the modal.</td></tr><tr><td>**There was an error while saving the product item. Please try again later.**</td><td>Label</td><td>This is to show that the saving of the product item information type was unsuccessful.</td></tr><tr><td>**OK**</td><td>Button</td><td>None</td></tr></table>
CONCLUSION:	This use case concludes once the product item has been successfully added on to the system
POST-CONDITION:	• The product item has been added to the ProductItem entity.
BUSINESS RULES:	1. Only the admin can add product items to the system
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	None
ASSUMPTIONS:	None
OPEN ISSUES:	None

NKAP BOLTING SYSTEM			
DATE:	**05/06/2021**		
VERSION:	**1.0**		
AUTHOR(S)	**JEROME AMENIGY**		
USE CASE NAME:	**Search Product Item**	**USE CASE TYPE**	
USE CASE ID:	**9.10**	**Business Requirements:**	☐
PRIORITY:	Medium	**System Analysis:**	☐
SOURCE:	NKAP Bolting System Requirements	**System Design:**	☑
PRIMARY BUSINESS ACTOR:	Admin		
PRIMARY THE SYSTEM ACTOR:	None		
OTHER PARTICIPATING ACTORS:	None		

<table>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case describes the event where the admin searches for a specific Product Item. Once the admin enters the search criteria the system will retrieve the matching results from the ProductItem entity the use case ends once the retrieved results have been displayed to the admin.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">The admin is logged in to the system
The ProductItem exists on the system</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The admin wants to search for a Product Item</td></tr>
<tr><td rowspan="4">TYPICAL COURSE OF EVENTS:</td><td rowspan="2">ACTOR ACTION:</td><td colspan="2">SYSTEM RESPONSE:</td></tr>
<tr><td>Manual Action</td><td>Automated Action</td></tr>
<tr><td>Step 1: The admin requests to search for a product item</td><td></td><td></td></tr>
<tr><td>Step 2: The admin clicks on the “Products” screen.</td><td></td><td>Step 3: The system responds by loading the “Products” screen with the following controls:
<table>
<tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Products</td><td>Heading</td><td>None</td></tr>
<tr><td>Search Bar</td><td>Textbox</td><td>Search bar used to search for product categories</td></tr>
<tr><td>Product Item</td><td>Label</td><td>Name of the selected product screen.</td></tr>
<tr><td>Product Item Table</td><td>Table</td><td>This is to show the Product Items that are saved in the database in a table.</td></tr>
<tr><td>Product Item Name</td><td>Column</td><td>This is the column for the Product Item names.</td></tr>
</table></td></tr>
</table>

			Product Item ID	Column	This is the column for the Product Item IDs.
			Product Item Description	Column	This is the column for the Product Item Descriptions
			Product Item Image	Column	This is the column for the Product Item images
			Product Item Cost	Column	This is the column for the Product Item cost prices
			Product Item Quantity	Column	This is the column for the Product Item quantities available
			Actions Column	Column	None
			Product Category/ Category Type/ Product Item	Button group	This is the button group whereby the admin can select which level of products he wants to work with
			Options	Icon button	This is to show the update and delete button for each row.
			Add	Button	None
			Write-off	Button	None
			Stock Take	Button	None
			Update	Button	None
			Delete	Button	None
	Step 4: The admin selects the				

<table>
<tr><td></td><td>“Product Item” radio button.</td><td></td><td></td></tr>
<tr><td></td><td>Step 5: The admin enters the product item search criteria.</td><td></td><td></td></tr>
<tr><td></td><td>Step 6: The admin clicks on the search icon.</td><td></td><td>Step 7: The system captures and validates the information against the Product Item table
.
<table>
<tr><th>Input Name</th><th>Attribute Name</th><th>Validation Requirements</th></tr>
<tr><td>Name:</td><td>ProductItem_Name</td><td>Maximum of 50 characters</td></tr>
<tr><td>ID:</td><td>ProductItem_ID</td><td>Characters must be integers</td></tr>
</table></td></tr>
<tr><td></td><td></td><td></td><td>Step 8: The system uses a read query to read the matching entries from the Product Item entity. The system retrieves:
ProductItem:
ProductItem_ID
CategoryType_ID
ProductItem_Name
ProductItem_Desc
ProductItem_Image
ProductItem_Cost
Quantity_on_Hand
CategoryType_Desc</td></tr>
<tr><td></td><td></td><td></td><td>Step 9: The system displays the retrieved search results in the table.</td></tr>
<tr><td></td><td>Step 10: The admin selects the product item they were searching for.</td><td></td><td></td></tr>
</table>

ALTERNATE	**Step 7b:** The validation fails, and the admin is prompted to enter a new search query.
CONCLUSION:	The use case concludes once the system displays the search results to the admin and the admin selects the searched product item
POST-CONDITION:	• The admin receives the results of the search query
BUSINESS RULES:	1. None
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	None
ASSUMPTIONS:	None
OPEN ISSUES:	None

NKAP BOLTING SYSTEM		
DATE:	**05/06/2021**	
VERSION:	**1.0**	
AUTHOR(S)	**JEROME AMENIGY**	
USE CASE NAME:	**Update Product Item**	USE CASE TYPE
USE CASE ID:	**9.11**	**Business Requirements:** ☐
PRIORITY:	Medium	**System Analysis:** ☐
SOURCE:	NKAP Bolting System Requirements	**System Design:** ☑
PRIMARY BUSINESS ACTOR:	Admin	
PRIMARY THE SYSTEM ACTOR:	None	
OTHER PARTICIPATING ACTORS:	None	
OTHER INTERESTED STAKEHOLDERS:	None	
DESCRIPTION:	This use case describes the event whereby an already existing Product Item is updated by an admin. Firstly the admin indicates that they want to update a Product Item, next the system invokes Use Case 9.10 Search Product Item, after the relevant information is displayed the admin chooses to edit the Product Item. After editing and saving the changes the system updates the ProductItem entity with the changes made.	
PRE-CONDITION:	• The ProductItem already exists on the system	

<table>
<tr><td></td><td colspan="3">

- The admin is logged in to the system

</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The admin wants to update the Product Item on the system.</td></tr>
<tr><td rowspan="4">TYPICAL COURSE OF EVENTS:</td><td rowspan="2">ACTOR ACTION:</td><td colspan="2">SYSTEM RESPONSE:</td></tr>
<tr><td>Manual Action</td><td>Automated Action</td></tr>
<tr><td>Step 1: The admin requests to update a Product Item.</td><td></td><td>Step 2: The system invokes UC 9.10 “Search Product Item”</td></tr>
<tr><td>Step 3: The admin clicks on the update button</td><td></td><td>Step 4: The system responds by loading the “Update Product Item” modal with the following controls:
<table>
<tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Update Product Item</td><td>Heading</td><td>None</td></tr>
<tr><td>Product Item Name</td><td>Label</td><td>Label for the name of the product item</td></tr>
<tr><td>Name</td><td>Textbox</td><td>The textbox through which the name for the product item will be provided</td></tr>
<tr><td>Description</td><td>Textbox</td><td>Textbox for which the changes to the description of a product item will be provided</td></tr>
<tr><td>Quantity</td><td>Textbox</td><td>Textbox for which the changes to the quantity of a product item will be provided</td></tr>
<tr><td>Cost</td><td>Textbox</td><td>Textbox for which the changes to the cost of a product item will be provided</td></tr>
<tr><td>Image</td><td>File upload</td><td>Control used to upload the product item images</td></tr>
</table>
</td></tr>
</table>

Upload Button	Button	Button used to upload the image
Update	Button	None
Cancel	Button	None

Step 5: The admin edits the product item

Step 6: The admin clicks the “Update” button

Step 7: The system captures and validates the changes made to the product item entered by the admin against the ProductItem table.

Input Name	Attribute in Table	Validation Requirements
Name	ProductItem_Name	Maximum of 50 characters
Description	ProductItem_Desc	Maximum of 150 characters
Image	ProductItem_Image	File Uploaded must be an image
Cost	ProductItem_Cost	Entered values must be integers
Quantity	Quantity_on_Hand	Entered values must be decimals

Step 8: The system updates the CategoryType table with the changes made.

Information to be updated	Attribute in table	Details
ProductItem _ID	ProductItem _ID	None
Name	ProductItem_Name	None
Description	ProductItem_Desc	None
Image	ProductItem_Image	None
Cost	ProductItem_Cost	None
Quantity	Quantity_on_Hand	None

ALTERNATE COURSES:	**Step 6b:** The admin clicks on the "Cancel" button which terminates the use case
	Step 7b: The validation of the entered information fails; the system will outline the information that does not meet the validation requirements.
	Step 8b: The changes made to the product item could not be saved. The system will display a modal stating the update was unsuccessful

Input Name	Control Type	Notes
Error	Label	This serves as a header for the modal.
There was an error while saving the new product item information. Please try again later.	Label	This is to show that the saving of the product item information was unsuccessful.
OK	Button	None

CONCLUSION:	This use case concludes when the category type information has been successfully updated on the system.
POST-CONDITION:	• The product item is updated in the ProductItem table
BUSINESS RULES:	1. Only an admin can update Product Items in the system
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	None
ASSUMPTIONS:	None
OPEN ISSUES:	None

NKAP BOLTING SYSTEM		
DATE:	**05/06/2021**	
VERSION:	**1.0**	
AUTHOR(S)	**JEROME AMENIGY**	
USE CASE NAME:	**Delete Product Item**	**USE CASE TYPE**
USE CASE ID:	**9.12**	**Business Requirements:** ☐
PRIORITY:	Medium	**System Analysis:** ☐
SOURCE:	NKAP Bolting System Requirements	**System Design:** ☑
PRIMARY BUSINESS ACTOR:	Admin	

<table>
<tr><td>PRIMARY THE SYSTEM ACTOR:</td><td colspan="3">None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case describes the process where the admin deletes a Product Item from the system. Upon indicating that the admin wants to delete a product item the system will invoke use case 9.10, the admin will then proceed to delete the selected product item and the system will remove the selected product item from the ProductItem entity.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">The admin is logged in
The product item already exists</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The admin wants to delete a product</td></tr>
<tr><td rowspan="4">TYPICAL COURSE OF EVENTS:</td><td rowspan="2">ACTOR ACTION:</td><td colspan="2">SYSTEM RESPONSE:</td></tr>
<tr><td>Manual Action</td><td>Automated Action</td></tr>
<tr><td>Step 1: The admin requests to delete a product item</td><td></td><td>Step 2: The system invokes use case 9.10 “Search Product Item”</td></tr>
<tr><td>Step 3: The admin clicks on the delete button</td><td></td><td>Step 4: The system displays a modal, with the following controls:
<table>
<tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Confirm deletion</td><td>Heading</td><td>This serves as a header for the modal.</td></tr>
<tr><td>Are you sure you want to delete this product item</td><td>Label</td><td>None</td></tr>
<tr><td>Yes</td><td>Button</td><td>This is to confirm that the admin wants to</td></tr>
</table></td></tr>
</table>

<table>
<tr><td></td><td></td><td></td><td><table><tr><td></td><td></td><td>delete a product item.</td></tr><tr><td>No</td><td>Button</td><td>This is selected when the admin does not want to delete a product item.</td></tr></table></td></tr>
<tr><td></td><td>Step 5: The admin clicks on the “Yes” button to confirm the deletion of the product item.</td><td></td><td>Step 6: The system removes the selected category type from the category type entity.
<table><tr><th>Information to be removed</th><th>Attribute in table</th><th>Details</th></tr><tr><td>ProductItem_ID</td><td>ProductItem_ID</td><td>None</td></tr><tr><td>CategoryType _ID</td><td>CategoryType_ID</td><td>None</td></tr><tr><td>Name</td><td>ProductItem _Name</td><td>None</td></tr><tr><td>Description</td><td>ProductItem_Desc</td><td>None</td></tr><tr><td>Image</td><td>ProductItem_Image</td><td>None</td></tr><tr><td>Cost</td><td>ProductItem_Cost</td><td>None</td></tr><tr><td>Quantity</td><td>Quantity_on_Hand</td><td>None</td></tr><tr><td>Supplier Order Line</td><td>SupplierOrderLine_ID</td><td>none</td></tr></table></td></tr>
<tr><td></td><td></td><td></td><td>Step 7: The system removes the product item from the table in the products screen.</td></tr>
<tr><td rowspan="2">ALTERNATE

COURSES:</td><td colspan="3">[Alt Step 5]: The admin clicks on the “No” button and the use case is terminated.</td></tr>
<tr><td colspan="3">[Alt Step 6]: product item could not be deleted from the system. The system will display a modal informing the admin that the deletion of the product item was unsuccessful.
<table><tr><th>Input Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Error</td><td>Label</td><td>This serves as a header for the modal.</td></tr><tr><td>There was an error while deleting the product item. Please try again later.</td><td>Label</td><td>This is to show that the deletion of the product item information was unsuccessful.</td></tr><tr><td>OK</td><td>Button</td><td>None</td></tr></table></td></tr>
</table>

CONCLUSION:	The use case concludes once the selected product item has been successfully deleted from the system.
POST-CONDITION:	• The category type has been removed from the ProductItem entity.
BUSINESS RULES:	1. Only the admin can delete a product item.
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	None
ASSUMPTIONS:	None
OPEN ISSUES:	None

2.1.10. SPECIALS SUBSYSTEM

<table>
<tr><th colspan="5">NKAP BOLTING SYSTEM</th></tr>
<tr><td>DATE:</td><td colspan="4">04-06-2021</td></tr>
<tr><td>VERSION:</td><td colspan="4">1.0</td></tr>
<tr><td>AUTHOR(S)</td><td colspan="4">S’nethemba Xulu</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Add Special</td><td colspan="2">USE CASE TYPE</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">10.1</td><td colspan="2">Business Requirements: ☐</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">High</td><td colspan="2">System Analysis: ☐</td></tr>
<tr><td>SOURCE:</td><td colspan="2">NKAP Bolting System Requirements</td><td colspan="2">System Design: ☑</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR:</td><td colspan="4">None</td></tr>
<tr><td>PRIMARY THE SYSTEM ACTOR:</td><td colspan="4">None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="4">None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="4">None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="4">This use case describes the event where the admin wants to add a new special on to the system. The use case begins when the admin requests to add the special. The admin will enter all the special details and the information will be captured in the Special table. The use case ends when the special has been successfully added.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="4">• The admin should be on the Specials screen</td></tr>
<tr><td>TRIGGER:</td><td colspan="4">The admin requesting to add a new special.</td></tr>
<tr><td rowspan="4">TYPICAL COURSE OF EVENTS:</td><td rowspan="2">ACTOR ACTION:</td><td colspan="3">SYSTEM RESPONSE:</td></tr>
<tr><td>Manual Action</td><td colspan="2">Automated Action</td></tr>
<tr><td>Step 1: The admin requests to add a new special.</td><td></td><td colspan="2"></td></tr>
<tr><td>Step 2: The admin will click on the “Add Special” button in the Special Screen.</td><td></td><td colspan="2">Step 3: The system responds by loading the “Add Special” modal with the following controls:
<table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Add Special</td><td>Heading</td><td>None</td></tr></table></td></tr>
</table>

<table>
<tr>
<td></td>
<td></td>
<td></td>
<td>
<table>
<tr><td>Special Image</td><td>Label</td><td>None</td></tr>
<tr><td>Special Image</td><td>Image</td><td>None</td></tr>
<tr><td>Special Description:</td><td>Label</td><td>This is the label for the description of the special</td></tr>
<tr><td>Description</td><td>Textbox</td><td>None</td></tr>
<tr><td>Special Price:</td><td>Label</td><td>This is the label for the price of the special</td></tr>
<tr><td>Price</td><td>Textbox</td><td>None</td></tr>
<tr><td>Start Date:</td><td>Label</td><td>This is the label for the start date of the special</td></tr>
<tr><td>Date</td><td>Calendar</td><td>None</td></tr>
<tr><td>End Date:</td><td>Label</td><td>This is the label for the end date of the special</td></tr>
<tr><td>Date</td><td>Calendar</td><td>None</td></tr>
<tr><td>Add</td><td>Button</td><td>None</td></tr>
<tr><td>Cancel</td><td>Button</td><td>This button is used when the admin decides to terminate the process of adding a new special.</td></tr>
</table>
</td>
</tr>
<tr>
<td></td>
<td>Step 4: The admin enters the details in the respective places:
• Description
• Special price
• Start Date
• End Date
• Image</td>
<td></td>
<td></td>
</tr>
</table>

<table>
<tr><td></td><td>Step 5: The admin clicks on the “Add” button [Alt]</td><td></td><td>Step 6: The system captures and validates the information entered by the admin against the Special table [Alt]:
<table>
<tr><th>Input Name</th><th>Attribute in table</th><th>Validation requirements</th></tr>
<tr><td>Image</td><td>Special_Image</td><td>Required</td></tr>
<tr><td>Price</td><td>Special_Price</td><td>Required, maximum of 10 characters</td></tr>
<tr><td>Description</td><td>Special_Description</td><td>Required, maximum of 200 characters.</td></tr>
<tr><td>Start Date</td><td>Special_StartDate</td><td>Required</td></tr>
<tr><td>End Date</td><td>Special_EndDate</td><td>Required</td></tr>
</table></td></tr>
<tr><td></td><td></td><td></td><td>Step 7: The system uses a SQL insert query to save the captured information in the Special table [Alt]:
<table>
<tr><th>Information to be saved:</th><th>Attribute in table</th><th>Details</th></tr>
<tr><td>Special_ID</td><td>Special_ID</td><td>Generated by the system by reading the last Special_ID from the Special table and adding it by one.</td></tr>
<tr><td>Special_ Price</td><td>Special_Price</td><td>None</td></tr>
<tr><td>Special_ Image</td><td>Special_Image</td><td>None</td></tr>
<tr><td>Special_ Description</td><td>Special_ Description</td><td>None</td></tr>
<tr><td>Special_ StartDate</td><td>Special_ StartDate</td><td>None</td></tr>
<tr><td>Special_ EndDate</td><td>Special_ EndDate</td><td>None</td></tr>
</table></td></tr>
</table>

ALTERNATE COURSES:	**[Alt Step 5]**: The admin clicks on the "Cancel" button which terminates the this use case. Return to step 4
	[Alt Step 6]: The details entered were not in the right format, so the validation failed. The system will display to the admin where the validation errors occurred by showing a red outline where the validation failed.
	[Alt Step 7]: The new special could not be added to the system. The system will display a modal informing the admin that the saving of the special information was unsuccessful.

Input Name	Control Type	Notes
Error	Label	This serves as a header for the modal.
There was an error while saving the new special. Please try again later.	Label	This is to show that the saving of the special information was unsuccessful.
OK	Button	None

CONCLUSION:	This use case concludes when the special has been successfully added on the system.
POST-CONDITION:	• The special has been added to the **Special** table.
BUSINESS RULES:	Only an admin can add a new special to the system
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	None
ASSUMPTIONS:	None
OPEN ISSUES:	None

NKAP BOLTING SYSTEM			
DATE:	04-06-2021		
VERSION:	**1.0**		
AUTHOR(S)	S'nethemba Xulu		
USE CASE NAME:	**Search Special**	USE CASE TYPE	
USE CASE ID:	**10.2**	**Business Requirements:**	☐
PRIORITY:	High	**System Analysis:**	☐
SOURCE:	NKAP Bolting System Requirements	**System Design:**	☑
PRIMARY BUSINESS ACTOR:	Admin		
PRIMARY THE SYSTEM ACTOR:	None		
OTHER PARTICIPATING ACTORS:	None		
OTHER INTERESTED STAKEHOLDERS:	None		
DESCRIPTION:	This use case describes the event where the admin wishes to search for a special. The admin will enter the search parameters like the suppler name. The system will search for the special based on the parameters the admin entered. This use case concludes when the special is shown the results of the search query.		
PRE-CONDITION:	• The admin should be logged on to the system. • The special should be already exist in the system.		
TRIGGER:	The admin wishes to search for a special.		

TYPICAL COURSE OF EVENTS:	ACTOR ACTION:	SYSTEM RESPONSE: Manual Action	SYSTEM RESPONSE: Automated Action
	Step 1: The admin would like to search for a special.		
	Step 2: The admin clicks on the "**Special**" screen.		**Step 3**: The system responds by loading the "Special" screen with the following controls:

Control Name	Control Type	Notes
Special	Heading	None
Search:	Label	This is the label to prompt the

		user to enter a search query.
Search	Textbox	None
Search icon	Icon button	None
Special Table	Table	This is to show the Specials that are saved in the database in a table.
Special Image	Column	This is the column for the Special image
Special Price	Column	This is the column for the special price
Special description	Column	This is the column for the description of the special
Options	Icon	This is to show the update and delete button for each row.
Update	Button	None
Delete	Button	None

	Step 4: The admin inputs the search criteria and clicks on the search icon.		**Step 5:** The system captures and validates the information against the **Special** table **[Alt]**:
			Step 6: A SQL read query is used to read the following attributes from the

<table>
<tr><td rowspan="2"></td><td></td><td></td><td>Special table based on the search criteria:

Special table:
Special_ID(PK)
Special_Image
Special_Price
Special_Description
Special_EndDate
Special_StartDate</td></tr>
<tr><td></td><td></td><td>Step 7: The system will show the search results received from step 6 in a table. [Alt]</td></tr>
<tr><td>ALTERNATE COURSE:</td><td colspan="3">[Alt Step 5]: The validation failed. The admin is prompted to enter a new search query.</td></tr>
<tr><td>CONCLUSION:</td><td colspan="3">The use case concludes when the admin receives the results of the searched criteria.</td></tr>
<tr><td>POST-CONDITION:</td><td colspan="3">The admin receives the results of the specials with the same searched criteria</td></tr>
<tr><td>BUSINESS RULES:</td><td colspan="3">1. None</td></tr>
<tr><td>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:</td><td colspan="3">None</td></tr>
<tr><td>ASSUMPTIONS:</td><td colspan="3">None</td></tr>
<tr><td>OPEN ISSUES:</td><td colspan="3">None</td></tr>
</table>

NKAP BOLTING SYSTEM			
DATE:	04-06-2021		
VERSION:	**1.0**		
AUTHOR(S)	S'nethemba Xulu		
USE CASE NAME:	**Update Special**	**USE CASE TYPE**	
USE CASE ID:	**10**	**Business Requirements:**	☐
PRIORITY:	3	**System Analysis:**	☐
SOURCE:	NKAP Bolting System Requirements	**System Design:**	☑
PRIMARY BUSINESS ACTOR:	Admin		
PRIMARY THE SYSTEM ACTOR:	None		
OTHER PARTICIPATING ACTORS:	None		
OTHER INTERESTED STAKEHOLDERS:	None		
DESCRIPTION:	This use case describes the event in which an admin wants to update the details of a special. The system will display the previous information that was added before and then the admin will add the new information. This use case ends when the information has been updated successfully.		
PRE-CONDITION:	• The admin should be logged on to the system.		
TRIGGER:	The admin wishes update the details of the special.		

TYPICAL COURSE OF EVENTS:

Actor Action:	Actor Action: Manual Action	Actor Action: Automated Action
Step 1: The admin requests to update the details of a special.		**Step 2**: The system invokes use case **10.2 Search Special.**
Step 3: The admin clicks on the "Update" button.		**Step 4**: The system responds by loading the "Update Special" modal with the following controls:

Control Name	Control Type	Notes
Update Special	Heading	None

<table>
<tr>
<td></td>
<td></td>
<td></td>
<td>
<table>
<tr><td>Special Image</td><td>Label</td><td>None</td></tr>
<tr><td>Special Image</td><td>Image</td><td>None</td></tr>
<tr><td>Special Description:</td><td>Label</td><td>This is the label for the description of the special</td></tr>
<tr><td>Description</td><td>Textbox</td><td>None</td></tr>
<tr><td>Special Price:</td><td>Label</td><td>This is the label for the price of the special</td></tr>
<tr><td>Price</td><td>Textbox</td><td>None</td></tr>
<tr><td>Start Date:</td><td>Label</td><td>This is the label for the start date of the special</td></tr>
<tr><td>Date</td><td>Calendar</td><td>None</td></tr>
<tr><td>End Date:</td><td>Label</td><td>This is the label for the end date of the special</td></tr>
<tr><td>Date</td><td>Calendar</td><td>None</td></tr>
<tr><td>Update</td><td>Button</td><td>None</td></tr>
<tr><td>Cancel</td><td>Button</td><td>This button is used when the admin decides to terminate the process of adding a new special.</td></tr>
</table>
</td>
</tr>
<tr>
<td></td>
<td>Step 5: The admin enters the new information and clicks on the “Update” button. [Alt]</td>
<td></td>
<td>
Step 6: The system captures and validates the information entered by the admin against the Special table [Alt]:
<table>
<tr><th>Input Name</th><th>Attribute in table</th><th>Validation requirements</th></tr>
<tr><td>Image</td><td>Special_Image</td><td>Required</td></tr>
<tr><td>Price</td><td>Special_Price</td><td>Required, maximum of 10 characters</td></tr>
</table>
</td>
</tr>
</table>

Description	Special_Description	Required, maximum of 200 characters.
Start Date	Special_StartDate	Required
End Date	Special_EndDate	Required

Step 7: The system uses a SQL insert query to save the captured information in the **Special** table **[Alt]**:

Information to be updated:	Attribute in table	Details
Special_ Price	Special_Price	None
Special_ Image	Special_Image	None
Special_ Description	Special_ Description	None
Special_ StartDate	Special_ StartDate	None
Special_ EndDate	Special_ EndDate	None

ALTERNATE COURSES:

[Alt Step 5]: The admin clicks on the "**Cancel**" button which terminates the this use case.

Return to **step 1.**

[Alt Step 6]: The details entered were not in the right format, so the validation failed. The system will display to the admin where the validation errors occurred by showing a red outline where the validation failed.

[Alt Step 7]: The updated special information could not be added to the system. The system will display a modal informing the admin that the saving of the special information was unsuccessful.

Input Name	Control Type	Notes
Error	Label	This serves as a header for the modal.
There was an error while saving the new special information. Please try again later.	Label	This is to show that the saving of the special information was unsuccessful.

	OK	Button	None
CONCLUSION:	This use case concludes when the special information has been successfully updated on the system.		
POST-CONDITION:	• The special has been updated in the **Special** table.		
BUSINESS RULES:	**1.** Only an admin can update the information of the special in the system.		
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	None		
ASSUMPTIONS:	None		
OPEN ISSUES:	None		

NKAP BOLTING SYSTEM			
DATE:	04-06-2021		
VERSION:	**1.0**		
AUTHOR(S)	S'nethemba Xulu		
USE CASE NAME:	**Delete Special**	**USE CASE TYPE**	
USE CASE ID:	**10.4**	**Business Requirements:**	☐
PRIORITY:	High	**System Analysis:**	☐
SOURCE:	NKAP Bolting System Requirements	**System Design:**	☑
PRIMARY BUSINESS ACTOR:	Admin		
PRIMARY THE SYSTEM ACTOR:	None		
OTHER PARTICIPATING ACTORS:	None		

<table>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case describes the process where the admin wants to delete the special from the system. This use case begins when the Administrator requests to delete a special. This use case concludes when the admin gets notified that the special has been successfully removed from the system.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">The admin should be logged in to the system.
The special should already exist on the systems database.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The admin requesting to delete a special.</td></tr>
<tr><td rowspan="4">TYPICAL COURSE OF EVENTS:</td><td rowspan="2">ACTOR ACTION:</td><td colspan="2">SYSTEM RESPONSE:</td></tr>
<tr><td>Manual Action</td><td>Automated Action</td></tr>
<tr><td>Step 1: The admin request to delete a special from the system.</td><td></td><td>Step 2: The system invokes Use case 10.2 Search Special.</td></tr>
<tr><td>Step 3: The admin clicks on the “Delete” button on the Special screen.</td><td></td><td>Step 4: The system displays a modal, with the following controls:
<table>
<tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Confirm deletion</td><td>Heading</td><td>This serves as a header for the modal.</td></tr>
<tr><td>Are you sure you want to delete this special?</td><td>Label</td><td>None</td></tr>
<tr><td>Yes</td><td>Button</td><td>This is to confirm that the admin wants to delete a special.</td></tr>
<tr><td>No</td><td>Button</td><td>This is selected when the admin does not want to delete a special.</td></tr>
</table></td></tr>
</table>

<table>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td>Step 5: The admin clicks on the “Yes” button to confirm the deletion of the special. [Alt]</td><td></td><td>Step 6: The system uses a SQL delete query to remove the special information from the Special table [Alt]:

Information to be deleted:	Attribute in table	Details
Special_ Price	Special_Price	None
Special_ Image	Special_Image	None
Special_ Description	Special_ Description	None
Special_ StartDate	Special_ StartDate	None
Special_ EndDate	Special_ EndDate	None

</td></tr>
<tr><td></td><td></td><td></td><td>Step 7: The system removes the deleted special from the table in the Special screen.</td></tr>
<tr><td>ALTERNATE COURSES:</td><td colspan="3">[Alt Step 5]: The admin clicks on the “No” button.
Terminate this use case.</td></tr>
<tr><td></td><td colspan="3">[Alt Step 6]: The special could not be deleted from the system. The system will display a modal informing the admin that the deletion of the special was unsuccessful.

Input Name	Control Type	Notes
Error	Label	This serves as a header for the modal.
There was an error while deleting the special. Please try again later.	Label	This is to show that the saving of the special information was unsuccessful.
OK	Button	None

</td></tr>
<tr><td>CONCLUSION:</td><td colspan="3">This use case ends when the special has been successfully deleted from the system.</td></tr>
<tr><td>POST-CONDITION:</td><td colspan="3">• The details of the special have been removed from the Special table.</td></tr>
</table>

BUSINESS RULES:	**1.** Only the admin can remove a special.
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	None
ASSUMPTIONS:	None
OPEN ISSUES:	None

NKAP BOLTING SYSTEM		
DATE:	04-06-2021	
VERSION:	**1.0**	
AUTHOR(S)	S'nethemba Xulu	
USE CASE NAME:	**Send Promotional Emails**	**USE CASE TYPE**
USE CASE ID:	**10.5**	**Business Requirements:** ☐
PRIORITY:	High	**System Analysis:** ☐
SOURCE:	NKAP Bolting System Requirements	**System Design:** ☑
PRIMARY BUSINESS ACTOR:	Admin	
PRIMARY THE SYSTEM ACTOR:	None	

<table>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case describes the event where the person wants to Send Promotional emails. The use case starts with the admin requesting to The use case starts with the admin requesting to post an announcement, the system then prompts the admin to select the specials that they want to promote from the existing specials. The admin sends the email to all the customers emails.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">The admin is already on the Supplier screen</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The admin wants to send promotional email.</td></tr>
<tr><td rowspan="5">TYPICAL COURSE OF EVENTS:</td><td rowspan="2">Actor Action:</td><td colspan="2">system Response:</td></tr>
<tr><td>Manual Action</td><td>Automated Action</td></tr>
<tr><td>Step 1: The admin requests to send promotional email.</td><td></td><td></td></tr>
<tr><td>Step 2: The admin clicks on the “Send Promotions” button.</td><td></td><td>Step 3: The system shows the “Send Promotions” screen with the following controls:
<table>
<tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Special</td><td>Heading</td><td>None</td></tr>
<tr><td>Send Promotion</td><td>Label</td><td>None</td></tr>
<tr><td>Special Table</td><td>Table</td><td>This is to show the Specials that are saved in the database in a table.</td></tr>
<tr><td>Select</td><td>Colum</td><td>This is to tick whether you want to include the special in the email.</td></tr>
<tr><td>Special Image</td><td>Column</td><td>This is the column for the</td></tr>
</table></td></tr>
</table>

<table>
<tr><td rowspan="4"></td><td></td><td></td><td><table>
<tr><td></td><td></td><td>Special image</td></tr>
<tr><td>Special Price</td><td>Column</td><td>This is the column for the special price</td></tr>
<tr><td>Special description</td><td>Column</td><td>This is the column for the description of the special</td></tr>
<tr><td>Send Email</td><td>Button</td><td>None</td></tr>
<tr><td>Cancel</td><td>Button</td><td>None</td></tr>
</table></td></tr>
<tr><td>Step 4: The admin selects the specials that they want in their email.</td><td></td><td>Step 5: The system reads all the specials in the SPECIAL table.
• Special_ID
• Special_Price
• Special_Description
• Special_StartDate
• Special_EndDate</td></tr>
<tr><td>Step 6: The admin clicks on the send email. [Alt]</td><td></td><td>Step 7: The system compiles the email and sends it to the customers.</td></tr>
<tr><td></td><td></td><td>Step 8: The system gets the details from The CUSTOMER table:
• Customer_ID
• Customer_Email</td></tr>
<tr><td rowspan="2">ALTERNATE COURSES:</td><td colspan="3">[Alt-Step 6]: The admin cancels. The use case terminates.</td></tr>
<tr><td colspan="3"></td></tr>
<tr><td>CONCLUSION:</td><td colspan="3">The promotional emails have been sent to the customers.</td></tr>
<tr><td>POST-CONDITION:</td><td colspan="3">The emails are sent to the customers.</td></tr>
<tr><td>BUSINESS RULES:</td><td colspan="3">None</td></tr>
<tr><td>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:</td><td colspan="3">None</td></tr>
<tr><td>ASSUMPTIONS:</td><td colspan="3">None</td></tr>
<tr><td>OPEN ISSUES:</td><td colspan="3">None</td></tr>
</table>

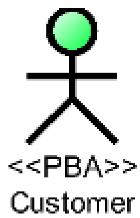
CONCLUSION

This section concludes. It contained the use cases in the form of a full set of use case diagrams as well as Design Use Case Documentation of each Use Case.

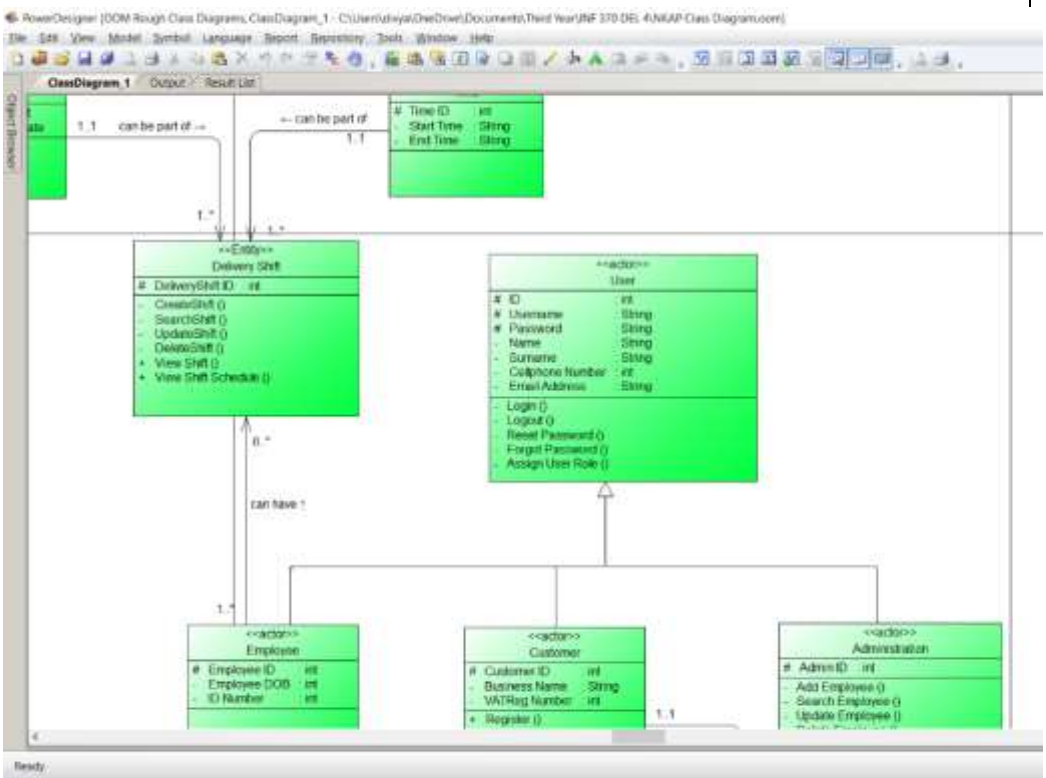
2. Process/Object-Oriented Design

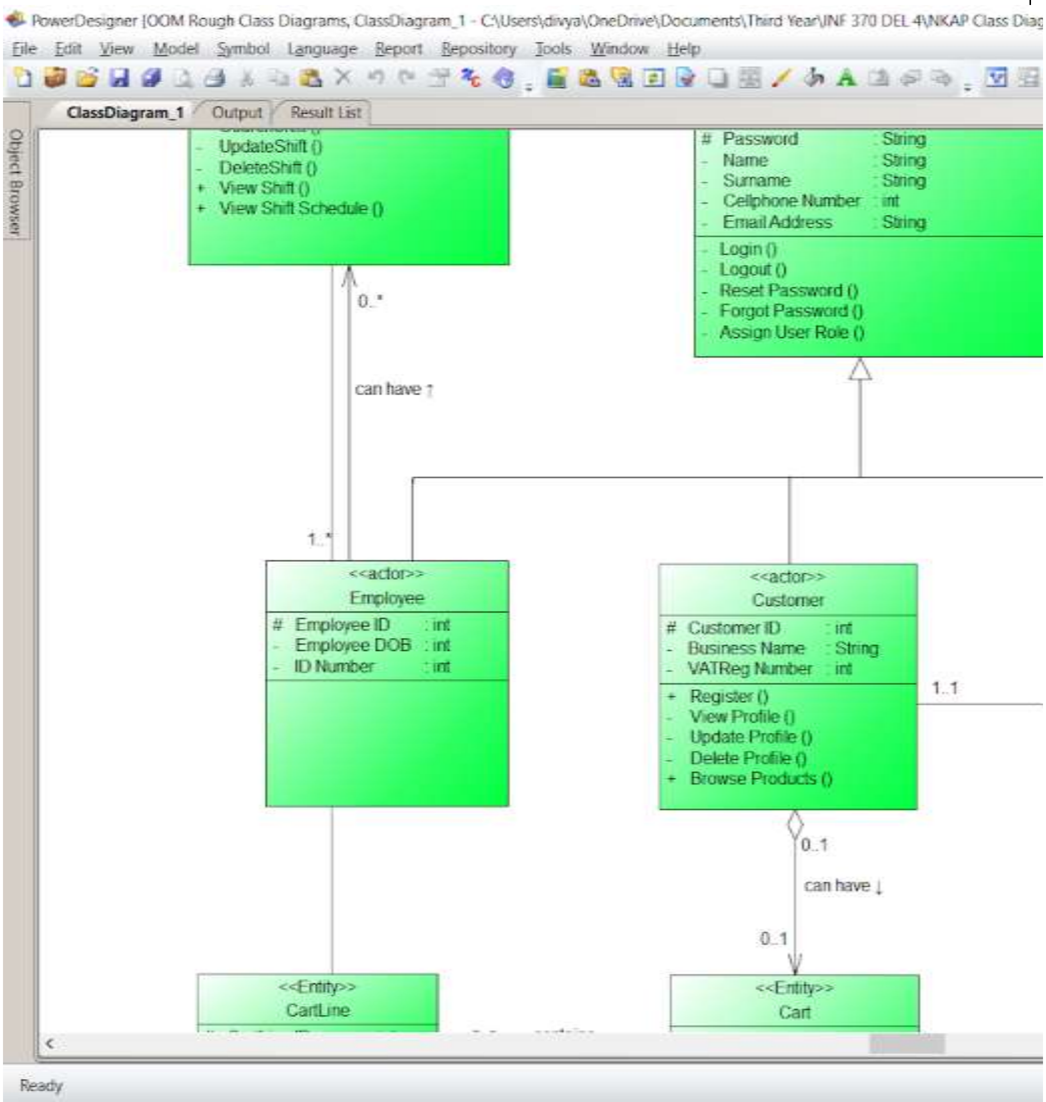
INTRODUCTION

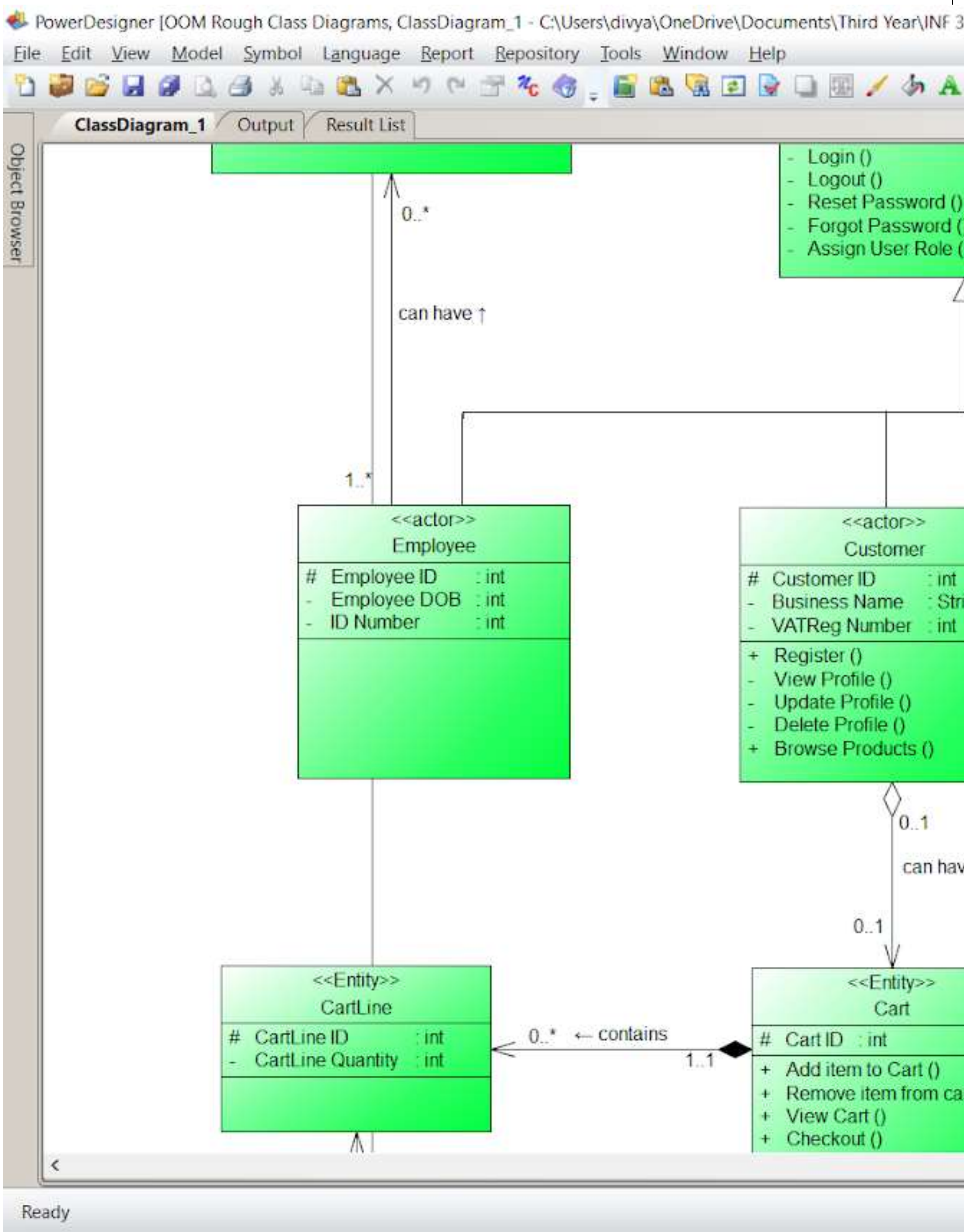
This section contains the Object-Oriented Design for the NKAP Bolting System. This includes a class diagram for the entire system, state diagrams for each object in the class diagram as well as sequence diagrams for each Use Case and Pseudo Code.

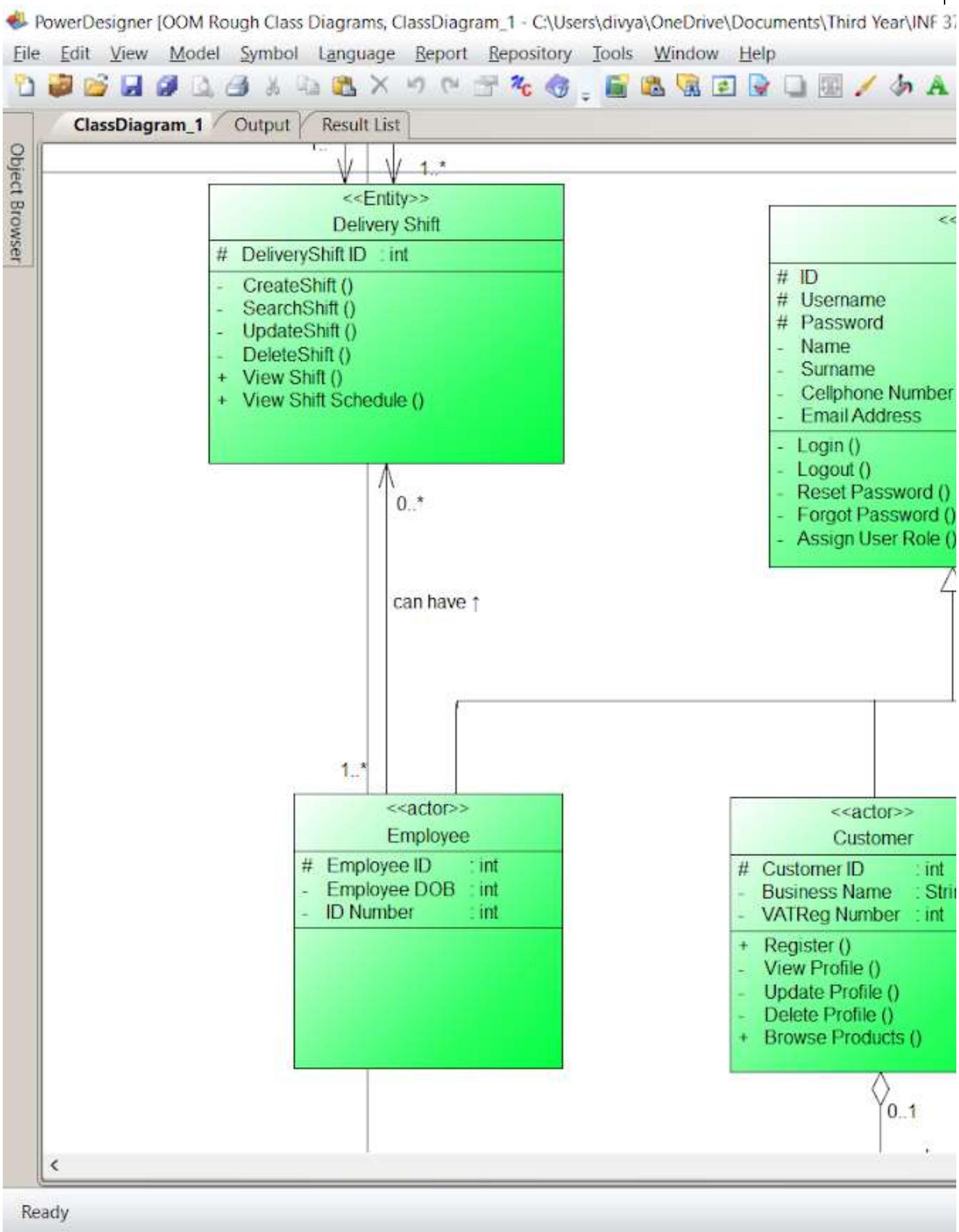
Use Case Diagram Key		
Element	**Description**	**Example**
System	Outer Rectangle Element with the system name, 0. NKAP Bolting system and black outline.	
Sub-System	Inner rectangle element with black outline and Sub-System name.	
Use Case	Use Case element with black outline and green fill.	
Actor	Use Case actor element with green fill and name of actor and actor type below.	

Association	Use Case relationship line with ‘initiates’ tag.	
Uses Association	Use case dependency relationship with ‘uses’ tag.	

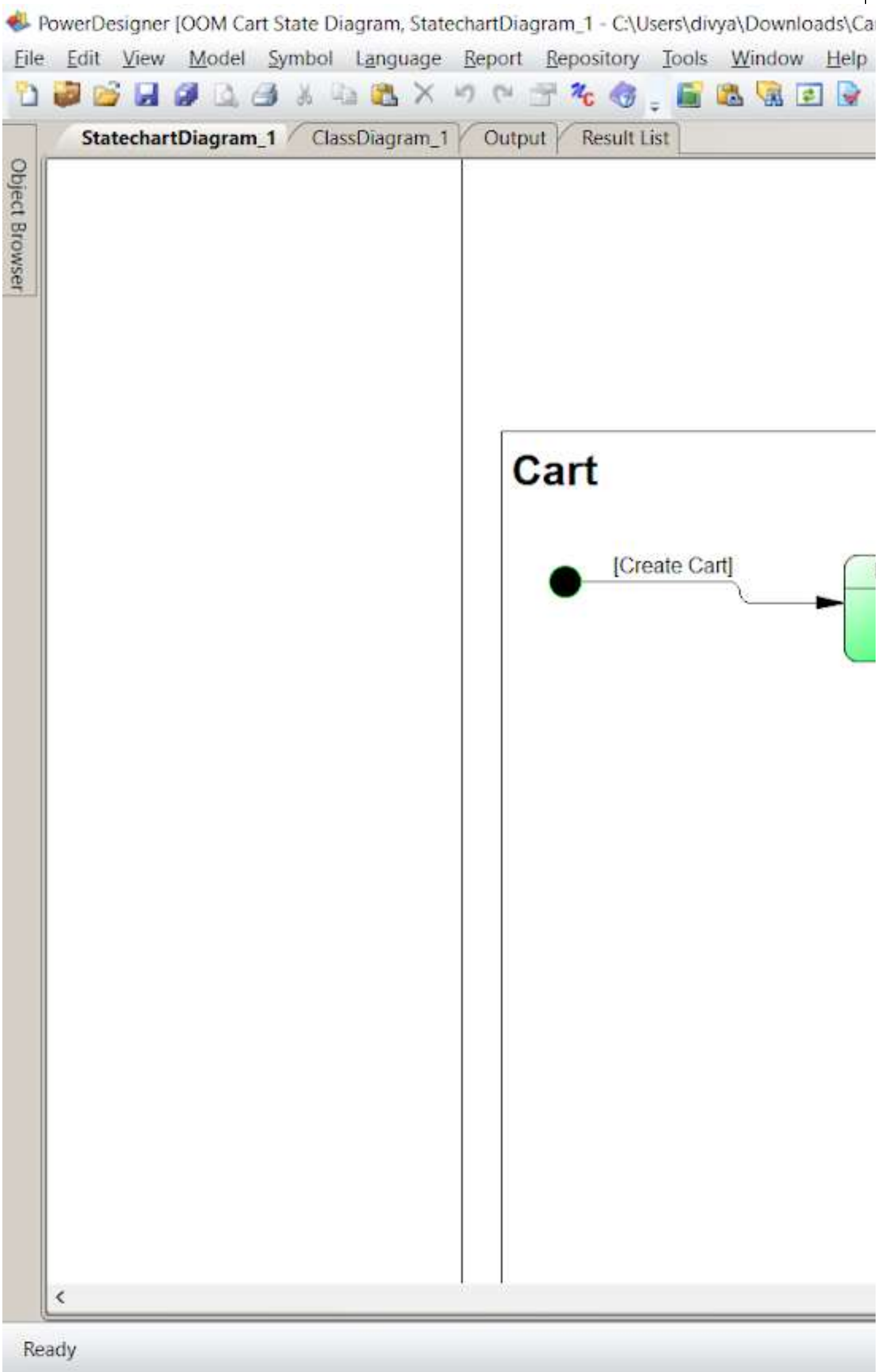
Class Diagram Key		
Element	**Description**	**Example**
Class	Family of objects that share common attributes and behaviour. Shown as green solid gradient square.	

Aggregation	Shows no dependency between Classes. Shown as an association decorated with a clear diamond	

Composition	Strong ownership between the “whole” class and its part. Shown as an association decorated with a filled black diamond.	

Inheritance	Inheritance lets the methods in one class pass down the hierarchy. Shown as an association with clear arrowhead.	

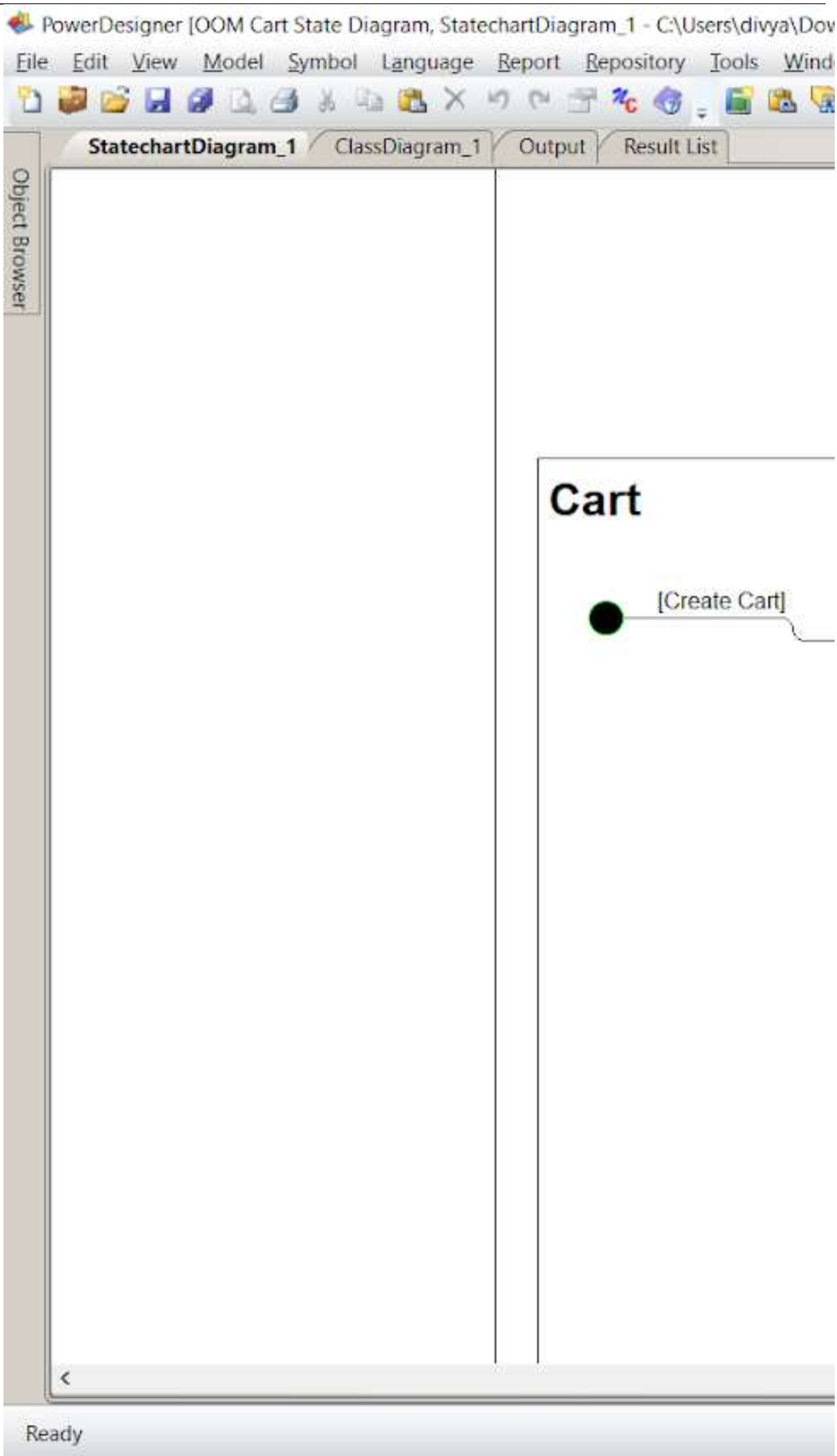
State Diagram Key		
Element	Description	Key

State	Rectangle with rounded edges. Filled with green sold gradient.	

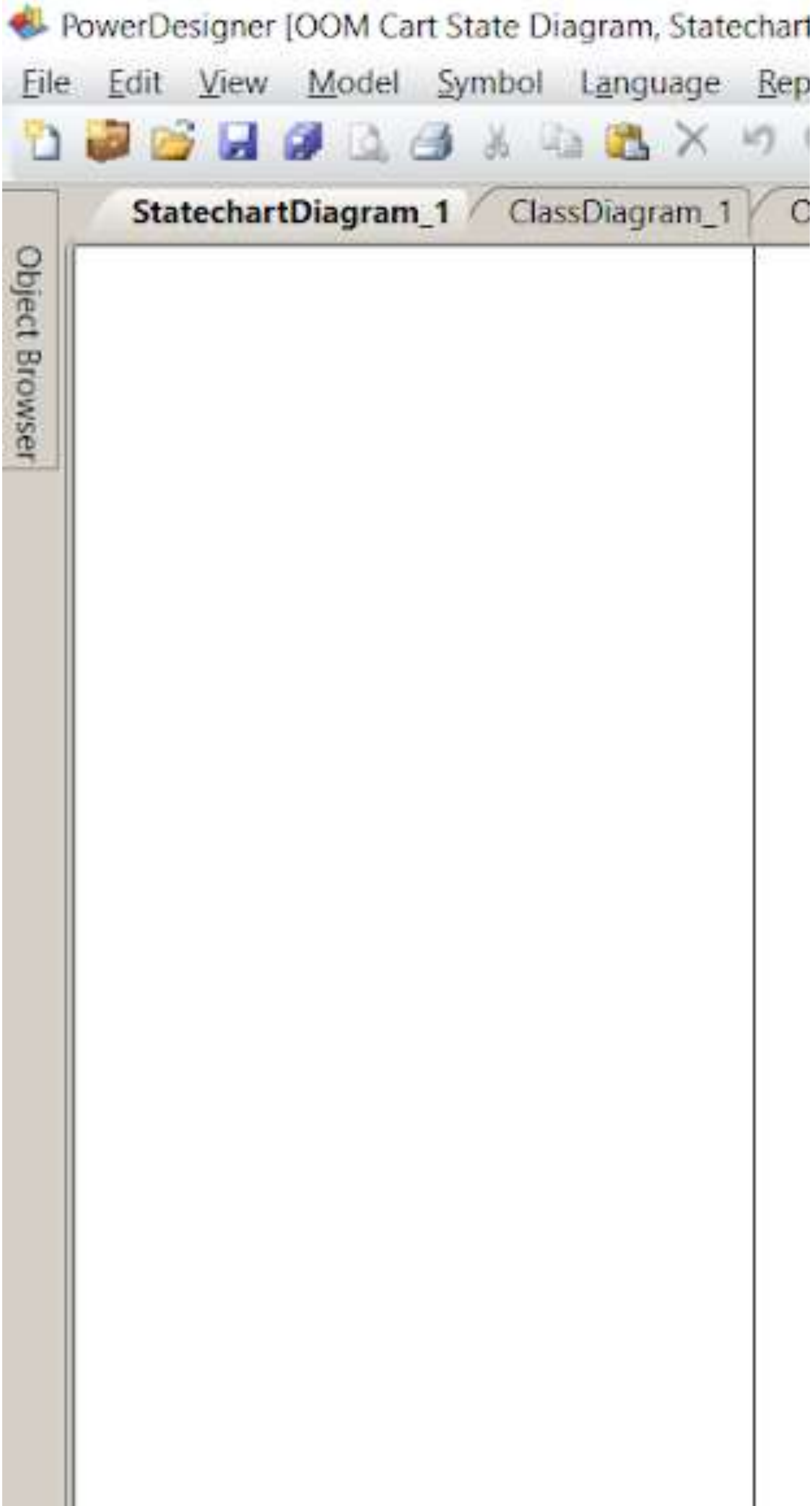
Transition	Black Line with solid black arrowhead and black text.	


PowerDesigner [OOM Cart State Diagram, StatechartDiagram_1 - C:\Use
File Edit View Model Symbol Language Report Repository
StatechartDiagram_1
ClassDiagram_1
Output
Object Browser
Cart

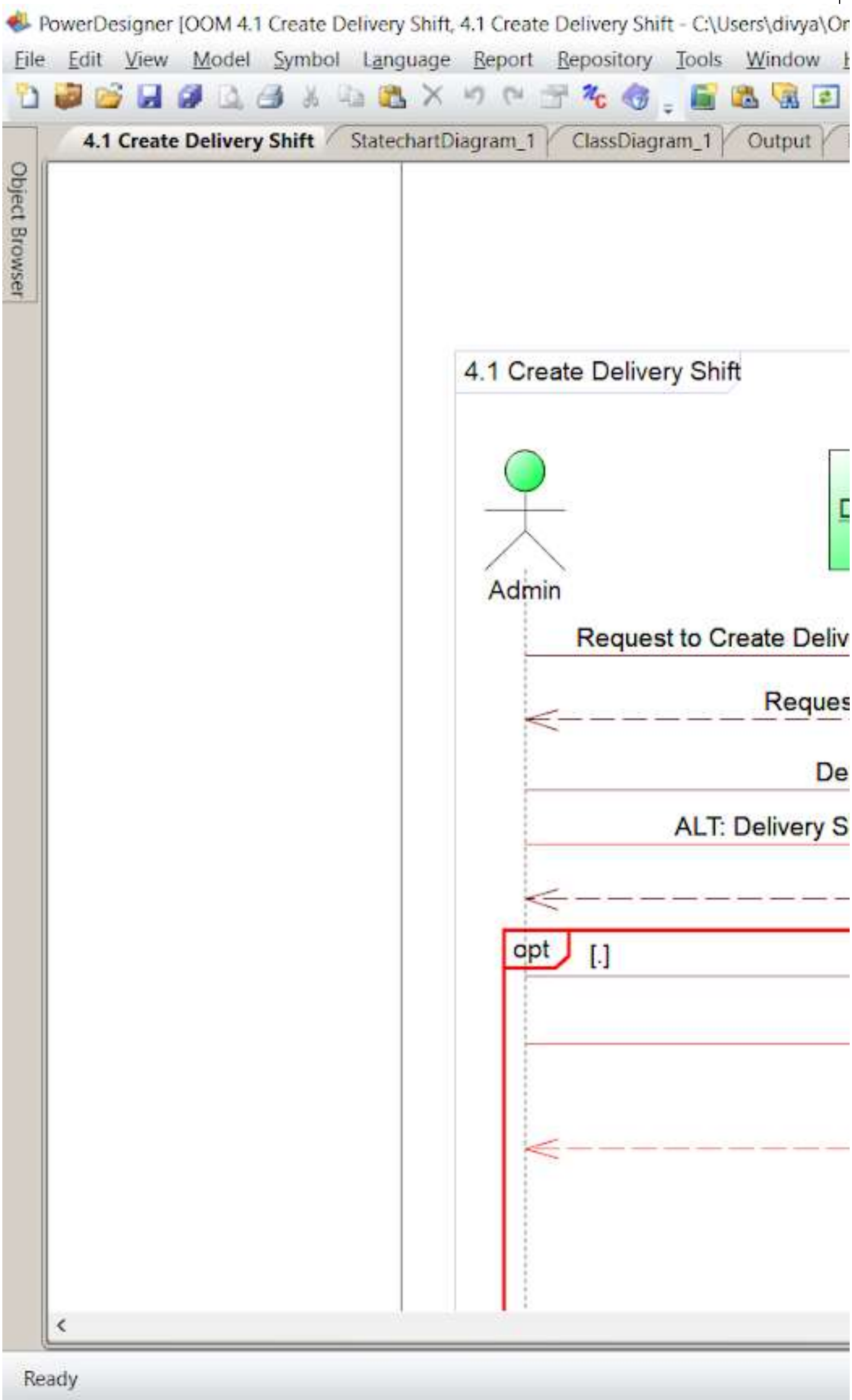
Start Node	Circle filled black with green outline.	

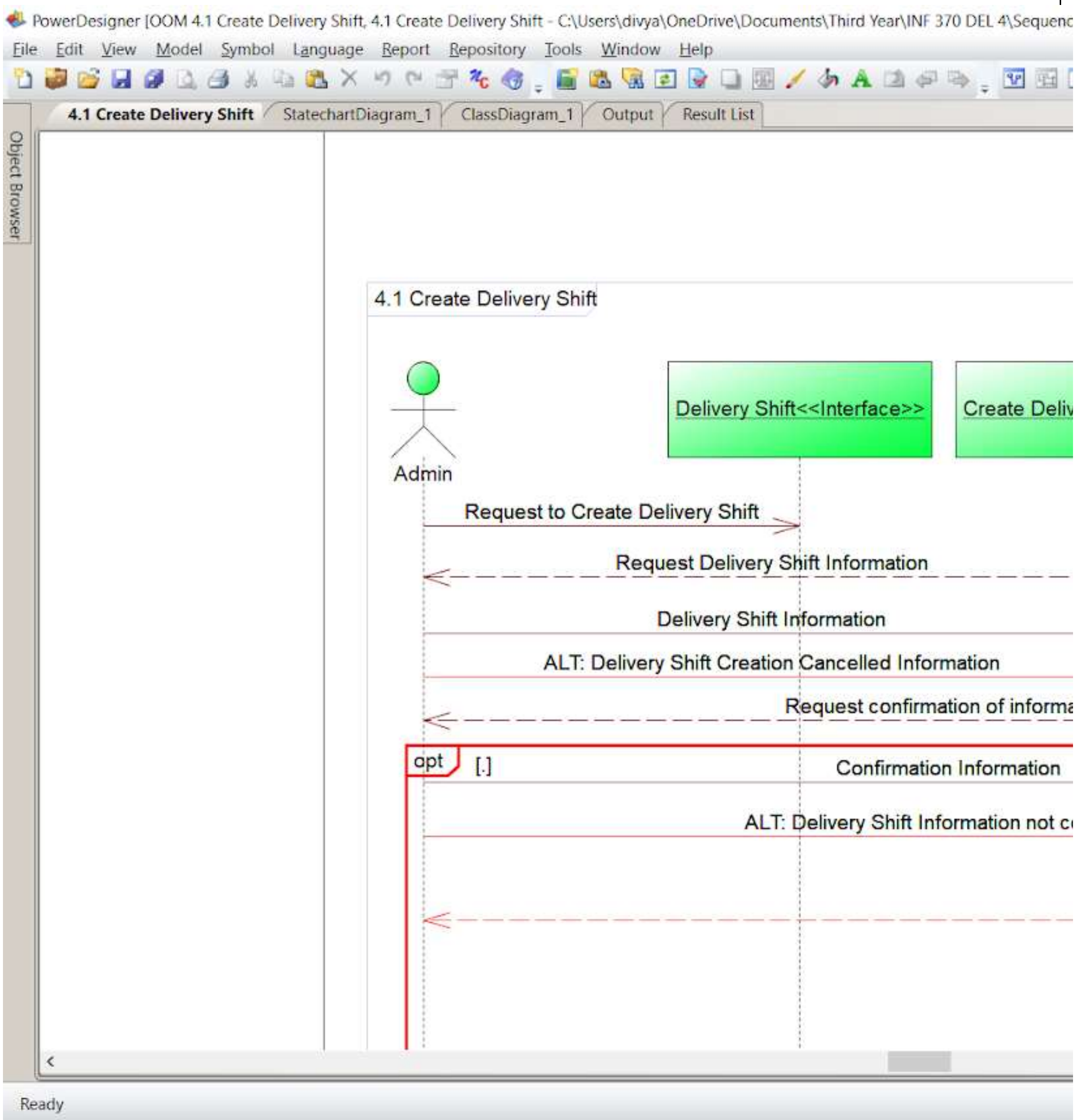

PowerDesigner [OOM Cart State Diagram, StatechartDiagram_1 - C:\Users\divya\Dov
File Edit View Model Symbol Language Report Repository Tools Wind
StatechartDiagram_1
ClassDiagram_1
Output
Result List
Object Browser
Cart
[Create Cart]
Ready

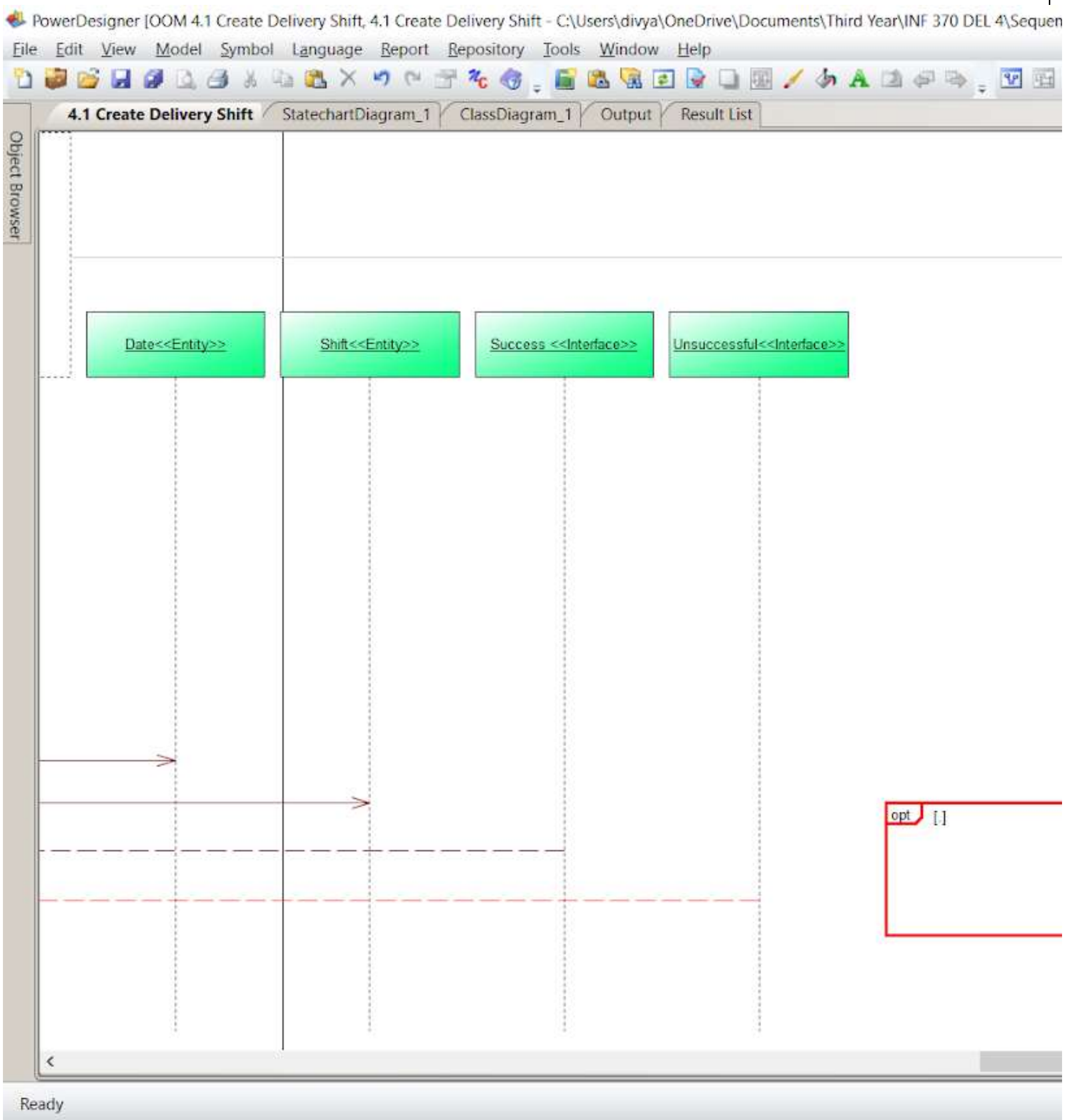
End Node	Solid Black Circle with Green outer circle.	

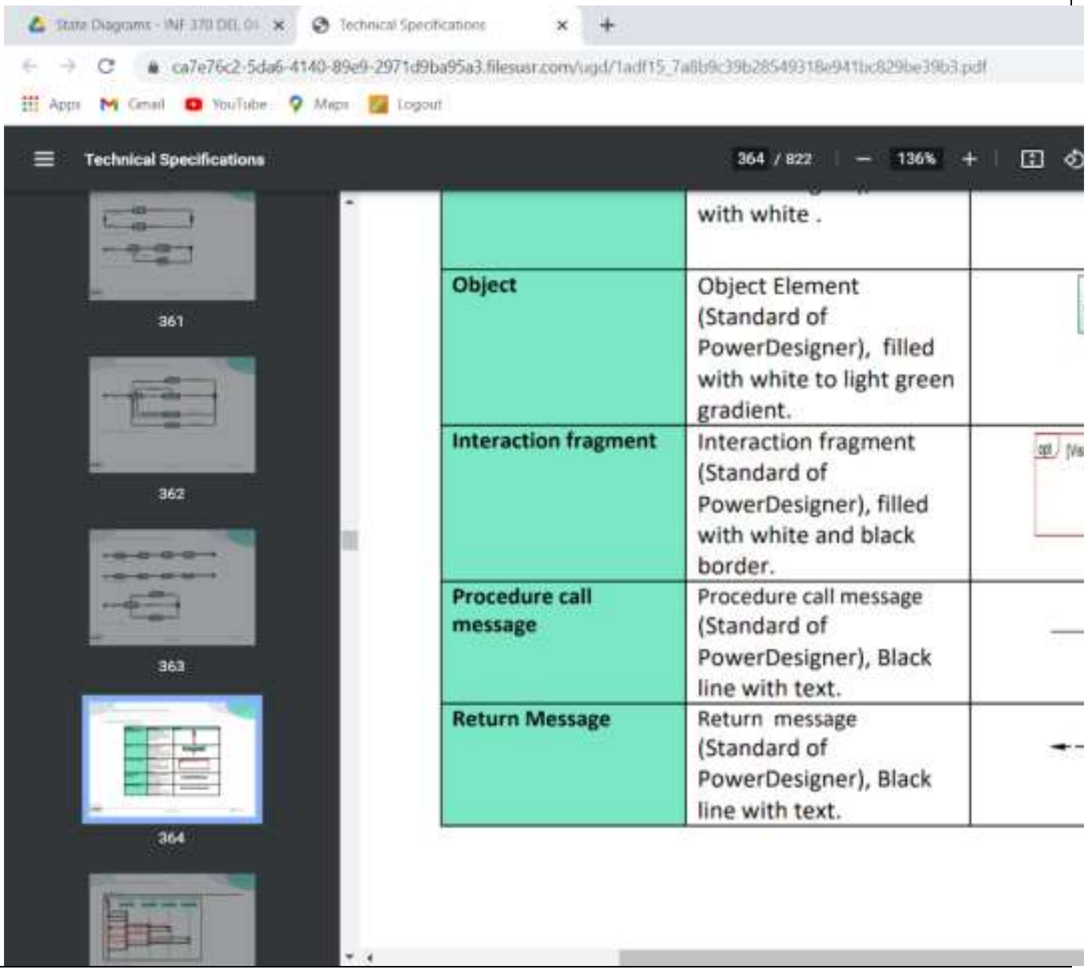

PowerDesigner [OOM Cart State Diagram, Statechart
File Edit View Model Symbol Language Rep
StatechartDiagram_1
ClassDiagram_1
Object Browser

Sequence Diagram Key		
Element	**Description**	**Key**

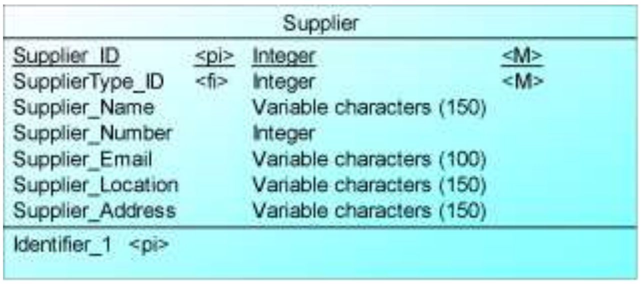
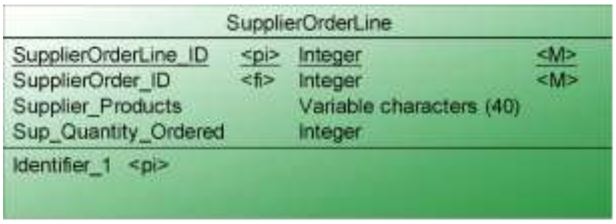
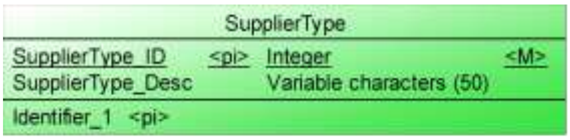
Actor	Stick Figure with black outline and green solid gradient circle.	

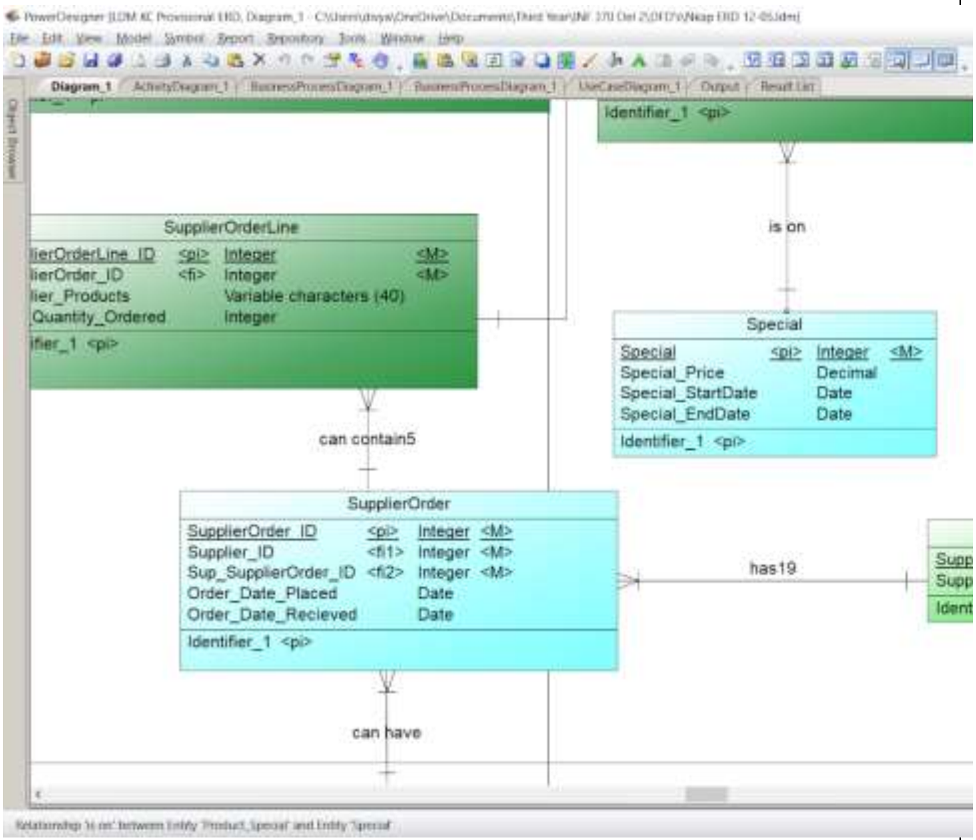
Object	Rectangle with black outline, solid green fill and black text.	

Integration Fragment	Rectangle with Red outline.	
Procedure Call Message	Black Line with solid arrow head and black text.	

Return Message	Black dotted line with solid arrow head and black text.	

Data Model Key		
Element	**Description**	**Example**

Entity	Entity element represented by rectangle with light blue fill.	
Associative Entity	Associative entity element represented by rectangle with dark green fill.	
Look-up Entity	Look-up entity element represented by rectangle with light green fill.	

Relationship	Relationship element represented by black line with appropriate relationship types.	

2.2. OBJECT CLASS DIAGRAM

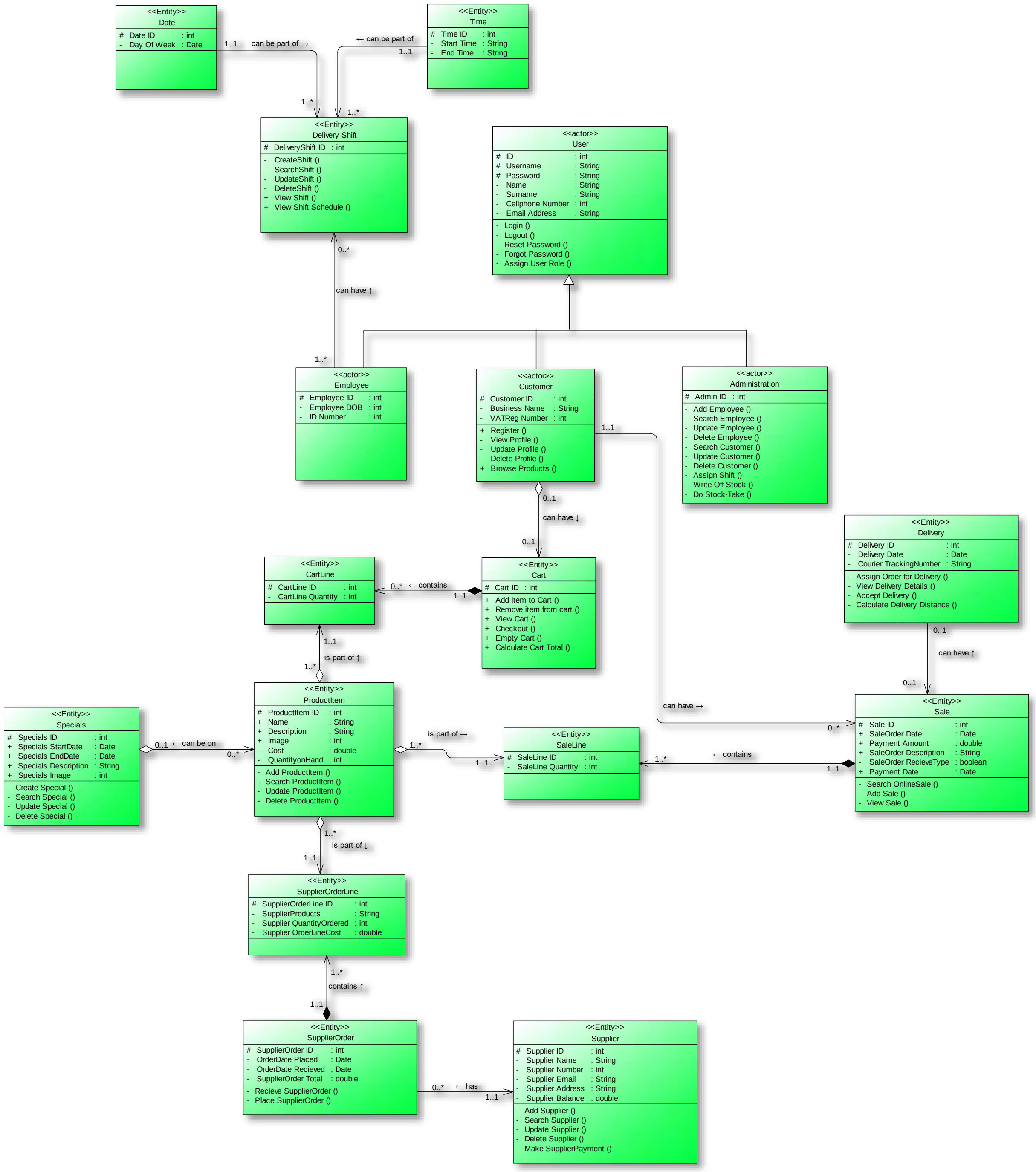


Figure 11 Object Class Diagram

2.3. STATE DIAGRAM

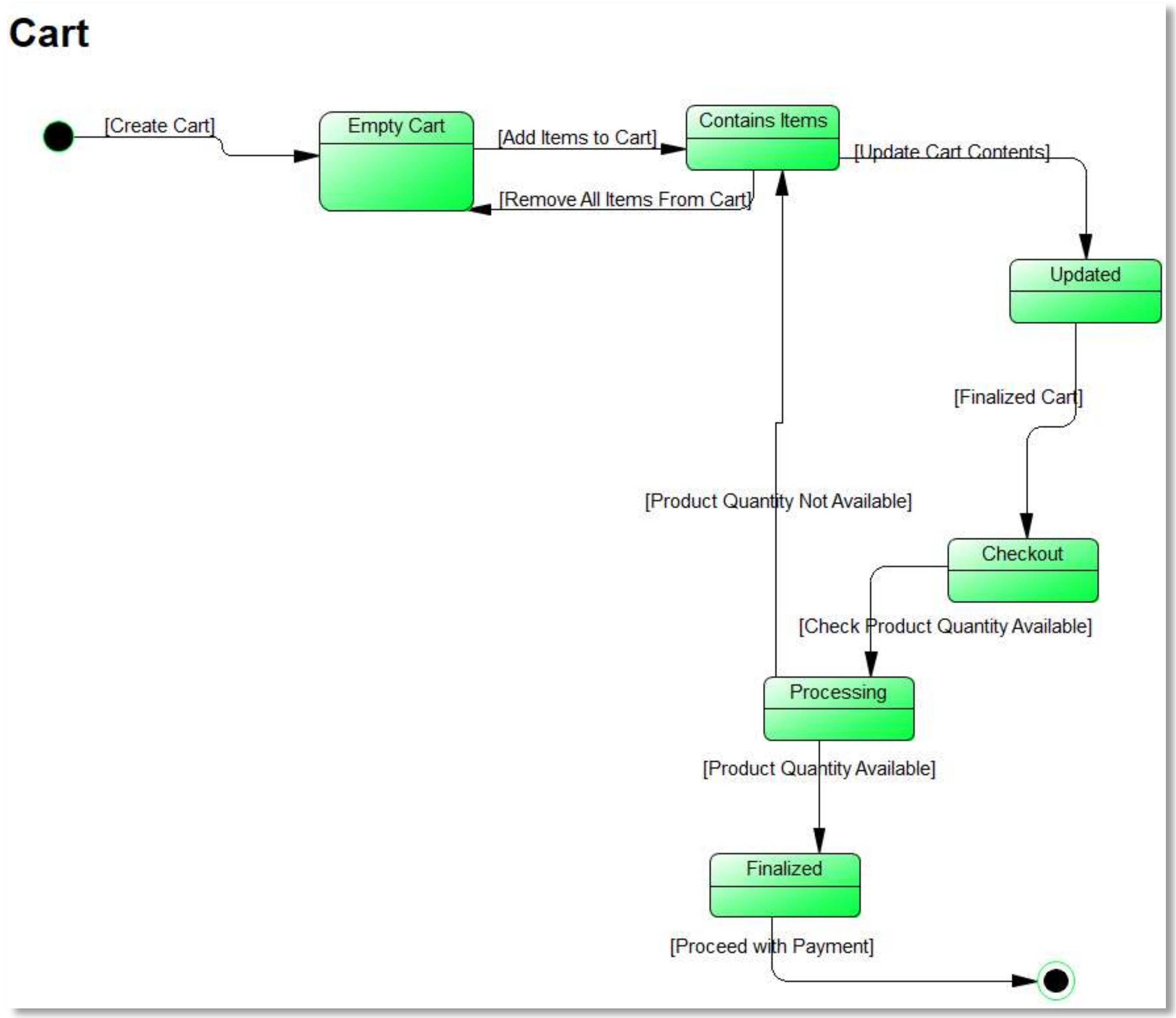


Figure 12 Cart State Diagram

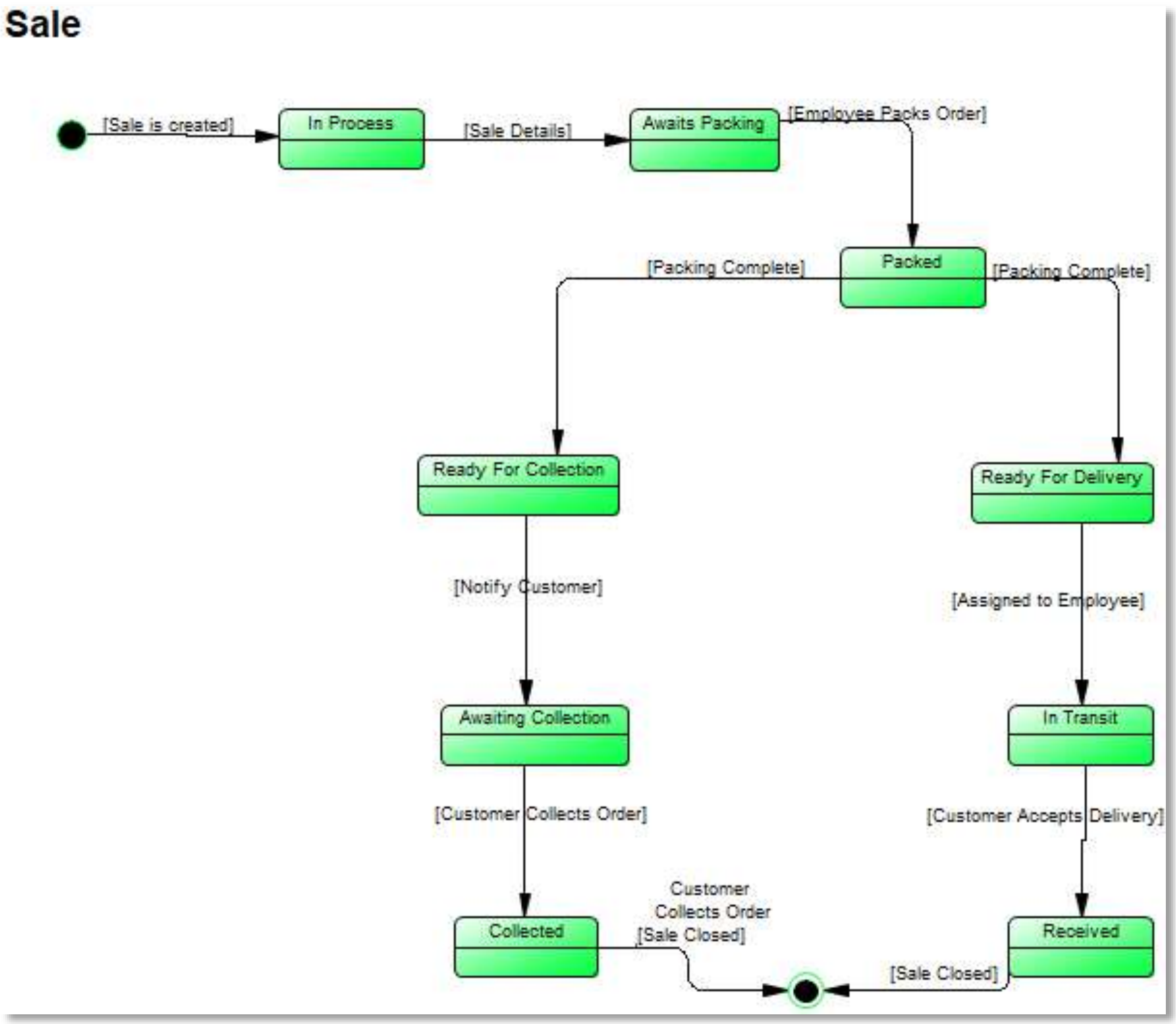


Figure 13 Sale State Diagram

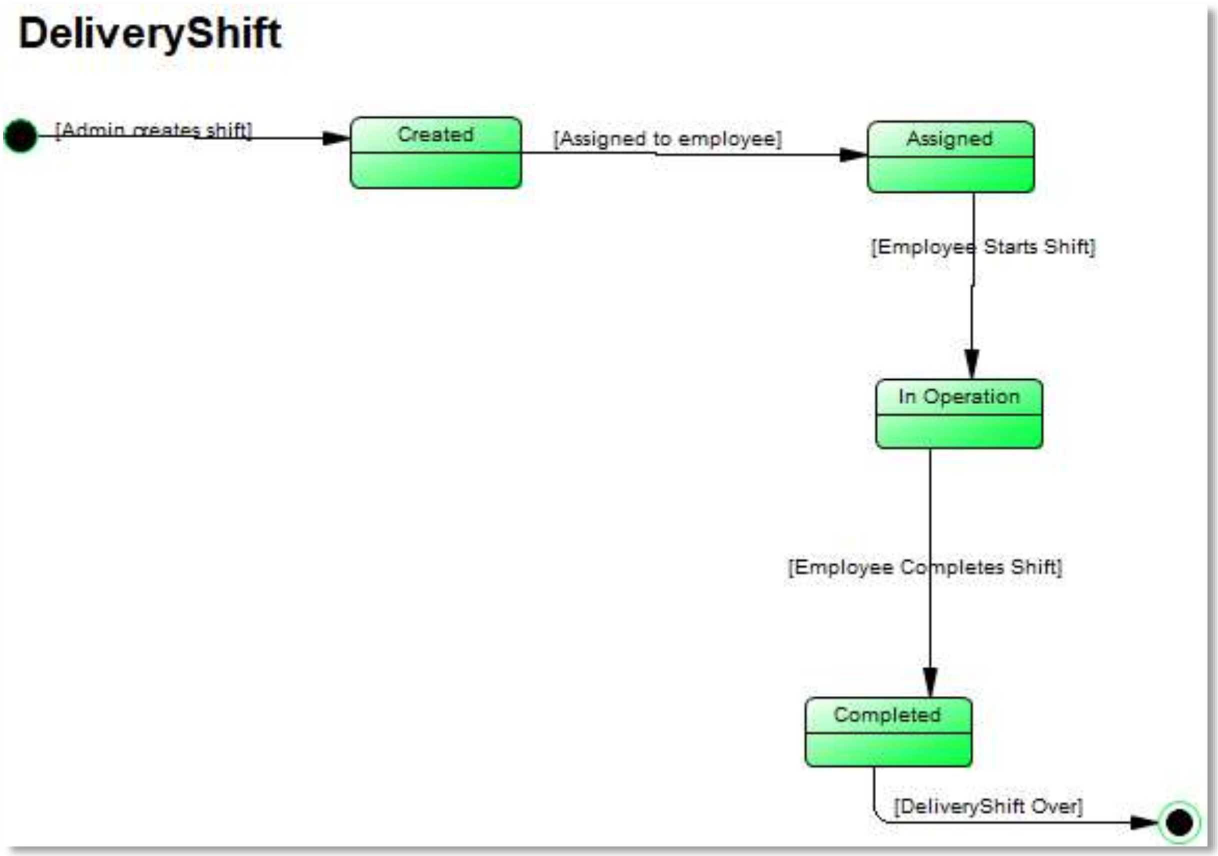


Figure 14 Delivery Shift State Diagram

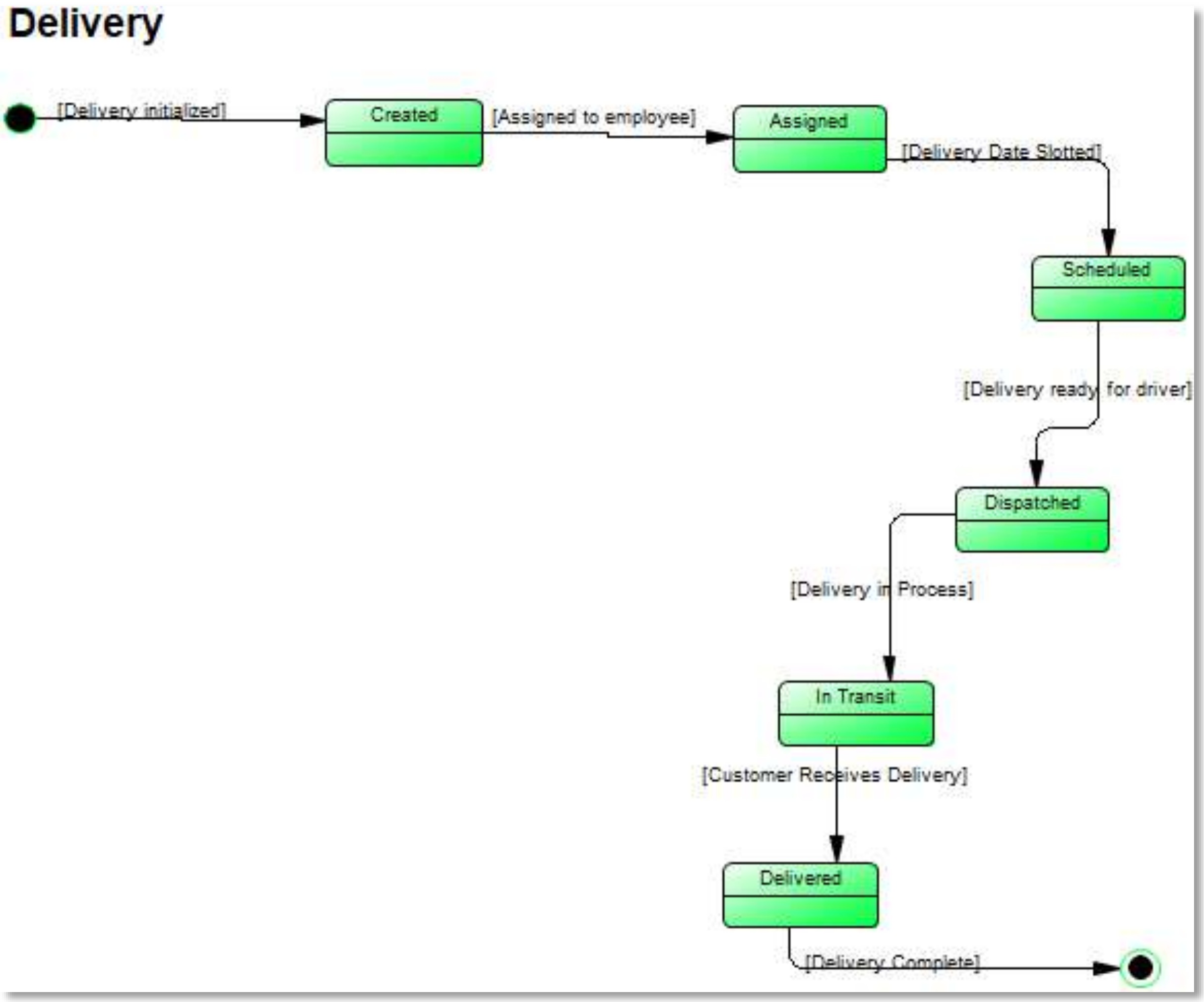


Figure 15 Delivery State Diagram

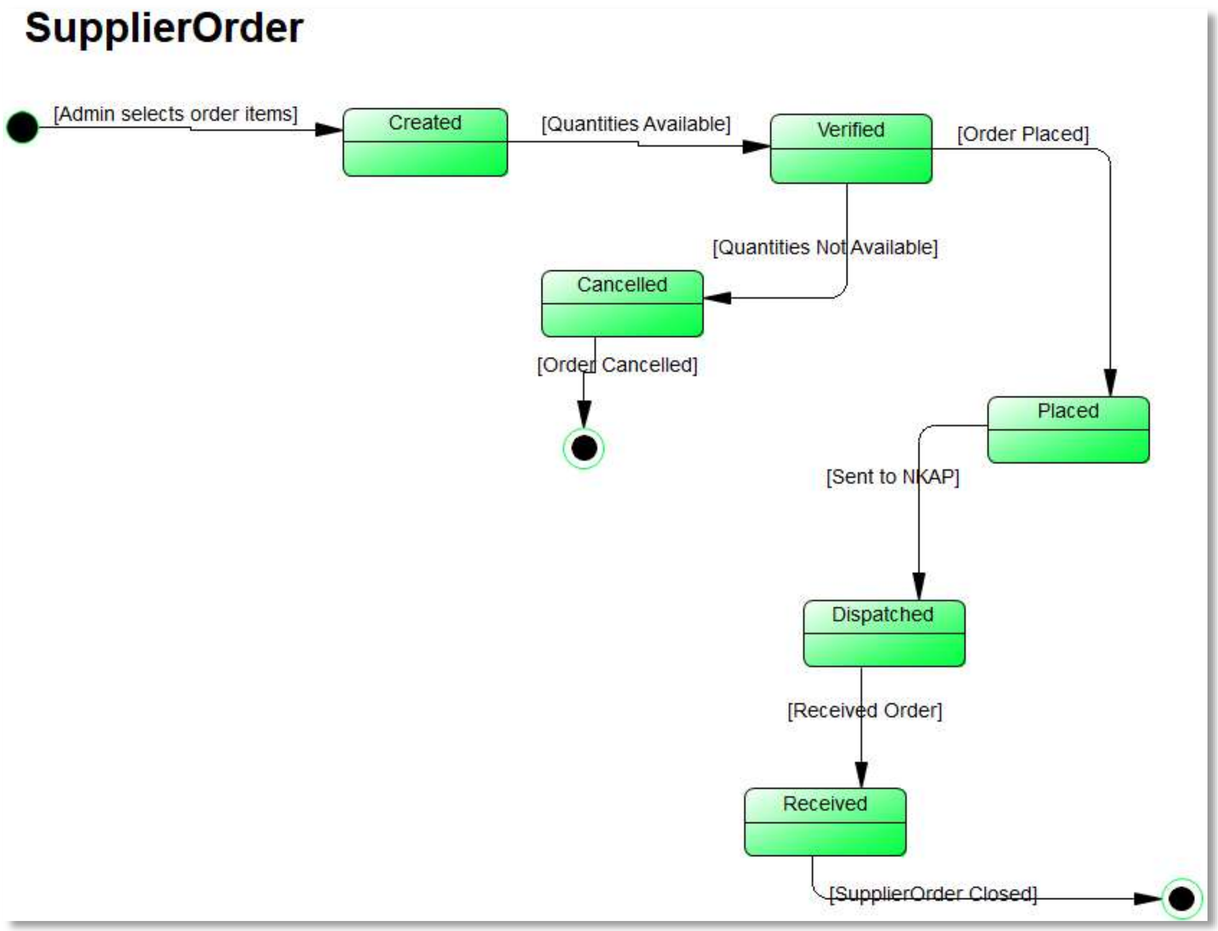


Figure 16 Supplier Order State Diagram

2.4. SEQUENCE DIAGRAM

2.4.1. Login Subsystem

Login

Figure 17 1.1. Login Sequence Diagram

Forgot Password

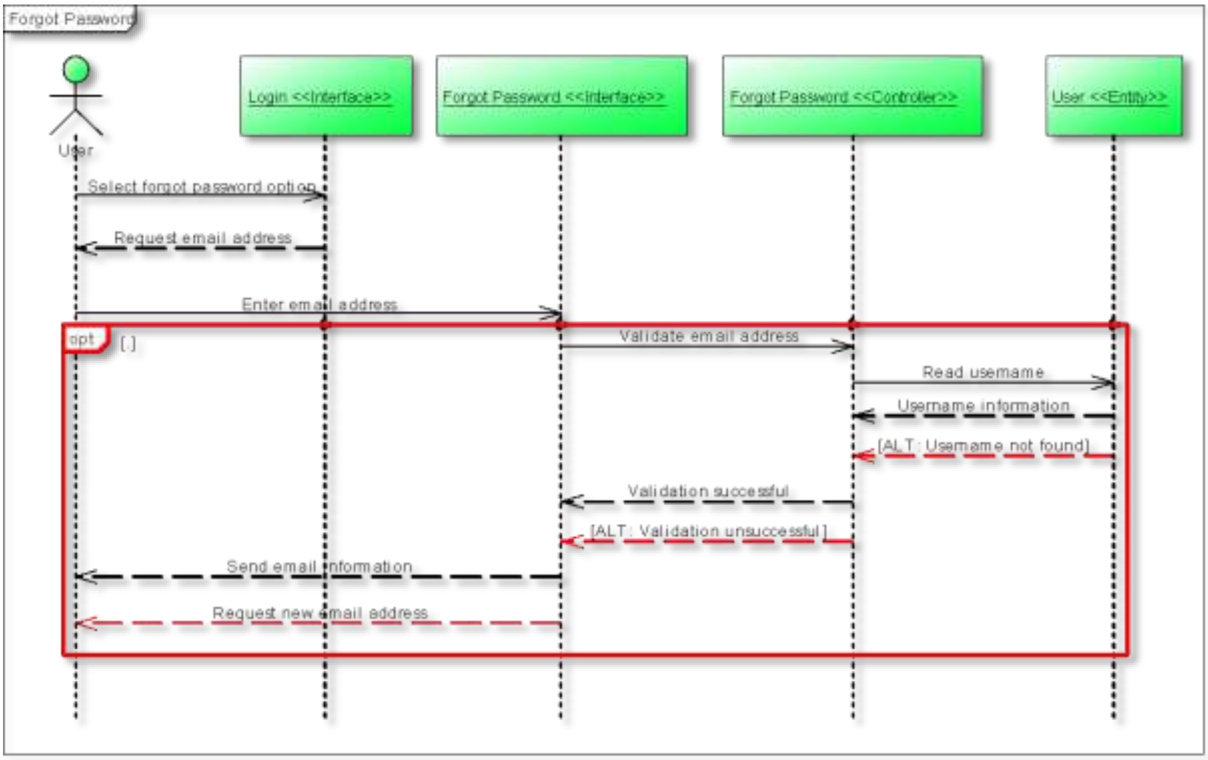


Figure 18 1.2. Forgot Password Sequence Diagram

Reset Password

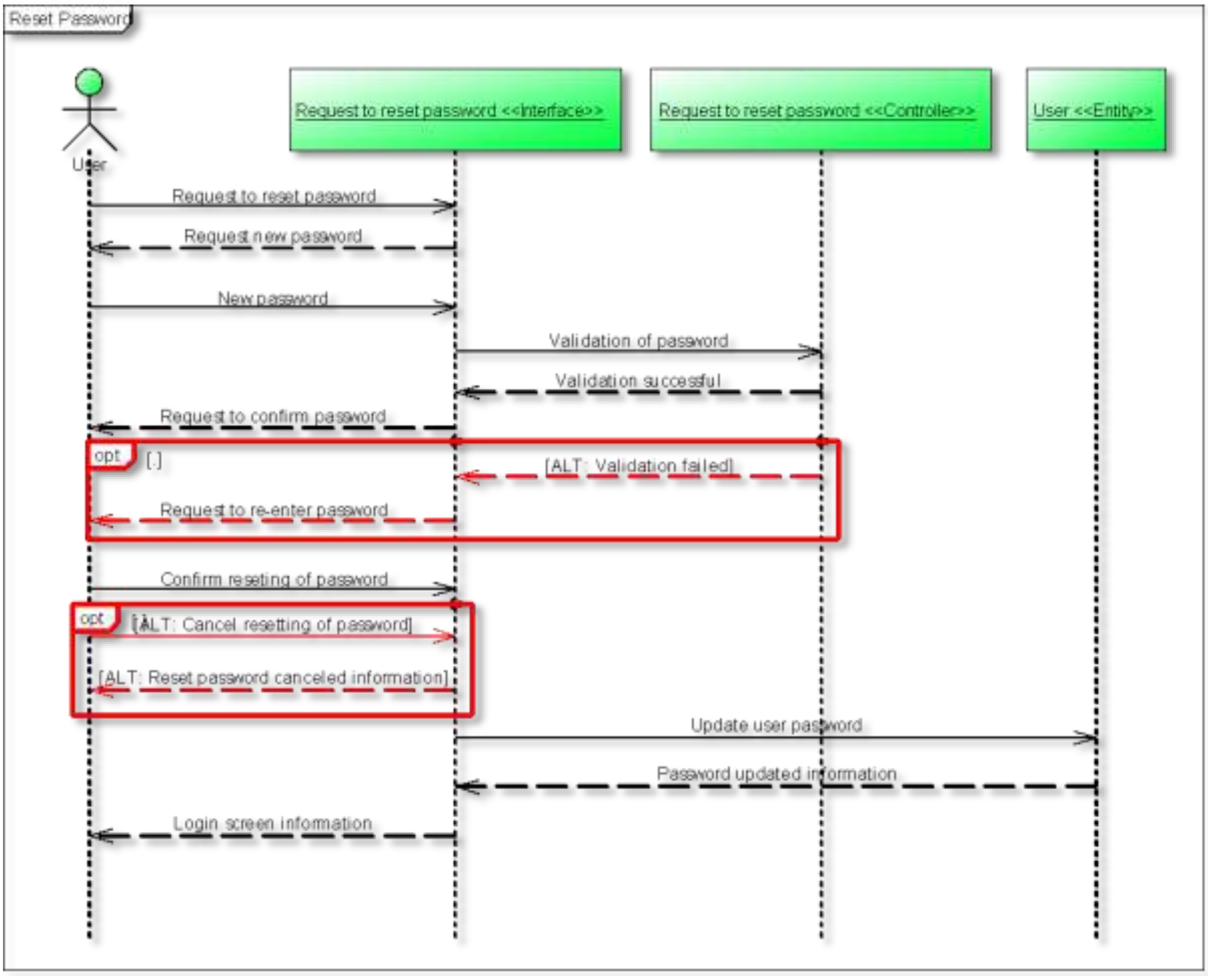


Figure 19 1.3. Reset Password Sequence Diagram

Logout

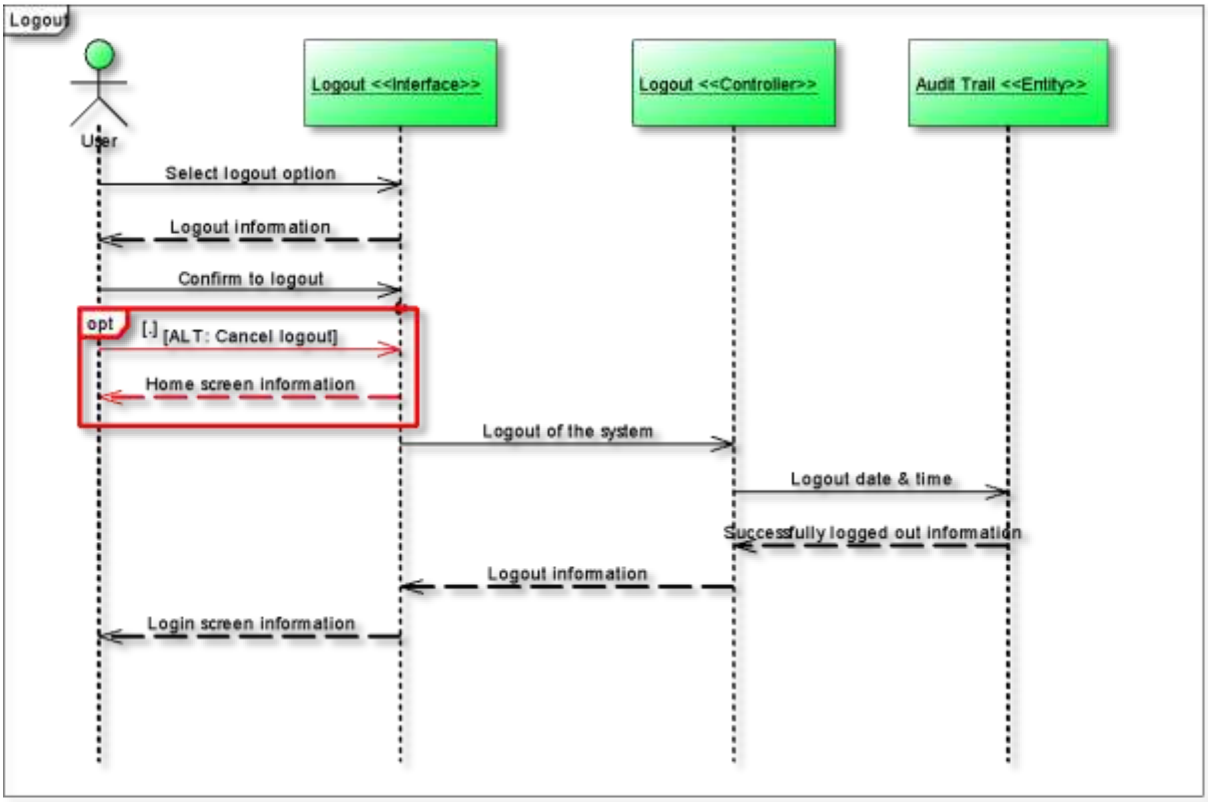


Figure 20 1.4. Logout Sequence Diagram

2.4.2. Supplier Subsystem

PASTE SEQUENCE DIAGRAMS ABOVE THE FOLLOWING CAPTIONS. DO NOT ERASE THE CAPTIONS!!

Add Supplier

<<insert Diagram here>>

Figure 21 2.1. Add Supplier Sequence Diagram

Search Supplier

<<insert Diagram here>>

Figure 22 2.2. Search Supplier Sequence Diagram

Update Supplier

<<insert Diagram here>>

Figure 23 2.3. Update Supplier Sequence Diagram

Delete Supplier

<<insert Diagram here>>

Figure 24 2.4. Delete Supplier Sequence Diagram

Place Supplier Order

<<insert Diagram here>>

Figure 25 2.5. Place Supplier Order Sequence Diagram

Receive Supplier Order

<<insert Diagram here>>

Figure 26 2.6. Receive Supplier Order Sequence Diagram

Capture Supplier Order

<<insert Diagram here>>

Figure 27 2.7. Capture Supplier Payment Sequence Diagram

2.4.3. User Subsystem

Register New Customer

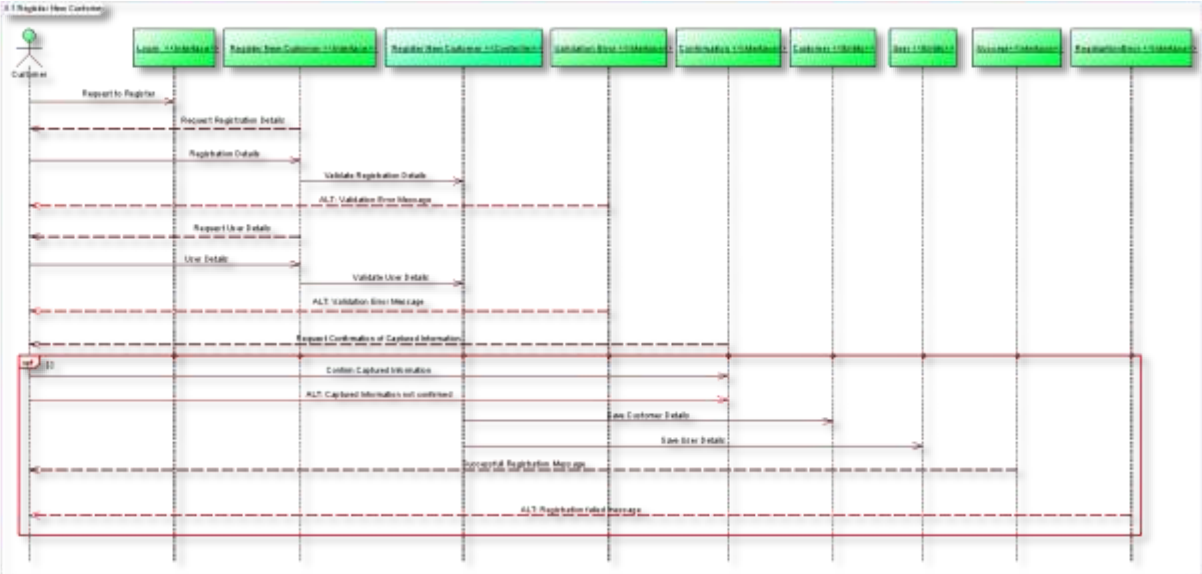


Figure 28 3.1. Register New Customer Sequence Diagram

Search Customer

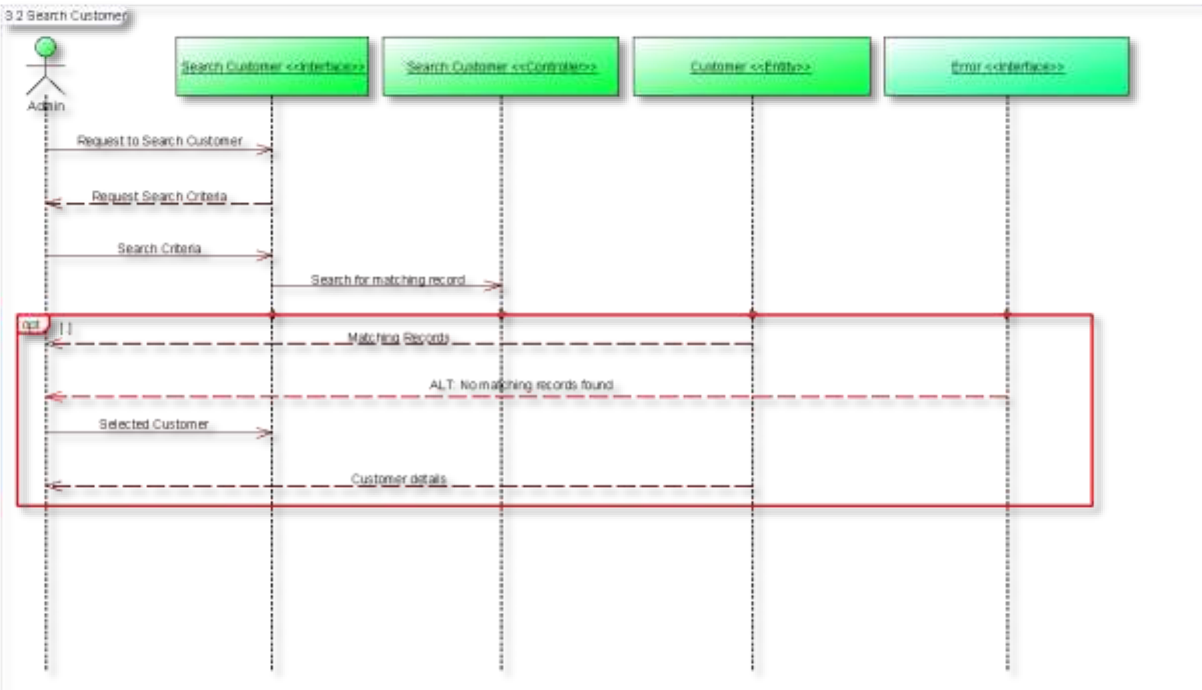


Figure 29 3.2. Search Customer Sequence Diagram

Update Customer

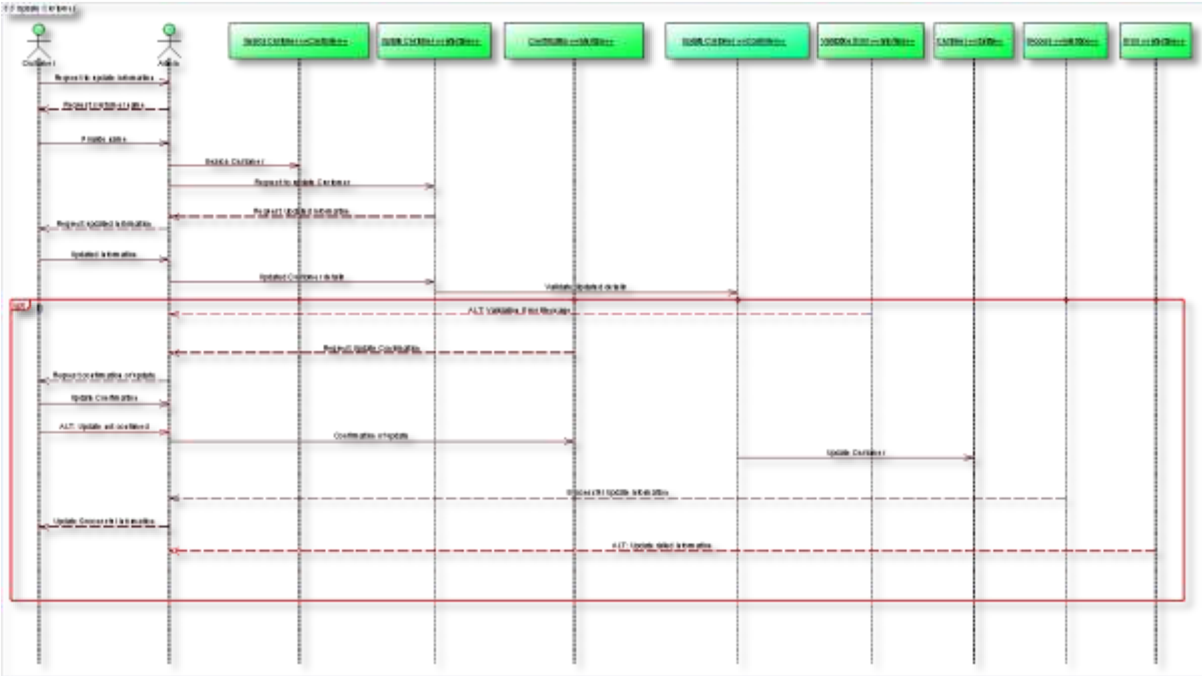


Figure 30 3.3. Update Customer Sequence Diagram

Delete Customer

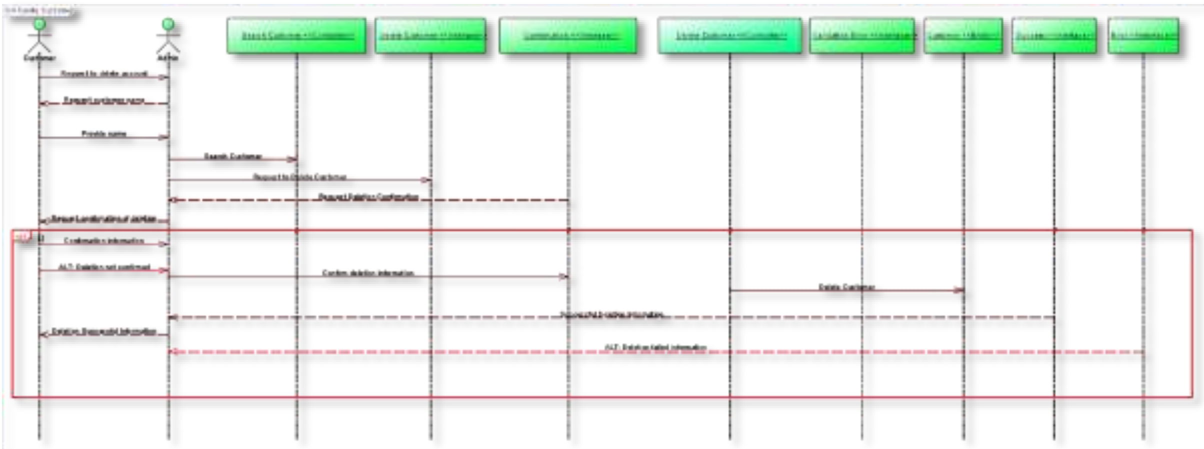


Figure 31 3.4. Delete Customer Sequence Diagram

View Profile

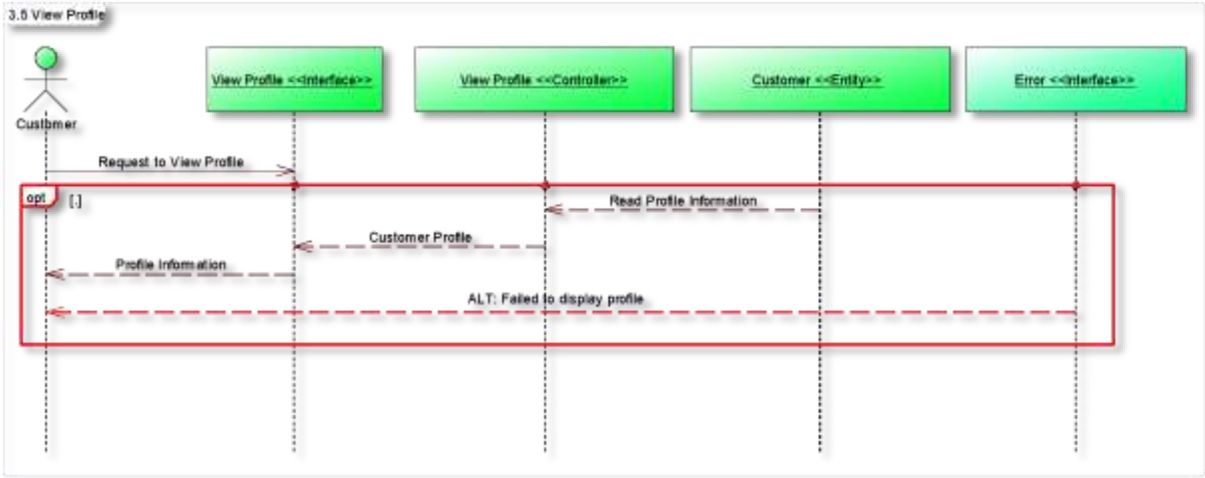


Figure 32 3.5. View Profile Sequence Diagram

Update Profile

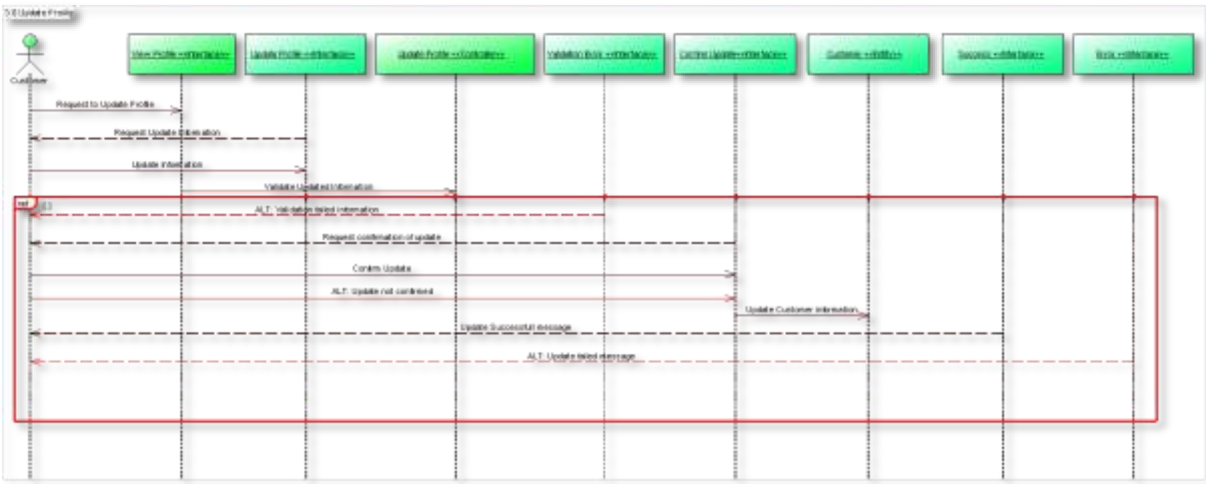


Figure 33 3.6. Update Profile Sequence Diagram

Delete Profile

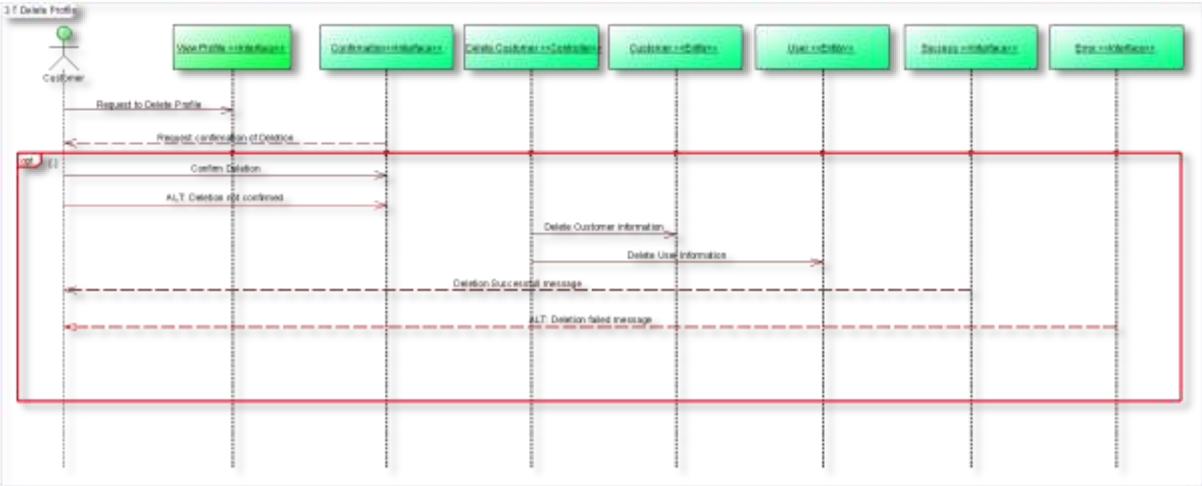


Figure 34 3.7. Delete Profile Sequence Diagram

Add User Role

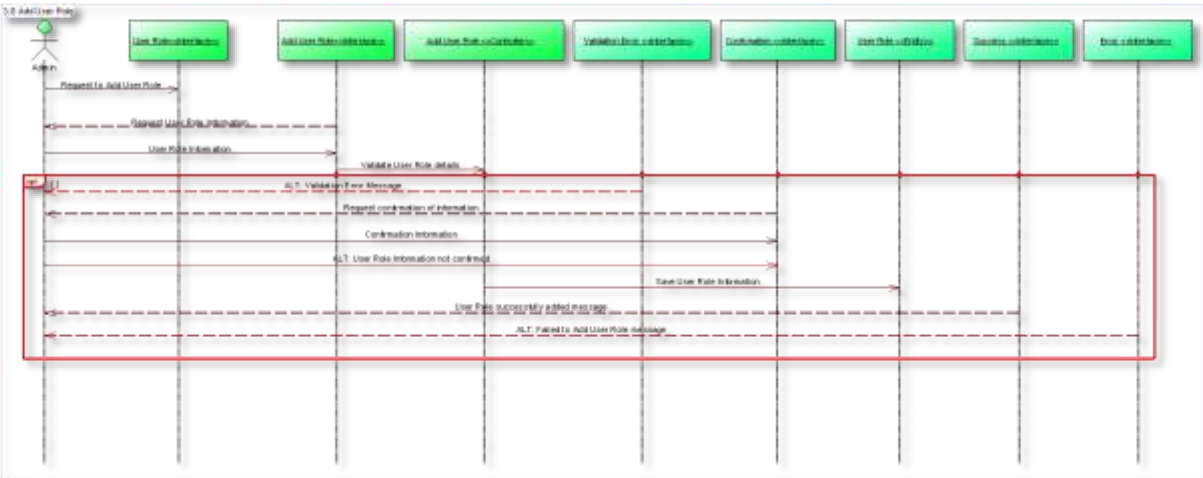


Figure 35 3.8. Add User Role Sequence Diagram

Search User Role

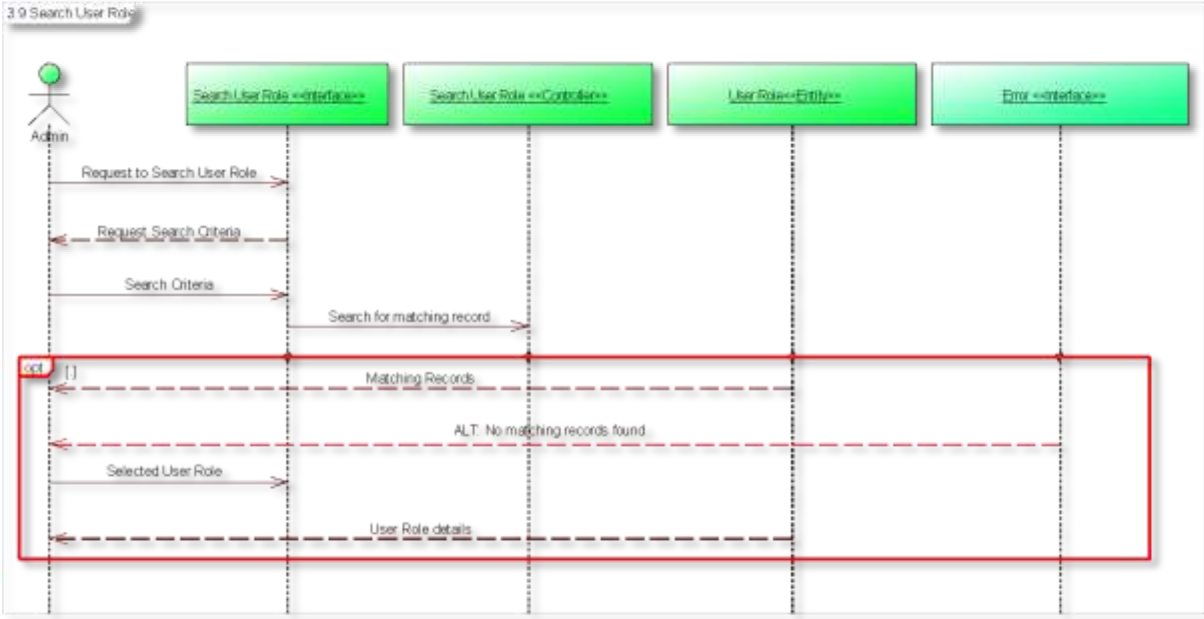


Figure 36 3.9. Search User Role Sequence Diagram

Update User Role

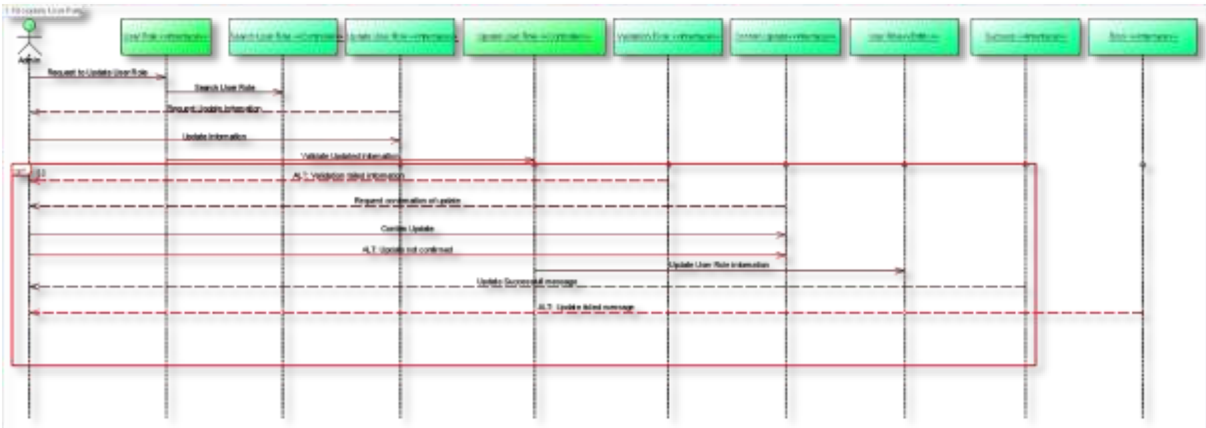


Figure 37 3.10. Update User Role Sequence Diagram

Delete User Role

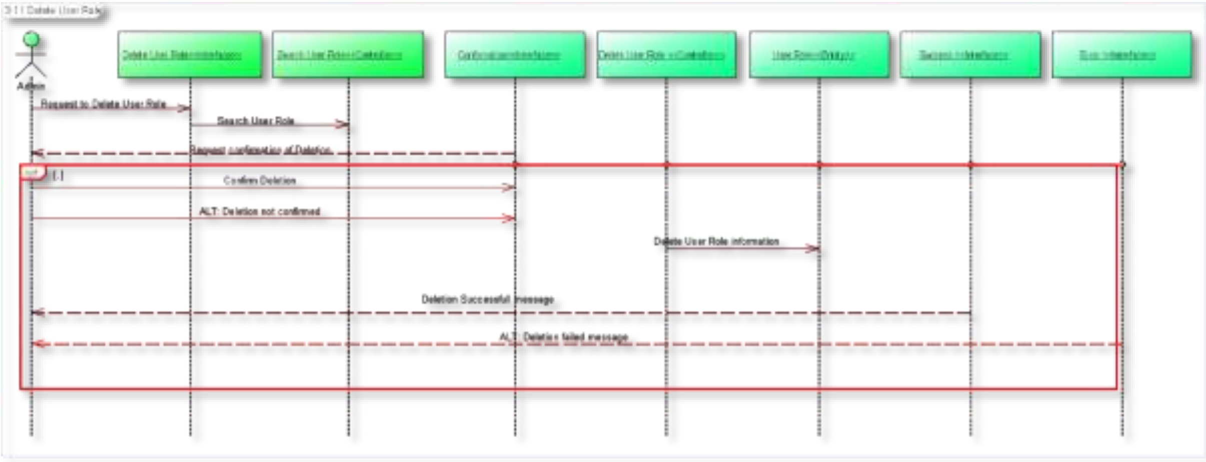


Figure 38 3.11. Delete User Role Sequence Diagram

2.4.4. Employee Delivery Shifts Subsystem

Create Delivery Shift

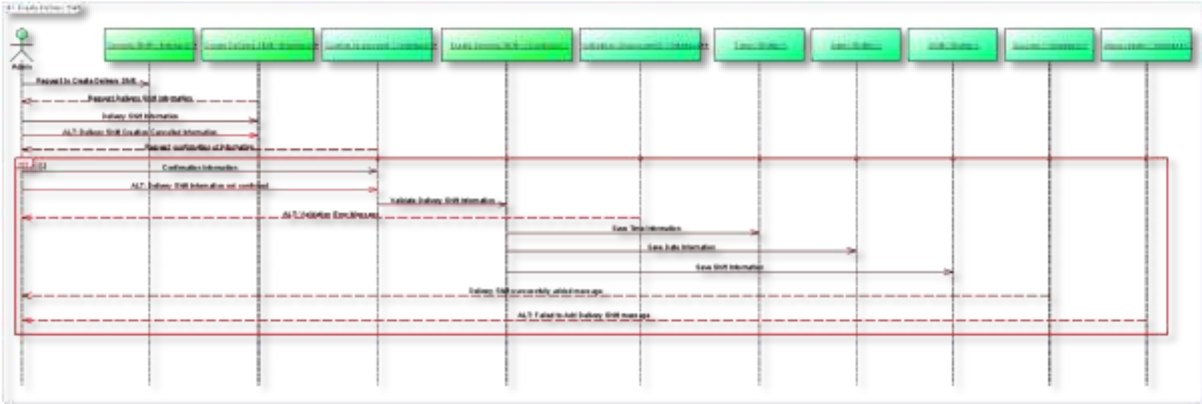


Figure 39 4.1. Create Delivery Shift Sequence Diagram

Search Delivery Shift

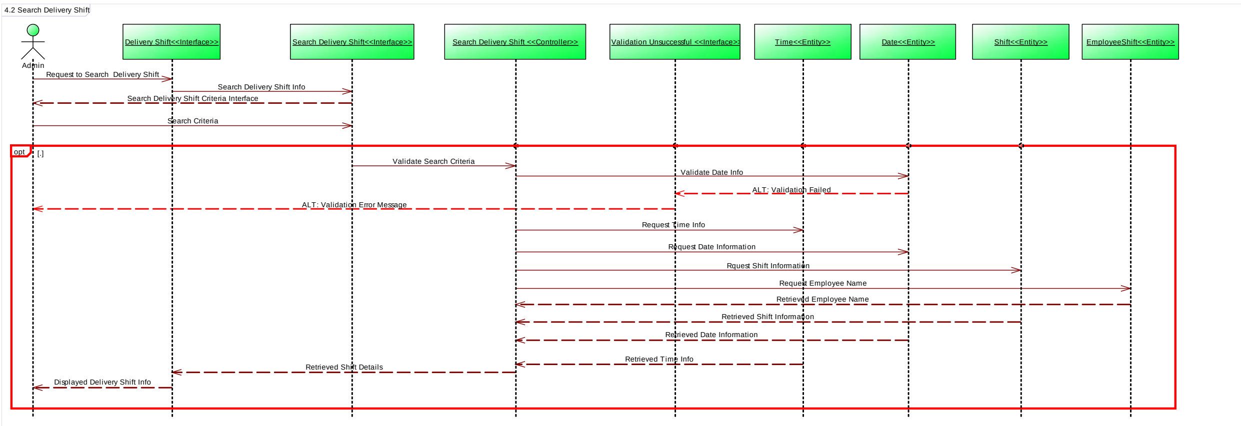


Figure 40 4.2. Search Delivery Shift Sequence Diagram

Update Delivery Shift

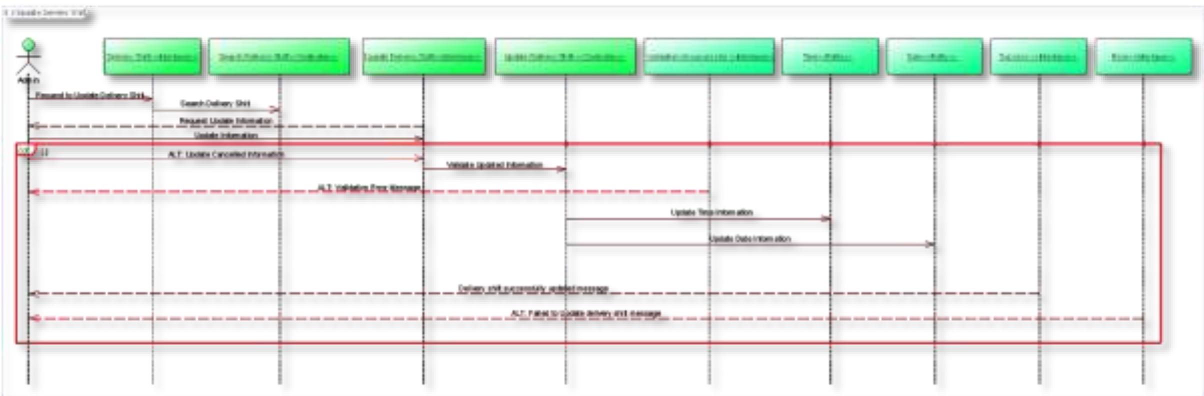


Figure 41 4.2. Update Delivery Shift Sequence Diagram

Delete Delivery Shift

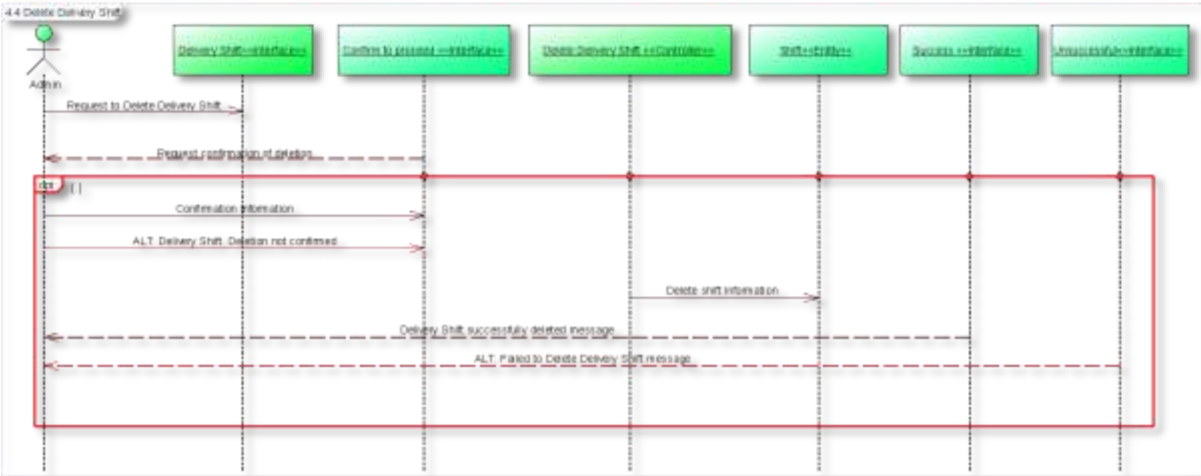


Figure 42 4.4. Delete Delivery Shift Sequence Diagram

Add Employee

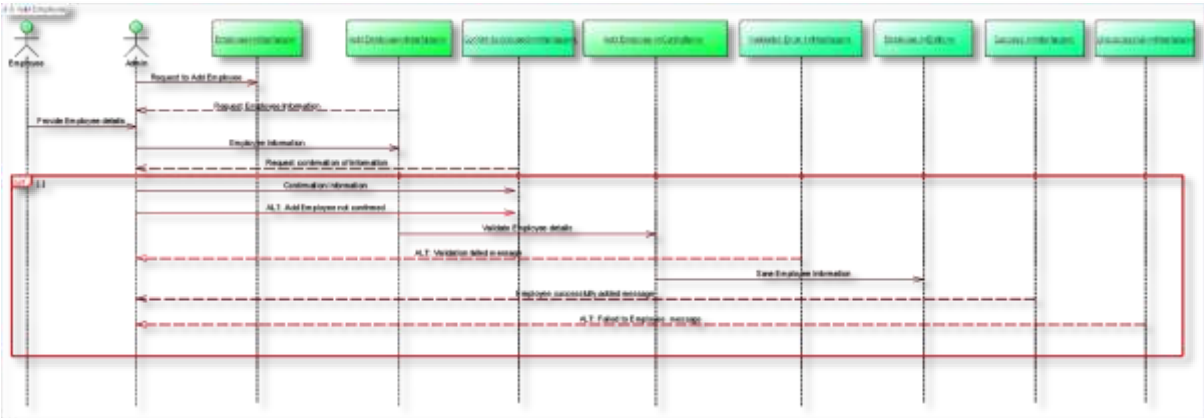


Figure 43 4.5. Add Employee Sequence Diagram

Search Employee

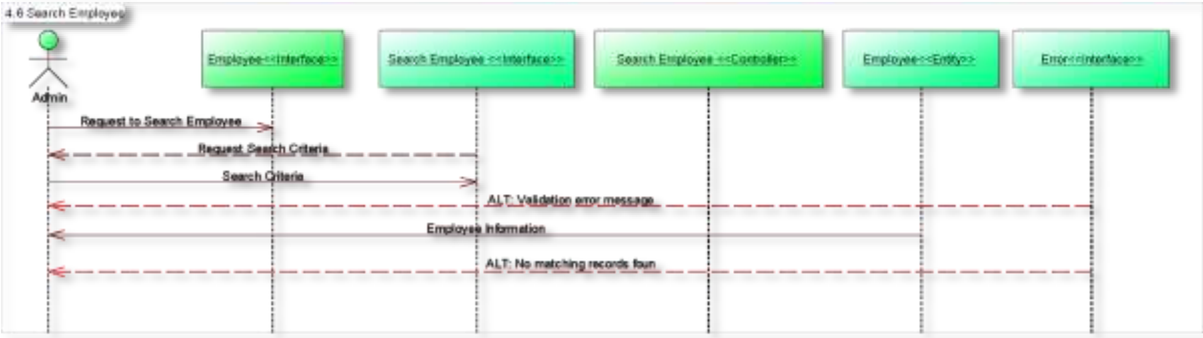


Figure 44 4.6. Search Employee Sequence Diagram

Update Employee

<<insert Diagram here>>

Figure 45 4.7. Update Employee Sequence Diagram

Delete Employee

<<insert Diagram here>>

Figure 46 4.8. Delete Employee Sequence Diagram

Assign Order for Delivery/Courier

<<insert Diagram here>>

Figure 47 4.9. Assign Order for Delivery/Courier Sequence Diagram

Assign Delivery Shift

<<insert Diagram here>>

Figure 48 4.10. Assign Delivery Shift Sequence Diagram

Cancel Delivery Shift

<<insert Diagram here>>

Figure 49 4.11. Cancel Delivery Shift Sequence Diagram

2.4.5. Admin Subsystem

Search Online Sales

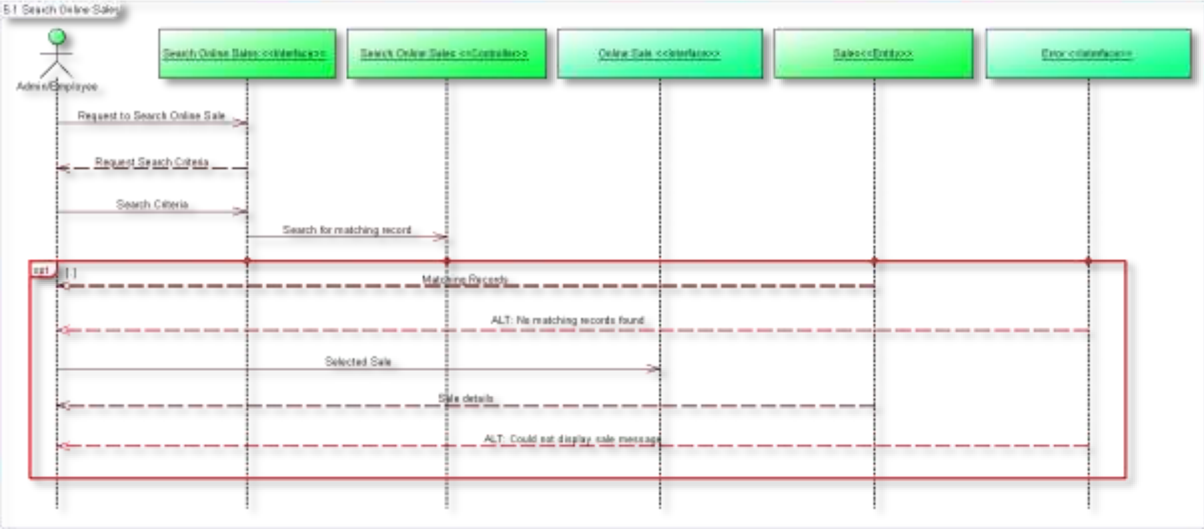


Figure 50 5.1. Search Online Sales Sequence Diagram

Pack Order

Figure 51 5.2. Pack Order Sequence Diagram

Do Stock-take

Figure 52 5.3. Do Stock-take Sequence Diagram

View Delivery Shift Schedule

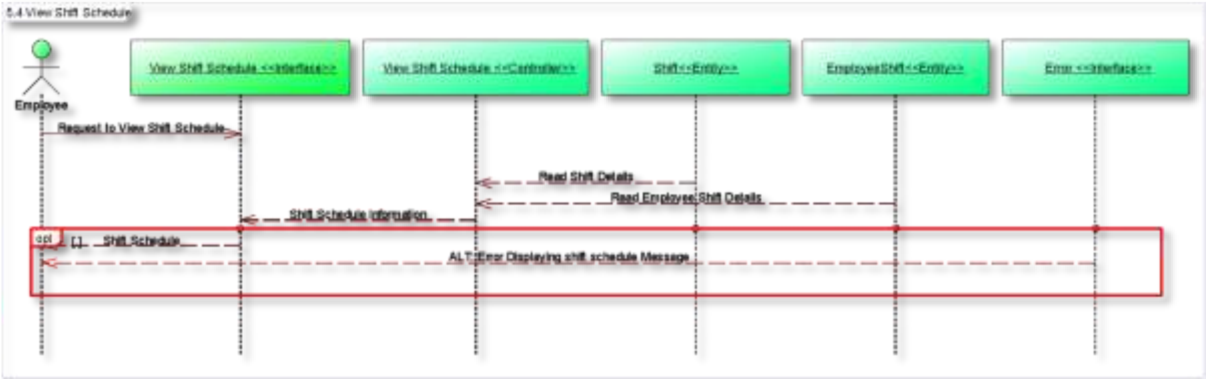


Figure 53 5.4. View Delivery Shift Schedule Sequence Diagram

Write-Off Stock

Figure 54 5.5. Write-off Stock Sequence Diagram

2.4.6. Customer Subsystem

View Products

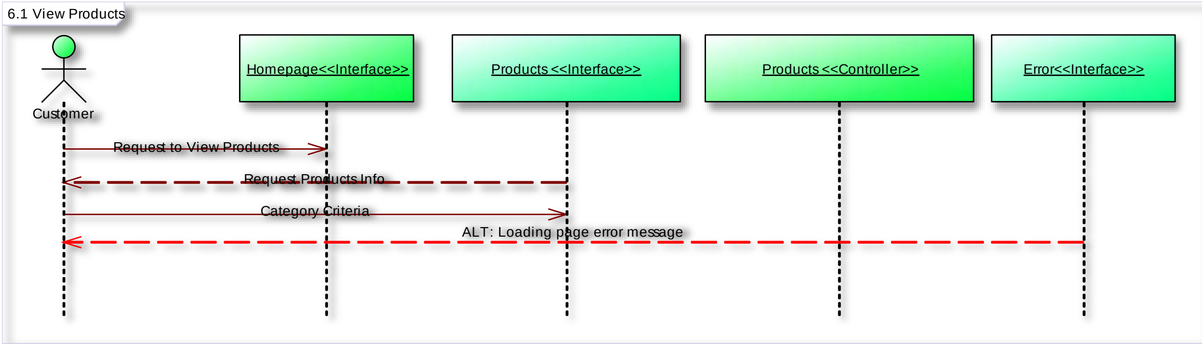


Figure 55 6.1. View Products Sequence Diagram

View Product Details

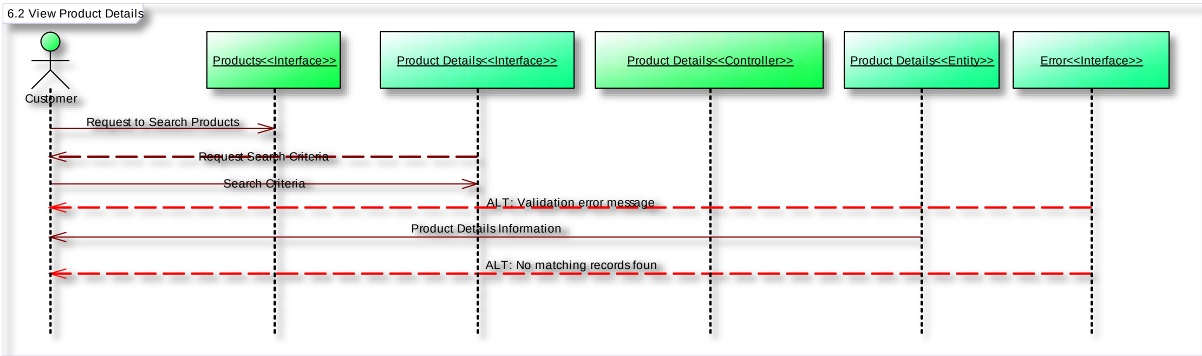


Figure 56 6.2. View Product Details Sequence Diagram

Search Specials and Promotions

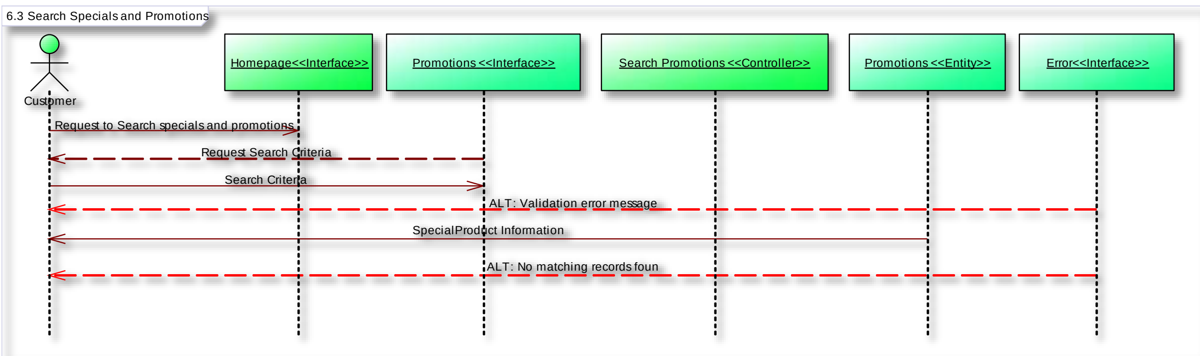


Figure 57 6.3. Search Specials and Promotions Sequence Diagram

Search Cart

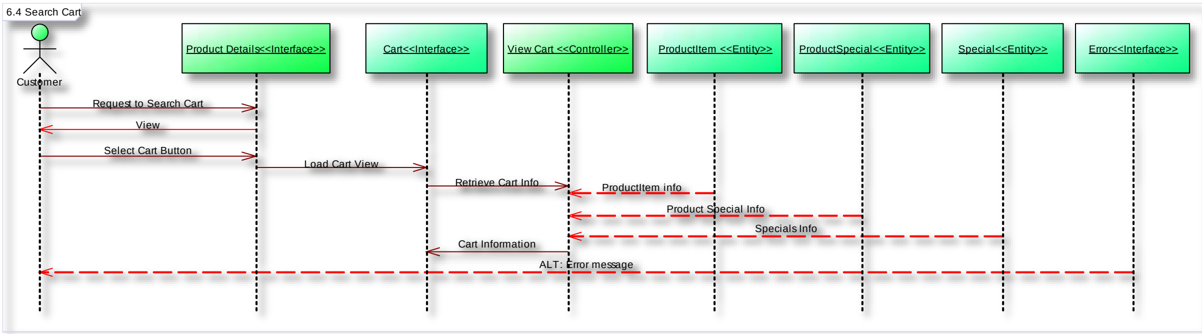


Figure 58 6.4. Search Cart Sequence Diagram

Add Item to Cart

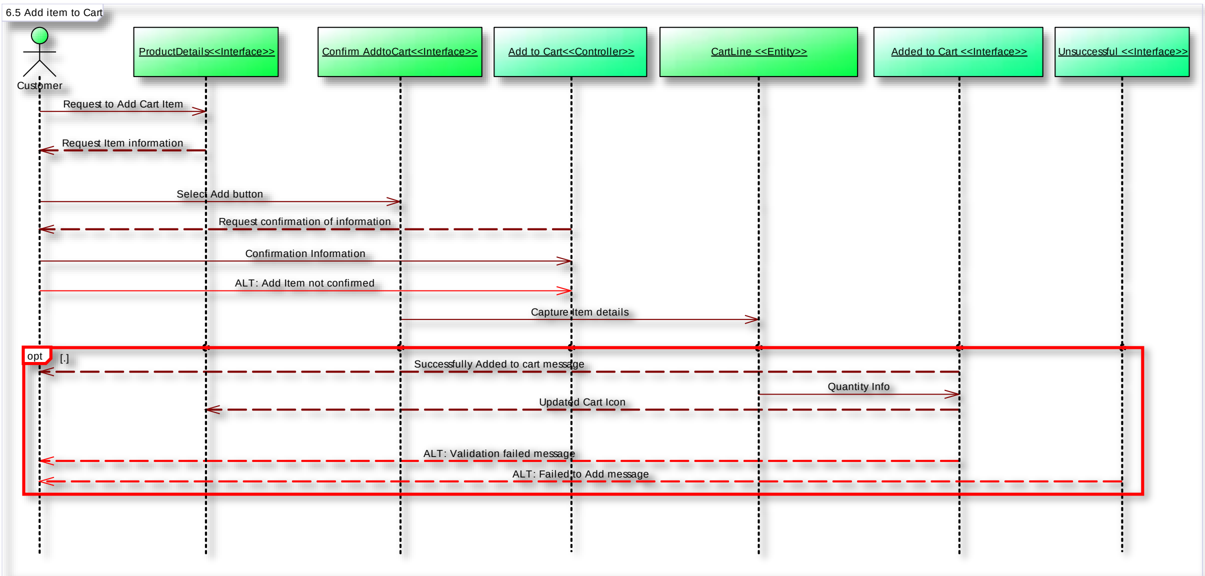


Figure 59 6.5. Add Item to Cart Sequence Diagram

Remove Item from Cart

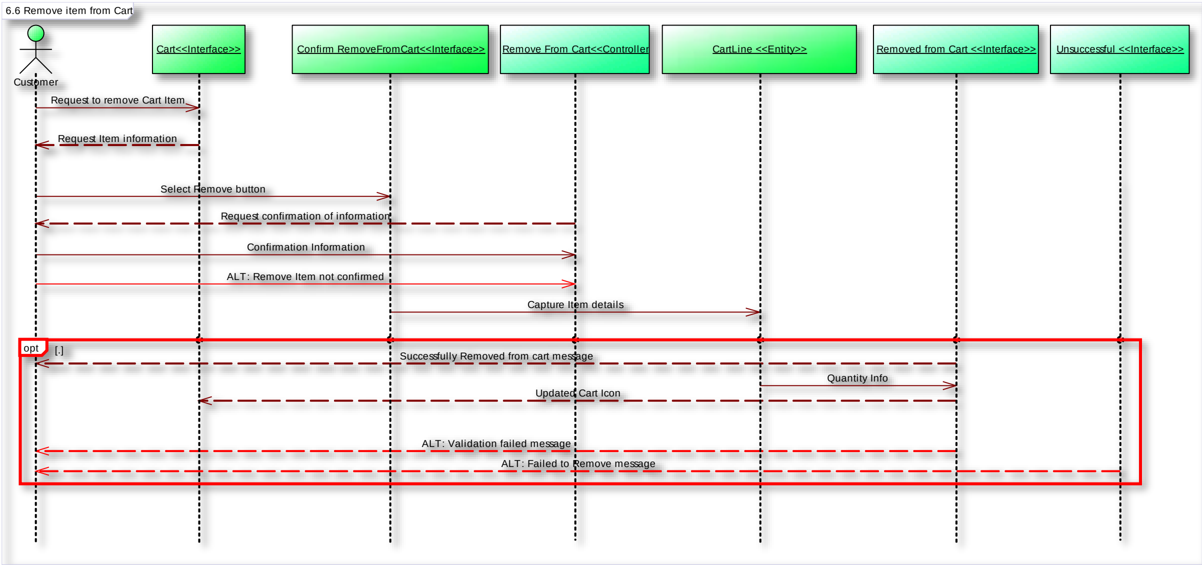


Figure 60 6.6. Remove Item from Cart Sequence Diagram

Checkout Order

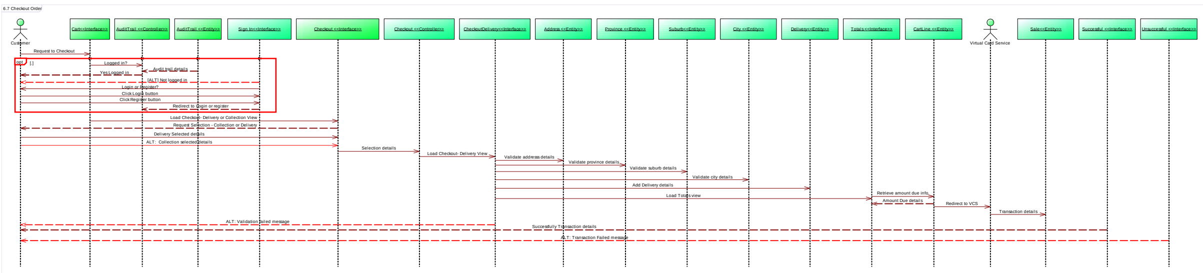


Figure 61 6.7. Checkout Order Sequence Diagram

2.4.7. Reports Subsystem

Generate Fast Selling Product List

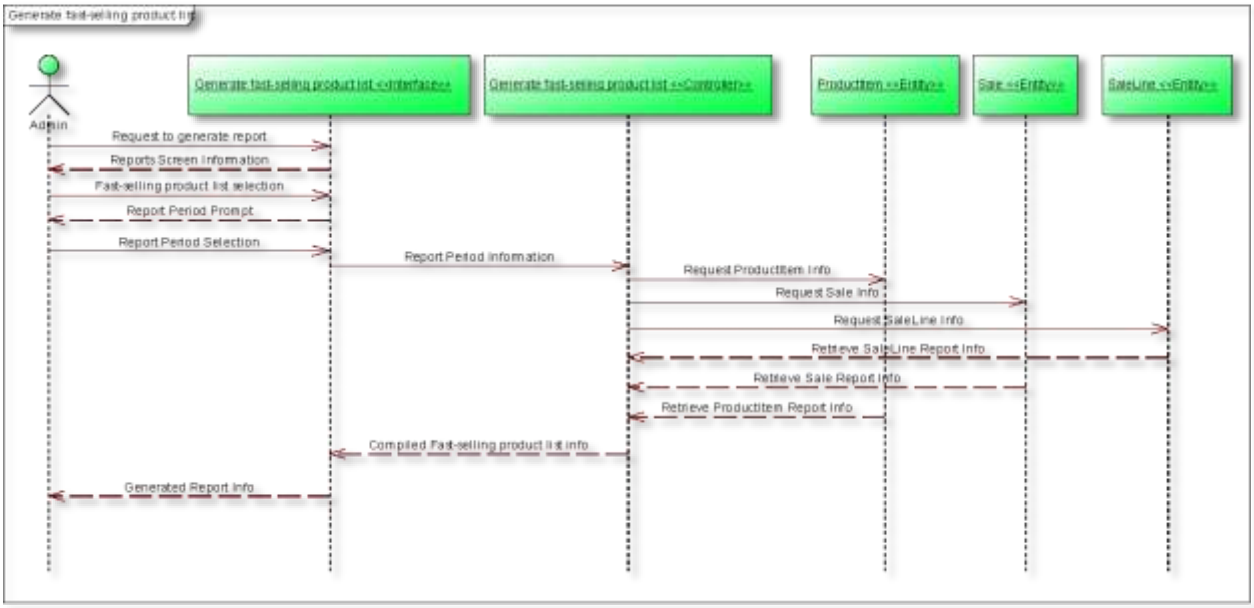


Figure 62 7.1. Generate Fast Selling Product List Sequence Diagram

Generate Slow Selling Product List

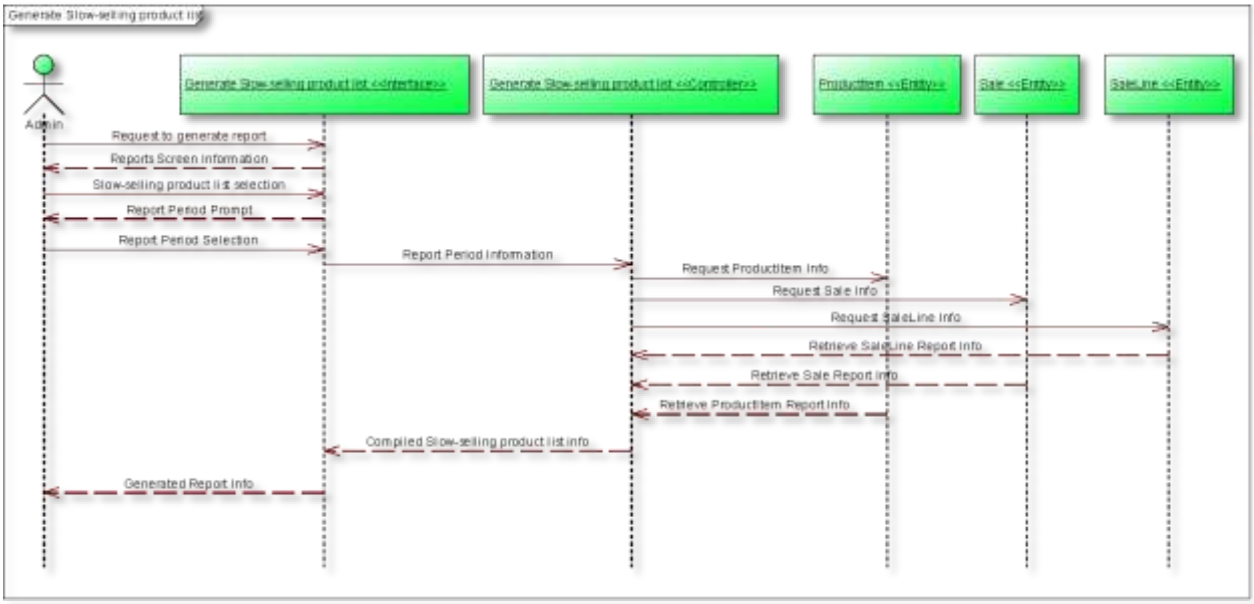


Figure 63 7.2. Generate Slow Selling Product List Sequence Diagram

Generate Most Frequent Buyers List

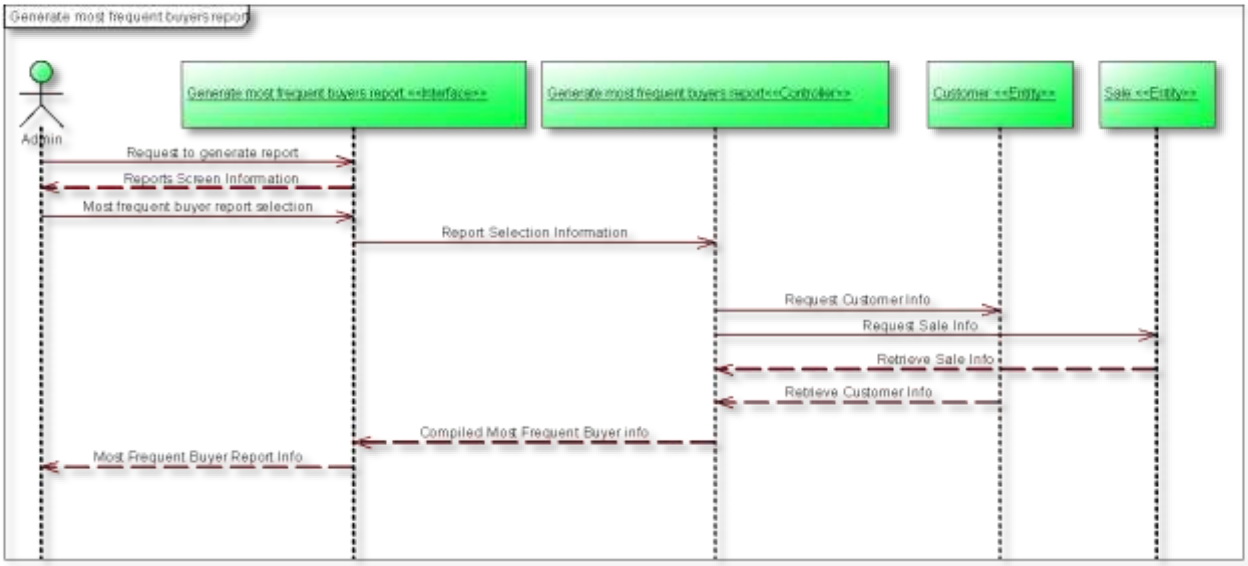


Figure 64 7.3. Generate Most Frequent Buyers List Sequence Diagram

Generate Most Popular Location Graph

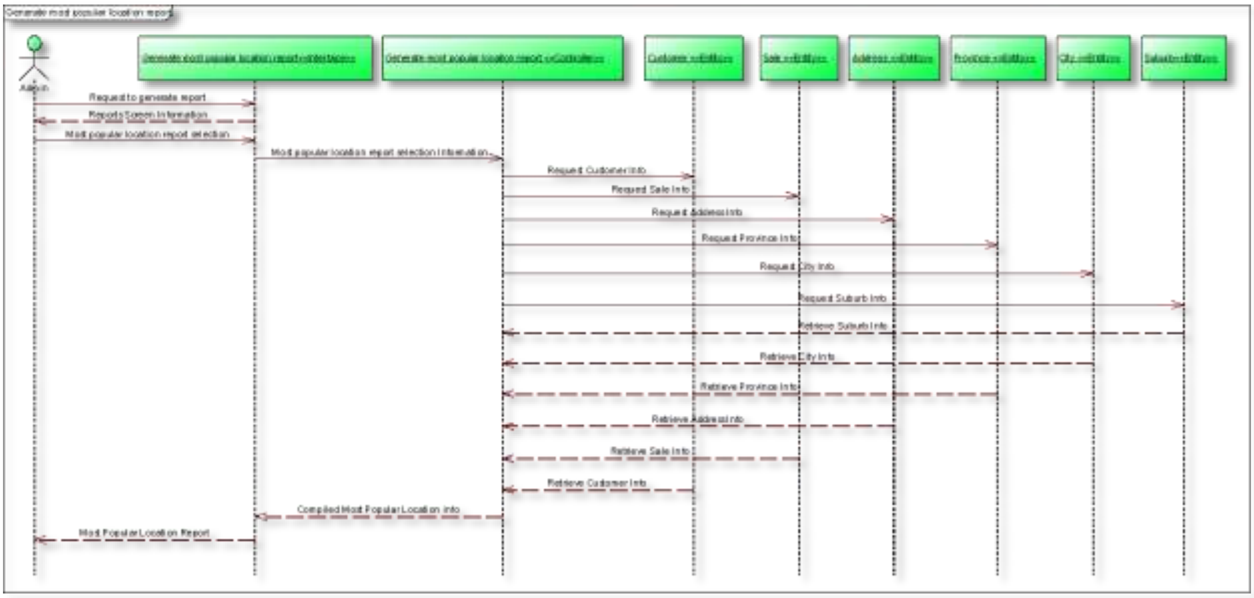


Figure 65 7.4. Generate Most Popular Location Graph Sequence Diagram

Generate Weekly Sale Orders List

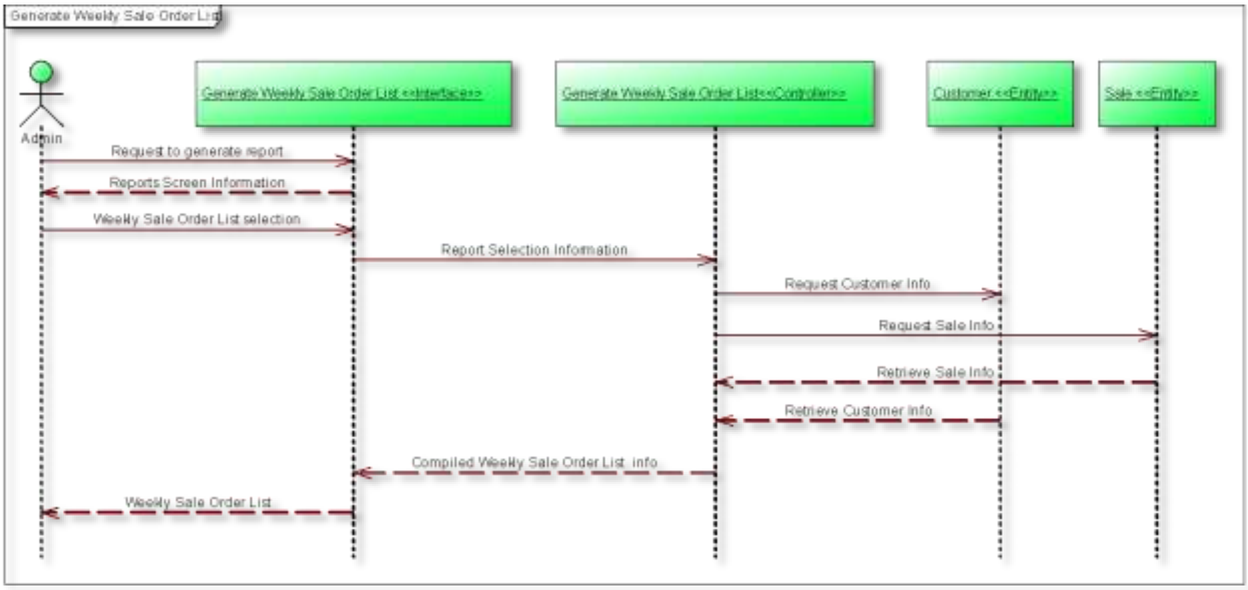


Figure 66 7.5. Generate Weekly Sale Order List Sequence Diagram

Generate Monthly Sale Orders List

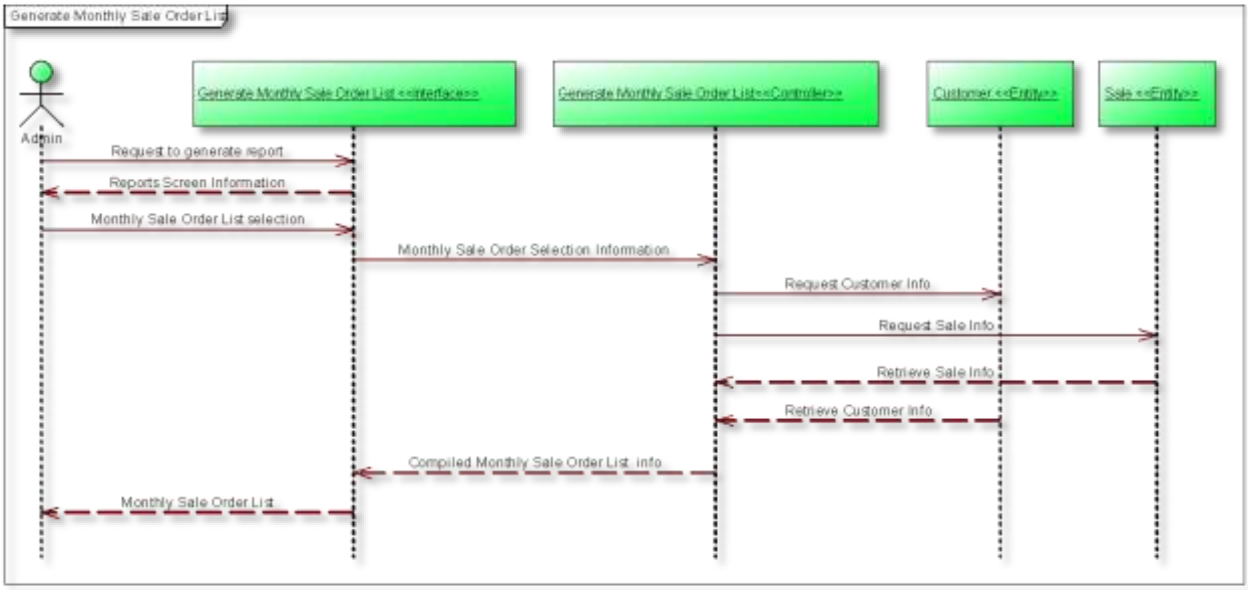


Figure 67 7.6. Generate Monthly Sale Orders List Sequence Diagram

Generate Stock-Level Report

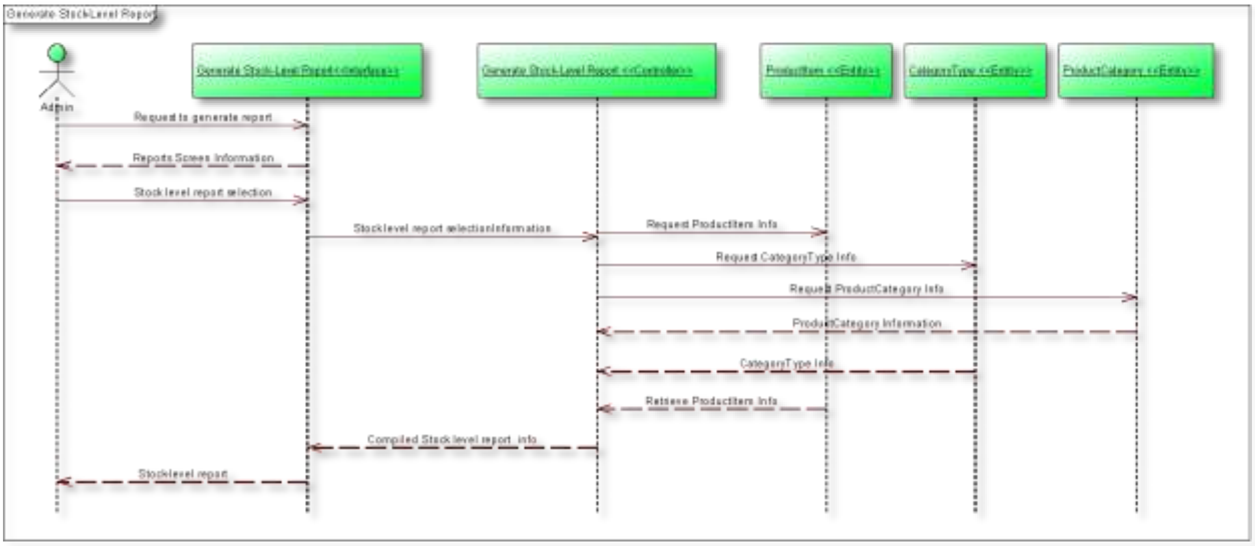


Figure 68 7.7. Generate Stock-level Report Sequence Diagram

Packing Report

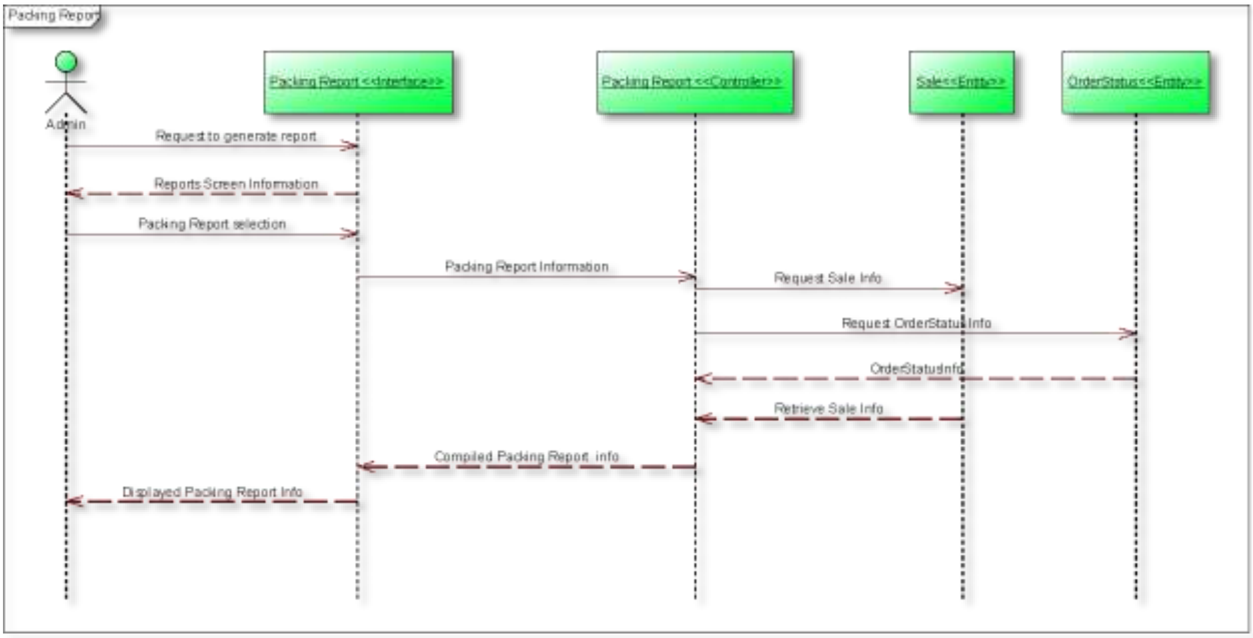


Figure 69 7.8. Packing Report Sequence Diagram

Delivery Report

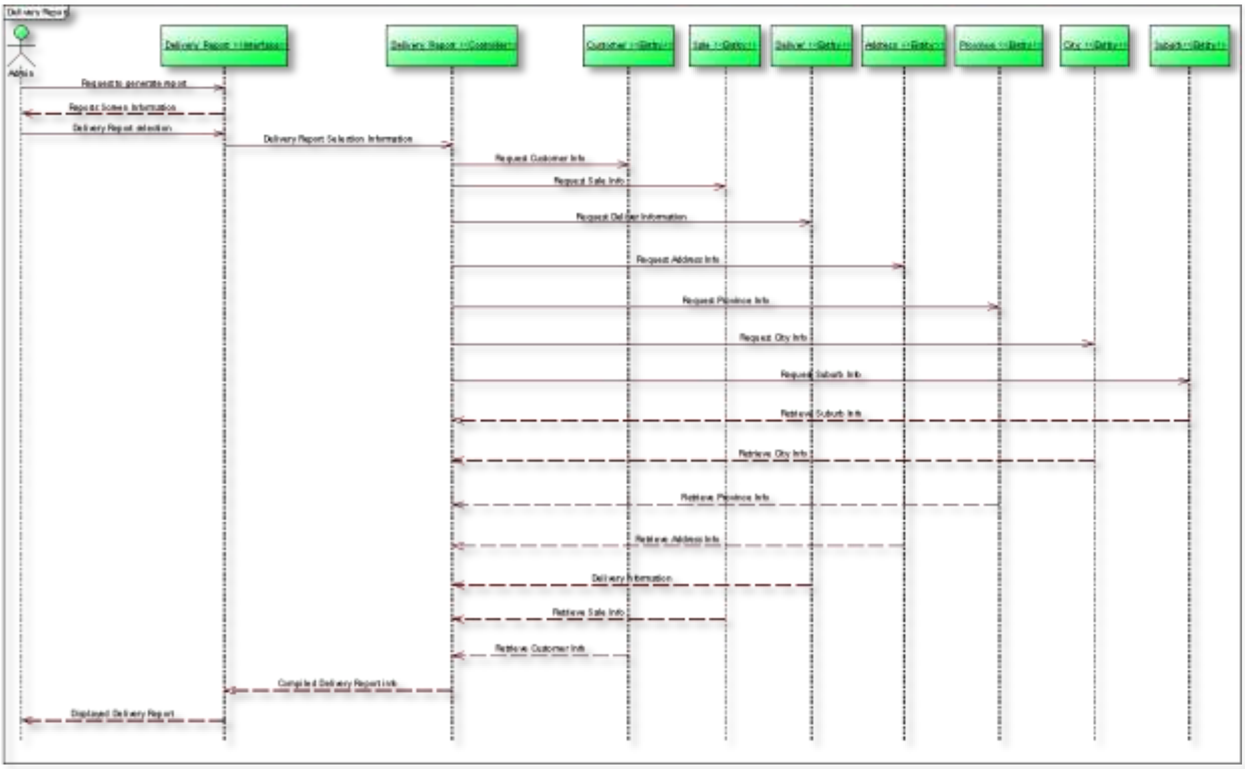


Figure 70 7.9. Delivery Report Sequence Diagram

2.4.8. Delivery Subsystem

8.1. Search Delivery Details

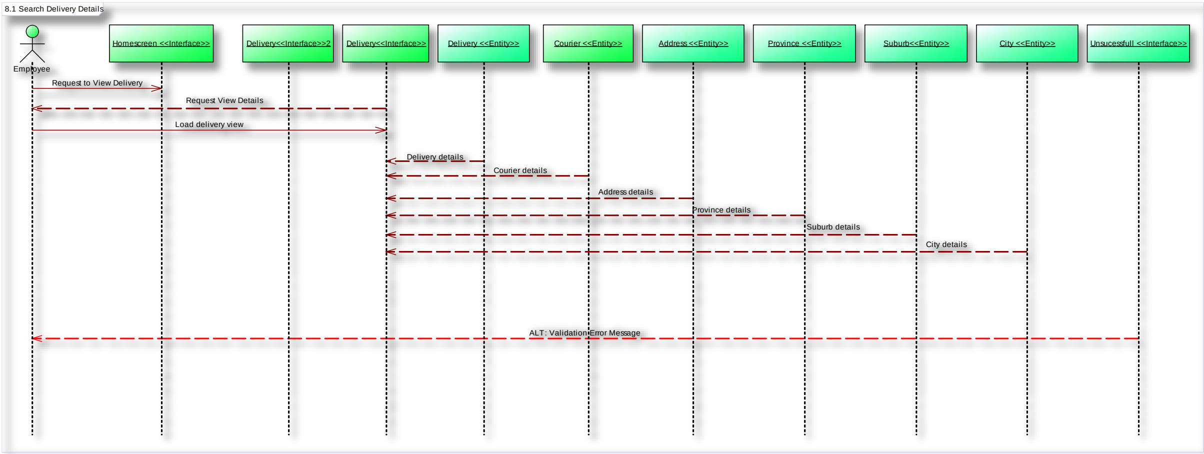


Figure 71 8.1. Search Delivery Details Sequence Diagram

8.2. Accept Delivery

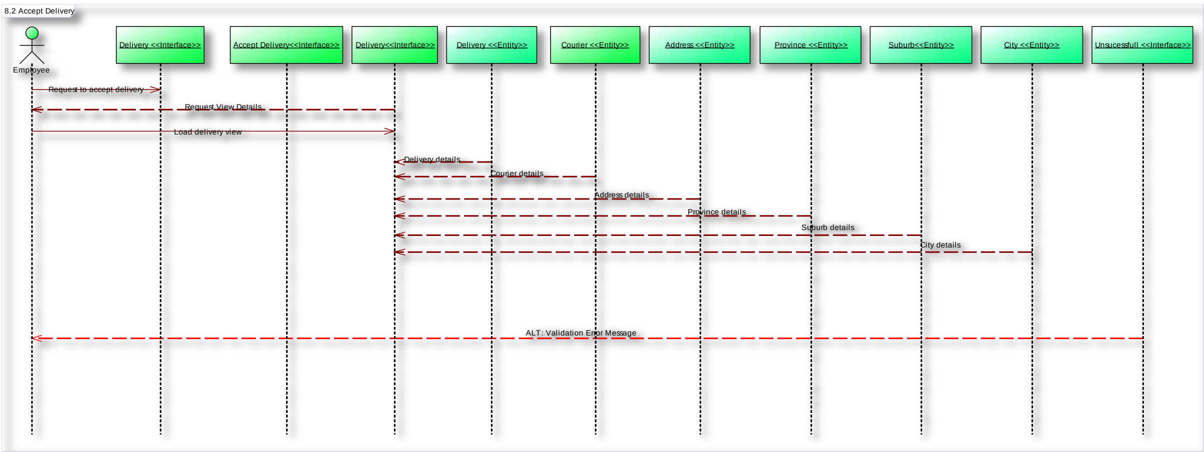


Figure 72 8.2. Accept Delivery Sequence Diagram

Add Courier

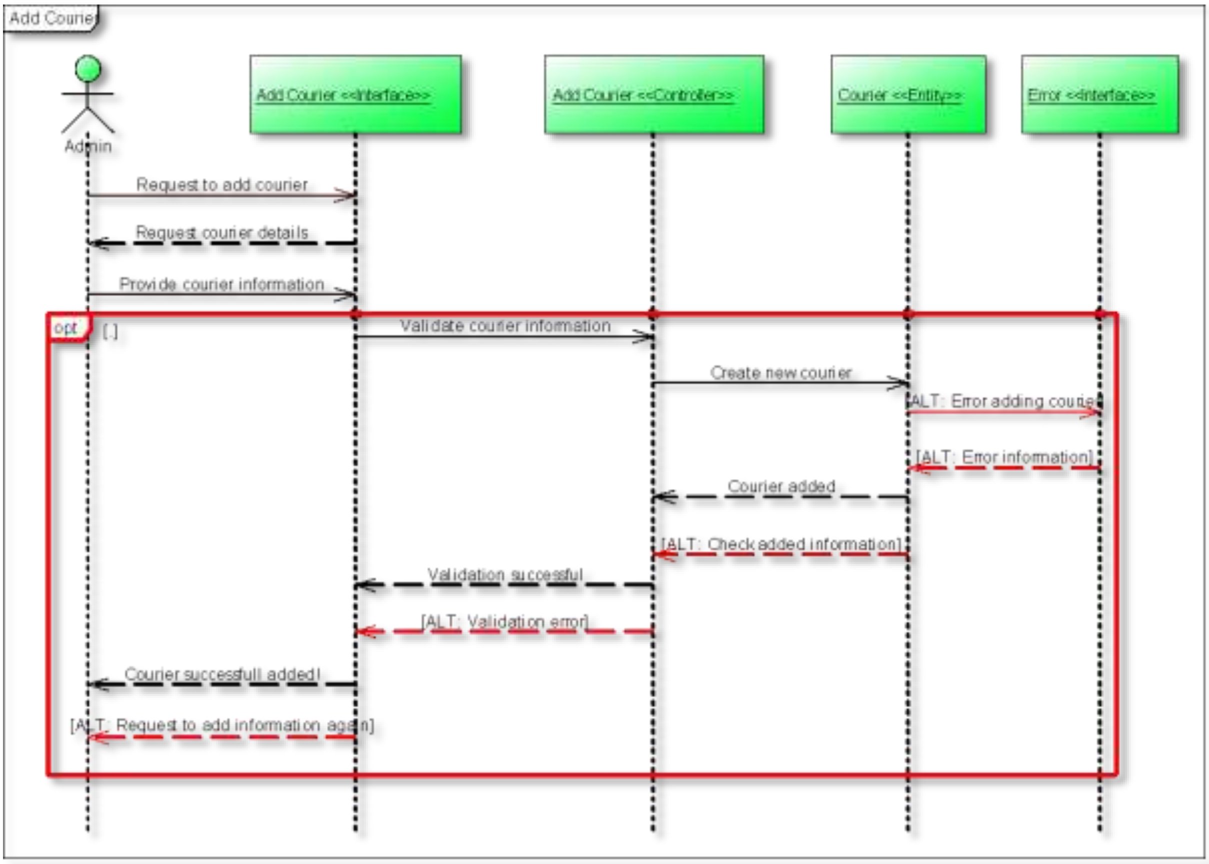


Figure 73 8.3. Add Courier Sequence Diagram

Search Courier

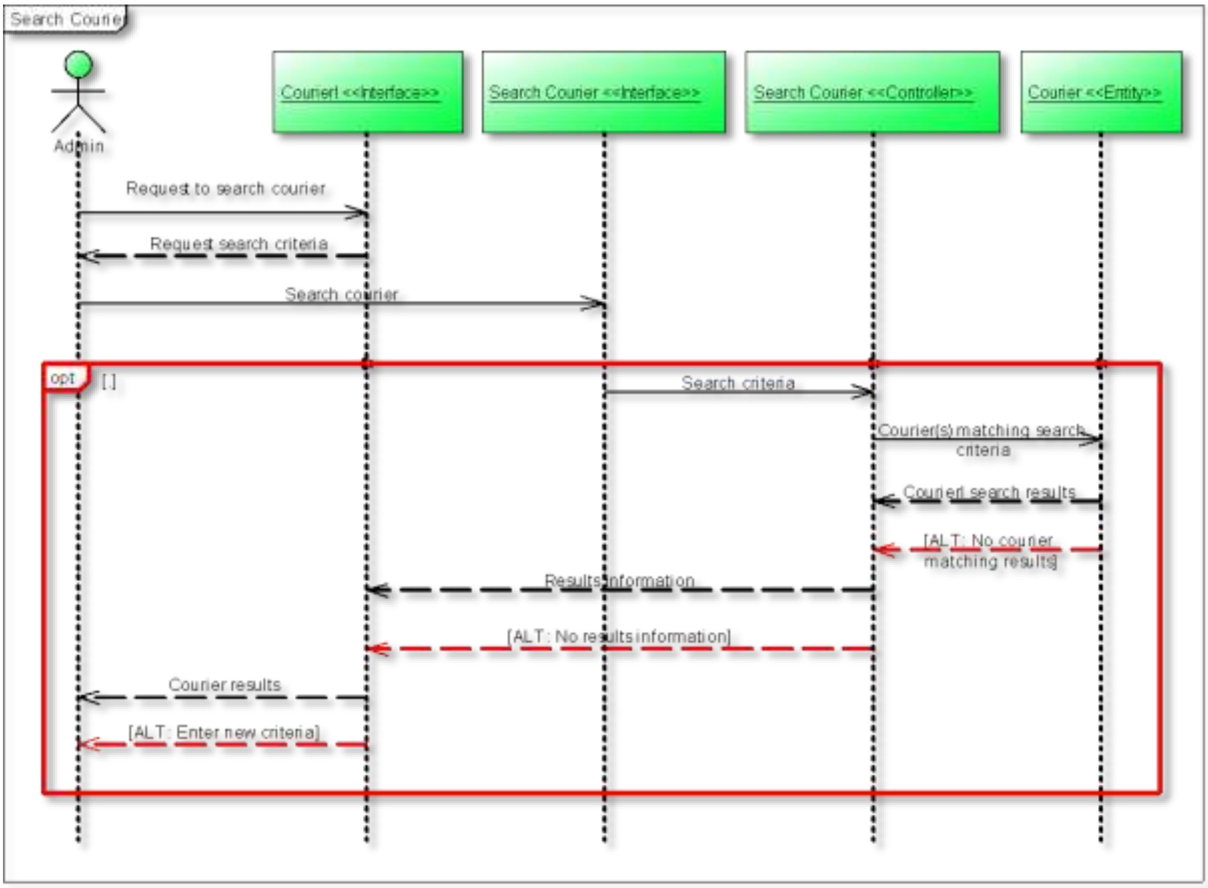


Figure 74 8.4. Search Courier Sequence Diagram

Update courier

Figure 75 8.5. Update Courier Sequence Diagram

Delete Courier

<<insert diagram here>>

Figure 76 8.6. Delete Courier Sequence Diagram

2.4.9. Products Subsystem

Add Product Category

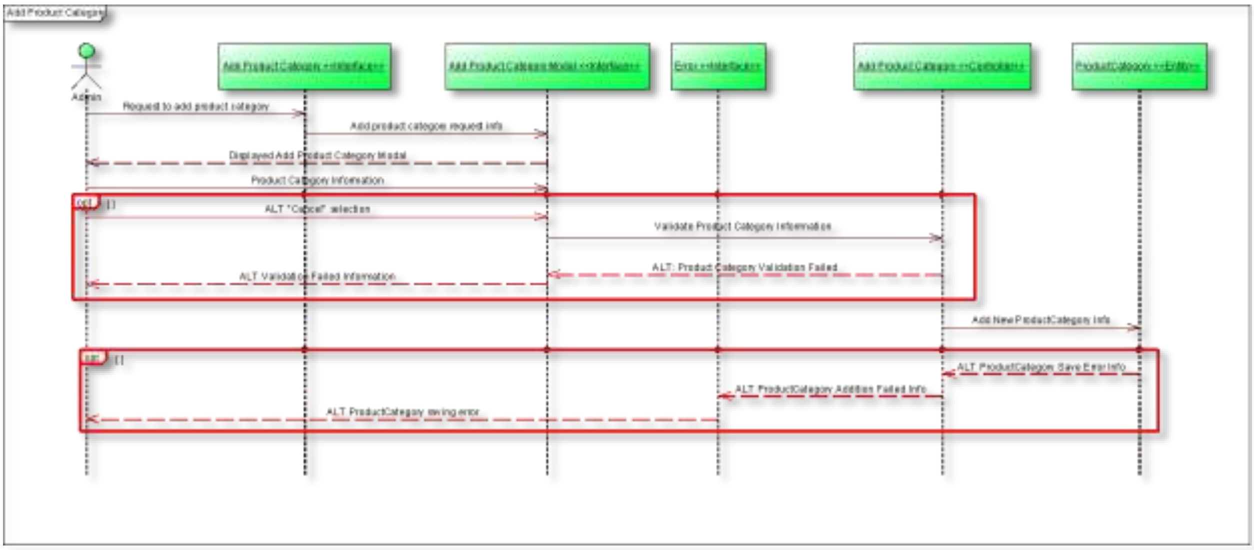


Figure 77 9.1. Add Product Category Sequence Diagram

Search Product Category

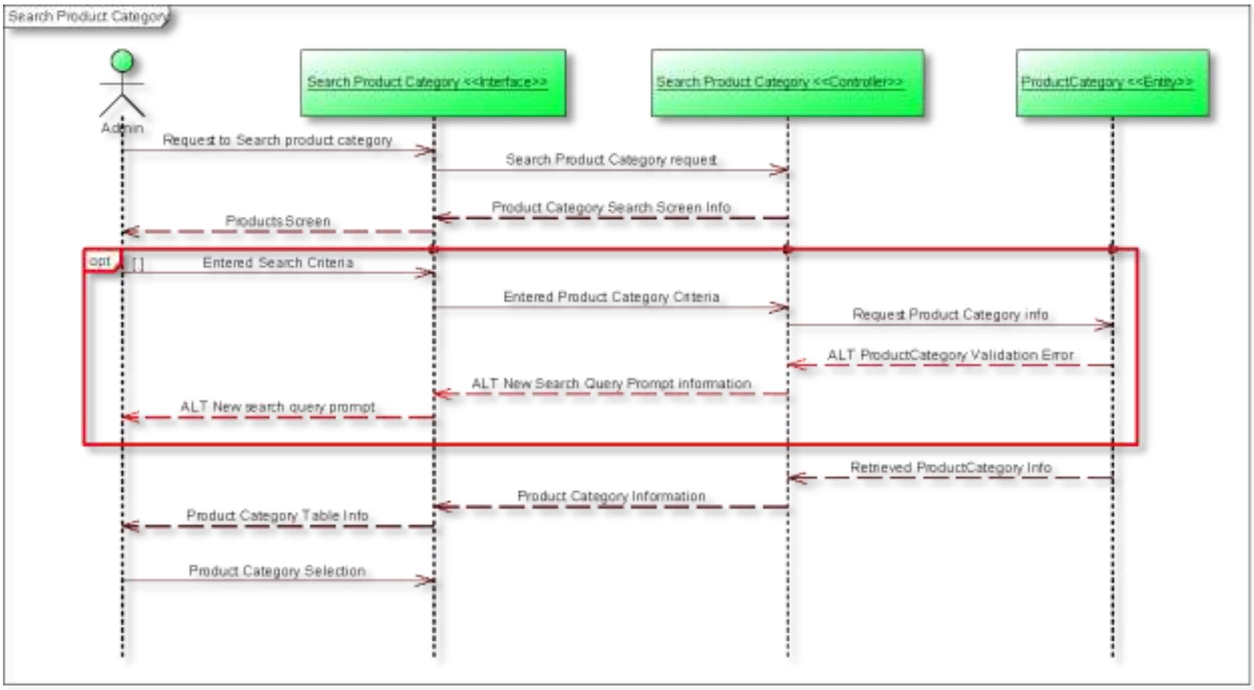


Figure 78 9.2. Search Product Category Sequence Diagram

Update Product Category

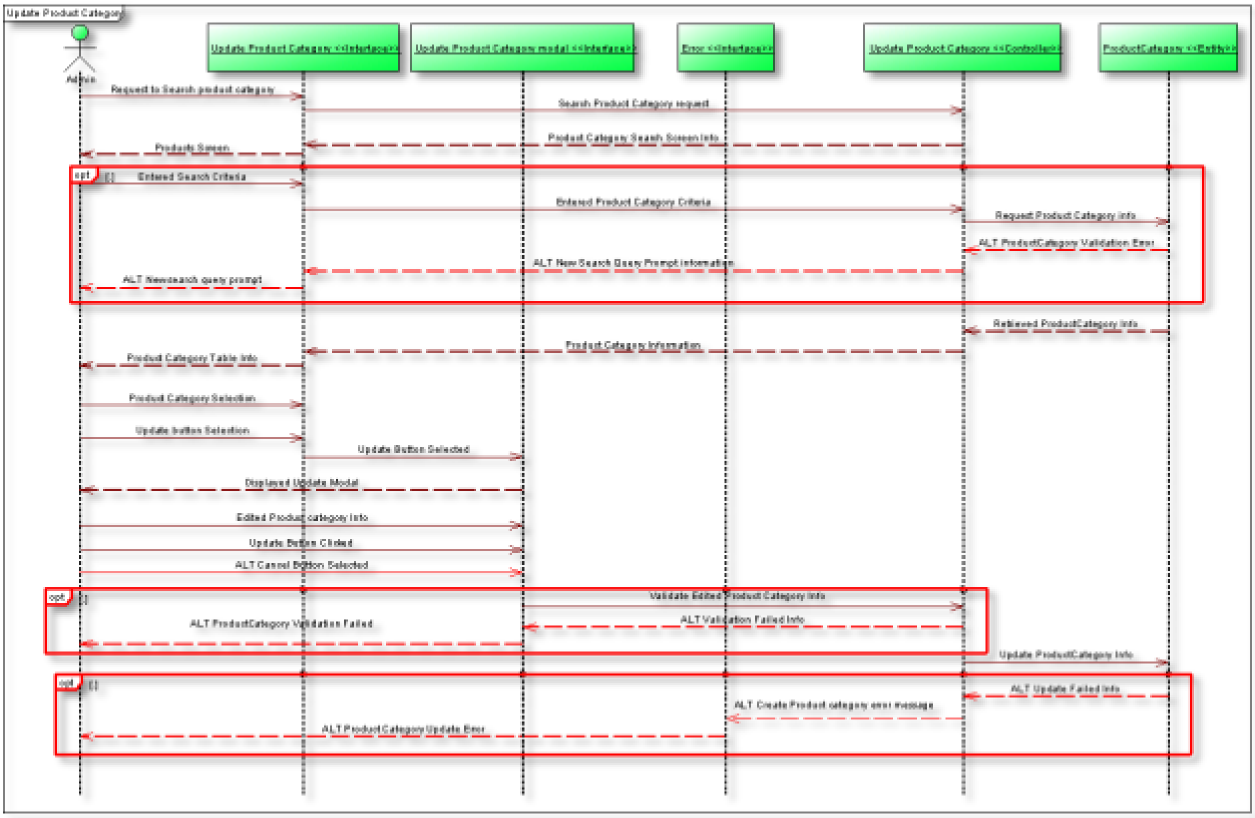


Figure 79 9.3. Update Product Category Sequence Diagram

Delete Product Category

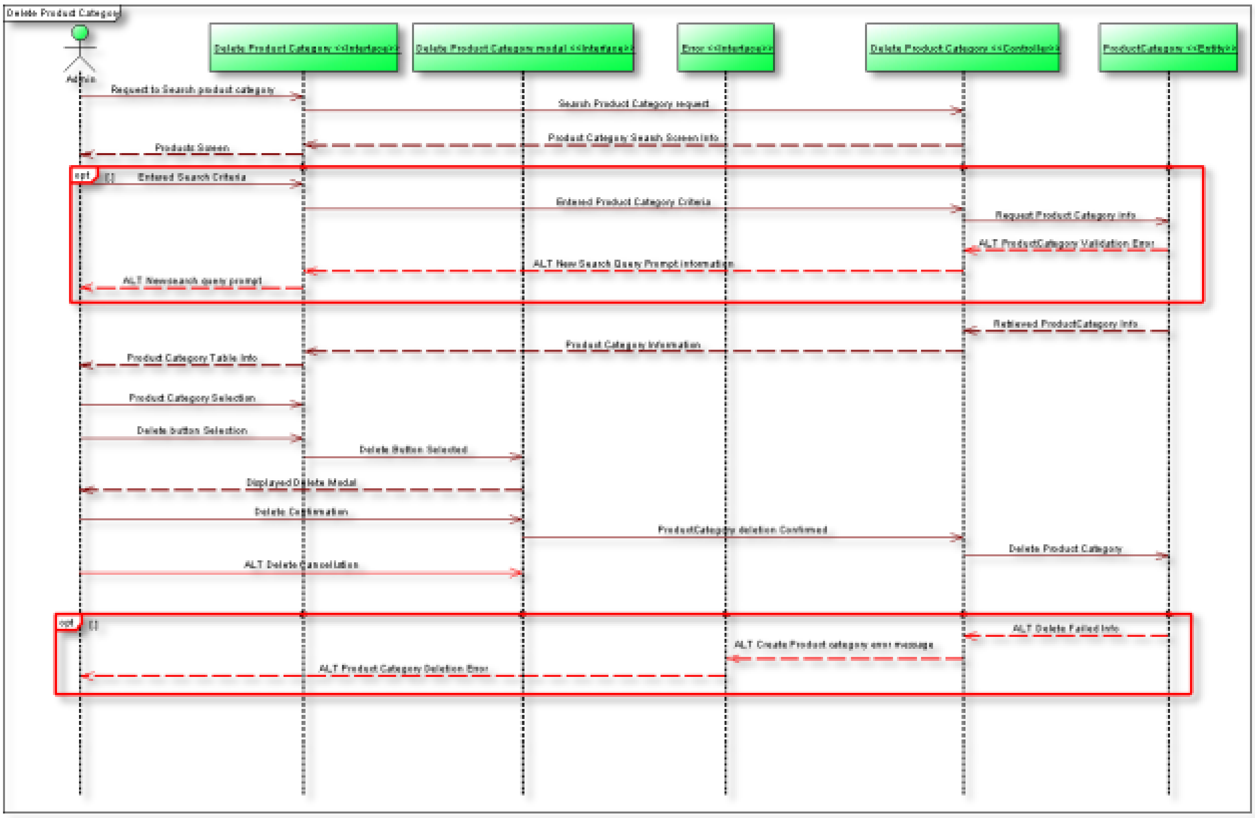


Figure 80 9.4. Delete Product Category Sequence Diagram

Add Category Type

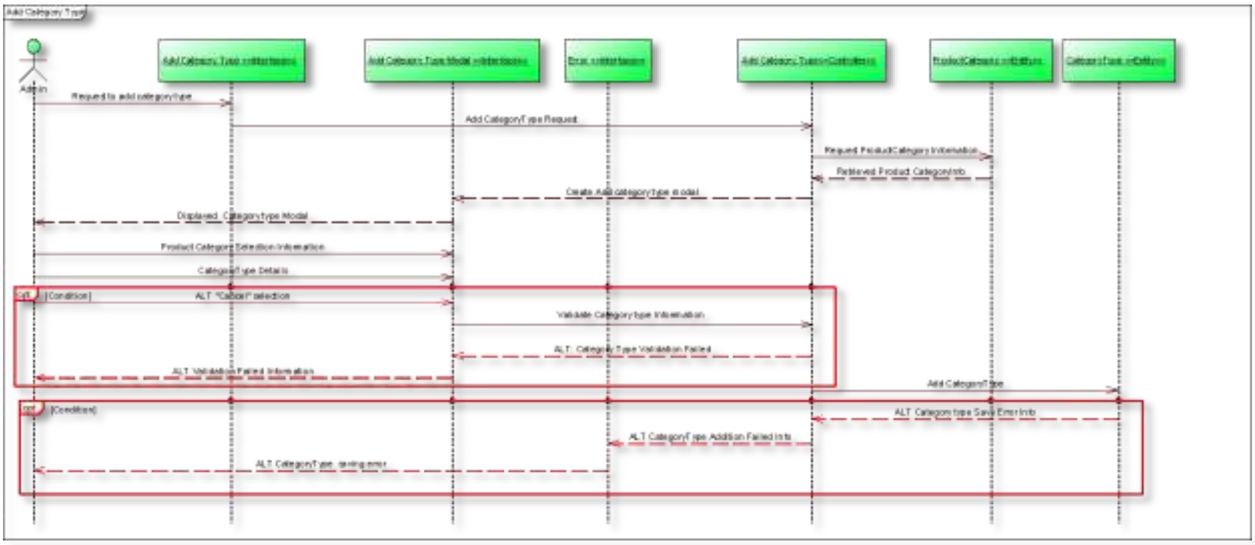


Figure 81 9.5. Add Category Type Sequence Diagram

Search Category Type

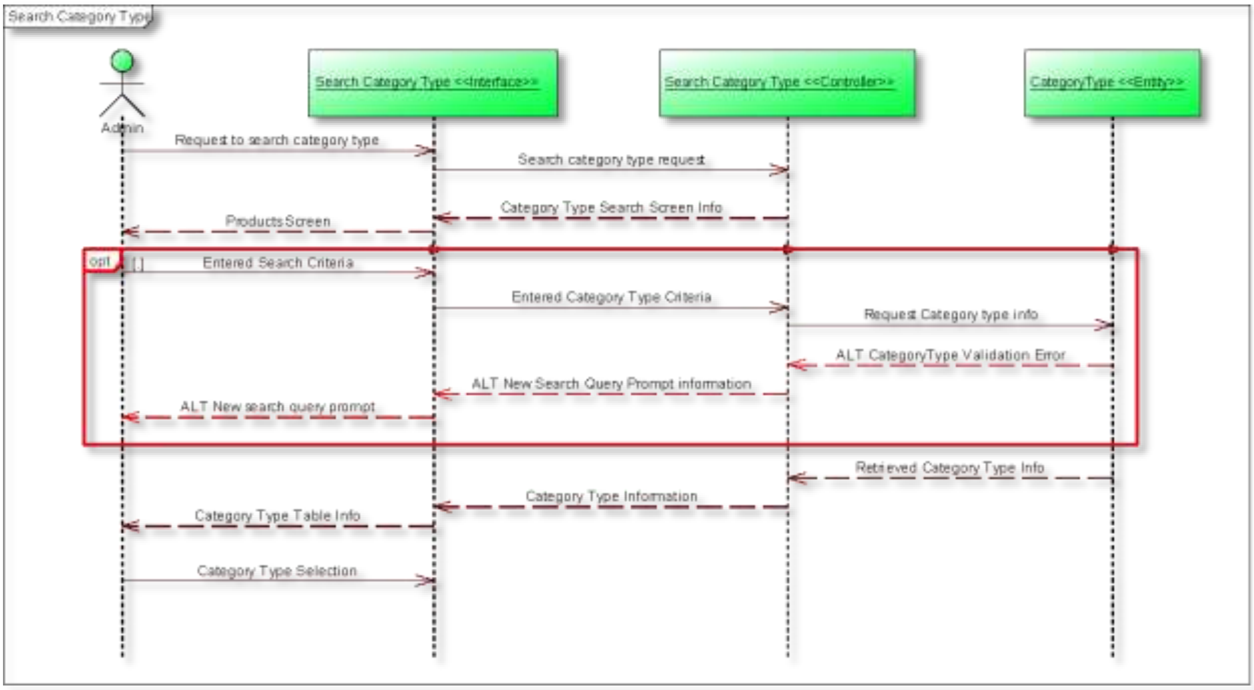


Figure 82 9.6. Search Category Type Sequence Diagram

Update Category Type

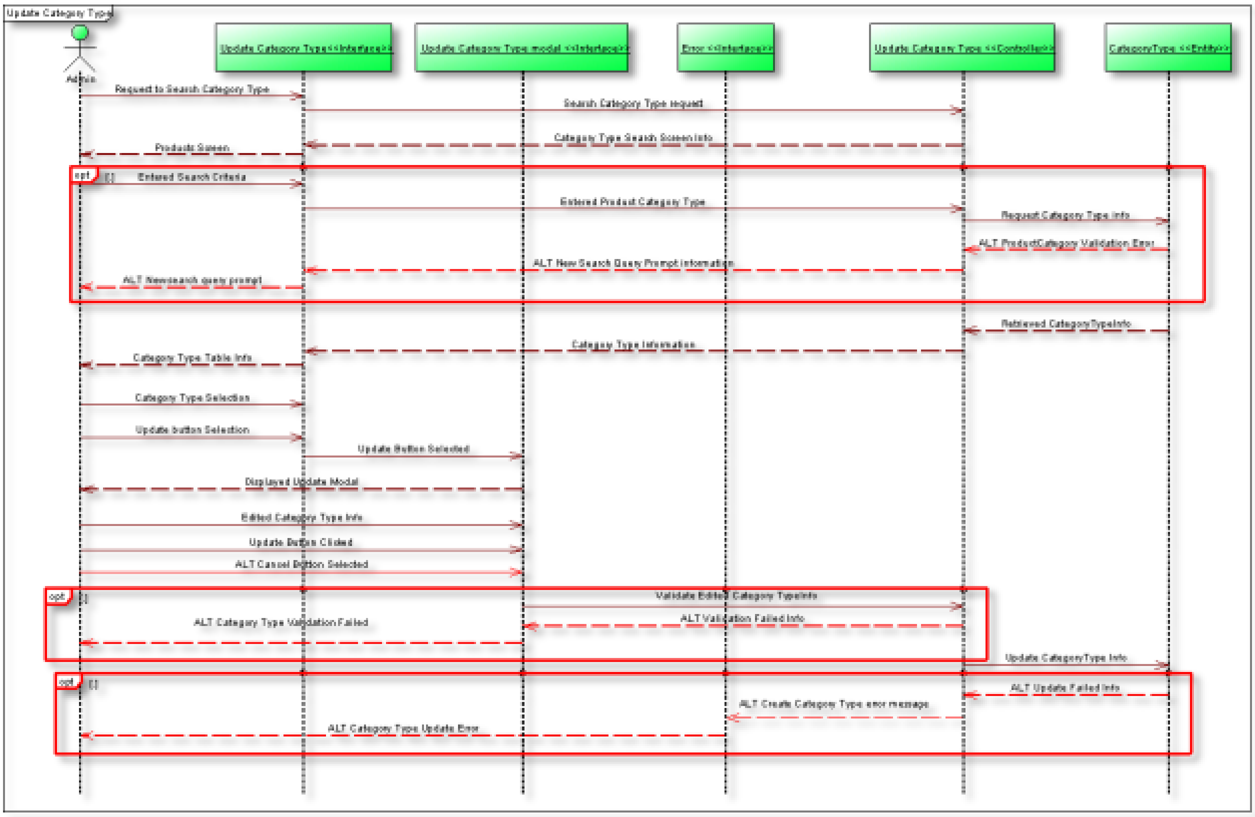


Figure 83 9.7. Update Category Type Sequence Diagram

Delete Category Type

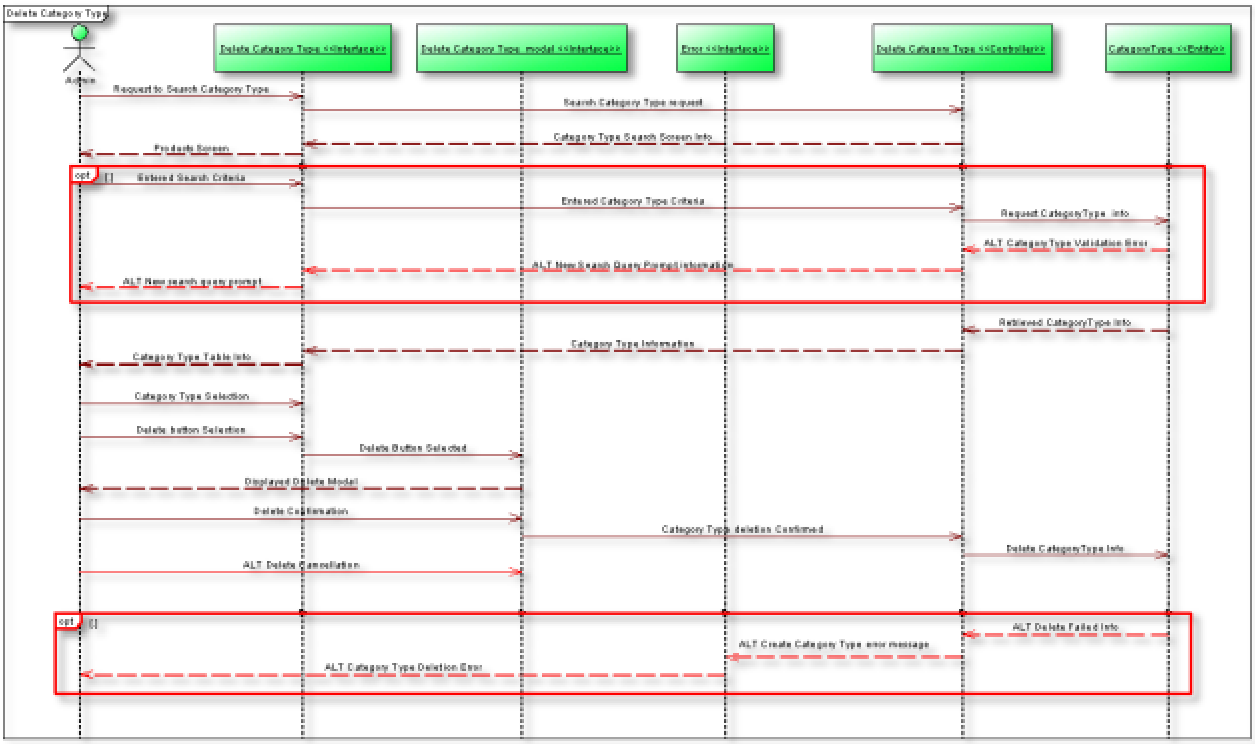


Figure 84 9.8. Delete Category Type Sequence Diagram

Add Product Item

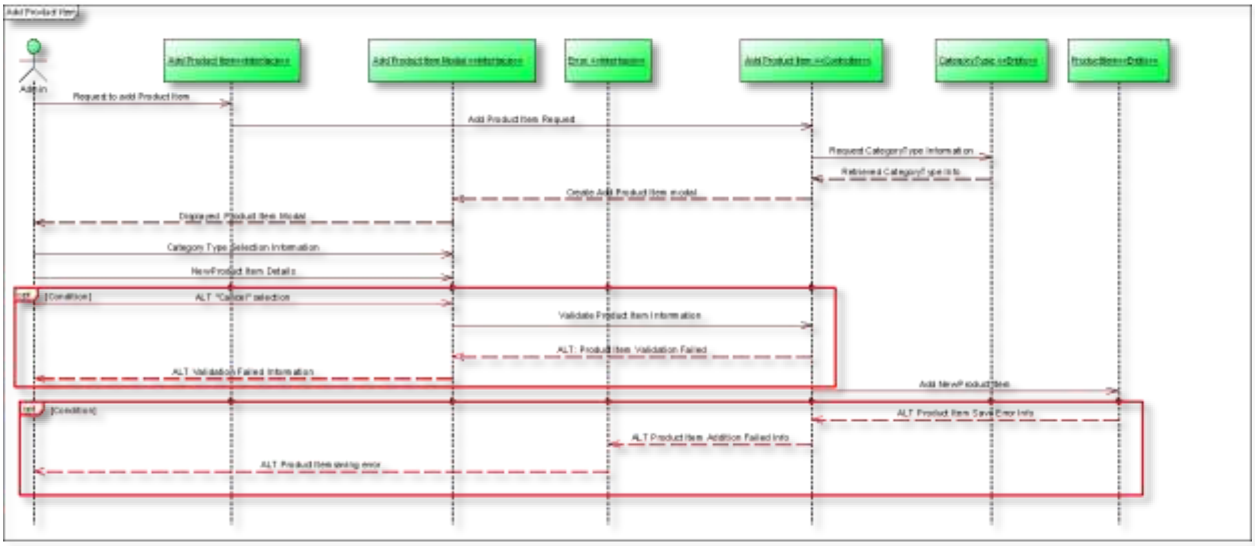


Figure 85 9.9. Add Product Item Sequence Diagram

Search Product Item

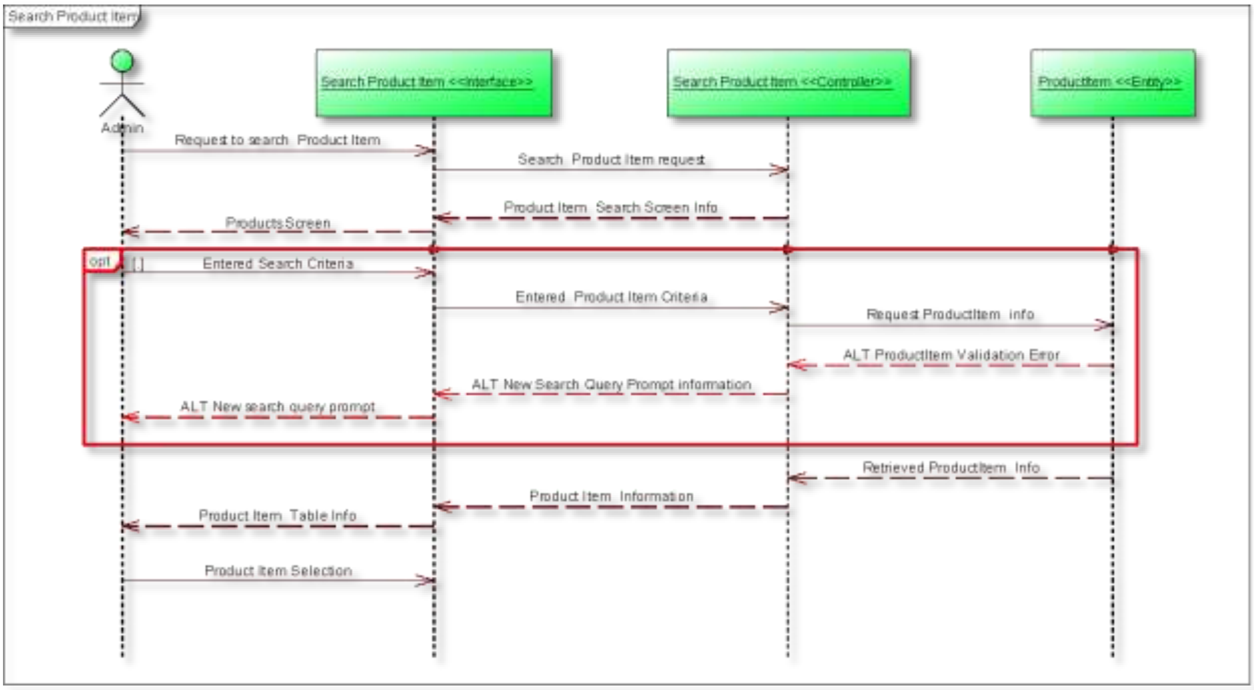


Figure 86 9.10. Search Product Item Sequence Diagram

Update Product Item

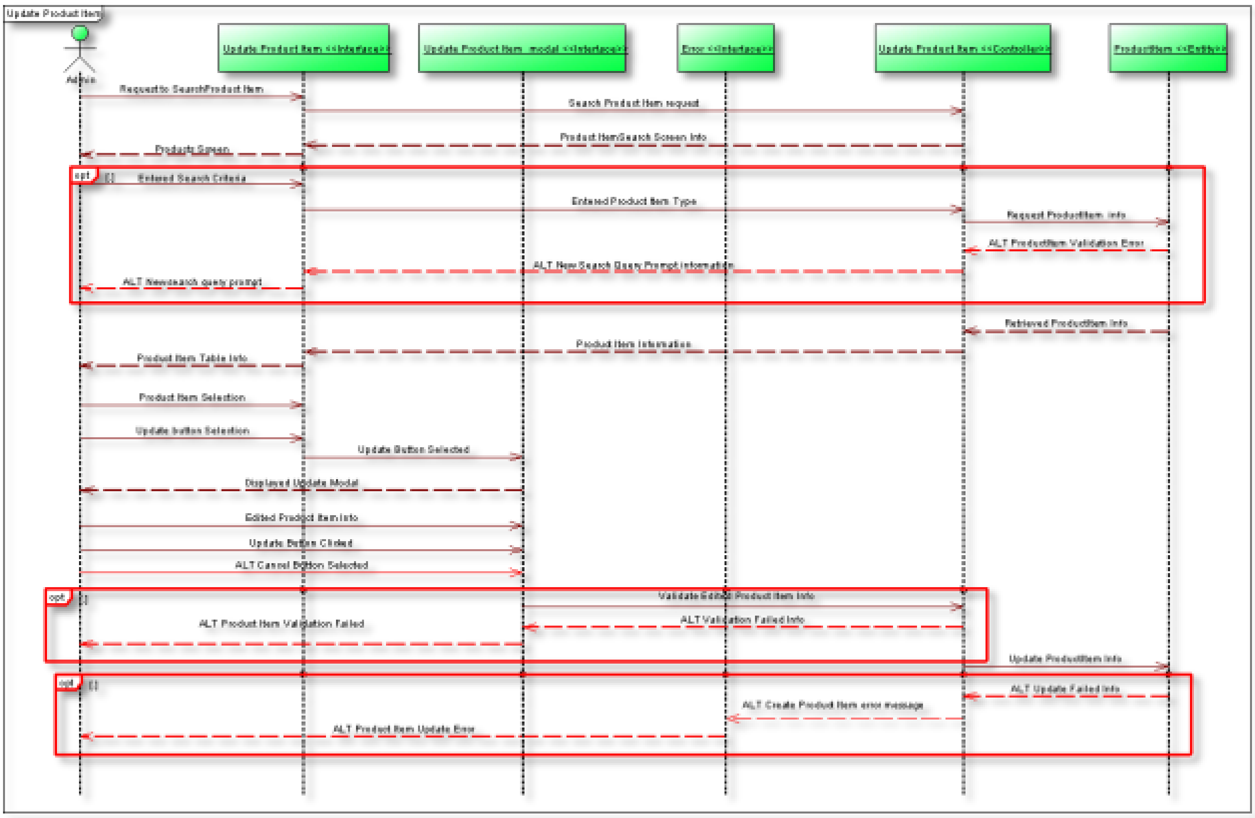


Figure 87 9.11. Update Product Item Sequence Diagram

Delete Product Item

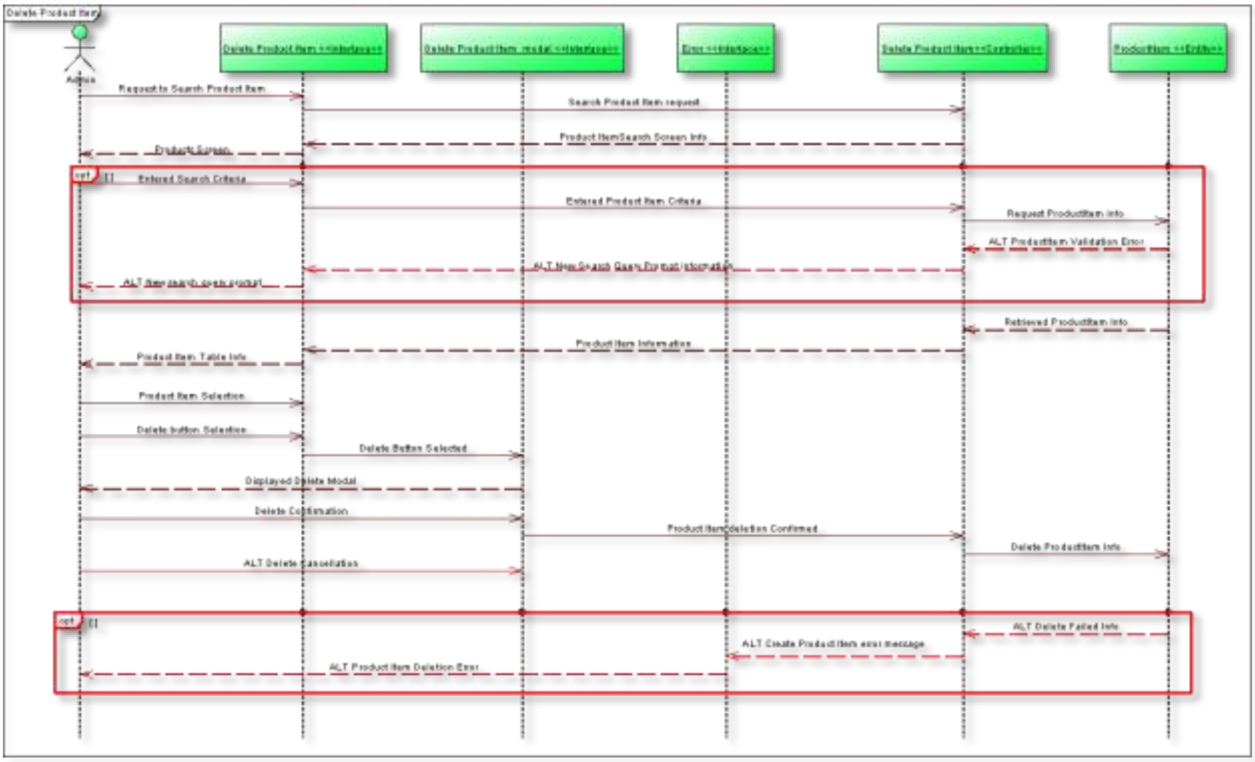


Figure 88 9.12. Delete Product Item Sequence Diagram

2.4.10. Specials Subsystem

Add Special

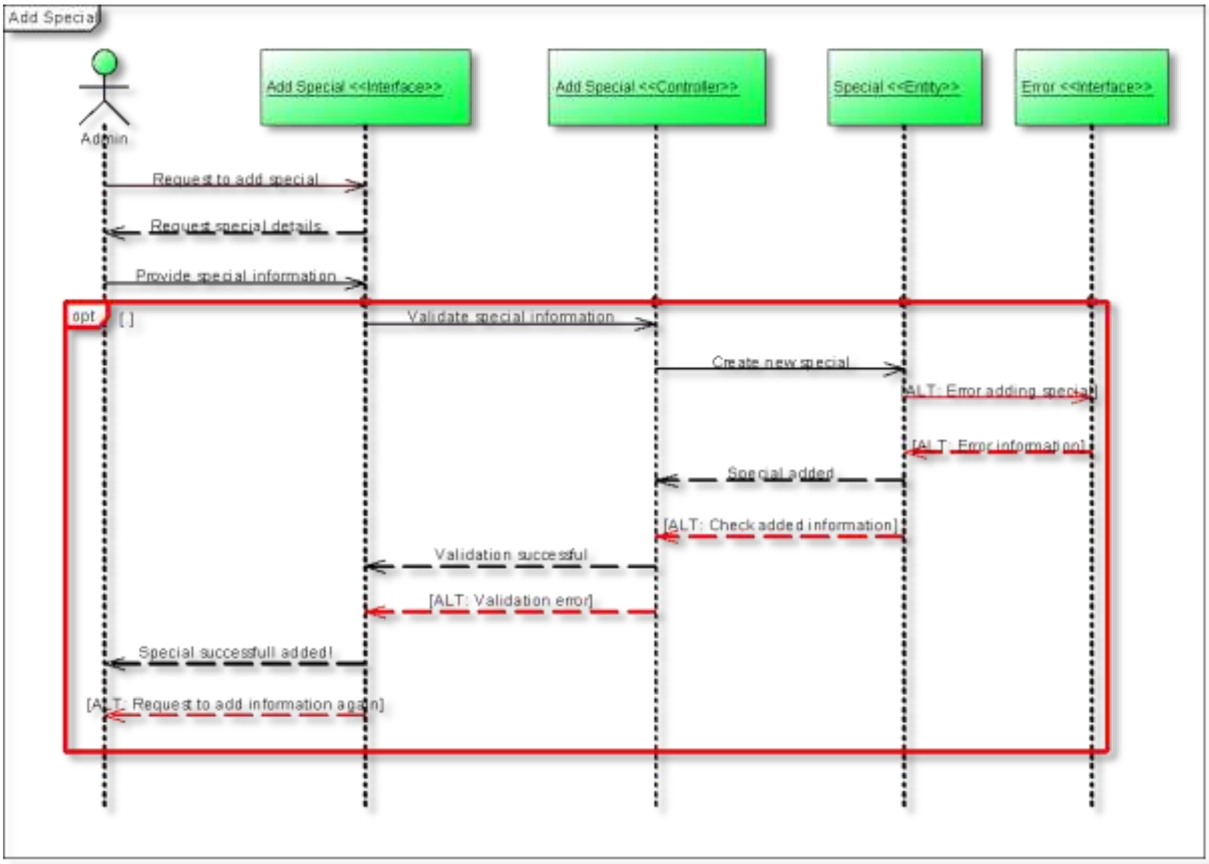


Figure 89 10.1. Add Special Sequence Diagram

Search Special

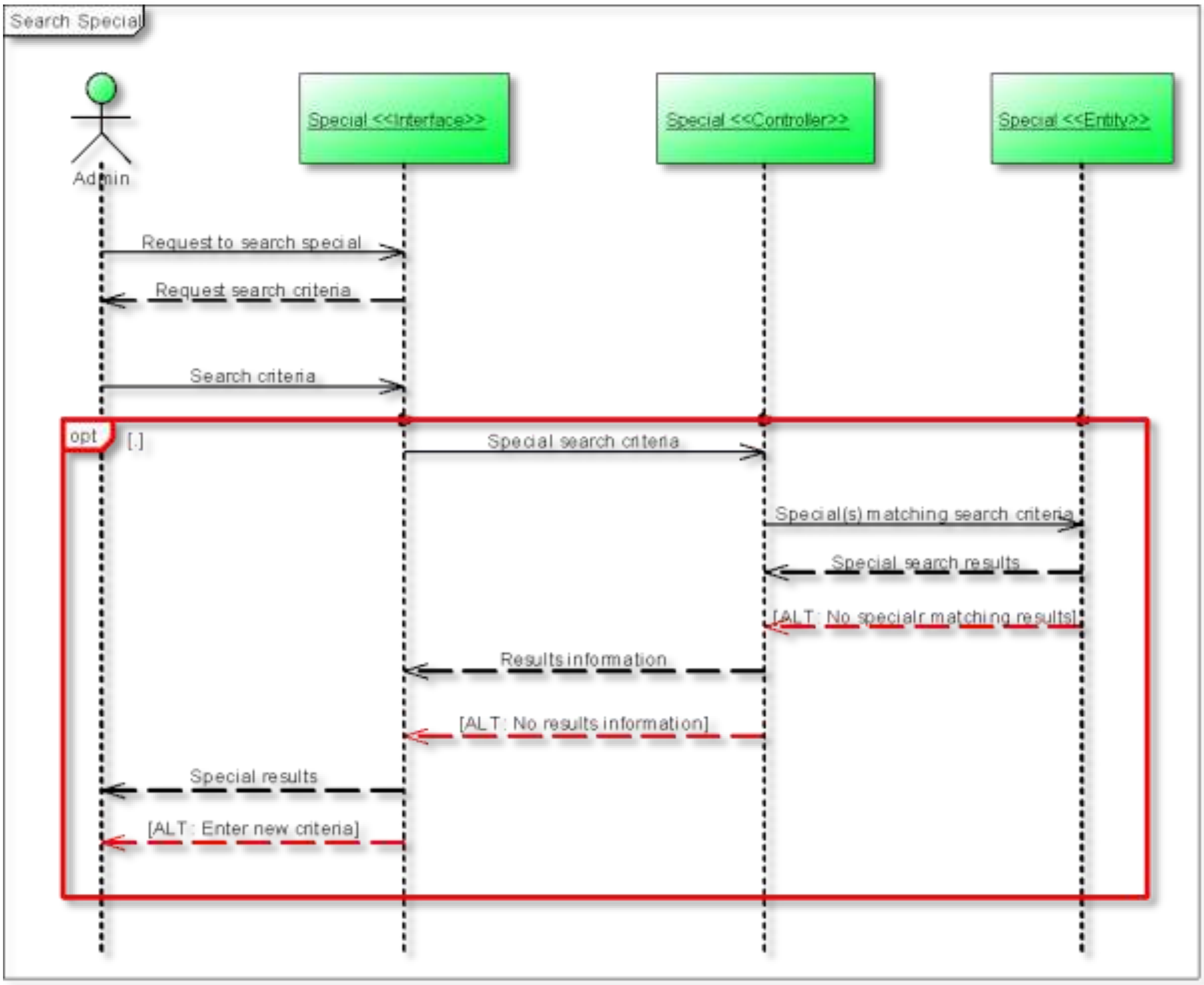


Figure 90 10.2. Search Special Sequence Diagram

Update Special

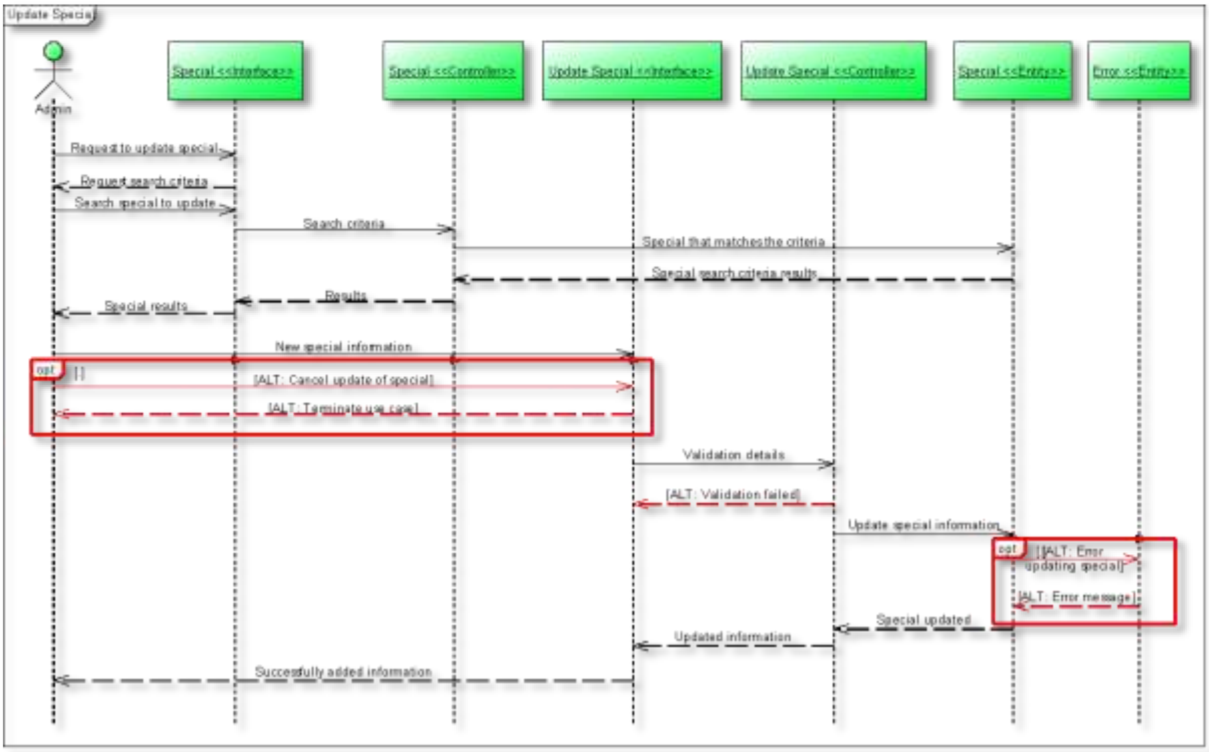


Figure 91 10.3. Update Special Sequence Diagram

Delete Special

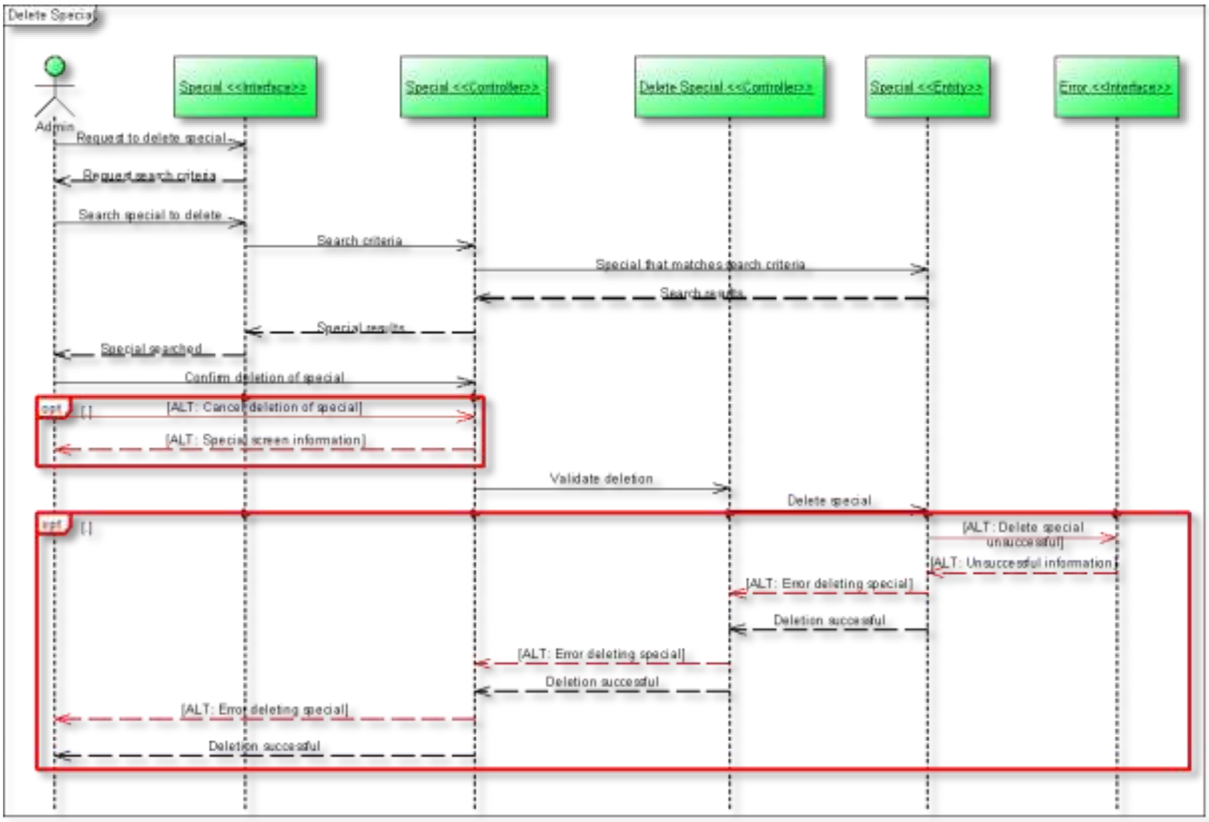


Figure 92 10.4. Delete Special Sequence Diagram

Send Promotional Emails

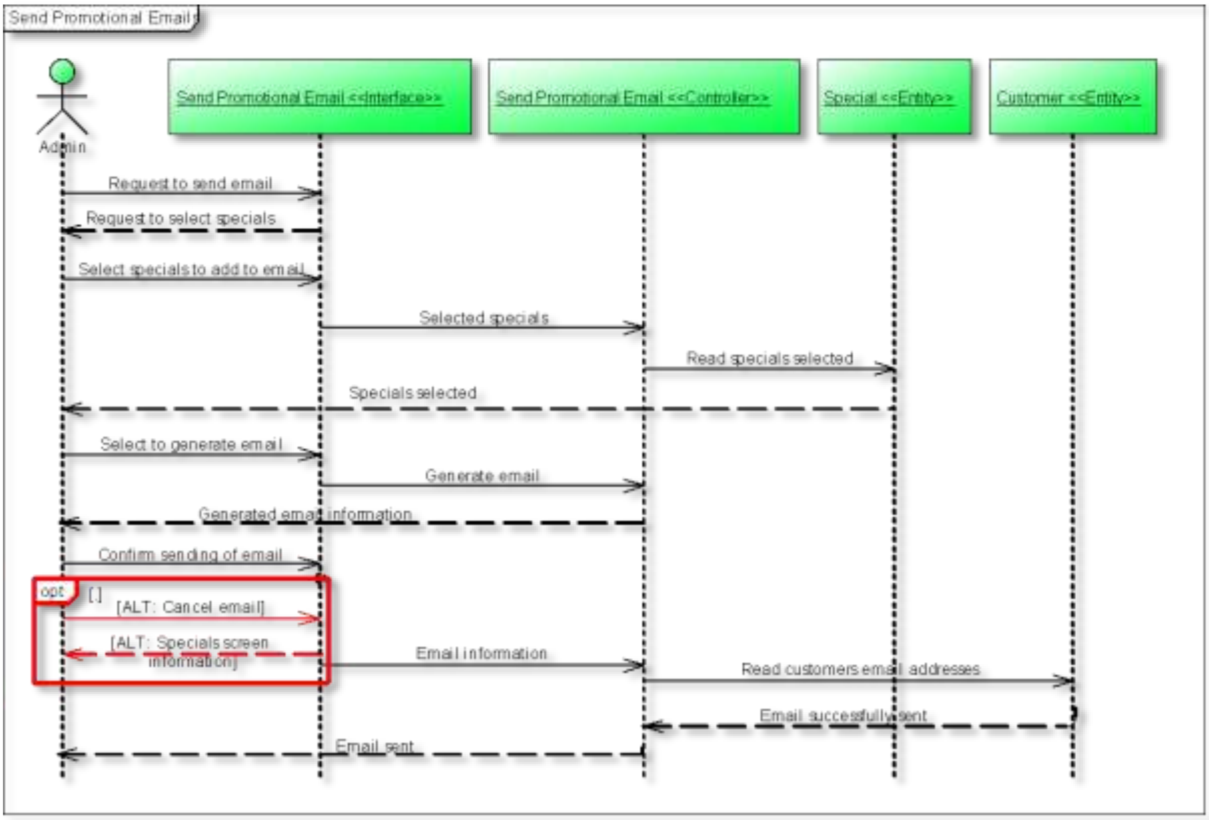


Figure 93 10.5. Send Promotional Emails Sequence Diagram

2.5. PSEUDO CODE

2.5.1. LOGIN SUBSYSTEM

3. Subsystem:	1. Login
Use Case:	1.1 Login
Language:	C#
Pseudo Code:	**Step 1: Request to login [Angular]** ON CLICK Login button **THEN** **Step 2: Display Login screen [Angular]** Display Login screen. **Step 3: Login [Angular]** ON CLICK Login button. **THEN** **Step 4: Validate login details** SQL SELECT Username and Password From User entity WHERE Username == User input && Password == User Input If (User input == Valid) Generate AuditTrail_ID **THEN** SQL ADD AuditTrail_ID INTO User entity

	[ALT Step 4] :Display validation error message [Angular] Else (User input == NotValid) Display error message **“**The entered login credentials are incorrect. Please try again.” **THEN** **Step 5: Add the login to the AuditTrail entity** SQL ADD Logged in users’ Username, Date and Time In the AuditTrail table. **THEN** System logs the user in to the system. **End use case.**

Subsystem:	1. Login
Use Case:	1.2 Forgot Password
Language:	C#
Pseudo Code:	**Step 1: User has forgotten their password.** **THEN** **Step 2: Display Login Screen [Angular]** Display Login screen. **Step 3: User clicks on the "Forgot Password" link** **ON CLICK** "Forgot Password" link **THEN** **Step 4: Display Forgot Password screen [Angular]** Display Forgot Password screen **THEN** **Step 5: User adds their email information** Input email address **ON CLICK** "Submit" button. **THEN** **Step 6: Validate reset details** SQL SELECT Email Address FROM Customer/Admin Entity WHERE Email Address == User Input If(User Input == Valid) **ALT Step 6]: Display invalid email message [Angular]**

	Else if(User Input == NotValid) Display **“**Invalid email address. Please try again.” **THEN** **Step 7: Send an email to the email the user entered.** **THEN** **Step 8: Notify the user that an email has been sent to them.** Display message **“**A password reset link has been sent to your email.” **End use case.**

Subsystem:	1. Login
Use Case:	1.3 Reset Password
Language:	C#
Pseudo Code:	**Step 1: User clicks on the link sent in the email** **ON CLICK** “Reset Password” link **THEN** **Step 2: Display Reset Password screen** Display Reset Password screen **THEN** **Step 3: User enters new password and confirmation** Input new password and confirmations **ONCLICK** “Submit” button **THEN** **Step 4: Validate password** SQL SELECT Password FROM User entity WHERE Password == User If(User Input == Valid) **[ALT Step 4]: Display validation error message [Angular]** Else if (User Input) == NotValid) Display error message “Password does not match with confirm password” THEN

	Step 5: Display Login screen. Display login screen. **End use case.**

Subsystem:	1. Login
Use Case:	1.4 Logout
Language:	C#
Pseudo Code:	**Step 1:** **Step 1: The user requests to Logout of the system [Angular]** **ON Click “Logout” option** **THEN** **Step 2: Display logout confirmation pop-up [Angular]** Display pop-up with the following message: “Are you sure you want to logout?” **THEN** **Step 3: User confirms log out [Angular]** **ON CLICK** “Yes” button. **THEN** **[ALT Step 3]: User cancels logout [Angular]** **ON CLICK** “No” button. **THEN** Take away the confirmation pop-up. **Step 4: User gets logged out of the system.** User is logged out. **THEN** **Step 5: Information is added to the Audit Trail** SQL ADD Logged out users’ Username, Date and Time In the AuditTrail table.

	THEN **Step 6: Take user to the login screen [Angualr]** Display login screen **End use case.**

3.1.1. SUPPLIER SUBSYTEM

3.1.2. USER SUBSYSTEM

Subsystem:	3. User
Use Case:	3.1 Register New Customer
Language:	C#
Pseudo Code:	**Step 1: Customer wants to Register.** **THEN** **Step 2: Customer clicks User Icon (Angular)** ONCLICK User icon on navigation bar **THEN** **Step 3: Display Login Dropdown menu item (Angular)** Display Login dropdown menu item **Step 4: Customer clicks Login dropdown menu item (Angular)** ONCLICK Login dropdown menu item THEN ONCLICK Register Button **THEN** **Step 5: Display Register New Customer Screen (Angular)** Display Register New Customer Screen **THEN** **Step 6: Provide Customer Details (Angular)** Provide Customer Details ONCLICK Next Button **THEN** **Step 7: Validate Details (C#)** If (Input fields == Valid) Continue to step 8 Else If (Input fields == false) Display Validation Error Modal “Entered details are in the incorrect format”

	THEN **Step 8: Display Register New Customer Screen (Angular)** Display Register New Customer Screen **THEN** **Step 9: Provide User Details (Angular)** Provide User Details ONCLICK Register Button **THEN** **Step 10: Validate Details (C#)** If (Input fields Valid == True) Continue to step 11 Else If (Input fields Valid == false) Display Validation Error Modal “Entered details are in the incorrect format” **THEN** **Step 11: Display Confirm Details Modal (Angular)** Display Confirm Details Modal ” Is your captured information correct?” **THEN** **Step 12: Confirm Details (Angular)** Confirm Details ONCLICK Confirm Button [ALT] ONCLICK Cancel Button → Terminate Use Case **THEN** **Step 13: Save Customer Details and Generate CustomerID and UserID (C#)** SQL INSERT Title, Name, Surname, Cell Phone Number, Email Address, VAT Registration Number, Business Name IN CUSTOMER ENTITY

	SQL INSERT Username, Password IN USER ENTITY **THEN** **Step 14: Display Success Modal (Angular)** IF (Registration Successful == True) Display Success Modal “You have successfully registered.” ELSE IF (Registration Successful == False) Display Error Modal “An unexpected error occurred when saving the customer.”

Subsystem:	3. User

Use Case:	3.2 Search Customer
Language:	C#
Pseudo Code:	**Step 1: Admin wants to Search for a Customer** **THEN** **Step 2: Admin clicks User Menu Item (Angular)** ONCLICK User Menu Item THEN ONCLICK Search Customer Dropdown menu Item **THEN** **Step 3: Display Search Customer Screen (Angular)** Display Search Customer Screen **THEN** **Step 4: Admin Provides Search Criteria (Angular)** Provide Search Criteria ONCLICK Search Button **THEN** **Step 5: Search for matching records (C#)** SQL READ Matching Records FROM CUSTOMER ENTITY **Step 6: Display matching records (Angular)** IF (Matching Records Found == True) Display Search Customer Table ELSE IF (Matching Records Found == False) Display Error Modal" No customers matching the search criteria was found." **THEN** **Step 7: Admin Selects Customer to View (Angular)** Select Customer to View ONCLICK View Button **THEN**

	Step 8: Display View Customer Screen (Angular) **SQL READ** Title, Name, Surname, Cell Phone Number, Email Address, VAT Registration Number, Business Name FROM CUSTOMER ENTITY

Subsystem:	3. User

Use Case:	3.3 Update Customer
Language:	C#
Pseudo Code:	**Step 1: The admin searches for a customer using the Customer Name Provided (C#)** **THEN** **Step 2: Invoke Use Case 3.2 Search Customer (C#)** **THEN** **Step 3: Admin requests to update Customer (Angular)** Admin requests to update Customer ONCLICK Update Button **THEN** **Step 4: Display Update Customer Screen (Angular)** Display Update Customer Screen **THEN** **Step 5: Admin provides update details (Angular)** Admin provides update details. ONCLICK Update Button **THEN** **Step 6: Validate Update Details (C#)** If (Input fields == Valid) Continue to step 7 Else If (Input fields == false) Display Validation Error Modal “Entered details are in the incorrect format.” **THEN** **Step 7: Display Confirm Update Modal (Angular)** Display Confirm Update Modal “Are you sure you want to update the customer information?” **THEN** **Step 8: Confirm Update (Angular)** Admin Confirms Update

	ONCLICK Confirm Button **THEN** **Step 9: Update Customer Details (C#)** SQL UPDATE Title, Name, Surname, Cell Phone Number, Email Address, VAT Registration Number, Business Name IN CUSTOMER ENTITY **THEN** **Step 10: Display Success Modal (Angular)** IF (Update Successful == True) Display Success Modal “Customer information was successfully updated.” ELSE IF (Update Successful == False) Display Error Modal “There was an unexpected error when trying to update the customer information.”

Subsystem:	3. User

Use Case:	3.4 Delete Customer
Language:	C#
Pseudo Code:	**Step 1: The admin searches for a customer using the Customer Name Provided (C#)** **THEN** **Step 2: Invoke Use Case 3.2 Search Customer (C#)** **THEN** **Step 3: Admin requests to Delete Customer (Angular)** Admin requests to Delete Customer ONCLICK Delete Button **THEN** **Step 4: Display Confirm Update Modal (Angular)** Display Confirm Deletion Modal “Are you sure you want to delete this customer?” **THEN** **Step 5: Confirm Deletion (Angular)** Admin Confirms Deletion ONCLICK Confirm Button **Step 6: Delete Customer Details (C#)** SQL DELETE Title, Name, Surname, Cell Phone Number, Email Address, VAT Registration Number, Business Name FROM CUSTOMER ENTITY SQL DELETE Username, Password IN USER ENTITY **THEN** **Step 7: Display Success Modal (Angular)** IF (Deletion Successful == True)

	Display Success Modal "The customer was successfully deleted." ELSE IF (Deletion Successful == False) Display Error Modal "There was an unexpected error when trying to delete the customer."

Subsystem:	3. User
Use Case:	3.5 View Profile
Language:	C#
Pseudo Code:	**Step 1: A customer requests to View Profile** **THEN** **Step 2: Customer Clicks User Icon (Angular)** ONCLICK User Icon THEN ONCLICK View Profile DropDown Menu Item **THEN** **Step 3: Read Customer Details (C#)** SQL Read Title, Name, Surname, Cell Phone Number, Email Address, VAT Registration Number, Business Name FROM CUSTOMER ENTITY **THEN** **Step 4: Display View Profile Screen (Angular)** IF (Read Customer Details == True) Display View Profile Screen ELSE IF (Read Customer Details == False) Display Error Modal “There was an unexpected error when trying to display Customer Profile.”

Subsystem:	3. User
Use Case:	3.6 Update Profile
Language:	C#
Pseudo Code:	**Step 1: A customer wants to Update Profile.** **THEN** **Step 2: Customer Clicks User Icon (Angular)** ONCLICK User Icon THEN ONCLICK View Profile DropDown Menu Item **THEN** **Step 3: Invoke Use Case 3.5 View Profile (C#)** Invoke Use Case 3.5 View Profile **THEN** **Step 4: Customer requests to update profile (Angular)** Customer requests to update profile. ONCLICK Update Button **THEN** **Step 5: Display Update Profile Screen (Angular)** Display Update Profile Screen **THEN** **Step 6: Customer provides updated details. (Angular)** Customer updates details. ONCLICK Update Button **THEN** **Step 7: Validate Update Details (C#)** If (Input fields Valid == True) Continue to step 8 Else If (Input fields Valid == false) Display Validation Error Modal “The information entered is in the incorrect format. Please try again. **THEN**

	Step 8: Display Confirm Update Modal (Angular) Display Confirm Update Modal “Are you sure you want to update Profile?” **THEN** **Step 9: Confirm Update (Angular)** Customer Confirms Update ONCLICK Confirm Button [ALT] ONCLICK Cancel Button → Terminate Use Case **THEN** **Step 10: Update Customer Details (C#)** SQL UPDATE Title, Name, Surname, Cell Phone Number, Email Address, VAT Registration Number, Business Name IN CUSTOMER ENTITY **THEN** **Step 11: Display Success Modal (Angular)** IF (Update Successful == True) Display Success Modal “Customer Profile was successfully updated.” ELSE IF (Update Successful == False) Display Error Modal “There was an unexpected error when trying to update the customer Profile.”

Subsystem:	3. User
Use Case:	3.7 Delete Profile
Language:	C#
Pseudo Code:	**Step 1: A customer wants to Delete Profile.** **THEN** **Step 2: Customer Clicks User Icon (Angular)** ONCLICK User Icon THEN ONCLICK View Profile DropDown Menu Item **THEN** **Step 3: Invoke Use Case 3.5 View Profile (C#)** Invoke Use Case 3.5 View Profile **THEN** **Step 4: Customer requests to delete profile (Angular)** Customer requests to delete profile. ONCLICK Delete Button **THEN** **Step 5: Display Confirm Update Modal (Angular)** Display Confirm Deletion Modal “Are you sure you want to delete your account? **THEN** **Step 6: Confirm Deletion (Angular)** Customer Confirms Update ONCLICK Confirm Button [ALT] ONCLICK Cancel Button → Terminate Use Case **THEN** **Step 7: Update Customer Details (C#)** SQL DELETE Title, Name, Surname, Cell Phone Number, Email Address, VAT Registration Number, Business Name

	FROM CUSTOMER ENTITY SQL DELETE Username, Password FROM USER ENTITY **THEN** **Step 8: Display Success Modal (Angular)** IF (Deletion Successful == True) Display Success Modal “Customer Profile was successfully deleted..” ELSE IF (Deletion Successful == False) Display Error Modal “There was an unexpected error when trying to delete the customer account.”

Subsystem:	3. User

Use Case:	3.8 Add User Role
Language:	C#
Pseudo Code:	**Step 1: An Admin wants to add a User Role** THEN **Step 2: Admin Clicks User Icon (Angular)** ONCLICK User Icon THEN ONCLICK Add User Role DropDown Menu Item **THEN** **Step 3: Display Add User Role Screen (Angular)** Display Add User Role Screen **THEN** **Step 4: Provide User Role Details (Angular)** Admin provides use role details. ONCLICK Add Button **THEN** **Step 5: Validate User Role Details (C#)** If (Input fields Valid == True) Continue to step 8 Else If (Input fields Valid == false) Display Validation Error Modal “The information entered is in the incorrect format. Please try again. **THEN** **Step 6: Display Confirmation Modal (Angular)** Display Confirmation Modal “Are the User Role details correct?” **THEN** **Step 7: Confirm User Role Details (Angular)** Admin Confirms Details ONCLICK Confirm Button [ALT] ONCLICK Cancel Button → Terminate Use Case **THEN**

	Step 8: Save User Role Details and generate UserRoleID (C#) SQL INSERT User Role Name, User Role Description IN USERROLE ENTITY **THEN** **Step 9: Display Success Modal (Angular)** IF (Add User Role Successful == True) Display Success Modal “The User Role was successfully added..” ELSE IF (Add User Role Successful == False) Display Error Modal “There was an unexpected error when trying to add the User Role.

Subsystem:	3. User

Use Case:	3.9 Search User Role
Language:	C#
Pseudo Code:	**Step 1: An Admin wants to Search for a User Role** THEN **Step 2: Admin Clicks User Icon (Angular)** ONCLICK User Icon THEN ONCLICK Search User Role DropDown Menu Item **THEN** **Step 3: Display Search User Role Screen (Angular)** Display Search User Role Screen **THEN** **Step 4: Provide Search criteria (Angular)** Admin Provides Search Criteria ONCLICK Search Button **THEN** **Step 5: Read Matching Records (C#)** SQL READ Matching Records FROM CUSTOMER ENTITY **THEN** **Step 6: Display matching records (Angular)** IF (Matching Records Found == True) Display Search User Role Table ELSE IF (Matching Records Found == False) Display Error Modal" No User Role's matching the search criteria was found." **THEN** **Step 7: Admin Selects User Role to View (Angular)** Select User Role to View ONCLICK View Button

	THEN **Step 8: Display View User Role Screen (Angular)** Display View User Role Screen

Subsystem:	3. User

Use Case:	3.10 Update User Role
Language:	C#
Pseudo Code:	**Step 1: An Admin wants to Update a User Role** **THEN** **Step 2: Admin Clicks User Icon (Angular)** ONCLICK User Icon THEN ONCLICK Update User Role DropDown Menu Item **THEN** **Step 3: Invoke Use Case 3.9 Search User Role (C#)** Invoke Use Case 3.9 Search User Role **THEN** **Step 4: Admin requests to update User Role** Admin requests to update the User Role ONCLICK Update Button **THEN** **Step 5: Display Update User Role Screen (Angular)** Display Update User Role Screen **THEN** **Step 6: Admin provides updated details. (Angular)** Admin updates details. ONCLICK Update Button **THEN** **Step 7: Validate Update Details (C#)** If (Input fields Valid == True) Continue to step 8 Else If (Input fields Valid == false) Display Validation Error Modal “The information entered is in the incorrect format. Please try again. **THEN** **Step 8: Display Confirm Update Modal (Angular)**

	Display Confirm Update Modal “Are you sure you want to update the User Role details.?” **THEN** **Step 9: Confirm Update (Angular)** Admin Confirms Update ONCLICK Confirm Button [ALT] ONCLICK Cancel Button → Terminate Use Case **THEN** **Step 10: Update User Role Details (C#)** SQL UPDATE User Role Name, User Role Description IN USERROLE ENTITY **THEN** **Step 11: Display Success Modal (Angular)** IF (Update Successful == True) Display Success Modal “The User Role was successfully Updated.” ELSE IF (Update Successful == False) Display Error Modal “There was an unexpected error when trying to update the User Role.”

Subsystem:	3. User

Use Case:	3.11 Delete User Role
Language:	C#
Pseudo Code:	**Step 1: An Admin wants to Delete a User Role** **THEN** **Step 2: Admin Clicks User Icon (Angular)** ONCLICK User Icon THEN ONCLICK Delete User Role DropDown Menu Item **THEN** **Step 3: Invoke Use Case 3.9 Search User Role (C#)** Invoke Use Case 3.9 Search User Role **THEN** **Step 4: Admin Requests to Delete User Role (Angular)** Admin requests to delete the user role. ONCLICK Delete Button **THEN** **Step 5: Display Confirmation Modal (Angular)** Display Confirmation Modal "Are you sure you want to delete the User Role?" **THEN** **Step 6: Confirm Deletion (Angular)** Admin Confirms Deletion ONCLICK Confirm Button [ALT] ONCLICK Cancel Button → Terminate Use Case **THEN** **Step 7: Delete User Role Details (C#)** SQL DELETE User Role Name, User Role Description FROM USERROLE ENTITY **THEN** **Step 8: Display Success Modal (Angular)**

	IF (Update Successful == True) Display Success Modal “The User Role was successfully deleted.” ELSE IF (Update Successful == False) Display Error Modal “There was an unexpected error when trying to delete the User Role.”

3.1.3. EMPLOYEE SUBSYSTEM

4. Subsystem:	4. Employee Delivery Shifts
Use Case:	4.1. Create Delivery Shift
Language:	C#
Pseudo Code:	**Step 1: An Admin wants to add a new delivery shift** **THEN** **Step 2: Admin Clicks Create Delivery Shift menu item (Angular)** ONCLICK Delivery Shift THEN ONCLICK Create Delivery Shift Drop Down Menu Item **THEN** **Step 3: Display Create Delivery Shift Screen (Angular)** Display Create Delivery Shift Screen **THEN** **Step 4: Provide details for new Delivery Shift (Angular)** Admin provides new delivery shift details. **THEN** **Step 5: Admin clicks Add button (Angular)** ONCLICK Add Button [ALT] ONCLICK Cancel Button → terminate use case (Angular) **THEN** **Step 6: Display Confirmation Modal (Angular)** Display Confirmation Modal “Would you like to proceed with the creation of a new delivery shift?” **THEN**

Step 7: Admin clicks Yes button (Angular)

ONCLICK Yes Button

[ALT] ONCLICK No Button → Display Add Delivery Shift Screen (Angular)

THEN

Step 8: Validate Delivery Shift Details (C#)

If (Input fields Valid == True)

Continue to step 9

Else If (Input fields Valid == false)

[ALT] Display Validation Unsuccessful Modal "The information provided in the date and time already exists in a delivery shift."

THEN

Step 9: Save delivery shift details and generates a Time_ID, Date_ID and Shift_ID (#C)

SQL INSERT

Start_Time, End_Time

IN TIME ENTITY

SQL INSERT

Date

IN THE DATE ENTITY

SQL INSERT

Time_ID, Date_ID

IN SHIFT ENTITY

THEN

	Step 10: Display Success Modal (Angular) IF (Create Delivery Shift Successful == True) Display Success Modal “Delivery Shift was successfully created.” ELSE IF (Create Delivery Shift Successful == False) [ALT] Display Unsuccessful Modal “Failed to create delivery shift. Please try again later.”

Subsystem:	4. Employee Delivery Shifts
Use Case:	4.2. Search Delivery Shift
Language:	C#
Pseudo Code:	**Step 1: An Admin wants to search for a delivery shift** **THEN** **Step 2: Admin Clicks Search Delivery Shift** ONCLICK Delivery Shift THEN ONCLICK Search Delivery Shift Drop Down Menu Item **THEN** **Step 3: Display Search Delivery Shift Screen (Angular)** Display Search Delivery Shift Screen **THEN**

Step 4: Input Search criteria (Angular)

Admin inputs Search Criteria

ONCLICK Search Icon

THEN

Step 5: Validate Delivery Shift Details (C#)

If (Input fields Valid == True)

Continue to step 6

Else If (Input fields Valid == false)

[ALT] Display Validation Unsuccessful Modal "No matches found. Please check the format of the entered information."

THEN

Step 6: Read Matching Records (C#)

SQL READ

Matching Records

FROM SHIFT ENTITY

SQL READ

Matching Records

FROM DATE ENTITY

SQL READ

Matching Records

FROM TIME ENTITY

SQL READ

Matching Records

	FROM EMPLOYEESHIFT ENTITY SQL READ Matching Records FROM EMPLOYEE ENTITY **THEN** **Step 7: Display matching records (Angular)** IF (Matching Records Found == True) Display Delivery Shift Table on Search Delivery Shift Screen ELSE IF (Matching Records Found == False) → terminate use case (Angular)

Subsystem:	4. Employee Delivery Shifts
Use Case:	4.3. Update Delivery Shift
Language:	C#
Pseudo Code:	**Step 1: An Admin wants to update delivery shift information** ONCLICK Delivery Shift THEN ONCLICK Search Delivery Shift Dropdown Menu Item **THEN** **Step 2: Invoke Use Case 4.2. Search Delivery Shift (C#)**

	Invoke Use Case 4.2. Search Delivery Shift Then ONCLICK Update Button in Option Column **THEN** **Step 3: Display Update Delivery Shift Screen (Angular)** Display Update Delivery Shift Screen **THEN** **Step 4: Admin provides new information. (Angular)** Admin edits textboxes to update information. ONCLICK Done Button [ALT] ONCLICK Cancel Button → terminate use case (Angular) **THEN** **Step 5: Validate Update Details (C#)** If (Input fields Valid == True) Continue to step 6 Else If (Input fields Valid == false) Display Validation Unsuccessful Modal “The information provided in the date and time already exists in a delivery shift.” **THEN** **Step 6: Update Delivery Shift Details (C#)** SQL UPDATE Start_Time, End_Time, Date IN SHIFT ENTITY **THEN**

	Step 7: Display Success Modal (Angular) IF (Update Successful == True) Display Success Modal "Delivery Shift information was successfully updated." ELSE IF (Update Successful == False) Display Unsuccessful Modal "Failed to update delivery shift information. Please try again later."

Subsystem:	4. Employee Delivery Shifts

Use Case:	4.4. Delete Delivery Shift
Language:	C#
Pseudo Code:	**Step 1: An Admin wants to delete delivery shift information** ONCLICK **Delivery Shift** THEN ONCLICK Search Delivery Shift Dropdown Menu Item **THEN** **Step 2: Invoke Use Case 4.2. Search Delivery Shift (C#)** Invoke Use Case 4.2. Search Delivery Shift **THEN** **Step 3: Admin Clicks Delete Button** Then ONCLICK Delete Button in Option Column **THEN** **Step 4: Display Confirm to Proceed Modal (Angular)** Display Confirm Proceed Modal “Would you like to proceed with the deletion of the selected delivery shift?” **THEN** **Step 5: Admin clicks Yes button (Angular)** ONCLICK Yes Button [ALT] ONCLICK No Button → Display Search Delivery Shift Screen (Angular) **THEN** **Step 6: Delete Delivery Shift** SQL DELETE Shift_ID IN SHIFT ENTITY

	THEN **Step 7: Remove Delivery Shift Information line on screen** Remove Delivery Shift Table on Search Delivery Shift Screen **THEN** **Step 8: Display Success Modal (Angular)** IF (Delete Successful == True) Display Success Modal “Delivery Shift information was successfully deleted.” ELSE IF (Delete Successful == False) Display Unsuccessful Modal “Failed to delete delivery shift. Please try again later.”

Subsystem:	4. Employee Delivery Shifts
Use Case:	4.5. Add Employee
Language:	C#
Pseudo Code:	**Step 1: The admin wants to add the employee.** **THEN** **Step 2: Admin clicks on Employee menu item(Angular)** ONCLICK Employee Menu item on navigation bar **THEN** **Step 3: Display Employee Dropdown menu item (Angular)** Display Employee dropdown menu item **THEN** **Step 4: Admin clicks Employee dropdown menu item (Angular)** ONCLICK Add Employee dropdown menu item **THEN** **Step 5: Display Add Employee Screen (Angular)** Display Add Employee Screen **THEN** *//Step 6: Employee Communicates details for the admin to enter into the system.* **Step 7: Provide Employee Details (Angular)** Provide Employee Details ONCLICK Next Button **THEN** **Step 8: Validate Details (C#)**

	If (Input fields == Valid)
	Continue to step 9
	Else If (Input fields == false)
	Display Validation Unsuccessful Modal "The information provided in the given parameters is not in the correct format."
	THEN
	
	Step 9: Enable and disable controls on Add Employee Screen (Angular)
	Title= combobox. (enable==false)
	Name= textbox. (enable==false)
	Surname= textbox. (enable==false)
	CellNumber= textbox. (enable==false)
	Address= textbox. (enable==false)
	DOB= calendar. (enable==false)
	IDNumber= textbox. (enable==false)
	Next=button. (enable==false)
	
	UserRole = combobox. (enable==true)
	Username = textbox. (enable==true)
	Password = textbox. (enable==true)
	Confirm Password = textbox. (enable==true)
	Add Employee = button. (enable==true)
	THEN
	
	Step 10: Provide User Details (Angular)
	Provide User Details
	ONCLICK Add Employee Button
	THEN
	
	Step 11: Validate Details (C#)
	If (Input fields Valid == True)

	Continue to step 12 Else If (Input fields Valid == false) Display Validation Unsuccessful Modal "The information provided in the given parameters is not in the correct format." **THEN** **Step 12: Display Confirm to Proceed Modal (Angular)** Display Confirm Proceed Modal "Would you like to proceed with the addition of a new employee?" **THEN** **Step 13: Admin clicks Yes button (Angular)** ONCLICK Yes Button [ALT] ONCLICK No Button → Display Add Employee Screen (Angular) **THEN** **Step 14: Save Employee Details and Generate EmployeeID and UserID (C#)** SQL INSERT Title, Employee_Name, Employee_Surname, Employee_Number, Employee_Address, Employee_DOB, Employee_IDNumber IN EMPLOYEE ENTITY SQL INSERT User_Username, User_Password, UserRole_ID IN USER ENTITY **THEN** **Step 15: Display Success Modal (Angular)** IF (Add Employee Successful == True) Display Success Modal "Employee information was successfully created." ELSE IF (Add Employee Successful == False)

	Display Unsuccessful Modal "Failed to create employee information. Please try again later."

Subsystem:	4. Employee Delivery Shifts
Use Case:	4.6. Search Employee
Language:	C#
Pseudo Code:	**Step 1: An Admin wants to search for an employee** **THEN** **Step 2: Admin Clicks Search Employee** ONCLICK Employee THEN ONCLICK Search Employee Drop Down Menu Item **THEN** **Step 3: Display Search Employee Screen (Angular)** Display Search Employee Screen **THEN** **Step 4: Input Search criteria (Angular)** Admin inputs Search Criteria ONCLICK Search Button **THEN** **Step 5: Validate Delivery Shift Details (C#)** If (Input fields Valid == True) Continue to step 6 Else If (Input fields Valid == false) [ALT] Display Validation Unsuccessful Modal "No matches found. Please check the format of the entered information." **THEN**

	Step 6: Read Matching Records (C#) SQL READ Matching Records FROM EMPLOYEE ENTITY **THEN** **Step 7: Display matching records (Angular)** IF (Matching Records Found == True) Display Employee Table on Search Employee Screen ELSE IF (Matching Records Found == False) → terminate use case (Angular)

Subsystem:	4. Employee Delivery Shifts
Use Case:	4.7. Update Employee
Language:	C#
Pseudo Code:	**Step 1: An Admin wants to update employee information** ONCLICK Employee THEN ONCLICK Search Employee Dropdown Menu Item **THEN** **Step 2: Invoke Use Case 4.6. Search Employee (C#)** Invoke Use Case 4.2. Search Employee Then ONCLICK Update Button in Option Column **THEN** **Step 3: Display Update Employee Screen (Angular)** Display Update Employee Screen

	THEN
	
	Step 4: Admin provides new information. (Angular)
	Admin edits textboxes to update information.
	ONCLICK Done Button
	[ALT] ONCLICK Cancel Button → terminate use case (Angular)
	THEN
	
	Step 5: Validate Update Details (C#)
	If (Input fields Valid == True)
	Continue to step 6
	Else If (Input fields Valid == false)
	Display Validation Unsuccessful Modal "The information provided in the given parameters is not in the correct format."
	THEN
	
	Step 6: Update Employee Details (C#)
	SQL UPDATE
	Title, Employee_Name, Employee_Surname, Employee_Number, Employee_Address, Employee_DOB, Employee_IDNumber
	IN EMPLOYEE ENTITY
	THEN
	
	Step 7: Display Success Modal (Angular)
	IF (Update Successful == True)
	Display Success Modal "Employee information was successfully updated."
	ELSE IF (Update Successful == False)

	Display Unsuccessful Modal “Failed to update employee information. Please try again later.”

Subsystem:	4. Employee Delivery Shifts
Use Case:	4.8. Delete Employee
Language:	C#
Pseudo Code:	**Step 1: An Admin wants to delete employee information** ONCLICK Employee THEN ONCLICK Search Employee Dropdown Menu Item **THEN** **Step 2: Invoke Use Case 4.6. Search Employee (C#)** Invoke Use Case 4.6. Search Employee **THEN** **Step 3: Admin Clicks Delete Button** Then ONCLICK Delete Button in Option Column **THEN** **Step 4: Display Confirm to Proceed Modal (Angular)** Display Confirm Proceed Modal "Would you like to proceed with the deletion of the selected employee?" **THEN** **Step 5: Admin clicks Yes button (Angular)** ONCLICK Yes Button [ALT] ONCLICK No Button → Display Search Employee Screen (Angular) **THEN** **Step 6: Delete Employee** SQL DELETE Employee_ID

	IN EMPLOYEE ENTITY **THEN** **Step 7: Remove Employee Information line on screen** Remove Employee Table on Search Employee Screen **THEN** **Step 8: Display Success Modal (Angular)** IF (Delete Successful == True) Display Success Modal “Employee information was successfully deleted.” ELSE IF (Delete Successful == False) Display Unsuccessful Modal “Failed to delete employee. Please try again later.”

Subsystem:	4. Employee Delivery Shifts
Use Case:	4.10. Assign Delivery Shift
Language:	C#
Pseudo Code:	**Step 1: An Admin wants to assign a delivery shift to an employee.** ONCLICK Delivery Shift THEN ONCLICK Search Delivery Shift Dropdown Menu Item **THEN** **Step 2: Invoke Use Case 4.2. Search Delivery Shift (C#)** Invoke Use Case 4.2. Search Delivery Shift **THEN** **Step 3: Admin selects Assign button** ONCLICK Assign button in Assign To Column

	THEN **Step 4: Display Assign Delivery Shift Modal (Angular)** Display Assign Delivery Shift Modal **THEN** **Step 5: Admin selects the employee name. (Angular)** Admin selects the employee name in dropdown. **THEN** **Step 6: Admin clicks on the Proceed button** ONCLICK Proceed Button [ALT] ONCLICK Cancel Button → Display Search Delivery Screen (Angular) **THEN** **Step 7: Display Confirm to Proceed Modal (Angular)** Display Confirm Proceed Modal "Would you like to proceed with assigning this employee to the selected delivery shift?" **THEN** **Step 8: Admin clicks Yes button (Angular)** ONCLICK Yes Button [ALT] ONCLICK No Button → Display Search Delivery Shift Screen (Angular) **THEN** **Step 9: Remove Delivery Shift Details (C#)** SQL DELETE EmployeeShift_ID, Employee_ID

	IN EMPLOYEESHIFT ENTITY **THEN** **Step 10: Display Success Modal (Angular)** IF (Removal Successful == True) Display Success Modal "Delivery shift was successfully assigned." ELSE IF (Remove Successful == False) Display Unsuccessful Modal "Failed to assign delivery shift to employee. Please try again later.."

Subsystem:	4. Employee Delivery Shifts
Use Case:	4.11. Cancel Delivery Shift
Language:	C#
Pseudo Code:	**Step 1: An Admin wants to cancel an assigned delivery shift** ONCLICK Delivery Shift THEN ONCLICK Search Delivery Shift Dropdown Menu Item **THEN** **Step 2: Invoke Use Case 4.2. Search Delivery Shift (C#)** Invoke Use Case 4.2. Search Delivery Shift **THEN** **Step 3: Admin Clicks Cancel Button** Then ONCLICK Cancel Button in Option Column **THEN** **Step 4: Display Confirm to Proceed Modal (Angular)** Display Confirm Proceed Modal "Would you like to proceed with the cancellation of the selected assigned delivery shift?"

	THEN **Step 5: Admin clicks Yes button (Angular)** ONCLICK Yes Button [ALT] ONCLICK No Button → Display Search Delivery Shift Screen (Angular) **THEN** **Step 6: Delete Employee shift information** SQL DELETE Employee_ID IN EMPLOYEESHIFT ENTITY **THEN** **Step 7: Display Success Modal (Angular)** IF (Delete Successful == True) Display Success Modal "Cancellation of the assigned shift was successful." ELSE IF (Delete Successful == False) Display Cannot Be Cancelled Modal "This delivery shift cannot be cancelled as it is not yet assigned to an employee. Please try again later." **THEN** **Step 8: Remove Employee Information line on screen** Remove Employee Name from Delivery Shift Table on Search Delivery Shift Screen **THEN**

4.1.1. ADMIN SUBSYSTEM

5. Subsystem:	5. Administration Back Office
Use Case:	5.1 Search Online Sales
Language:	C#
Pseudo Code:	**Step 1: Employee/Admin wants to Search an Online Sale** **THEN** **Step 2: Employee/Admin selects Administration Back Office Menu Item (Angular)** ONCLICK Administration Back Office Menu Item THEN ONCLICK Search Online Sales dropdown menu item **THEN** **Step 3: Display Search Online Sales Screen (Angular)** Display Search Online Sales Screen **THEN** **Step 4: Admin provides search criteria (Angular)** Admin Selects filter option. ONCLICK Search Button **THEN** **Step 5: Read Matching Record (C#)** SQL READ Matching Records FROM SALE ENTITY **THEN** **Step 6:** IF (Matching Records Found == True) Display Search Online Sales Table ELSE IF (Matching Records Found == False) Display Error Modal “No sales matching the search criteria was found.” **THEN** **Step 7: Admin selects Sale to View**

	Admin selects sale. ONCLICK View Button **THEN** **Step 8: Display View Sale Screen** IF (Read Sale Details Successfully == True) Display View Sale Screen ELSE IF (Read Sale Details Successfully == False) Display Error Modal “There was an unexpected error when trying to display the sales.”

Subsystem:	5. Administration Back Office

Use Case:	5.2 Pack Order
Language:	C#
Pseudo Code:	**Step 1: Employee wants to Pack an Order** **THEN** **Step 2: Employee selects Administration Back Office Menu Item (Angular)** ONCLICK Administration Back Office Menu Item THEN ONCLICK Pack Order dropdown menu item **THEN** **Step 3: Invoke Use Case 5.1 Search Sale Order (C#)** Invoke Use Case 5.1 Search Sale Order **Step 4: Employee requests to pack Order (Angular)** Employee requests to pack an order ONCLICK Pack Order Button **Step 5: Display Pack Order Screen (Angular)** Display Search Online Sales Screen **THEN** **Step 6: Employee Provides Pack Order Details (Angular)** Employee Provides Pack Order Details ONCLICK Confirm Button **THEN** **Step 7: Update Order Status(C#)** IF (Order For Collection Checkbox == Checked) SQL UPDATE Order Status == “Ready for Collection” IN SALE ENTITY [ALT] ELSE IF (Order For Delivery Checkbox == Checked) SQL UPDATE

	Order Status == “Ready for Delivery” IN SALE ENTITY Skip to Step 9. **THEN** **Step 8: Notify Customer that Order is Ready for Collection (C#)** Notify customer via SMS that Order is Ready for Delivery. SQL READ Customer CellPhone Number From Customer Entity **THEN** **Step 9: Display Success Modal** IF (Pack Order Successful == True) Display Success Modal “Order has been Packed.” ELSE IF (Pack Order Successful == False) Display Error Modal “There was an unexpected error when trying to complete the pack order process.

Subsystem:	5. Administration Back Office
Use Case:	5.3 Do Stock Take
Language:	C#
Pseudo Code:	**Step 1: Admin wants to do Stock Take** **THEN** **Step 2: Admin selects Administration Back Office Menu Item (Angular)** ONCLICK Administration Back Office Menu Item THEN ONCLICK Do Stock Take dropdown menu item **THEN** **Step 3: Invoke Use Case 9.10 Search Product (C#)** Invoke Use Case 9.10 Search Product **Step 4: Admin requests to do stock take (Angular)** Employee requests to do stock take ONCLICK Do Stock-Take Button **Step 5: Display Do Stock-Take Screen (Angular)** Display Do Stock-Take Screen **THEN** **Step 6: Admin Provides Stock Take Details (Angular)** Admin Provides Stock Take Details ONCLICK Save Button **THEN** **Step 7: Validate Stock Take Details (C#)** **Validate Details (C#)** If (Input fields Valid == True) Continue to step 8 Else If (Input fields Valid == false) Display Validation Error Modal “The information entered is in the incorrect format. Please try again.” **THEN** **Step 8: Display Confirmation Modal (Angular)**

Display Confirmation Modal " Do you want to proceed with the stock take?"

THEN

Step 9: Confirm Stock Take (Angular)

Admin Confirms to proceed

ONCLICK Confirm Button

[ALT] ONCLICK Cancel Button → Terminate Use Case

THEN

Step 10: Save Stock Take Details and Generate Stock Take ID (C#)

SQL INSERT

Stock Take Date

IN STOCKTAKE ENTITY

SQL INSERT

Stock Take Quantity

IN PRODUCTITEM_STOCKTAKE ENTITY

THEN

Step 11: Update Quantity on Hand (C#)

SQL UPDATE

Quantity on Hand

IN PRODUCTITEM ENTITY

THEN

Step 12: Display Success Modal (Angular)

IF (StockTake Successful == True)

Display Success Modal "Stock Take was completed successfully."

ELSE IF (StockTake Successful == False)

Display Error Modal "There was an unexpected error when trying to do the stock-take."

Subsystem:	5. Administration Back Office
Use Case:	5.4 View Delivery Shift Schedule
Language:	C#
Pseudo Code:	**Step 1: Employee wants to View Delivery Shift Schedule** **THEN** **Step 2: Employee selects Administration Back Office Menu Item (Angular)** ONCLICK Administration Back Office Menu Item THEN ONCLICK View Delivery Shift dropdown menu item **THEN** **Step 3: Read Delivery Shift Schedule Details (C#)** SQL Read Shift Date, Shift Start Time, Shift End Time FROM SHIFT ENTITY SQL READ Employee Name FROM EMPLOYEE_SHIFT ENTITY **THEN** **Step 4: Display Delivery Shift Schedule (Angular)** IF (Read Delivery Shift Details Successfully == True) Display Delivery Shift Schedule Screen ELSE IF (Read Delivery Shift Details Successfully == False) Display Error Modal “There was an unexpected error when trying to display the Delivery Shift Schedule.”

Subsystem:	5. Administration Back Office
Use Case:	5.5 Write-Off Stock
Language:	C#
Pseudo Code:	**Step 1: Admin wants to Write-Off Stock** **THEN** **Step 2: Admin selects Administration Back Office Menu Item (Angular)** ONCLICK Administration Back Office Menu Item THEN ONCLICK Write-Off Stock dropdown menu item **THEN** **Step 3: Invoke Use Case 9.10 Search Product (C#)** Invoke Use Case 9.10 Search Product **Step 4: Admin requests to do stock take (Angular)** Admin requests to Write-Off Stock ONCLICK Write-Off Stock Button **Step 5: Display Do Write-Off Stock Screen (Angular)** Display Do Stock-Take Screen **THEN** **Step 6: Admin Provides Stock Take Details (Angular)** Admin Provides Stock Take Details ONCLICK Save Button **THEN** **Step 7: Validate Write-Off Stock Details (C#)** **Validate Details (C#)** If (Input fields Valid == True) Continue to step 8 Else If (Input fields Valid == false) Display Validation Error Modal “The information entered is in the incorrect format. Please try again.” **THEN** **Step 8: Display Confirmation Modal (Angular)**

	Display Confirmation Modal " Do you want to proceed with the Write-off? **THEN** **Step 9: Confirm Write-off Stock (Angular)** Admin Confirms to proceed ONCLICK Confirm Button [ALT] ONCLICK Cancel Button → Terminate Use Case **THEN** **Step 10: Save Write-off Stock Details and Generate WriteOffID (C#)** SQL INSERT Write-Off Date IN WRITEOFF ENTITY SQL INSERT Write-Off Quantity, Write-off Reason IN PRODUCTITEM_WRITEOFFSTCOK ENTITY **THEN** **Step 11: Update Quantity on Hand (C#)** SQL UPDATE Quantity on Hand IN PRODUCTITEM ENTITY **THEN** **Step 12: Display Success Modal (Angular)** IF (StockTake Successful == True) Display Success Modal "Write-Off was completed successfully." ELSE IF (StockTake Successful == False) Display Error Modal "There was an unexpected error when trying to do the write-off,

5.1.1. CUSTOMER SUBSYSTEM

Subsystem:	6. Customer
Use Case:	6.1 View Products
Language:	C#
Pseudo Code:	**Step 1: Customer request to browse through the products by category.** **THEN** **Step 2: Customer clicks Products Menu Item (Angular)** **ON CLICK** Products Menu Item **THEN ON CLICK** Products Dropdown menu Item **THEN** **Step 3: Load Products Screen (Angular)** SQL Select ProductItem information From ProductItem table Where ProductCategory matches with ProductCategory in ProductCategory table Retrieve the following: o ProductCategory_ID o ProductCategory_Desc o ProductItem_Name o ProductItem_Image **THEN** display the information for each product in that category in a card. Display Products Screen **[ALT STEP 3]: The system fails to load delivery page(Angular)** **ON CLICK** Product Menu Item **THEN ON CLICK** Products Dropdown menu Item System fails to load Products page. **THEN** display Error message, “There was an error loading the Products page. Please try again later.” Terminate

Subsystem:	6. Customer
Use Case:	6.2 Search Product Details
Language:	C#
Pseudo Code:	**Step 1: Customer request to search for a product.** The system invokes 6.1 Search Products **Step 2: Customer provides search criteria (Angular)** **ON CLICK** Search Menu bar **THEN** Customer enters search criteria **THEN** The customer provides Product information(Angular) **Step 3: Validate and Capture search details (Angular)** **ON CLICK** Search icon-button **THEN** *If(input not valid== true)* Display error message, "Please ensure that the details you have entered are valid" *else* SQL Read ProductItem information From ProductItem table Where ProductItem matches with ProductItem in ProductItem table Read the following: o ProductItem_ID o ProductItem_Name **THEN** **Step 3: Display retrieved results Screen (Angular)** **THEN** display the information for each product in that category in a card. Display Products Screen with retrieved results Customer selects product they were looking for.

	THEN **Step 4: Display retrieved results Screen (Angular)** **ON CLICK** product card SQL Select ProductItem information From ProductItem table SQL Select CategoryType information From CategoryType table SQL Select ProductCategory information From ProductCategory table Where ProductItem matches with ProductItem in ProductItem table Read the following: ○ CategoryType_Desc ○ ProductCategory_Desc ○ ProductItem_Name ○ ProductItem_Desc ○ ProductItem_Image ○ ProductItem_Cost **THEN** Display Product Details Screen with retrieved results **[ALT STEP 4]: The system fails to load Product Details page(Angular)** **ON CLICK** Product Card System fails to load Product Details page. **THEN** display Error message, “There was an error loading the Product Details page. Please try again later.” Terminate

<table>
<tr><td>Subsystem:</td><td>6. Customer</td></tr>
<tr><td>Use Case:</td><td>6.3 Search Product Details</td></tr>
<tr><td>Language:</td><td>C#</td></tr>
<tr><td>Pseudo Code:</td><td>Step 1: Customer request to search for a product.

The customer request to browse through the specials and promotions.

Step 2: Customer clicks Promotion Menu Item (Angular)

ON CLICK Promotions Menu item

THEN

Step 3: Load Promotions Screen (Angular)

SQL Select

ProductItem information

From ProductItem table

SQL Select

Product_Special information

From Product_Special table

SQL Select

Special information

From Special table

Retrieve the following:
○ ProductItem_Name
○ ProductItem_Image
○ Special_Price
○ Special_EndDate

THEN display the information for each product in that category in a card.
Display Promotions Screen

Step 4: Customer provides search criteria (Angular)

ON CLICK Search Menu bar</td></tr>
</table>

	THEN Customer enters search criteria **THEN** The customer provides ProductItem information(Angular) **Step 5: Validate and Capture search details (Angular)** **ON CLICK** Search icon-button **THEN** *If(input not valid== true)* Display error message, “Please ensure that the details you have entered are valid” *else* SQL Read ProductItem information From ProductItem table Where ProductItem matches with ProductItem in ProductItem table Read the following: o ProductItem_ID o ProductItem_Name **THEN** **Step 6: Display retrieved results Screen (Angular)** **THEN** display the information for each product in that category in a card. Display Products Screen with retrieved results Customer selects product they were looking for. **THEN** **Step 7: Display retrieved results Screen (Angular)** **ON CLICK** Promotions card SQL Select ProductItem information From ProductItem table SQL Select

	CategoryType information From CategoryType table SQL Select ProductCategory information From ProductCategory table Where ProductItem matches with ProductItem in ProductItem table Read the following: o CategoryType_Desc o ProductCategory_Desc o ProductItem_Name o ProductItem_Desc o ProductItem_Image o Special_Price o Special_EndDate **THEN** Display SpecialPromotion Details Screen with retrieved results **[ALT STEP 3]: The system fails to load delivery page(Angular)** **ON CLICK** Promotions Menu Item **THEN ON CLICK** Promotions Dropdown menu Item System fails to load Promotions page. **THEN** display Error message, “There was an error loading the Promotions page. Please try again later.” Terminate **[ALT STEP 6]: The system fails to load Product Details page(Angular)** **ON CLICK** Promotion Card System fails to load SpecialPromotion Details page. **THEN** display Error message, “There was an error loading the Product Details page. Please try again later.” Terminate

Subsystem:	6. Customer
Use Case:	6.4 Add to cart
Language:	C#
Pseudo Code:	**Step 1: Customer request to add product item to cart.** The customer requests to add a product item to cart. **Step 2: Customer clicks add to cart button (Angular)** **ON CLICK** 'Add to cart' button **THEN** **Step 3: Load Confirm Add to cart prompt (Angular)** **ON CLICK** Add button **THEN** **Step 4: Capture ProductItem details (Angular)** SQL Select CartLine_information From CartLine_table SQL Select ProductItem information From ProductItem _table Add the following: ○ CartLine_Quantity ○ ProductItem_ID **THEN** **Step 5: Update cart icon using added details (Angular)** Update the cart icon in the page header. Using the CartLine_Quantity attribute in the **CartLine entity** **[ALT Step 3:]** The user clicks the cancel button. Terminated **[ALT STEP 4]: The system fails add product items to cart (Angular)** **ON CLICK** Add button System fails add product items to cart. **THEN** Display Error message, "There was an error while added the product to cart.

	Please try again." Terminate

Subsystem:	6. Customer
Use Case:	6.5 Search Cart Item
Language:	C#
Pseudo Code:	**Step 1: Customer request to see the cart.** **THEN** **Step 2: Customer clicks Cart icon on the menu-strip(Angular)** **ON CLICK** Products Menu Item **THEN** **Step 3: Load Cart Screen (Angular)** SQL Select ProductItem information From ProductItem table SQL Select Product_Special information From Product_Special table SQL Select Special information From Special table Where ProductCategory matches with ProductCategory in ProductCategory table Retrieve the following: ○ ProductItem_Name ○ ProductItem_Image ○ Special_Price or ProductItem_Cost ○ Special_EndDate (if Special_Price is shown) **THEN** display the information for each product in a tablerow of the CartTable. Display Cart Screen **[ALT STEP 3]: The system fails to load Cart page(Angular)**

	ON CLICK Cart icon-button **THEN** System fails to load Cart page. **THEN** Display Error message, “There was an error loading the Cart page. Please try again later.” Terminate

Subsystem:	6. Customer
Use Case:	6.6 Remove Cart Item
Language:	C#
Pseudo Code:	**Step 1: Customer request to remove a product item from cart.** The customer requests to add a product item to cart. **Step 2: Customer clicks remove icon-button (Angular)** In the Cart view the user navigates to Cart table **ON CLICK** Remove icon-button of specific cart item **THEN** **Step 3: Load Confirm Remove the item from cart prompt (Angular)** **ON CLICK** Remove button **THEN** **Step 4: Remove ProductItem details (Angular)** SQL Delete CartLine_information From CartLine_table SQL Select ProductItem information From ProductItem table Add the following: o CartLine_Quantity o ProductItem_ID **THEN** **Step 5: Update cart icon using added details (Angular)** Update the cart icon in the page header. Using the CartLine_Quantity attribute in the **CartLine entity** **[ALT Step 3:]** The user clicks the cancel button. Terminated **[ALT STEP 4]: The system fails remove product items from cart (Angular)** **ON CLICK** Remove button

	System fails remove product items from cart. **THEN** Display Error message, “There was an error while removing the product from cart. Please try again.” Terminate

Subsystem:	6. Customer
Use Case:	6.7 Checkout
Language:	C#
Pseudo Code:	**Step 1: Employee request to go to checkout.** **THEN** **Step 2: Employee clicks the Checkout button (Angular)** In the Cart View the user navigates to Checkout button **ON CLICK** Checkout button **THEN** **Step 3: System checks if customer is logged (Angular)** SQL Read AuditTrail information From table *If(audittrail description == valid)* *Continue to next step* *else* THEN **[ALT STEP 3]: Customer is not logged in prompt(Angular)** **ON CLICK** sign in button Load login page Else if **ON CLICK** register button Load register page **THEN** System fails to update delivery status. **THEN** display Error message, “There was an error while updating. Please try again later.” **Step 4: Load Checkout- Delivery or Collect prompt (Angular)**

	The customer selects the delivery optton **ON CLICK** Proceed button **THEN** The system loads and displays delivery form. **Step 5: Validate and Update delivery information (Angular)** **ON CLICK** Proceed button **THEN** *If(input incomplete== true)* Display error message, "Please ensure that information has been filled in" *else* SQL Read Address information From Address table SQL Read Province information From Province table SQL Read City information From City table SQL Read Suburb information From Suburb table Validate the following: ○ Delivery_ID ○ Courier_TrackingNumber

- Delivery_Date
- Courier_Name
- Courier_Number
- Courier_Email
- Address_Line1
- Address_Line2
- Address_Line3
- Address_PostalCode
- Province_Description
- City_Description
- Suburb_Description

THEN

Step 6: The system captures and updates Delivery and Sale

The system uses a SQL insert to capture the AddressID using the Delivery table and generates a new deliveryID, the system also uses SQL Update to update the SaleOrder_ReceiveType in the Sale Entity

SQL Insert

Delivery information

From Delivery table

SQL Insert

Sale information

From Sale table

Update the following:

- AddressID (Delivery entity)
- DeliveryID (Sale entity)
- SaleOrder_ReceiveType

THEN

Step 7: Load Amount due prompt (Angular)

SQL Select

CartLine information

From CartLine table

	Step 8: The user clicks the 'Checkout with PayPal' button (Angular) **ON CLICK** VCS **THEN** The system system loads the payment gateway with the amount due. **THEN** The customer makes a payment through the Virtual card services (PayPal) and that system sends us the transaction details. **Step 8:** The system uses a SQL insert to record the transaction in the sale table and generates a new SaleID. SQL Update Sale information From Sale table Update the following: ○ SaleID ○ Customer ID ○ PaymentType ID ○ OrderStatus ID ○ SaleOrder Desc ○ SaleOrder Date using the current Date. ○ Payment Date using the current Date. ○ Payment Amount calculating the total of the cart items. The system generates a new SaleID by retrieving the last SaleID D in the Sale table and incrementing it by one. **Step 9: The system displays a successful message** **THEN** Display message, "Payment Successful" Terminate **[ALT Step 4]: Customer clicks the collection radio button(Angular)** Proceed to next step. **[ALT Step 5:] The customer selects the cancel button** **ON Click** cancel Terminate

	[ALT STEP 6]: The system fails update delivery (Angular)
	        ON CLICK No
	
	System fails to update delivery.
	
	THEN display Error message, “There was an error while updating the delivery details.
	Please try again.”
	
	Terminate
	
	[ALT STEP 8]: An error occurs on the VCS system and the payment failed (Angular)
	        ON CLICK No
	
	System fails to update delivery.
	
	THEN display Error message, “Payment unsuccessful
	Please try again.”
	
	Terminate

5.1.2. REPORTS SUBSYSTEM

6. Subsystem:	7. Reports Subsystem
Use Case:	7.1 Generate fast-selling product list
Language:	C#
Pseudo Code:	**Step 1: The requests to generate a report** Then **Step 2: The admin clicks on the Reports menu option (Angular)** Onclick the Reports Menu item **Then** **Step 3: The system displays the "Reports" screen (Angular)** Display Reports Screen **Then** **Step 4: The admin selects the "Fast-selling product list" (Angular)** Onclick Report type dropdown **Then** **Step 5: The admin selects the required period.** Select Report start & end date **Then** **Step 6: The admin clicks the "Generate" button** Onclick Generate Button **Then** **Step 7: System retrieves report records from the database (C#)** SQL Select ProductItem_ID, ProductItem_Name, ProductItem_Desc, CategoryType_ID from ProductItem entity, SQL Select SaleID, SaleOrder_Date, SaleOrderDesc from SALE entity and SQL Select

	SaleLine_ID, ProductItem_ID, Sale_ID, SaleLine_Quantity from SaleLine Entity **Then** **Step 8: The system sorts through the retrieved information and filters for the most purchased products (C#)** Calculate the most purchased product items by checking the SaleLine_Quantity present in each order. With the items being most ordered at the top. The system also counts how many times the product items appeared in the different orders **Then** **Step 9: The system displays the Fast-selling product list (Angular)** Display the fast-selling product list with columns for the quantity of product items ordered and also for the number of times the products were placed in unique orders.

Subsystem:	7. Reports Subsystem
Use Case:	7.2 Generate slow-selling product list
Language:	C#
Pseudo Code:	**Step 1: The requests to generate a report** Then **Step 2: The admin clicks on the Reports menu option (Angular)** Onclick the Reports Menu item **Then** **Step 3: The system displays the "Reports" screen (Angular)** Display Reports Screen **Then** **Step 4: The admin selects the "Slow-selling product list" (Angular)** Onclick Report type dropdown **Then**

	Step 5: The admin selects the required period. Select Report start & end date **Then** **Step 6: The admin clicks the "Generate" button** Onclick Generate Button **Then** **Step 7: System retrieves report records from the database (C#)** SQL Select ProductItem_ID, ProductItem_Name, ProductItem_Desc, CategoryType_ID from ProductItem entity, SQL Select SaleID, SaleOrder_Date, SaleOrderDesc from SALE entity and SQL Select SaleLine_ID, ProductItem_ID, Sale_ID, SaleLine_Quantity from SaleLine Entity **Then** **Step 8: The system sorts through the retrieved information and filters for the least purchased products (C#)** Calculate the most purchased product items by checking the SaleLine_Quantity present in each order. With the items not purchased often at the top. The system also counts how many times the product items appeared in the different orders **Then** **Step 9: The system displays the Slow-selling product list (Angular)** Display the slow-selling product list with columns for the quantity of product items ordered and for the number of times the products were placed in unique orders.

Subsystem:	7. Reports Subsystem

Use Case:	7.3 Generate most frequent buyers report
Language:	C#
Pseudo Code:	**Step 1: The requests to generate a report** Then **Step 2: The admin clicks on the Reports menu option (Angular)** Onclick the Reports Menu item **Then** **Step 3: The system displays the “Reports” screen (Angular)** Display Reports Screen **Then** **Step 4: The admin selects the “Most frequent buyer report” (Angular)** Onclick Report type dropdown **Then** **Step 5: The admin clicks the “Generate” button (Angular)** Onclick Generate Button **Then** **Step 6: System retrieves report records from the database (C#)** SQL Select Sale ID, SaleOrder Date, SaleOrder Desc, Delivery ID, Courier ID, Payment Date, Customer ID from the SALE entity and SQL Select Customer ID, Customer Name, Customer Surname, Customer Cellphone Number, Customer Email Address and Customer Business name from the CUSTOMER entity **Then** **Step 7: The system sorts through the retrieved information and filters for the top results by using a count (C#)**

	Calculate the most frequent buyers by counting the total number of Sales made a particular customer in the past 4 weeks. Sort from most purchases to least purchases. **Then** **Step 8: The system displays the most frequent buyers report (Angular)** Display customers who have made the most purchases and the number of purchases they have made as a report on the screen.

Subsystem:	7. Reports Subsystem
Use Case:	7.4 Generate most popular location report
Language:	C#
Pseudo Code:	**Step 1: The requests to generate a report** Then **Step 2: The admin clicks on the Reports menu option (Angular)** Onclick the Reports Menu item **Then** **Step 3: The system displays the “Reports” screen (Angular)** Display Reports Screen **Then** **Step 4: The admin selects the “most popular location report” (Angular)** Onclick Report type dropdown **Then** **Step 5: The admin clicks the “Generate” button (Angular)** Onclick Generate Button **Then** **Step 6: System retrieves report records from the database (C#)** SQL Select

	Sale ID, SaleOrder Date, SaleOrder Desc, Delivery ID, Courier ID, Payment Date, Customer ID from the SALE entity SQL Select Customer ID, Customer Name, Customer Surname, Customer Cellphone Number, Customer Email Address and Customer Business name from the CUSTOMER entity SQL Select Address ID, Customer ID, Province ID, Address line 1, Address line 2, Address line 3, Address Postal Code from the Address entity SQL Select Province ID, Province Description from the Province entity SQL Select City ID, Province ID, City Description from the City entity SQL Select Suburb ID, City ID, Suburb Description from the Suburb entity **Then** **Step 7: The system sorts through the retrieved information and filters for the areas which had the most sales in the past month (C#)** Calculate the number customers which have been placing the most orders. Calculate the locations where these customers are located Sort the information by the province where the most orders have been placed over the last month. **Then** **Step 8: The system displays the most popular location report (Angular)** Display the locations (province – city – suburb) where the most purchases have been made in the past month

Subsystem:	7. Reports Subsystem
Use Case:	7.5 Generate Weekly Sale Order List
Language:	C#
Pseudo Code:	**Step 1: The requests to generate a report** Then **Step 2: The admin clicks on the Reports menu option (Angular)** Onclick the Reports Menu item **Then** **Step 3: The system displays the "Reports" screen (Angular)** Display Reports Screen **Then** **Step 4: The admin selects the "Weekly Sale Order List" (Angular)** Onclick Report type dropdown **Then** **Step 5: The admin clicks the "Generate" button (Angular)** Onclick Generate Button **Then** **Step 6: System retrieves report records from the database (C#)** SQL Select Sale ID, SaleOrder Date, SaleOrder Desc, Delivery ID, Courier ID, Payment Date, Customer ID from the SALE entity and SQL Select Customer ID, Customer Name, Customer Surname, Customer Cellphone Number, Customer Email Address and Customer Business name from the CUSTOMER entity **Then** **Step 7: The system then selects the Sales which have occurred in the past week and puts them in a list with the corresponding customers who made those sale orders also in the list. (C#)**

	Select sales made in the past 7 days order by SaleOrder date (last placed first) **Then** **Step 8: The system then displays the weekly sale order list to the admin (Angular)** Display the table list of sales made over the past week according to the date the sales where placed. Table list contains customer and matching sale information.

Subsystem:	7. Reports Subsystem
Use Case:	7. 6 Generate Weekly Sale Order List
Language:	C#
Pseudo Code:	**Step 1: The requests to generate a report** Then **Step 2: The admin clicks on the Reports menu option (Angular)** Onclick the Reports Menu item **Then** **Step 3: The system displays the "Reports" screen (Angular)** Display Reports Screen **Then** **Step 4: The admin selects the "Weekly Sale Order List" (Angular)** Onclick Report type dropdown **Then** **Step 5: The admin clicks the "Generate" button (Angular)** Onclick Generate Button **Then** **Step 6: System retrieves report records from the database (C#)** SQL Select

	Sale ID, SaleOrder Date, SaleOrder Desc, Delivery ID, Courier ID, Payment Date, Customer ID from the SALE entity and SQL Select Customer ID, Customer Name, Customer Surname, Customer Cellphone Number, Customer Email Address and Customer Business name from the CUSTOMER entity **Then** **Step 7: The system then selects the Sales which have occurred in the past month and puts them in a list with the corresponding customers who made those sale orders also in the list. (C#)** Select sales made in the past 7 days order by SaleOrder date (last placed first) **Then** **Step 8: The system then displays the monthly sale order list to the admin (Angular)** Display the table list of sales made over the past month (4 weeks) according to the date the sales were placed. Table list contains customer and matching sale information.

Subsystem:	7. Reports Subsystem
Use Case:	7.7 Generate Stock-Level Report
Language:	C#
Pseudo Code:	**Step 1: The requests to generate a report** Then **Step 2: The admin clicks on the Reports menu option (Angular)** Onclick the Reports Menu item **Then** **Step 3: The system displays the "Reports" screen (Angular)** Display Reports Screen **Then**

	Step 4: The admin selects the "Stock-Level Report" (Angular) Onclick Report type dropdown **Then** **Step 5: The admin clicks the "Generate" button (Angular)** Onclick Generate Button **Then** **Step 6: System retrieves report records from the database (C#)** SQL Select ProductItem ID, ProductItem Name, ProductItem Desc, ProductItem Image, CategoryType ID, Quantity on Hand from the ProductItem entity SQL Select CategoryType ID, CategoryType Desc, ProductCategory ID from CategoryType entity SQL Select ProductCategory ID, ProductCategory Desc from ProductCategory entity **Then** **Step 7: The system categorizes each ProductItem into its Category type and each category type into its Product Category (C#)** Sort retrieved product items into category types and then into product categories. Order the information according to quantity on hand with the items with the highest quantity on hand being at the top. **Then** **Step 8: The system then displays the Stock-level report with the necessary columns (Angular)** Display Product Category Desc, Category Type Desc, Product Item Name and Quantity on Hand as table columns with the rows populated according to the entries in the ProductItem table

Subsystem:	7. Reports Subsystem
Use Case:	7.8 Packing Report
Language:	C#
Pseudo Code:	**Step 1: The requests to generate a report** Then **Step 2: The admin clicks on the Reports menu option (Angular)** Onclick the Reports Menu item **Then** **Step 3: The system displays the "Reports" screen (Angular)** Display Reports Screen **Then** **Step 4: The admin selects the "Packing Report" (Angular)** Onclick Report type dropdown **Then** **Step 5: The admin clicks the "Generate" button (Angular)** Onclick Generate Button **Then** **Step 6: System retrieves report records from the database (C#)** SQL Select Sale ID, SaleOrder Date, SaleOrder Desc, OrderStatus ID, SaleOrder Assign, Payment Date, Payment Amount from the SALE entity SQL Select OrderStatus ID, OrderStatus Desc from the OrderStatus entity **Then** **Step 7: The system sorts the retrieved sales from the first made to the most recent** Select Sales which will have the order status of "ready for packing"

	Sort retrieved sales by sale order date with the sales placed first at the top. **Then** **Step 8: The system displays the list of Sales to the admin as the Packing Report (Angular)** Display all the Sale information of the sales with an order status of “ready for packing” as the Packing report

Subsystem:	7. Reports Subsystem
Use Case:	7.9 Deliver Report
Language:	C#
Pseudo Code:	**Step 1: The requests to generate a report** Then **Step 2: The admin clicks on the Reports menu option (Angular)** Onclick the Reports Menu item **Then** **Step 3: The system displays the “Reports” screen (Angular)** Display Reports Screen **Then** **Step 4: The admin selects the “Delivery Report” (Angular)** Onclick Report type dropdown **Then** **Step 5: The admin clicks the “Generate” button (Angular)** Onclick Generate Button **Then** **Step 6: System retrieves report records from the database (C#)** SQL Select Sale ID, SaleOrder Date, SaleOrder Desc, OrderStatus ID, SaleOrder Assign, Payment Date, Payment Amount from the SALE entity

	SQL Select OrderStatus ID, OrderStatus Desc from the OrderStatus entity SQL Select Delivery ID, Courier ID, Delivery Date, Address ID SQL Select Address ID, Customer ID, Province ID, Address line 1, Address line 2, Address line 3, Address Postal Code from the Address entity SQL Select Province ID, Province Description from the Province entity SQL Select City ID, Province ID, City Description from the City entity SQL Select Suburb ID, City ID, Suburb Description from the Suburb entity SQL Select Customer ID, Customer Name from the Customer entity **Then** **Step 7: The system then filters through the retrieved sales and sorts them according to SaleOrder_Date.** Select Sales which will have the order status of "ready for delivery" Sort retrieved sales by sale order date with the sales placed first at the top. **Then** **Step 8: The system then displays the list of retrieved Sales to the admin in the correct order as the Delivery Report (Angular)** Display all the Sale, delivery, customer and location information of the sales with an order status of "ready for delivery" as the Delivery report

6.1.1. DELIVERY SUBSYSTEM

Subsystem:	8. Employee Delivery Shift
Use Case:	8.1 Search Delivery Details
Language:	C#
Pseudo Code:	**Step 1: Employee request to View deliveries** **THEN** **Step 2: Employee clicks Delivery Menu Item (Angular)** **ON CLICK** Delivery Menu Item **THEN ON CLICK** Delivery Dropdown menu Item **THEN** **Step 3: Load Delivery Screen (Angular)** SQL Select Delivery information From Delivery table SQL Select Courier information From Courier table SQL Select Address information From Address table SQL Select Province information From Province table SQL Select City information From City table

	SQL Select Suburb information From Suburb table SQL Select OrderStatus information From OrderStatus table Retrieve the following: ○ Delivery_ID ○ Courier_TrackingNumber ○ Delivery_Date ○ Courier_Name ○ Courier_Number ○ Courier_Email ○ Address_Line1 ○ Address_Line2 ○ Address_Line3 ○ Address_PostalCode ○ Province_Description ○ City_Description ○ Suburb_Description ○ OrderStatus_Desc **THEN** display the information in a table. Display Delivery Screen [ALT STEP 3]: The system fails to load delivery page(Angular) **ON CLICK** Delivery Menu Item **THEN ON CLICK** Delivery Dropdown menu Item System fails to load delivery page. **THEN** display Error message, "There was an error loading the Delivery page. Please try again later." Terminate

6.1.2. PRODUCTS SUBSYSTEM

6.1.3. SPECIALS SUBSYSTEM

Subsystem:	10. Special
Use Case:	10.1 Add Special
Language:	C#
Pseudo Code:	**Step 1: Request to add a special [Angular]** **ON CLICK** “Add Special” button **THEN** **Step 2: Display Add Special screen [Angular]** Display Add Special Screen **ON CLICK** “Add” button **THEN** **Step 3: Capture, validate and save new special [Angular]** SQL SELECT Special FROM Special entity WHERE Special_Image == User Input, Special_Description == User Input, Special_Price == User Input, Special_StartDate == User Input, Special_EndDate == User Input If(User Input == Valid) **THEN** **[ALT Step 3]: Display validation error areas [Angular]** Else if(User Input == NotValid) Display error areas by circling the input areas that need to be entered correctly.

	[ALT Step 3]: Admin clicks on the "Cancel" button [Angular] **ONCLICK** "Cancel" button **THEN** **Step 4: The system adds the new special on to the system.** SQL ADD Special_ID**,** Special_Description, Special_Image, Special_StartDate, Special_EndDate, Special_Price TO Special entity **THEN** **Step 5: The new special is added on to the table** Display Special screen **Terminate.**

Subsystem:	10. Special
Use Case:	10.2 Search Special
Language:	C#
Pseudo Code:	**Step 1: Request to search special [Angular]** **ON CLICK** “Special” option **THEN** **Step 2: Display Special screen [Angular]** SQL SELECT All Specials FROM Special table **THEN** **Step 3: Admin enters search criteria [Angular]** Enter search criteria **THEN** **Step 4: Capture and validate search criteria [Angular]** **ON CLICK** Search input field SQL SELECT Any Special FROM Special table WHERE Special == Search Criteria **THEN** **[ALT Step 4]: Matching special could not be found [Angular]** If (Search criteria == NoMatch) Display “Enter a new search criteria”.

	Step 5: Display results of search criteria [Angular] Display Special that is similar to the search criteria **THEN** **Terminate.**

Subsystem:	10. Special
Use Case:	10.3 Update Special
Language:	C#
Pseudo Code:	**Step 1: Request to update special [Angular]** **ON CLICK** “Special” option **THEN** **Step 2: Display special screen [Angular]** SQL SELECT All Specials FROM Special table **THEN** **Step 3: Invoke Use Case 10.10 Search Special [Angular]** Search for special **THEN** **Step 4: Click on the update button [Angular]** **ONCLICK** “Update” button **THEN** **Step 5: Display Update special screen [Angular]** Display update special screen **Step 6: Input updated details [Angular]** Fill input fields **THEN** **Step 7: Admin clicks on the update button [Angular]** **ON CLICK** “Update” button

	[ALT Step 7]: Cancel update of special [Angular]
	ON CLICK “Cancel” button
	THEN
	
	Step 8: Capture and validate special [Angular]
	SQL UPDATE
	Special
	FROM Special Table
	If(Admin Input == Filled && Valid)
	Capture details
	
	[ALT Step 8]: Validation failed
	If (Admin Input == NotValid)
	THEN
	
	Step 9: Save updated special details [Angular]
	SQL UPDATE
	Special_Description, Special_Image, Special_StartDate, Special_EndDate, Special_Price
	IN Special entity
	THEN
	
	Terminate

Subsystem:	10. Special
Use Case:	10.4 Delete Special
Language:	C#
Pseudo Code:	**Step 1: Request to delete special [Angular]** **ON CLICK** “Special” option **THEN** **Step 2: Display special screen [Angular]** SQL SELECT All Specials FROM Special table **THEN** **Step 3: Invoke Use Case 10.10 Search Special [Angular]** Search for special **THEN** **Step 4: Click on the delete button [Angular]** **ONCLICK** “Delete” button **THEN** **Step 5: Confirm deletion of special [Angualr]** Display confirm delete message “Are you sure you want to delete this special”. **THEN** **Step 6: Confirm deletion of special [Angular]** **ON CLICK** “Confirm” button **THEN**

	[ALT Step 6]: Admin clicks on the "Cancel" button [Angular] **ON CLICK** "Cancel" button **Step 8: Delete special** SQL DELETE Special_ID**,** Special_Description, Special_Image, Special_StartDate, Special_EndDate, Special_Price IN Special entity **THEN** **Terminate**

Subsystem:	10. Special
Use Case:	10.5
Language:	C#
Pseudo Code:	**Step 1: Request to send promotional emails [Angular]** **ON CLICK** "Special" option **THEN** **Step 2: Display special screen [Angular]** SQL SELECT All Specials FROM Special table **THEN** **Step 3: Click on the Send Promotions button [Angular]** **ONCLICK** "Send Promotions" button. **THEN** **Step 4: Display Send promotions screen [Angular]** Display send promotions screen **THEN** **Step 5: Select the specials that the admin wants to add in the email. [Angular]** Select the relevant specials. **THEN** **Step 6: Read the information on the selected specials [Angular]** SQL READ Special_ID**,** Special_Description, Special_Image, Special_StartDate, Special_EndDate, Special_Price FROM Special table

	THEN **Step 7: The admin clicks on the Send email button [Angular]** **ONCLICK** “Send email” button. **[ALT Step 7]: Admin clicks on the cancel button [Angular]** **ONCLICK** “Cancel” button **THEN** **Step 8: The system sends the email to the customers.** SQL READ Customer_EmailAddress FROM Customer table System generates an email and sends it to the emails of the customers that are registered on the system. **THEN** **Terminate.**

CONCLUSION

This section concludes. It contained the Object-Oriented Design for the NKAP Bolting System. This included a class diagram for the entire system, state diagrams for each object in the class diagram as well as sequence diagrams for each Use Case and Pseudo Code.

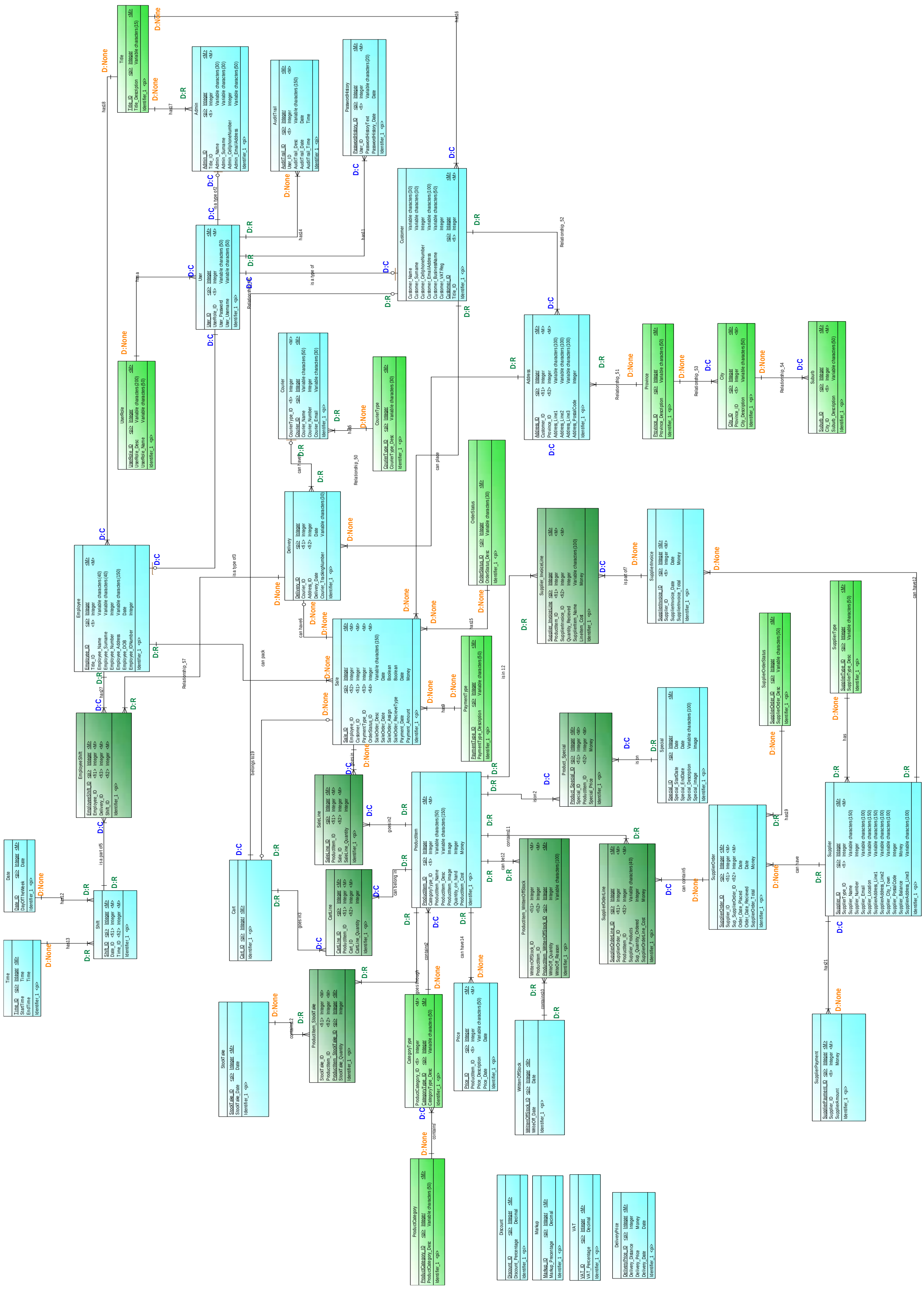
SYSTEM DATABASE DESIGN

7. Entity Relationship Model

INTRODUCTION

In this section, we will be discussing the technical Entity Relationship Diagram for NKAP Bolting Systems database.

7.1. ENTITY RELATIONSHIP DIAGRAM



CONCLUSION

This section contains the finalized third normal form technical Entity Relationship Diagram for NKAP Bolting.

8. Size Estimation of Proposed Database

INTRODUCTION

The size of the planned NKAP Bolting database will be estimated in this part, with each table in the Entity Relationship Diagram and their expected sizes multiplied by the expected rows for that table to generate a total size for that table, as well as a 5-year projection.

8.1. SIZE ESTIMATION OF PROPOSED DATABASE

Entity Name	Expected Rows	Entity (in bytes)	Total Size (In bytes)
Address	410	300	123000
Admin	3	140	420
Audit Trail	7600	182	1383200
Cart	410	10	4100
CartLine	410	40	16400
Category Type	65	70	4550
City	74	70	5180
Courier	5	110	550
Courier Type	4	40	160
Customer	410	250	102500
Date	6	18	108
Delivery	574	68	39032
Delivery Price	7	38	266
Discount	5	14	70
Employee	4	278	1112
EmployeeShift	4	40	160
Markup	7	14	98
OrderStatus	6	40	240

PasswordHistory	517	48	24816
PaymentType	1	60	60
Price	845	78	65910
Product_Special	150	40	6000
ProductCategory	13	60	780
ProductItem	845	250	211250
ProductItem_StockTake	52	40	2080
ProductItem_WrittenOffStock	48	140	6720
Entity Name	Expected Rows	Entity (in bytes)	Total Size (In bytes)
Province	9	60	540
Sale	2870	230	660100
SaleLine	2870	40	114800
Shift	5	30	150
Special	150	136	20400
StockTake	52	18	936
Suburb	740	70	6800
Supplier	93	900	83700
Supplier_InvoiceLine	83	150	12450
SupplierInvoice	83	38	3154
SupplierOrder	112	46	5152
SupplierOrderLine	112	90	10080
SupplierOrderStatus	2	60	180
SupplierPayment	90	30	3600
SupplierType	8	60	480
Time	4	18	72
Title	6	25	200
User	417	120	50040
UserRole	3	260	780
VAT	3	16	48

WrittenOffStock	48	18	864
Total Bytes			2 973 288
Total bytes (MB)			2.973288

8.2. FIVE YEAR PROJECTION

Year	Expected Growth	Expected Total Growth	Expected Size
Current	N/A	N/A	2 973 288
2022	5%	105%	3 121 952
2023	8%	113%	3 359 815
2024	13%	126%	3 746 341
2025	15%	143%	4 251 802
2026	20%	160%	4 757 261
Overall Total Bytes			4 757 261
Overall Total (MB)			4.757261

CONCLUSION

This section now contains the NKAP Bolting database will be estimated in this part, with each table in the Entity Relationship Diagram and their expected sizes multiplied by the expected rows for that table to generate a total size for that table, as well as a 5-year projection.

SYSTEM INTERFACE DESIGN

9. Input Design

INTRODUCTION

This section will cover the screens, web pages, and menus for the Website Application and Desktop Application, as well as their controls and functionality for the suggested NKAP Bolting System's functionality

9.1.1. LOGIN SUBSYSTEM
9.1.2. SUPPLIER SUBSYTEM
9.1.3. USER SUBSYSTEM
9.1.4. EMPLOYEE SUBSYSTEM
9.1.5. ADMIN SUBSYSTEM
9.1.6. CUSTOMER SUBSYSTEM
9.1.7. REPORTS SUBSYSTEM
9.1.8. DELIVERY SUBSYSTEM
9.1.9. PRODUCTS SUBSYSTEM
9.1.10. SPECIALS SUBSYSTEM

CONCLUSION

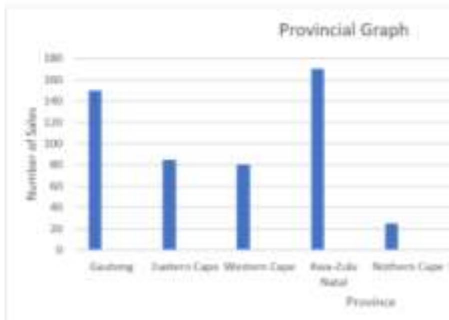
This section now contains the screens, web pages, and menus for the Website Application and Desktop Application, as well as their controls and functionality for the suggested NKAP Bolting System's functionality

10. Output Design

INTRODUCTION

In this section we will be looking at the output design of our system. The section will contain the reports and outputs that are generated by the system as well as a description of each report, when they will be run, and the possible entities involved with the reports.

Design Principles

No.	Design Issue	Design Guideline	Example
1.	Page Size	A4	N/A
2.	Page Orientation	Portrait	N/A
3.	NKAP Bolting Logo		NKAP BOLTING SPECIALISTS IN FASTENING SYSTEMS
4.	Date Report was generated	Font: Arial Bold Font Size: 12 Positioning: Left	Date Generated :21/06/21
5.	Company Name	Font: Arial Bold Font Size: 12 Positioning: Left	Company Name: NKAP Bolting CC
6.	Font	Arial	Collection
7.	Font Size	12	Collection
8.	Bar Chart		
9.	Table Columns	Font: Arial Bold Font Size: 12	**Collection/Delivery**
10.	Table Rows	Font: Arial Font Size: 12	Collection
11.	Report Heading	Font: Arial Font Size: 36	Packing Report

Report Number:	7.1
Report Name:	Generate Fast Selling Product List

12.	Font Color	Black	Collection

Short Description:	This report displays the products which had the highest number of sales over a selected period.
Frequency:	Period is adjustable. Usually generated on a monthly basis.
Special Features:	The Report period is adjustable.

Selection Criteria					
Criteria	**Default**	**Type**	**'All' Sections?**	**Multiple Sections**	**Values**
Start Date	N/A	Date picker	No	No	Any valid date
End Date	N/A	Date picker	No	No	Any valid date

Report: Fast Selling Product List				
Report Fields				
Field Name	**Format**	**Max Length**	**Justified**	**Source**
Company Name	Text	50	Left	Hardcoded
Date	dd/mm/yyyy	10	Left	System.DateTime.Now on generation of the report.
Logo	Image	N/A	Right	Hardcoded
Fast-Selling Product List	Text	30	Centre	Hardcoded
Report Summary	Text	13	N/A	Hardcoded
Ranking	Numeric	1	N/A	Hardcoded
Product	Text	50	N/A	ProductItem Entity ProductItem_Name
Full Report	Text	10	Centre	Hardcoded
Product Item ID	Numeric	6	N/A	ProductItem Entity ProductItem_ID
Product Item Name	Text	50	N/A	ProductItem Entity ProductItem_Name
Quantity Purchased	Numeric	1000	N/A	Generated from Sale ProductItem Entity
Number of Purchases	Numeric	1000	N/A	Generated from Sale entity
For the period:	Text	13	Centre	Hardcoded
Report period	dd/mm/yyyy	20	Centre	Start and End date inputted

Report Layout:

Company Name: xxxxxxxxxxxxxx

Date Generated :99/99/9999

XXXX-XXXXXXX XXXXXXX XXXX

Report Summary	
Ranking	Product
9	Xxxxxxxxxxxxxxxxxxx
9	Xxxxxxxxxxxxxxxxxxx
9	Xxxxxxxxxxxxxxxxxxx

XXXX XXXXX

Product Item ID	Product Item Name	Quantity Purchased	Number of Purchases
999999	Xxxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxxx	9999	9999
999999	Xxxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxxx	9999	9999
999999	Xxxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxxx	9999	9999
999999	Xxxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxxx	9999	9999
999999	Xxxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxxx	9999	9999
999999	Xxxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxxx	9999	9999
999999	Xxxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxxx	9999	9999
999999	Xxxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxxx	9999	9999

Example Report:

Company Name: NKAP Bolting CC

Date Generated :21/06/2021

Fast-Selling Product List

Report Summary	
Ranking	Product
1	M8 Cap Nut
2	3mm Socket Cap Screw
3	M40 Hex Head Bolt

Full Report

Product Item ID	Product Item Name	Quantity Purchased	Number of Purchases
023	3mm Socket Cap Screw	300	20
024	3.5mm Deck Screw	250	19
025	4mm Sheet Metal Screw	230	17
007	M10 Square Head Bolt	160	26
008	M16 Shoulder Head Bolt	150	23
009	M40 Hex Head Bolt	145	48
063	M8 Cap Nut	80	26
067	M3.5 Hex Nut	75	27

For the period: 03/04/2020 – 24/06/2020

Report Number:	7.2
Report Name:	Generate slow-selling product list

Short Description:	This report displays the products which had the lowest number of sales over a selected period.
Frequency:	Period is adjustable. Usually generated on a monthly basis.
Special Features:	The Report period is adjustable.

Selection Criteria					
Criteria	**Default**	**Type**	**'All' Sections?**	**Multiple Sections**	**Values**
Start Date	N/A	Date picker	No	No	Any valid date
End Date	N/A	Date picker	No	No	Any valid date

Report: Slow Selling Product List				
Report Fields				
Field Name	**Format**	**Max Length**	**Justified**	**Source**
Company Name	Text	50	Left	Hardcoded
Date	dd/mm/yyyy	10	Left	System.DateTime.Now on generation of the report.
Logo	Image	N/A	Right	Hardcoded
Slow-Selling Product List	Text	30	Centre	Hardcoded
Product Item ID	Numeric	6	N/A	ProductItem Entity ProductItem_ID
Product Item Name	Text	50	N/A	ProductItem Entity ProductItem_Name
Quantity Purchased	Numeric	1000	N/A	Generated from Sale ProductItem Entity
Number of Purchases	Numeric	1000	N/A	Generated from Sale entity
For the period:	Text	13	Centre	Hardcoded
Report period	dd/mm/yyyy	20	Centre	Start and End date inputted

Report Layout:

Company Name: xxxxxxxxxxxxxx

Date Generated :99/99/9999

XXXX-XXXXXXX XXXXXX XXXX

Product Item ID	Product Item Name	Quantity Purchased	Number of Purchases
999999	Xxxxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxxxx	9999	9999
999999	Xxxxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxxxx	9999	9999
999999	Xxxxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxxxx	9999	9999
999999	Xxxxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxxxx	9999	9999
999999	Xxxxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxxxx	9999	9999
999999	Xxxxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxxxx	9999	9999
999999	Xxxxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxxxx	9999	9999
999999	Xxxxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxxxx	9999	9999

Example Report:

Company Name: NKAP Bolting CC

Date Generated :21/06/2021

Slow-Selling Product List

Product Item ID	Product Item Name	Quantity Purchased	Number of Purchases
026	8mm Wood Screw	10	1
027	12mm Machine Screw	13	1
028	12mm Thread Cutting Screw	14	2
005	M40 Carriage Bolt	20	1
006	M8 Plow Bolt	20	2
004	M30 Eye Head Bolt	22	3
064	M12 Tee Nut	25	2
068	M20Acorn Nut	25	4

For the period: 03/04/2020 – 24/06/2020

Report Number:	7.3
Report Name:	Generate most frequent buyers report
Short Description:	This report shows the customers who have purchased most frequently during the course of one month.
Frequency:	Monthly
Special Features:	None

Report: Most Frequent Buyer				
Report Fields				
Field Name	**Format**	**Max Length**	**Justified**	**Source**
Company Name	Text	50	Left	Hardcoded
Date	dd/mm/ yyyy	10	Left	System.DateTime.Now on generation of the report.
Logo	Image	N/A	Right	Hardcoded
Most Frequent Buyer Report	Text	30	Centre	Hardcoded
Report Summary	Text	13	N/A	Hardcoded
Customer:	Text	9	N/A	Customer Entity Customer_Name Customer_Surname
Purchases Made	Numeric	4	N/A	Hardcoded
Full Report	Text	10	Centre	Hardcoded
Customer ID	Numeric	6	N/A	Customer Entity Customer_ID
Customer Name	Text	60	N/A	Customer Entity Customer_Name Customer_Surname
Business Name	Text	50	N/A	Customer Entity Customer_BusinessName
Contact Number	Numeric	10	N/A	Customer Entity Customer_CellphoneNumber
Email Address	Text	100	N/A	Customer Entity Customer_EmailAddress
No. of purchases made	Numeric	3	N/A	Generated from Sale and Customer entities

Report Layout:

Company Name: xxxxxxxxxxxxxx

Date Generated :99/99/9999

XXXX XXXXXXXX XXXXX XXXXXX

Report Summary	
Customer:	**Purchases Made**
xxxxxxxxxxxxxxxxxx	9999
xxxxxxxxxxxxxxxxxx	9999
xxxxxxxxxxxxxxxxxx	9999

XXXX XXXXX

Customer ID	Customer Name	Business Name	Contact Number	Email Address	No. of purchases made
999999	xxxxxxxxxx xxxxxxxxxx xxxxxxxxxx xxxxxxxxxx	xxxxxxxxxx xxxxxxxxxx xxxxxxxxxx xxxxxxxxxx	999 999 9999	xxxxxxxxxxxx xxxxxxxxxxxx xxxxxxxxxxxx xxxxxxxxxxxx	999
999999	xxxxxxxxxx xxxxxxxxxx xxxxxxxxxx xxxxxxxxxx	xxxxxxxxxx xxxxxxxxxx xxxxxxxxxx xxxxxxxxxx	999 999 9999	xxxxxxxxxxxx xxxxxxxxxxxx xxxxxxxxxxxx xxxxxxxxxxxx	999
999999	xxxxxxxxxx xxxxxxxxxx xxxxxxxxxx xxxxxxxxxx	xxxxxxxxxx xxxxxxxxxx xxxxxxxxxx xxxxxxxxxx	999 999 9999	xxxxxxxxxxxx xxxxxxxxxxxx xxxxxxxxxxxx xxxxxxxxxxxx	999
999999	xxxxxxxxxx xxxxxxxxxx xxxxxxxxxx xxxxxxxxxx	xxxxxxxxxx xxxxxxxxxx xxxxxxxxxx xxxxxxxxxx	999 999 9999	xxxxxxxxxxxx xxxxxxxxxxxx xxxxxxxxxxxx xxxxxxxxxxxx	999
999999	xxxxxxxxxx xxxxxxxxxx xxxxxxxxxx xxxxxxxxxx	xxxxxxxxxx xxxxxxxxxx xxxxxxxxxx xxxxxxxxxx	999 999 9999	xxxxxxxxxxxx xxxxxxxxxxxx xxxxxxxxxxxx xxxxxxxxxxxx	999
999999	xxxxxxxxxx xxxxxxxxxx xxxxxxxxxx xxxxxxxxxx	xxxxxxxxxx xxxxxxxxxx xxxxxxxxxx xxxxxxxxxx	999 999 9999	xxxxxxxxxxxx xxxxxxxxxxxx xxxxxxxxxxxx xxxxxxxxxxxx	999
999999	xxxxxxxxxx xxxxxxxxxx xxxxxxxxxx xxxxxxxxxx	xxxxxxxxxx xxxxxxxxxx xxxxxxxxxx xxxxxxxxxx	999 999 9999	xxxxxxxxxxxx xxxxxxxxxxxx xxxxxxxxxxxx xxxxxxxxxxxx	999

Example Report:

Company Name: NKAP Bolting CC

Date Generated :21/06/2021

Most Frequent Buyer Report

Report Summary	
Customer:	**Purchases Made**
S'nethemba Xulu	30
Jerome Amenigy	26
Divya Bagratee	20

Full Report

Customer ID	Customer Name	Business Name	Contact Number	Email Address	No. of purchases made
4579	S'nethemba Xulu	SAX Industries	085 478 0789	snethembax@gmail.com	22
3591	Jerome Amenigy	Rome EJ & Co Incorporated	061 349 6490	jeromea@gmail.com	18
6845	Divya Bagratee	Divs PTY (Ltd)	023 871 4841	divyab@gmail.com	16
9471	Charde Vries	DC PTY (Ltd)	085 692 0954	chardev@gmail.com	14
6218	Sonali Badrinath	Nikita's Bolts Limited	089 761 4502	sonalib@gmail.com	9
9045	Jane Doe	J&J Doe Industries	093 720 5919	janed@yahoo.com	8
8723	John Doe	J&J Doe Industries	063 401 7630	johnnyd@gmail.com	7

Report Number:	7.4
Report Name:	Generate most popular location report
Short Description:	This report displays a list of areas where there is the highest concentration of sales over a month.
Frequency:	Monthly
Special Features:	Done

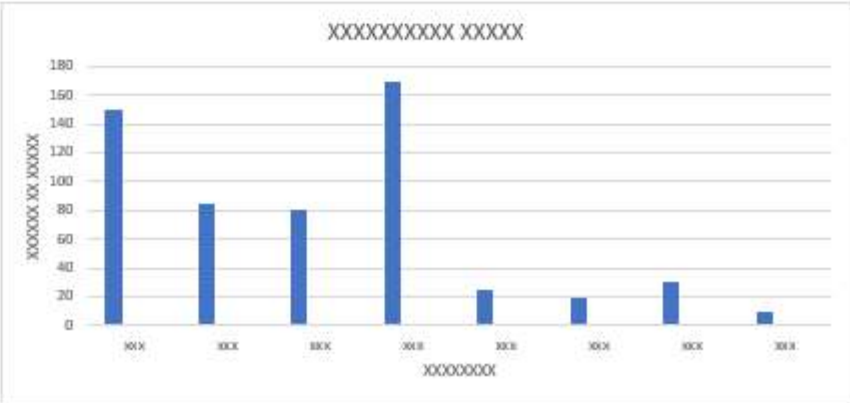
Report: Most Popular Location				
Report Fields				
Field Name	**Format**	**Max Length**	**Justified**	**Source**
Company Name	Text	50	Left	Hardcoded
Date	dd/mm/yyyy	10	Left	System.DateTime.Now on generation of the report.
Logo	Image	N/A	Right	Hardcoded
Most Popular Location Graph	Text	50	Centre	Hardcoded
Bar Chart	Chart	N/A	Centre	Generate Most Popular Location from Address Entity and Province Entity
Province	Text	50	N/A	Province Entity Proivince_Desc
Number of Sales	Numeric	1000	N/A	Generate from Sale Entity

Report Layout:

Company Name: xxxxxxxxxxxxxx

Date Generated :99/99/9999

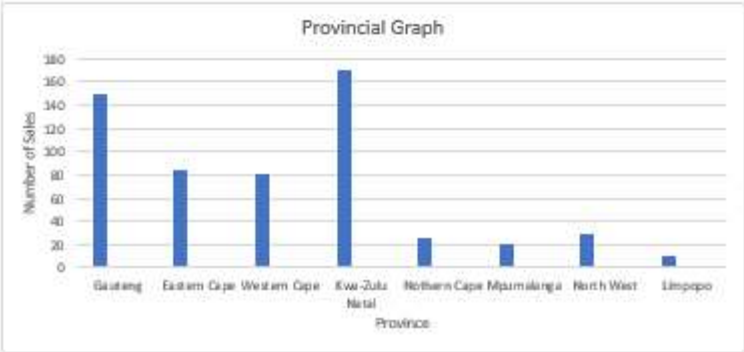
XXXX XXXXXXX XXXXXXXX XXXXXX



Example Report:

Company Name: NKAP Bolting CC
Date Generated :21/06/2021

Most Popular Location Report



Report Number:	7.5
Report Name:	Generate Weekly Sale Order List
Short Description:	This report shows a list of orders placed over a 7-day period.
Frequency:	Weekly
Special Features:	None

Report: Weekly Sale Order				
Report Fields				
Field Name	**Format**	**Max Length**	**Justified**	**Source**
Company Name	Text	50	Left	Hardcoded
Date	dd/mm/ yyyy	10	Left	System.DateTime.Now on generation of the report.
Logo	Image	N/A	Right	Hardcoded
Weekly Sale Order Report	Text	30	Centre	Hardcoded
Customer ID	Numeric	6	N/A	Customer Entity Customer_ID
Customer Name	Text	60	N/A	Customer Entity Customer_Name Customer_Surname
Business Name	Text	50	N/A	Customer Entity Customer_BusinessName
Contact Number	Numeric	10	N/A	Customer Entity Customer_CellphoneNumber
Email Address	Text	100	N/A	Customer Entity Customer_EmailAddress
SaleID	Numeric	6	N/A	Sale Entity Sale_ID
Sale Date	dd/mm/yyyy	10	N/A	Sale Entity SaleOrder_Date

Report Layout:

Company Name: xxxxxxxxxxxxxx

Date Generated :99/99/9999

XXXXXX XXXX XXXXX XXXXXX

Customer ID	Customer Name	Business Name	Contact Number	Email Address	Sale ID	Date
999999	Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx xxxxxxxxxx	Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx	999 999 9999	Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx	999999	99/99/9999
999999	Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx xxxxxxxxxx	Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx	999 999 9999	Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx	999999	99/99/9999
999999	Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx xxxxxxxxxx	Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx	999 999 9999	Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx	999999	99/99/9999
999999	Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx xxxxxxxxxx	Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx	999 999 9999	Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx	999999	99/99/9999
999999	Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx xxxxxxxxxx	Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx	999 999 9999	Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx	999999	99/99/9999
999999	Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx xxxxxxxxxx	Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx	999 999 9999	Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx	999999	99/99/9999

Example Report:

Company Name: NKAP Bolting CC

Date Generated :21/06/2021

Weekly Sale Order Report

Customer ID	Customer Name	Business Name	Contact Number	Email Address	Sale ID	Date
4579	S'nethemba Xulu	SAX Industries	085 478 0789	snethembax@gmail.com	001	14/06/2021
3591	Jerome Amenigy	Rome EJ & Co Incorporated	061 349 6490	jeromea@gmail.com	002	14/06/2021
6845	Divya Bagratee	Divs PTY (Ltd)	023 871 4841	divyab@gmail.com	003	17/06/2021
9471	Charde Vries	DC PTY (Ltd)	085 692 0954	chardev@gmail.com	004	19/06/2021
6218	Sonali Badrinath	Nikita's Bolts Limited	089 761 4502	sonalib@gmail.com	005	19/06/2021
9045	Jane Doe	J&J Doe Industries	093 720 5919	janed@yahoo.com	006	19/06/2021
8723	John Doe	J&J Doe Industries	063 401 7630	johnnyd@gmail.com	007	20/06/2021

Report Number:	7.6
Report Name:	Generate Monthly Sale Order List
Short Description:	This report shows a list of orders placed over a 4-week period.
Frequency:	Monthly
Special Features:	None

Report: Monthly Sale Order				
Report Fields				
Field Name	**Format**	**Max Length**	**Justified**	**Source**
Company Name	Text	50	Left	Hardcoded
Date	dd/mm/yyyy	10	Left	System.DateTime.Now on generation of the report.
Logo	Image	N/A	Right	Hardcoded
Monthly Sale Order Report	Text	30	Centre	Hardcoded
Customer ID	Numeric	6	N/A	<u>Customer Entity</u> Customer_ID
Customer Name	Text	60	N/A	<u>Customer Entity</u> Customer_Name Customer_Surname
Business Name	Text	50	N/A	<u>Customer Entity</u> Customer_BusinessName
Contact Number	Numeric	10	N/A	<u>Customer Entity</u> Customer_CellphoneNumber
Email Address	Text	100	N/A	<u>Customer Entity</u> Customer_EmailAddress
SaleID	Numeric	6	N/A	<u>Sale Entity</u> Sale_ID
Date	dd/mm/yyyy	10	N/A	<u>Sale Entity</u> SaleOrder_Date

Report Layout:

Company Name: xxxxxxxxxxxxxx

Date Generated :99/99/9999

XXXXXXX XXXX XXXXX XXXXXX

Customer ID	Customer Name	Business Name	Contact Number	Email Address	Sale ID	Date
999999	Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx xxxxxxxxxx	Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx	999 999 9999	Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx	999999	99/99/9999
999999	Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx xxxxxxxxxx	Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx	999 999 9999	Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx	999999	99/99/9999
999999	Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx xxxxxxxxxx	Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx	999 999 9999	Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx	999999	99/99/9999
999999	Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx xxxxxxxxxx	Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx	999 999 9999	Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx	999999	99/99/9999
999999	Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx xxxxxxxxxx	Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx	999 999 9999	Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx	999999	99/99/9999
999999	Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx xxxxxxxxxx	Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx	999 999 9999	Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx	999999	99/99/9999

Example Report:

Company Name: NKAP Bolting CC

Date Generated :21/06/2021

Monthly Sale Order Report

Customer ID	Customer Name	Business Name	Contact Number	Email Address	Sale ID	Date
4579	S'nethemba Xulu	SAX Industries	085 478 0789	snethembax@gmail.com	001	14/06/2021
3591	Jerome Amenigy	Rome EJ & Co Incorporated	061 349 6490	jeromea@gmail.com	002	14/06/2021
6845	Divya Bagratee	Divs PTY (Ltd)	023 871 4841	divyab@gmail.com	003	17/06/2021
9471	Charde Vries	DC PTY (Ltd)	085 692 0954	chardev@gmail.com	004	19/06/2021
6218	Sonali Badrinath	Nikita's Bolts Limited	089 761 4502	sonalib@gmail.com	005	19/06/2021
9045	Jane Doe	J&J Doe Industries	093 720 5919	janed@yahoo.com	006	19/06/2021
8723	John Doe	J&J Doe Industries	063 401 7630	johnnyd@gmail.com	007	20/06/2021

Report Number:	7.7
Report Name:	Generate Stock-Level Report
Short Description:	This report displays all the products available on the system as well as the quantity on hand of those products at a given time.
Frequency:	Monthly
Special Features:	None

Report: Stock-Level				
Report Fields				
Field Name	**Format**	**Max Length**	**Justified**	**Source**
Company Name	Text	50	Left	Hardcoded
Date	dd/mm/ yyyy	10	Left	System.DateTime.Now on generation of the report.
Logo	Image	N/A	Right	Hardcoded
Stock-level Report	Text	30	Centre	Hardcoded
Product Category	Text	50	N/A	ProductItem Entity ProductCategory_Desc
Category Type	Text	50	N/A	ProductItem Entity CategoryType_ID
Product Item ID	Numeric	6	N/A	ProductItem Entity ProductItem_ID
Product Item Name	Text	50	N/A	ProductItem Entity ProductItem_Name
Quantity on Hand	Numeric	1000	N/A	ProductItem Entity Quantity_on_Hand

Report Layout:

Company Name: xxxxxxxxxxxxxxx

Date Generated :99/99/9999

XXXXXXXXXX XXXXXX

Product Category	Category Type	Product Item ID	Product Item Name	Quantity on Hand
XXXXXXXXXX	xxxxxxxxxxxxxxxxxxxx	999999	Xxxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxxx xxxxxxxxxx	9999
	xxxxxxxxxxxxxxxxxxxx	999999	Xxxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxxx xxxxxxxxxx	9999
	xxxxxxxxxxxxxxxxxxxx	999999	Xxxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxxx xxxxxxxxxx	9999
XXXXXXXXXX	xxxxxxxxxxxxxxxxxxxx	999999	Xxxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxxx xxxxxxxxxx	9999
	xxxxxxxxxxxxxxxxxxxx	999999	Xxxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxxx xxxxxxxxxx	9999
	xxxxxxxxxxxxxxxxxxxx	999999	Xxxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxxx xxxxxxxxxx	9999
XXXXXXXXXX	xxxxxxxxxxxxxxxxxxxx	999999	Xxxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxxx xxxxxxxxxx	9999
	xxxxxxxxxxxxxxxxxxxx	999999	Xxxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxxx xxxxxxxxxx	9999

Example Report:

Company Name: NKAP Bolting CC

Date Generated :21/06/2021

Stock-Level Report

Product Category	Category Type	Product Item ID	Product Item Name	Quantity on Hand
Screws	Socket Cap	023	3mm Socket Cap Screw	360
	Deck	024	3.5mm Deck Screw	1150
	Sheet Metal	025	4mm Sheet Metal Screw	130
Bolts	Square Head	007	M10 Square Head Bolt	600
	Shoulder	008	M16 Shoulder Head Bolt	50
	Hex Head	009	M40 Hex Head Bolt	45
Nuts	Cap	063	M8 Cap Nut	600
	Hex	067	M3.5 Hex Nut	750

Report Number:	7.8
Report Name:	Packing Report
Short Description:	This report shows a list of orders which require packing (where the Sale order status = “Needs packing”)
Frequency:	Daily
Special Features:	None

Report: Packing Report				
Report Fields				
Field Name	**Format**	**Max Length**	**Justified**	**Source**
Company Name	Text	50	Left	Hardcoded
Date	dd/mm/ yyyy	10	Left	System.DateTime.No w on generation of the report.
Logo	Image	N/A	Right	Hardcoded
Packing Report	Text	14	Centre	Hardcoded
SaleID	Numeric	6	N/A	Sale Entity Sale_ID
Description	Text	150	N/A	Sale Entity SaleOrder_Desc
Date	dd/mm/yyyy	10	N/A	Sale Entity SaleOrder_Date
Collection/Delivery	Text	20	N/A	SaleEntity SaleOrder_RecieveType
Assigned To	Text	30	N/A	Sale Entity SaleOrder_Assign
Payment Date	dd/mm/yyyy	10	N/A	Sale Entity Payment_Date

Report Layout:

Company Name: xxxxxxxxxxxxxx

Date Generated :99/99/9999

XXXXXXX XXXXXX

Sale ID	Description	Date	Collection/Delivery	Assigned To	Payment Date
999999		99/99/9999	xxxxxxxxxx	-	99/99/9999
999999		99/99/9999	xxxxxxxxxx	-	99/99/9999
999999		99/99/9999	xxxxxxxxxx	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	99/99/9999
999999		99/99/9999	xxxxxxxxxx	-	99/99/9999
999999		99/99/9999	xxxxxxxxxx	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	99/99/9999
999999		99/99/9999	xxxxxxxxxx	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	99/99/9999
999999		99/99/9999	xxxxxxxxxx	-	99/99/9999

Example Report:

Company Name: NKAP Bolting CC

Date Generated :21/06/2021

Packing Report

Sale ID	Description	Date	Collection/Delivery	Assigned To	Payment Date
001		14/06/2021	Collection	-	14/06/2021
002		14/06/2021	Collection	-	14/06/2021
003		17/06/2021	Delivery	S'nethemba Xulu	17/06/2021
004		19/06/2021	Collection	-	19/06/2021
005		19/06/2021	Delivery	Jerome Amenigy	19/06/2021
006		19/06/2021	Delivery	Divya Bagratee	19/06/2021
007		20/06/2021	Collection	-	20/06/2021

Report Number:	7.9
Report Name:	Delivery Report
Short Description:	This report shows a list of orders which require packing (where the Sale order status = “Ready for delivery”)
Frequency:	Daily
Special Features:	None

Report: Delivery Report				
Report Fields				
Field Name	**Format**	**Max Length**	**Justified**	**Source**
Company Name	Text	50	Left	Hardcoded
Date	dd/mm/yyyy	10	Left	System.DateTime.Now on generation of the report.
Logo	Image	N/A	Right	Hardcoded
Delivery Report	Text	30	Centre	Hardcoded
SaleID	Numeric	6	N/A	Sale Entity Sale_ID
Date	dd/mm/yyyy	10	N/A	Sale Entity SaleOrder_Date
Customer ID	Numeric	6	N/A	Customer Entity Customer_ID
Customer Name	Text	60	N/A	Customer Entity Customer_Name Customer_Surname
Contact Number	Numeric	10	N/A	Customer Entity Customer_CellphoneNumber
Email Address	Text	100	N/A	Customer Entity Customer_EmailAddress
Province	Text	50	N/A	Address Entity Province_ID
City & Suburb	Text	100	N/A	City Entity City_ID Suburb Entity Suburb_ID
Address	Text	300	N/A	Address Entity Address_Line1 Address_Line2 Address_Line3 Address_PostalCode

Report Layout:

Company Name: xxxxxxxxxxxxxx

Date Generated :99/99/9999

XXXXXXXX XXXXXX

Sale ID	Date	Customer ID	Customer Name	Contact Number	Email Address	Address
999999	99/99/9999	999999	Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx xxxxxxxxxx	999 999 9999	Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx	Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx
999999	99/99/9999	999999	Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx xxxxxxxxxx	999 999 9999	Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx	Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx
999999	99/99/9999	999999	Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx xxxxxxxxxx	999 999 9999	Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx	Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx
999999	99/99/9999	999999	Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx xxxxxxxxxx	999 999 9999	Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx	Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx
999999	99/99/9999	999999	Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx xxxxxxxxxx	999 999 9999	Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx	Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx
999999	99/99/9999	999999	Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx xxxxxxxxxx	999 999 9999	Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx	Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx Xxxxxxxxxxxxxxxxxx

Example Report:

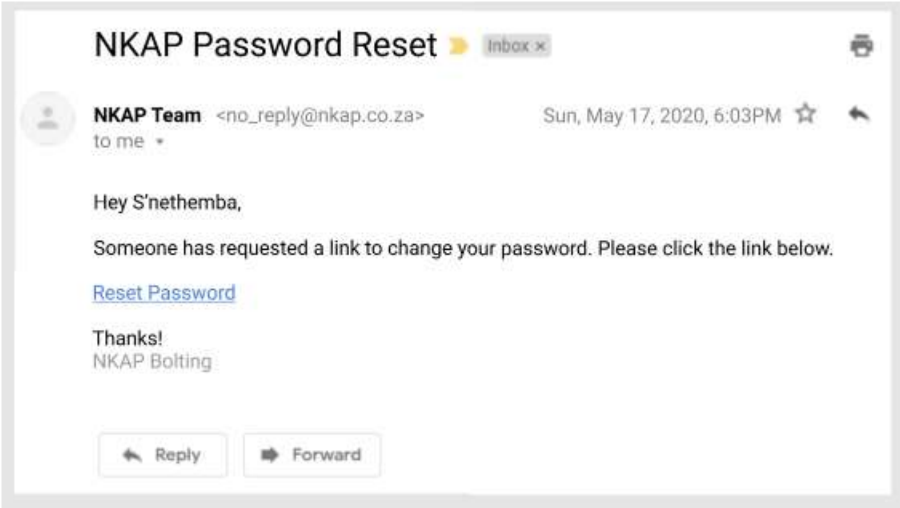
Company Name: NKAP Bolting CC

Date Generated :21/06/2021

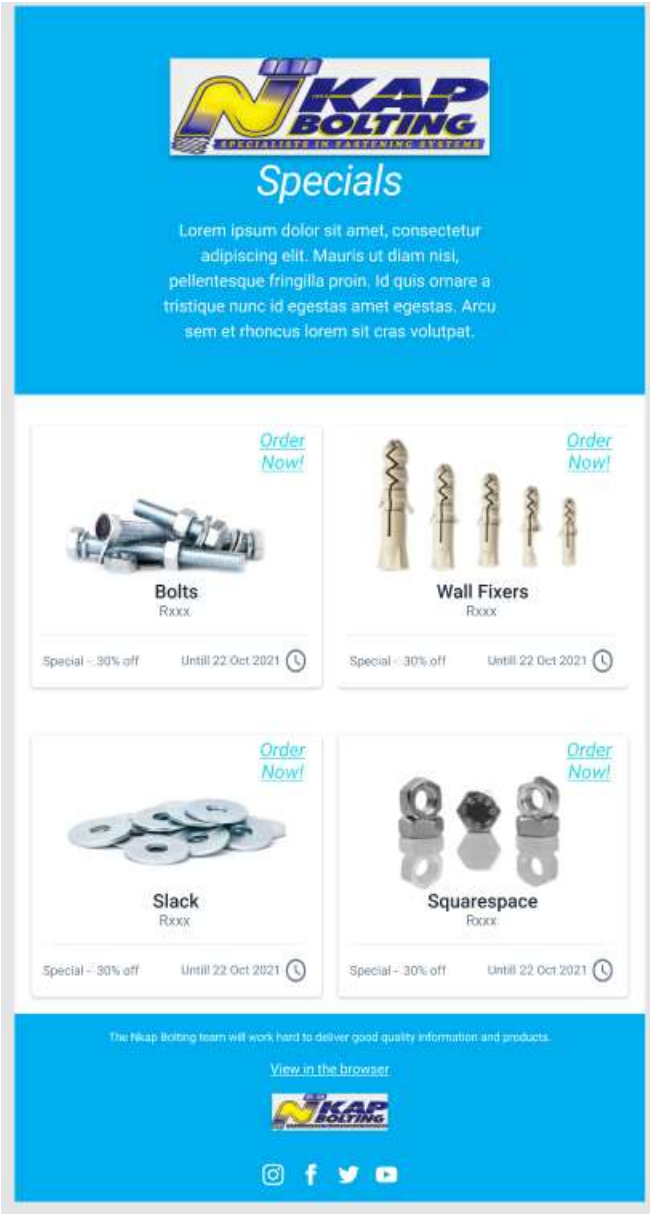
Delivery Report

Sale ID	Date	Customer ID	Customer Name	Contact Number	Email Address	Address
001	14/06/2021	4579	S'nethemba Xulu	085 478 0789	snethembax@gmail.com	313 Etsy Street Lynnwood Pretoria Gauteng
002	14/06/2021	3591	Jerome Amenigy	061 349 6490	jeromea@gmail.com	314 Etsy Street Lynnwood Pretoria Gauteng
003	17/06/2021	6845	Divya Bagratee	023 871 4841	divyab@gmail.com	315 Etsy Street Lynnwood Pretoria Gauteng
004	19/06/2021	9471	Charde Vries	085 692 0954	chardev@gmail.com	316 Etsy Street Lynnwood Pretoria Gauteng
005	19/06/2021	6218	Sonali Badrinath	089 761 4502	sonalib@gmail.com	317 Etsy Street Lynnwood Pretoria Gauteng
006	19/06/2021	9045	Jane Doe	093 720 5919	janed@yahoo.com	13 Etsy Street Lynnwood Pretoria Gauteng
007	20/06/2021	8723	John Doe	063 401 7630	johnnyd@gmail.com	13 Etsy Street Lynnwood Pretoria Gauteng

Reset Password Email

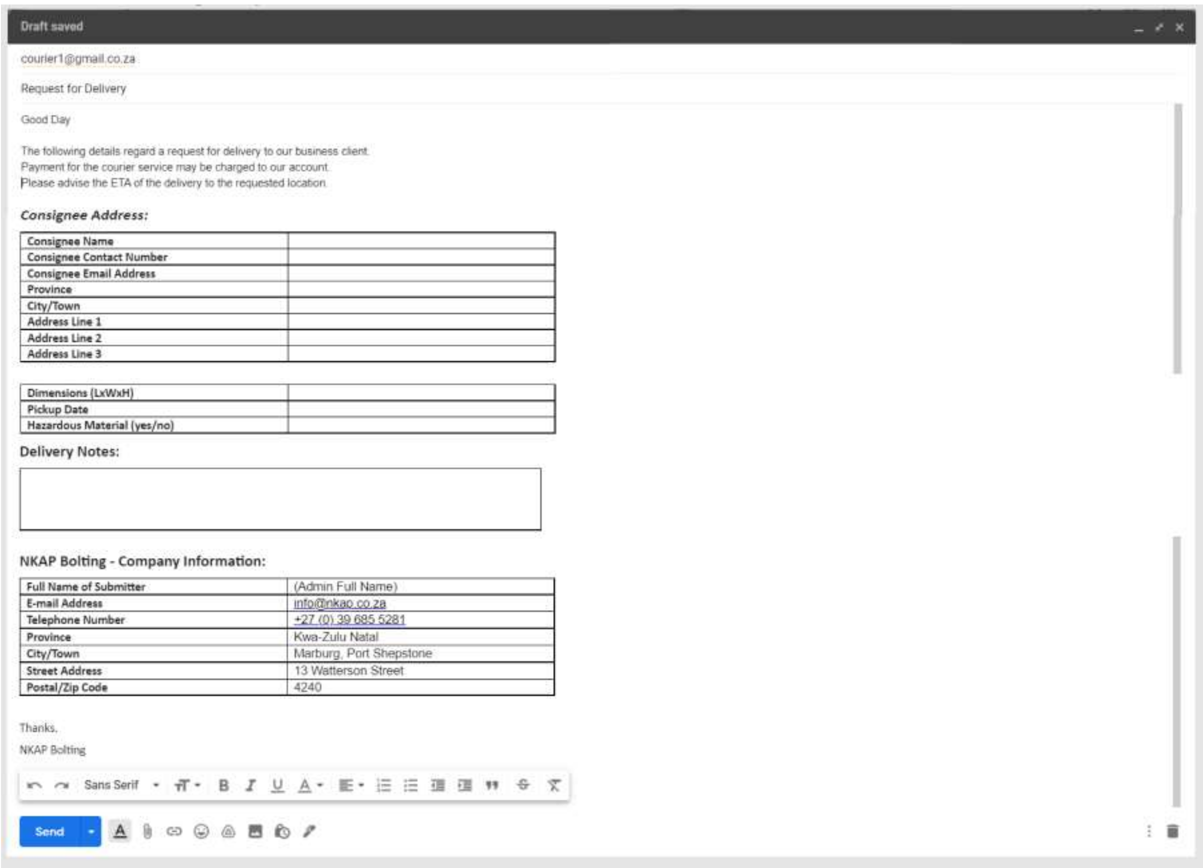


Specials Email



Order Ready for Collection Notification

Courier Delivery Request Email



Draft saved

courier1@gmail.co.za

Request for Delivery

Good Day

The following details regard a request for delivery to our business client.
Payment for the courier service may be charged to our account.
Please advise the ETA of the delivery to the requested location.

Consignee Address:

Consignee Name	
Consignee Contact Number	
Consignee Email Address	
Province	
City/Town	
Address Line 1	
Address Line 2	
Address Line 3	

Dimensions (LxWxH)	
Pickup Date	
Hazardous Material (yes/no)	

Delivery Notes:

NKAP Bolting - Company Information:

Full Name of Submitter	(Admin Full Name)
E-mail Address	info@nkap.co.za
Telephone Number	+27 (0) 39 685 5281
Province	Kwa-Zulu Natal
City/Town	Marburg, Port Shepstone
Street Address	13 Watterson Street
Postal/Zip Code	4240

Thanks,

NKAP Bolting

Sans Serif

Send

CONCLUSION

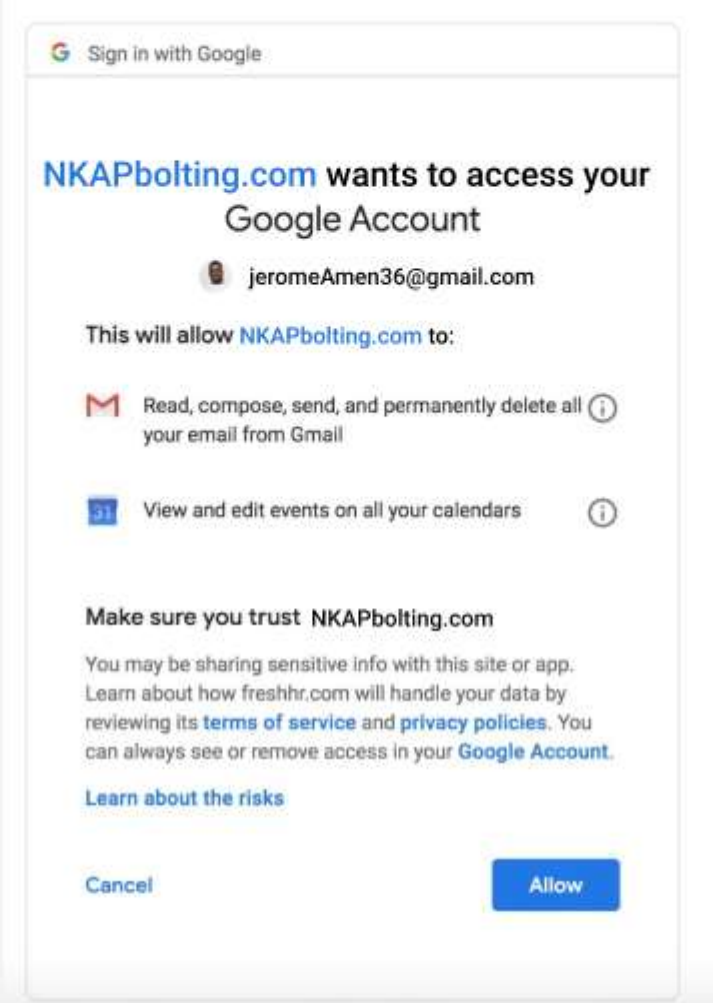
This concludes the output design section of the deliverable, the section contained descriptions of each report, when they will be run, and the possible entities involved in the reports. All reports have been listed and explained with examples

11. Other Interfaces

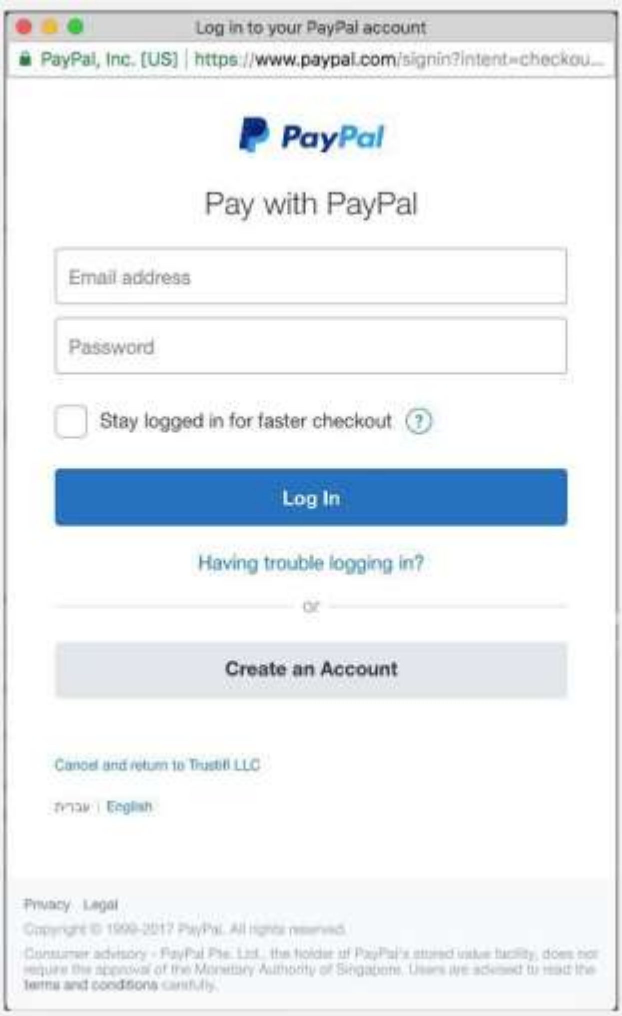
INTRODUCTION

11.1.EXTERNAL APPLICATION INTERFACE TECHNICAL

The system will make use of Gmail to send promotional emails.



The system will make use of PayPal as a virtual card system used during checkout



11.2. DESCRIPTION

CONCLUSION

12. Test Specifications

INTRODUCTION

This section will contain the detailed test specifications for 6 core use cases in our system. It includes test data, test scenarios, procedures, and the test plan. Each test scenario will account for correct and incorrect scenarios.

12.1. TEST PLAN

12.2. TEST SCENARIOS & PROCEDURES

12.3. TEST DATA

Test Case 1	
Use Case:	1.1 Login
System:	NKAP Bolting System
Sub-System:	1
Short Description:	The use case describes the event where the user wants to log in to the system. The use case begins when the user enters their Log-in details (username and password). The system will verify the details. The use case ends when the user is logged in to the system.
Pre-Conditions:	• The user should have registered previously.
Expected Functionality:	• The user should be logged in to the system if their login details are correct.

Test Scenario	**Test Data**		**Expected Result**		**Actual Result**		**Pass/Fail**
	Correct Test Data	**Incorrect Test Data**	**Correct Test Data**	**Incorrect Test Data**	**Correct Test Data**	**Incorrect Test Data**	
Sonali Badrinath wishes to login to the system. She enters these login details: Username: SonaliB Password: S_onali234	Username: SonaliB Password: S_onali234	Username: sonaliB Password: S_onali234	System logs user in to the system.	Pop-up message: Incorrect usernameor password			☐
Divya Bagratee wishes to login to	Username: DivsB Password:	Username: DIVSB	Pop-up message: Incorrect	System logs user in to the system.			☐

the system. She enters these login details: Username: DivsB Password: wassupWorld	wassupWorld	Password: Worldwassup	usernameor password				
Devine Vries wishes to login to the system. She enters these login details: Username: Devine_V01 Password: **[Blank]**	Username: Devine_V01 Password: 370Rocks	Username: Devine_V01 Password: **[Blank]**	Pop-up: All fields are required.	System logs user in to the system.			☐
Jerome Amenigy wishes to login to the system. He enters these login details: Username: **[Blank]** Password: **[Blank]**	Username: JJAmen* Password: positiveEnergy	Username: **[Blank]** Password: **[Blank]**	Pop-up: All fields are required	System logs user in to the system.			☐
S'nethemba Xulu wishes to login to	Username: sne_Xulu Password: Sne1234	Username: Sne_Xulu	System logs user in to the system.	Pop-up message: Incorrect			

☐

the system. She enters these login details: Username: sne_Xulu Password: Sne1234		Password: Sne1234		usernameor password			

Test Case 2	
Use Case:	2.5 Place Supplier Order
System:	NKAP Bolting System
Sub-System:	2
Short Description:	This use case describes the event where the admin navigates to the "**Place Supplier Order**" screen in order to generate a list of what the business wants to order from the supplier. The admin will select all the items they wish to purchase and the quantity. This list will be saved on to the system in the **SupplierOrder** and **SupplierOrderLine** tables.
Pre-Conditions:	• The admin is logged on to the system.
Expected Functionality:	• The admin should have placed a supplier order and the information should be saved in the **SupplierOrder** and **SupplierOrderLine** tables

Test Scenario	**Test Data**		**Expected Result**		**Actual Result**		**Pass/Fail**
	Correct Test Data	**Incorrect Test Data**	**Correct Test Data**	**Incorrect Test Data**	**Correct Test Data**	**Incorrect Test Data**	
Select Place Supplier order button	-	-	Display Place Supplier Order screen.	-			☐
Choose Supplier	Supplier: Makro	Supplier: Null	Display Supplier	Circle the control type in red			☐
Choose Category	Category: Bolts	Category: Null	Display category	Circle the control			

				type in red			
Select confirm button	-	-	Validation passed. Show the product items list.	Circle the control types in red			
Select product items	-	-	Select items.	-			
Add quantity	Quantity : 0 and above	Quantity : Below 0	Display quantity amount	Circle the control type in red			
Select place order	-	-	Validation passed. Save the information.	-			

Test Case 3	
Use Case:	2.6 Receive Supplier Order
System:	NKAP Bolting System
Sub-System:	2. Supplier
Short Description:	This use case describes the event where the admin receives the order from the supplier. The admin will check if the order is correct by comparing it with the invoice. Then they will capture the order on to the system. The use case ends when the product item quantity on hand is updated.
Pre-Conditions:	• The order should have been sent already.
Expected Functionality:	• The received supplier order should have been captured and the quantity on hand is updated.

Test Scenario	**Test Data**		**Expected Result**		**Actual Result**		**Pass/Fail**
	Correct Test Data	**Incorrect Test Data**	**Correct Test Data**	**Incorrect Test Data**	**Correct Test Data**	**Incorrect Test Data**	
Select order date	Order Date: 5/15/2021	Order Date: Null	Display order date	Circle the control type in red			
Enter invoice number	Invoice Number: 01055PQR	Invoice Number: Null	Display invoice number	Circle the control			

				type in red			
Select invoice date	Invoice Date: 12/6/2021	Invoice Date: Null	Display invoice date	Circle the control type in red			
Enter invoice total	Invoice Total: 12 565	Invoice Total: Null	Display invoice total	Circle the control type in red			☐
Select Add Button	-	-	Display modal	Circle the control types in red			☐
Enter product item name	Name: Screw bolt	Name: null	Display product item name	Circle the control type in red			☐
Enter the quantity	Quantity: 25	Quantity : 0 or -2	Display the quantity	Circle the control type in red			☐
Enter product item cost	Cost: 12	Cost: Null	Display the cost	Circle the control type in red			☐
Select the add button	-	-	Display the Receive Supplier order screen.	-			☐
Select cancel button	-	-	Display the Receive Supplier order screen.	-			☐
Select Save button	-	-	Save the informatio n in the tables.	Circle the control types in red			☐

Test Case 3	
Use Case:	5.2 Pack Order
System:	NKAP Bolting System
Sub-System:	Administration Back Office
Short Description:	This use case describes the event where an employee searches for orders which require packing and proceeds to pack the order. Once the employee has packed the order, they will indicate so on the system and the system will update the order status of the order appropriately in the **Sale Entity.** If the Order is for collection, the system will notify the customer that their order is ready for collection via SMS. The use case concludes when the Order has been packed and the status has been updated accordingly.
Pre-Conditions:	• The Employee must be logged in to the system. • The sale must exist on the system and have an order status of “Needs Packing”.
Expected Functionality:	• The system should update the status of the sale to “Ready for Delivery” or “Ready for Collection” depending on the Receival type of the sale in the **Sale Entity**. • The system should notify the customer via SMS that their order is ready for collection if the Receival Type is “Collection”. • The system should display a success notification once the order has been successfully packed.

Test Scenario	Test Data		Expected Result		Actual Result		Pass/Fail
	Correct Test Data	Incorrect Test Data	Correct Test Data	Incorrect Test Data	Correct Test Data	Incorrect Test Data	
1. Click the Administration Back Office Menu item on the navigation bar and click the “Pack Order” dropdown menu item.			Invoke Use Case 5.1 Search Online Sales				☐
2. Click “Pack Order” Button			Display Pack Order Screen				☐
3. Capture Pack Order Information and click	**Has this order been packed?**		• Update Order Status to	Display Error Modal if Pack			

“Proceed” button.	Check Checkbox **For Delivery** Check Checkbox **For Collection** Check Checkbox		“Ready for Collection”. • [ALT] Update Order Status to “Ready for Delivery”. • (If For Collection Checkbox = Checked) Notify Customer via SMS that their order is Ready for Collection. • Display Success Modal Once Pack Order Process is complete.	Order Process Failed.			

Test Case 5	
Use Case:	Checkout Order
System:	NKAP Bolting System
Sub-System:	6.7
Short Description:	This use case describes the event where a customer would like to Checkout all the items in their cart. The system checks if the customer is logged in. The system asks if they would like to pick up the order or if it should be delivered, which is followed by a delivery form that the customer has to fill in if they want it to be delivered. The system displays a modal with the order total. The customer requests to pay through the Virtual Card Service (PayPal). The transaction takes place on their system and once complete we save the transaction details on our system. The use case ends when the system displays a Payment successful popup.
Pre-Conditions:	The customer should have product items in their cart.

Expected Functionality:	• The system should update the **Delivery Entity** with the delivery address of the customer • The system should calculate and display the amount due • The system should load and redirect the customer to the payment gateway, in order to complete the transaction. • The system should retrieve the transaction details from the VCS system • The system should update the **Sale Entity** with the transaction information • The system should record the transaction. • The system should display a message informing the customer that the transaction was successful

Test Scenario	**Test Data**		**Expected Result**		**Actual Result**	
	Correct Test Data	**Incorrect Test Data**	**Correct Test Data**	**Incorrect Test Data**	**Correct Test Data**	**Incorre Test Data**
1. The customer clicks the 'Checkout' button			The system checks to see if the customer is logged in. The system prompts the customer to enter delivery details (Alt)	Not logged in- please sign in or register as a new user prompt shown by the system.		
3. The customer selects the delivery option and clicks the proceed button.	One radio button option selected, proceed button clicked.		The system displays delivery form. The system uses a SQL insert to capture the AddressID using the Delivery table and generates a new deliveryID, the system also uses SQL Update to update the SaleOrder_ReceiveType in the Sale Entity. The system validates the information using a SQL read from the • **Delivery entity** • **Address entity**	The customer selects the cancel button Terminate Use Case The customer chooses Collection radio button. The system uses SQL Update query to update SaleOrder_ReceiveType to 'collection' in the Sale Entity.		

			• **Province entity** • **City entity** • **Suburb entity**			
4. The customer enters their address information and click the proceed button.	4 textboxes filled in, Two dropdown box items selected, Proceed button clicked	Not all information valid. Not all required information provided.	The system captures and validates the information to the **Address entity**, **Province entity**, **Suburb entity** and **City entity** The system displays a message box with the amount due and prompts the user to select a payment method. The system displays a message box with the amount due and prompts the user to use available payment method.	System fails to update the delivery details- There was an error while updating the delivery details message.		
5. The user clicks the 'Checkout with PayPal' button			The system loads the payment gateway with the amount due.			
6. The customer makes a payment through the Virtual card services (PayPal).	Select Virtual Card Service to checkout		The system uses a SQL insert to record the transaction in the sale table and generates a new SaleID.			
7.Virtual card services: Their system displays the payment			The system uses a SQL insert to record the transaction in the sale table and generates a new SaleID.	System shows message –Payment unsuccessful as there was an error on the VCS system and the payment failed.		

successful message			The system displays a Payment successful message box.			

CONCLUSION

This section contained the test plan for the systems 6 core use cases, test scenarios and procedures, and test data.

13. Hardware & Software Requirements

INTRODUCTION

This section contains descriptions of the required hardware and software for the system to run.

Minimum Requirements:

Computer Hardware		
Hardware Requirements	**Requirements**	**Explanation**
Storage	4GB	For a Windows 10 operating system, the minimum requirements will be 4GB. Storage is required to access and store data.
RAM	2GB	Minimum system requirement for 64-bit Windows 10 is 2GB.
CPU	Intel Core i3	Required to execute system instructions.
Input Device	Keyboard & Mouse	None
Output Device	Monitor & Printer	None
Computer Software		
Software Requirements	**Requirements**	**Explanation**
Operating System	Microsoft Windows 10 x64	To manage hardware and software resources.
Database Management Server	SQL Server Management studio 2014	Required to operate the system.
Browser	Google Chrome Microsoft Edge	Browser on which system can be accessed.
Development Environment	Visual Studio 2015 (16.10.1) Visual Studio Code (1.5.7)	These environments will be used to develop and maintain the system.

Recommended Requirements:

Computer Hardware		
Hardware Requirements	**Requirements**	**Explanation**
Storage	10GB	This will be used to store the data.
RAM	4GB RAM	Enough to process the requests in the system.
CPU	Intel Core i5 or higher	Higher CPU generates better performance.
Input Device	Keyboard & Mouse	Main input device.
Output Device	Monitor & Printer	Main output device and printing out the reports.
Computer Software		
Software Requirements	**Requirements**	**Explanation**
Operating System	HTML5 Compatible operating system. Eg. Microsoft Windows	The operating system that can support the system.
Database Management Server	SQL Server Management studio 2014	Required to operate the system.
Browser	Google Chrome Microsoft Edge	Browser on which system can be accessed.
Development Environment	Visual Studio 2015 (16.10.1) Visual Studio Code (1.5.7)	These environments will be used to develop and maintain the system.

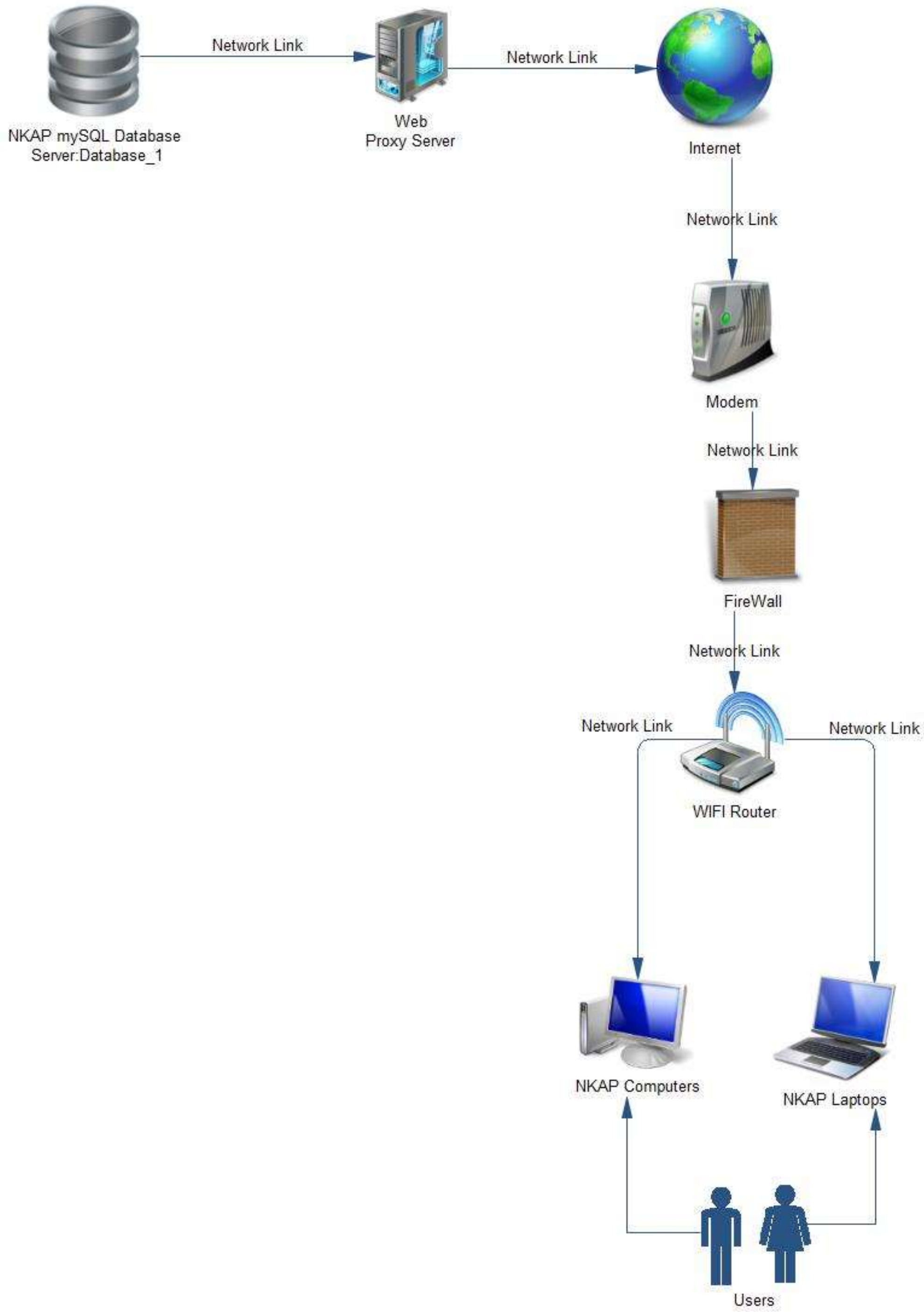
CONCLUSION

This section contained the hardware and software requirements needed to run the system. It specified the minimum requirements for a computer.

14. Network / Web Layout Specifications

INTRODUCTION

In this section we will be looking at the network specifications needed to implement and operate the system.



CONCLUSION

This section included the network specifications and layout.

15. Validation

INTRODUCTION

This section will detail the validation of each procedure for each of the requirements in the NKAP Bolting system.

15.1.1. LOGIN SUBSYSTEM

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Login	1.1 Login	Step 1: Request to login to the system.	None	Login	None	None
		Step 2: Show login screen.	None	Login	None	None
		Step 3: Enter login details	None	Login	None	None
		Step 4: User clicks on the log-in button	None	Login	None	None
		Step 5: The system validates the login details [ALT: Login details are incorrect]	User	None	None	None
		Step 6: The user logs in to the system.	None	None	Login()	None

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Forgot Password	1.2. Forgot Password	Step 1: User clicks on the forgot password link.	None	Login	None	None
		Step 2: Display the Forgot Password screen.	None	Forgot Password	None	None
		Step 3: User enters their email address.	None	Forgot Password	None	None
		Step 4: User clicks on the Submit button. [ALT: user clicks on the Cancel button]	None	Forgot Password	None	None
		Step 5: System validates the email address.	Customer Admin	None	None	None
		Step 6: The system reads the email address of the user.	None	None	None	None
		Step 7: Generate an email with a link.	None	None.	None	None

		Step 8: The system notifies the user of the email sent to them.	None	Outcome pop-up.	None	None

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Reset Password	1.3.	Step 1: User clicks on the Reset Password link.	None	None	None	None
		Step 2: Display Reset Password page.	None	Reset Password	None	None
		Step 3: Enter new password.	None	Reset Password	None	None
		Step 4: The user clicks on submit. [ALT: User clicks on Cancel button]	None	Reset Password	None	None
		Step 5: Validate new password. [ALT: Passwords don’t match]	User	Reset Password	None	None
		Step 6: The system saves the new password.	User	None	ResetPassword()	None

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Logout	1.4.	Step 1: User requests to logout	None	The current screen the user logs out from.	None	None
		Step 2: The user clicks on the logout option.			None	None
		Step 3: Display logout confirmation pop-up.	None	The current screen the user logs out from.	None	None
		Step 4: User confirms the logout.	None	The current screen the	None	None

		[ALT: User cancels log-out]		user logs out from.		
		Step 5: Logout information is added to the audit trail.	Audit Trail	None.	None	None
		Step 6: The user gets logged out of the system.	None	None	Logout()	None

15.1.2. SUPPLIER SUBSYTEM

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Add Supplier	2.1	Step 1: Request to add new supplier.	None	Supplier	None	None
		Step 2: Select add supplier button.	None	Supplier	None	None
		Step 3: Display Add Supplier screen.	None	Add Supplier	None	None
		Step 4: Enter supplier details.	None	Add Supplier	None	None
		Step 5: Admin clicks on the add button. [ALT: Admin clicks on the Cancel button]	None	Add Supplier	None	None
		Step 6: The system validates the information	Supplier	None	None	None
		Step 7: System saves the information	Supplier	None	AddSupplier()	None

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Search Supplier	2.2.	Step 1: The admin requests to search a supplier.	None	None	None	None
		Step 2: Enter search criteria	None	Supplier	None	None
		Step 3: Validate the search criteria	Supplier	None	SearchSupplier()	None
		Step 4: Read the suppliers matching the search criteria	Supplier	None	None	None
		Step 5: Display the search results.	None	Supplier	None	None

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Update Supplier	2.3.	Step 1: Request to update a supplier.	None	None	None	None
		Step 2: Invoke Use Case 2.2 Search Supplier	None	None	None	None
		Step 3: Click on the update button.	None	Supplier	None	None

		Step 4: Display Update Supplier screen	Supplier	Update Supplier	None	None
		Step 5: Enter new supplier information. [Alt: Admin clicks on the Cancel button]	None	Update Supplier	None	None
		Step 6: Validate the new information [ALT: Validation unsuccessful]	Supplier	None	None	None
		Step 7: System saves updated information. [ALT: Couldn't save updated supplier information]	Supplier	None	UpdateSupplier()	None

Functional Requirement	**Use Case No.**	**Narrative Steps**	**ERD Entities**	**Screens**	**Methods**	**Reports**
Delete Supplier	2.4.	Step 1: Admin requests to delete a supplier.	None	None	None	None

		Step 2: Invoke Use Case 2.2 Search Supplier	None	None	None	None
		Step 3: The admin clicks on the Delete button.	None	Supplier	None	None
		Step 4: Display the delete modal.	None	Delete Modal	None	None
		Step 5: Admin confirms deletion. [ALT: Admin clicks on the cancel button]	None	Delete modal	None	None
		Step 6: The system deletes the supplier from the system.	Supplier	None	DeleteSupplier()	None
		Step 7: The rows get updated in the table.	None	Supplier	None	None

Functional Requirement	**Use Case No.**	**Narrative Steps**	**ERD Entities**	**Screens**	**Methods**	**Reports**
Place Supplier Order	2.5.	Step 1: Request to place supplier order.	None	None	None	None
		Step 2: Select the Place	None	Home	None	None

		Supplier Order option.				
		Step 3: Display the Place Supplier Order screen.	None	Place Supplier Order	None	None
		Step 4: Admin selects the supplier and category.	Supplier Category Items	Place Supplier Order	None	None
		Step 5: They click on the submit button.	None	Place Supplier Order	None	None
		Step 6: Display a list of the product items in that category.	None	Place Supplier Order	None	None
		Step 7: Select the product items and quantity.	None	Place Supplier Order	None	None
		Step 8: Click on Place Order button. [ALT: Admin clicks on the Cancel button]	None	Place Supplier Order	None	None
		Step 9: System captures the items.	None	None	None	None
		Step 10: The system will save the information.	Supplier OrderLine	None	PlaceSupplierOrder()	None

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Receive Supplier Order	2.6.	Step 1: Check supplier invoice matches with items delivered.	None	None	None	None
		Step 2: Admin accepts the delivery order.	None	None	None	None
		Step 3: Request to receive supplier order.	None	None	None	None
		Step 4: Invoke use case 2.2 Search Supplier.	None	None	None	None
		Step 5: Display search results	None	None	None	None
		Step 6: Click on the Receive Order button.	None	Supplier	None	None
		Step 7: Display Receive Supplier Order screen.	None	Receive Supplier Order	None	None
		Step 8: Select supplier order they want.	None	Receive Supplier Order	None	None
		Step 9: Enter relevant order and date information	None	Receive Supplier Order	None	None

		Step 10: Click on the Add button	None	Receive Supplier Order	None	None
		Step 11: Display Add invoice items modal	None	Add Invoice Items	None	None
		Step 12: Enter the valid information.	None	Add Invoice Items	None	None
		Step 13: Click on the add button	None	Add Invoice Items	None	None
		Step 14: System save and update information	SupplierInvoiceLine	None	None	None
		Step 15: Click on save button.	None	Receive Supplier Order	None	None
		Step 16: The information is saved.	Supplier SupplierInvoice	None	None	None
		Step 17: The system updates the quantity on hand.	ProductItem	None	None	None

Functional Requirement	**Use Case No.**	**Narrative Steps**	**ERD Entities**	**Screens**	**Methods**	**Reports**
Capture Supplier Payment	2.7	Step 1: Request to capture payment	None	None	None	None

		Step 2: Invoke Use case 2.2 search Supplier	None	None	None	None
		Step 3: Display the search results.	None	Supplier	None	None
		Step 4: Click on the Capture Payment button.	None	Supplier	None	None
		Step 5: Display Capture Payment screen.	None	Capture Payment	None	None
		Step 6: Add amount paid to supplier	None	Capture Payment	None	None
		Step 7: Validate information	None	None	None	None
		Step 8: Calculate the supplier balance	Supplier	None	None	None
		Step 9: Add information to the system.	Supplier Supplier Payment	None	CapturePayment()	None

15.1.3. USER SUBSYSTEM

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Register New Customer	3.1	Step 1: Request to Register	None	None	None	None
		Step 2: Click User Icon	None	Homescreen	None	None
		Step 3: Display User Dropdown menu	None	Homescreen	None	None
		Step 4: Click Register button on login screen	None	Login	None	None
		Step 5: Display Register new customer screen	None	Register New Customer	None	None
		Step 6: Provide Customer Details	None	Register New Customer	None	None
		Step 7: Validate Customer Details [ALT] Validation failed	None	None [ALT] Validation Error Modal	Validate Customer details()	None
		Step 8: Display User Details Screen	None	User Details	None	None
		Step 9: Provide User Details	None	Register Customer	None	None
		Step 10: Validate User Details [ALT] Validation failed	None	None [ALT] Validation Error Modal	Validate User Details()	None
		Step 11: Display Confirmation Screen	None	Confirm Details	None	None
		Step 12: Confirm details [ALT] Details not confirmed	None	Confirm Details	None	None
		Step 13: Save Customer and User details in database	Customer User	None	Register Customer()	None
		Step 14: Display success modal [ALT] Registration failed	None	Success modal [ALT] Error Modal	None	None

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Search Customer	3.2	Step 1: Request to search for customer	None	None	None	None
		Step 2: Click Search Customer dropdown menu item.	None	Homescreen	None	None
		Step 3: Display search customer screen	None	Search Customer	None	None
		Step 4: Provide Search Criteria	None	Search Customer	None	None
		Step 5: Search for matching record in database	Customer	None	Search Customer()	None
		Step 6: Display matching records in datagridview [ALT] No matching records found	None	Search Customer [ALT] Error Modal	None	None
		Step 7: Select Customer to View	None	Search Customer	None	None
		Step 8: Display View Customer Screen.	None	View Customer	None	None

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Update Customer	3.3	Step 1: Customer wants to update details	None	None	None	None
		Step 2: Admin requests customer name	None	None	None	None
		Step 3: Customer Provides name	None	None	None	None
		Step 4: Admin searches for customer	Customer	Search Customer	Search Customer()	None
		Step 5: System invokes Use Case 3.2 Search Customer	None	None	None	None
		Step 6: Admin clicks update button	None	Search Customer	None	None
		Step 7: Display Update customer screen	None	Update Customer	None	None
		Step 8: Admin request customer information	None	None	None	None
		Step 9: Customer Provides information	None	None	None	None
		Step 10: Admin provides updated customer information	None	None	None	None
		Step 11: Validate Customer Information [ALT] Validation failed	None	None [ALT] Validation error modal	Validate Updated details()	None
		Step 12: Display Confirm Update modal	None	Confirm Update modal	None	None
		Step 13: Admin request confirmation from customer	None	None	None	None
		Step 14: Customer provides confirmation	None	None	None	None

		[ALT] Update not confirmed				
		Step 15: Admin clicks confirm button	None	Confirm Update modal	None	None
		Step 16: Update Customer details in database	Customer	None	Update Customer()	None
		Step 17: Display success modal [ALT] Update failed	None	Success modal [ALT] Error modal	None	None
		Step 18: Admin informs customer that information was updated	None	None	None	None

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Delete Customer	3.4	Step 1: Customer wants to delete account	None	None	None	None
		Step 2: Admin requests customer name	None	None	None	None
		Step 3: Customer Provides name	None	None	None	None
		Step 4: Admin searches for customer	Customer	Search Customer	Search Customer()	None
		Step 5: System invokes Use Case 3.2 Search Customer	None	None	None	None
		Step 6: Admin clicks delete button	None	Search Customer	None	None
		Step 7: Display confirm deletion modal	None	Confirm Deletion Modal	None	None
		Step 8: Admin request confirmation from customer	None	None	None	None
		Step 9 Customer provides confirmation [ALT] Update not confirmed	None	None	None	None
		Step 10: Admin clicks confirm button	None	Confirm Deletion modal	None	None
		Step 11: Delete Customer details from database	Customer User	None	Delete Customer()	None
		Step 12: Display success modal [ALT] Update failed	None	Success Modal [ALT] Error modal	None	None
		Step 13: Admin informs customer that their account was deleted	None	None	None	None

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
View Profile	3.5	Step 1: Customer Request to view profile	None	None	None	None
		Step 2: Customer clicks User icon then View Profile dropdown menu item	None	Homescreen	None	None
		Step 3: Read Customer information from database	Customer	None	View Profile()	None
		Step 4: Display View Profile [ALT] Error displaying profile	None	View Profile [ALT] Error Modal	None	None

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Update Profile	3.6	Step 1: Customer Request to view profile	None	None	None	None
		Step 2: Click User icon then View Profile dropdown menu item	None	Homescreen	None	None
		Step 3: Invoke Use Case 3.5 View Profile	Customer	None	View Profile()	None
		Step 4: Click Update Button	None	View Profile	None	None
		Step 5: Display Update Profile screen	None	Update Profile	None	None
		Step 6: Provided updated profile information	None	Update Profile	None	None
		Step 7: Validate Profile Information [ALT] Validation failed	None	None [ALT] Validation Error Modal	None	None
		Step 8: Display Confirm Update Modal	None	Confirm Update Modal	None	None
		Step 9: Customer Clicks confirm button [ALT] Update not confirmed	None	Confirm Update Modal	None	None
		Step 10: Update customer details in database	Customer	None	Update Customer()	None
		Step 11: Display success modal [ALT] Update failed	None	Success Modal [ALT] Error Modal	None	None

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Delete Profile	3.7	Step 1: Customer Request to view profile	None	None	None	None
		Step 2: Click User icon then View Profile dropdown menu item	None	Homescreen	None	None
		Step 3: Invoke Use Case 3.5 View Profile	Customer	None	View Profile()	None
		Step 4: Click Delete Button	None	View Profile	None	None
		Step 5: Display confirmation modal	None	Confirmation modal	None	None
		Step 6: Customer clicks confirm [ALT] Deletion not confirmed	None	Confirmation modal	None	None
		Step 7: Delete Customer from database	Customer User	None	Delete Customer()	None
		Step 8: Display success modal [ALT] Deletion failed	None	None	Success Modal [ALT] Error Modal	None

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Add User Role	3.8	Step 1: Admin Request to Add User Role	None	None	None	None
		Step 2: Click User Menu Item and Add User Role dropdown menu item	None	Homescreen	None	None
		Step 3: Display Add User Role Screen	None	Add User Role	None	None
		Step 4: Admin Provides User Role information	None	Add User Role	None	None
		Step 5: Validate User Role details	None	None	Validate User Role()	None
		Step 6: Display confirmation modal	None	Confirmation Modal	None	None
		Step 7: Admin clicks confirm button [ALT] Add User Role not confirmed	None	Confirmation Modal	None	None
		Step 8: Add User Role to database	User Role	None	Add UserRole()	None
		Step 9: Display Success Modal [ALT] Add User Role failed	None	Success Modal [ALT] Error Modal	None	None

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Search User Role	3.9	Step 1: Admin Request to Search User Role	None	None	None	None
		Step 2: Click User Menu Item and Searcg User Role dropdown menu item	None	Homescreen	None	None
		Step 3: Display Search User Role Screen	None	Search User Role	None	None
		Step 4: Admin Provides Search Criteria	None	Search User Role	None	None
		Step 5: Search for matching record in database	UserRole	None	Search UserRole ()	None
		Step 6: Display matching records in datagridview [ALT] No matching records found	None	Search User Role [ALT] Error Modal	None	None
		Step 7: Select User Role to View	None	Search User Role	None	None
		Step 8: Display User Role	UserRole	View User Role	None	None

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Update User Role	3.10	Step 1: Admin Request to Update User Role	None	None	None	None
		Step 2: Click User Menu Item and Search User Role dropdown menu item	None	Homescreen	None	None
		Step 3: Invoke Use Case 3.9 Search Role	None	None	Search User Role()	None
		Step 4: The Admin clicks the update button	None	None	None	None
		Step 5: Display Update User Role Screen	None	Update UserRole	None	None
		Step 6: Provide updated UserRole information	None	Update UserRole	None	None
		Step 7: Validate UserRole Information [ALT] Validation failed	None	None [ALT] Validation Error Modal	None	None
		Step 8: Display Confirm Update Modal	None	Confirm Update Modal	None	None
		Step 9: Admin Clicks confirm button [ALT] Update not confirmed	None	Confirm Update Modal	None	None
		Step 10: Update UserRole details in database	UserRole	None	Update UserRole()	None
		Step 11: Display success modal [ALT] Update failed	None	Success Modal [ALT] Error Modal	None	None

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Delete User Role	3.11	Step 1: Admin Request to Delete User Role	None	None	None	None
		Step 2: Click User Menu Item and Search User Role dropdown menu item	None	Homescreen	None	None
		Step 3: Invoke Use Case 3.9 Search Role	None	None	Search User Role()	None
		Step 4: The Admin clicks the delete button	None	None	None	None
		Step 5: Display confirmation modal	None	Confirmation modal	None	None
		Step 6: Admin clicks confirm [ALT] Deletion not confirmed	None	Confirmation modal	None	None
		Step 7: Delete UserRole from database	UserRole	None	Delete User Role()	None
		Step 8: Display success modal [ALT] Deletion failed	None	None	Success Modal [ALT] Error Modal	None

15.1.4. EMPLOYEE SUBSYSTEM

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Create Delivery Shift	4.1	Step 1: Admin wants to add Delivery Shift	None	None	None	None
		Step 2: admin will click on the “Create Delivery Shift” menu item.	None	Home screen	None	None
		Step 3: Display Create Delivery Shift Modal	None	Create Delivery Shift	None	None
		Step 4: Admin Provides Delivery Shift details	None	Create Delivery Shift	None	None
		Step 5: Admin clicks Add button [ALT] Admin clicks cancel	None	Create Delivery Shift	None	None
		Step 6: Display Confirm to Proceed modal	None	Proceed to Confirm Modal	None	None
		Step 7: Admin clicks Yes button [ALT] Admin clicks No button	None	Proceed to Confirm Modal	None	None
		Step 8: Validate Delivery shift information [ALT] Validation failed.	Date Time Shift	None [ALT[Validation unsuccessful modal	Validate Delivery Shift Details()	None
		Step 9: Save Delivery Shift Information	Date Time Shift	None	Save Delivery Shift Details()	None
		Step 10: Display success modal [ALT] Delivery Shift creation failed	None	Success Modal [ALT] Unsuccessful modal	None	None

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Search Delivery Shift	4.2	Step 1: Admin wants to search for a delivery shift.	None	None	None	None
		Step 2: Admin clicks Search Delivery Shift	None	Home screen	None	None
		Step 3: Display Search Delivery Shift Screen	None	Search Delivery Shift	None	None
		Step 4: Admin inputs search criteria and clicks search icon.	None	Search Delivery Shift	None	None
		Step 5: Validate Search Criteria [ALT] Error Searching for matching record	Date	None [ALT] Error Modal	Validate Search()	None
		Step 6: Read Delivery Shift details from Database	Shift Date Time Employee-Shift	None	Search Employee( )	None
		Step 7: Display Search Results in a table	None	Search Delivery Shift	None	None

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Update Delivery Shift	4.3	Step 1: Admin wants to search then Update Delivery Shift	None	Search Delivery Shift	None	None
		Step 2: System Invokes Use Case 4.2 Search Delivery Shift	None	Search Delivery Shift	Search Delivery Shift()	None
		Step 3: Display Update Delivery Shift Modal	None	Update Delivery Shift Modal	None	None
		Step 4: Provide Delivery Shift Information & clicks 'Done' button [ALT] Admin Clicks cancel button	None	Update Delivery Shift Modal	None	None
		Step 5: Validate Updated Information [ALT] Validation failed	Time Date	None [ALT] Validation Unsuccessful Modal	Validate Delivery Details()	None
		Step 6: Update Delivery Shift in Database	Time Date	None	Update Delivery Shift()	None
		Step 7: Display Success modal [ALT] Update Failed	None	Success modal [ALT] Error Modal	None	None

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Delete Delivery Shift	4.4	Step 1: The admin wans to delete a Delivery Shift	None	None	None	None
		Step 2: System Invokes Use Case 4.2 Search Delivery Shift	None	Search Delivery Screen	Search Delivery Shift()	None
		Step 3: Admin selects delete option	None	Search Delivery Screen	None	None
		Step 4: Display Confirm to Proceed modal	None	Confirm to Proceed Modal	None	None
		Step 5: Admin clicks Yes button [ALT] Admin clicks No button	None	Confirm to Proceed Modal [ALT] Search Delivery Screen	None	None
		Step 6: Delete Delivery Shift from database	Shift	None	Delete Delivery Shift()	None
		Step 7: Remove Delivery Shift from Search Delivery Shift table	None	Search Delivery Shift	None	None
		Step 8: Display success modal [ALT] Deletion Failed	None	Success Modal [ALT] Unsuccessful Modal	None	None

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Add Employee	4.5	Step 1: Admin requests to add a new employee	None	None	None	None
		Step 2: Admin clicks “Employee” menu item on the navigation bar.	None	Home screen	None	None
		Step 3: The system displays the Employee dropdown menu item	None	Home screen	None	None
		Step 4: The customer clicks the “Add Employee” dropdown menu item.	None	Home screen	None	None
		Step 5: The system responds by loading the “Add Employee” modal	None	Add Employee Modal	None	None
		Step 6: The employee communicates their details to the admin	None	None	None	None
		Step 7: Admin enters employee details and clicks the next button	None	Add Employee Modal	None	None
		Step 8: The system validates the captured employee information [Alt Step 8]: The system fails to validate the inputted information and displays a Validation Unsuccessful Modal.	Employee	None Validation Unsuccessful Modal	Validate Employee Details ()	None
		Step 9: The system enables and	None	Add Employee Modal	None	None

		disables controls in the modal				
		Step 10: The admin enters the temporary login details for the employee	None	Add Employee Modal	None	None
		Step 11: The system validates the captured information [Alt Step 11]: The system fails to validate the inputted information and displays a Validation Unsuccessful Modal.	User	None Validation Unsuccessful Modal	Validate Employee Details ()	None
		Step 12: The system displays a modal for the admin to confirm the addition of the employee information	None	Confirm to proceed modal	None	None
		Step 13: The admin selects the “Yes” button [Alt Step 13]: The admin clicks on the “No” button which makes the admin Return to step 7	None	Confirm to proceed modal Add Employee Modal	None	None
		Step 14: The system uses a SQL insert query to save the new employee information in the Employee and User Tables	Employee User	None	Add Employee ()	None
		Step 15: The system saves the new employee details in the relevant entities and displays a success modal	Employee User	Success modal	None	None

		[Alt Step 15]: The new employee could not be added to the system. The system will display the unsuccessful modal		Unsuccessful modal		

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Search Employee	4.6	Step 1: Admin wants to search for an employee	None	None	None	None
		Step 2: Admin clicks Search Employee	None	Home screen	None	None
		Step 3: Display Search Employee Screen	None	Search Employee	None	None
		Step 4: Input search criteria & clicks Search button	None	Search Employee	None	None
		Step 5: Validate Search Criteria [ALT] No matches found.	Employee	None [ALT] Error Modal	Validate Search()	None
		Step 6: Read Employee Details from Database	Employee	None	Search Employee( )	None
		Step 7: Display Search Results in a table	None	Search Employee	None	None

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Update Employee	4.7.	Step 1: Admin wants to search then Update Employee	None	Search Employee	None	None
		Step 2: System Invokes Use Case 4.6. Search Employee	None	Search Employee	Search Employee( )	None
		Step 3: Display Update Employee Modal	None	Update Employee Modal	None	None
		Step 4: Provide Employee Information & clicks 'Done' button [ALT] Admin Clicks Cancel button	None	Update Employee Modal [ALT] Search Employee	None	None
		Step 5: Validate Updated Information [ALT] Validation failed	Employee	None [ALT] Validation Unsuccessful Modal	Validate Employee Details()	None
		Step 6: Update Delivery Shift in Database	Employee	None	Update Employee( )	None
		Step 7: Display Success modal [ALT] Update Failed	None	Success modal [ALT] Unsuccessful Modal	None	None

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Delete Delivery Shift	4.8.	Step 1: The admin requests to delete an Employee	None	Search Employee	None	None
		Step 2: System Invokes Use Case 4.6. Search Employee	None	Search Employee	Search Employee( )	None
		Step 3: Admin clicks on delete option	None	Search Employee	None	None
		Step 4: Display Confirm to Proceed modal	None	Confirm to Proceed Modal	None	None
		Step 5: Admin clicks Yes button [ALT] Admin clicks No button	None	Confirm to Proceed Modal [ALT] Search Employee Screen	None	None
		Step 6: Delete Employee from database	Employee	None	Delete Employee( )	None
		Step 7: Remove Employee from Search Delivery Shift table	None	Search Employee Screen	None	None
		Step 8: Display success modal [ALT] Deletion Failed	None	Success Modal [ALT] Unsuccessful Modal	None	None

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Assign Order for Delivery/Courier	4.9.	Step 1: The admin wants to assign an order to be delivered to the customer	None	None	None	None
		Step 2: The admin clicks on the “Assign Order Delivery” sub menu item.	None	Home Screen	None	None
		Step 3: Display Assign Order Delivery Modal	None	Assign Order Delivery Modal	None	None
		Step 4: The admin clicks on the Courier button. [ALT delivery] The admin clicks on the Delivery Button	None None	Assign Order Delivery Modal None	None None	None None
		Step 5: Display Courier Modal [ALT delivery] Display Delivery Modal	None None	Courier Modal [ALT delivery] Delivery Modal	None None	None None
		Step 6: Admin wants to search for the courier. [ALT courier] Admin clicks on Cancel Button. [a. ALT delivery] Admin clicks employee name from dropdown & clicks ‘Done’ button. [b. ALT delivery] Admin clicks Cancel button.	None None None None	None [ALT Courier] Courier Modal to Assign Order Delivery [a. ALT delivery] Delivery Modal [b. ALT delivery] Delivery Modal to	None None None None	None None None None

				Assign Order Delivery		
		Step 7: System Invokes Use Case 8.4. Search Courier [ALT delivery] Display Confirmation Modal	None None	Courier Modal [ALT delivery] Confirm to Proceed Modal	Search Courier () None	None None
		Step 8: Admin clicks email address of chosen courier. [a. ALT delivery] Admin clicks on Yes button. [b. ALT delivery] Admin clicks on No button.	None None None	Courier Modal [a. ALT delivery] Confirm to Proceed Modal [b. ALT delivery] Confirm to Proceed Modal to Assign Order Delivery.	None None None	None None None
		Step 9: Directs admin to Web browser to compose email. [ALT delivery] Save delivery information in database.	None [ALT delivery] Delivery Sale Employee Employee-Shift	Web Browser Email Web Application Screen [ALT delivery] None	None None	None None
		Step 10: Admin enters order information for delivery. [a. ALT delivery] Display Success Modal	None None	Web Browser Email Web Application Screen [a. ALT delivery] Success Modal	None None	None None

		[b. ALT delivery] Order could not be assigned for delivery. [c. ALT delivery] Employee cannot be assigned to deliver order as they are not assigned to an appropriate shift. Display Error Modal.	None None	[b. ALT delivery] Error Modal [c. ALT delivery] Error Modal	None None	None None
		Step 11: Admin sends email & navigates back to Courier modal.	None	Courier modal	None	None
		Step 12: Admin clicks ‘Done’ button [ALT] Admin Clicks Cancel button	None	Courier Modal [ALT courier] Assign Order Delivery Modal	None	None
		Step 13: Display message	None	Attention Modal	None	None
		Step 14: Admin clicks on the OK button. [ALT] Admin clicks on the Cancel button.	None	Attention Modal [ALT courier] Courier modal		
		Step 15: Save courier information in the database.	Delivery Sale	None	Assign Courier Delivery ()	None
		Step 16: Display success modal [ALT] Saving of information was unsuccessful.	None	Success Modal [ALT courier] Error Modal	None	None

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Assign Delivery Shift	4.10.	Step 1: Admin wants to assign a delivery shift an employee	None	None	None	None
		Step 2: System Invokes Use Case 4.2. Search Delivery Shift	None	Search Delivery Shift	Search Delivery Shift()	None
		Step 3: Admin clicks on Assign button	None	Search Delivery Shift	None	None
		Step 4: Display Assign Delivery Shift Modal.	None	Assign Delivery Shift Modal	None	None
		Step 5: Admin selects employee name.	None	Assign Delivery Shift Modal	None	None
		Step 6: Admin clicks Proceed button [ALT] Admin clicks Cancel button	None	Assign Delivery Shift Modal [ALT] Home screen	None	None
		Step 7: Display Confirm to proceed modal	None	Confirm To Proceed modal	None	None
		Step 8: Admin clicks Yes button [ALT] Admin clicks No button	None	Confirm to Proceed [ALT] Assign Delivery Shift Modal	None	None
		Step 9: Save Employee details in database	Employee-Shift	None	Assign Employee( )	None
		Step 10: Display success modal [ALT] Failed to Assign Employee to delivery shift.	None	Success Modal [ALT] Unsuccessful Modal	None	None

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Cancel Delivery Shift	4.11.	Step 1: Admin wants to cancel a delivery shift that was assigned to an employee	None	None	None	None
		Step 2: System Invokes Use Case 4.2. Search Delivery Shift	None	Search Delivery Shift	Search Delivery Shift()	None
		Step 3: Admin clicks on Cancel button	None	Search Delivery Shift	None	None
		Step 4: Display Confirm Cancellation modal	None	Confirm To Proceed modal	None	None
		Step 5: Admin clicks Yes button [ALT] Admin clicks No button	None	Confirm to Proceed [ALT] Assign Delivery Shift Modal	None	None
		Step 6: Remove Employee details in database	Employee-Shift	None	Cancel Employee Shift()	None
		Step 7: Display success modal [ALT] Failed to remove employee information from delivery shift.	None	Success Modal [ALT] Cannot be Cancelled Modal	None	None
		Step 8: Remove employee name in search results table	None	Search Delivery Shift	None	None

15.1.5. ADMIN SUBSYSTEM

16. Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Search Online Sales	5.1	Step 1: An employee/ admin requests to search for an online sale	None	None	None	None
		Step 2: An employee/ admin Clicks Administration Back Office Menu item and clicks Search Online Sales dropdown menu item	None	Homescreen	None	None
		Step 3: Display Search Online Sales Screen	None	Search Online Sales	None	None
		Step 4: An employee/ admin selects a filter option	None	Search Online Sales	None	None
		Step 5: Retrieve matching sale records from database	Sale	None	Search Online Sale()	None
		Step 6: Display matching sale records in datagrid view	None	Online Sales	None	None
		Step 7: Employee/Admin selects sale to view	None	Online Sales	None	None
		Step 8: Display View Online Sale Screen	Sale	View Sale	None	None

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Pack Order	5.2	Step 1: An employee requests to pack an order	None	None	None	None
		Step 2: The employee clicks Administration Back Office Menu item and clicks Pack Order dropdown menu item	None	Homescreen	None	None
		Step 3: Invoke Use Case 5.1 Search Online Sales	None	None	Search Online Sale()	None
		Step 4: Employee clicks Pack Order Button	None	View Sale	None	None
		Step 5: Display Pack Order Screen	None	Pack Order	None	None
		Step 6: Employee provides Pack Order Details [ALT] Employee clicks cancel	None	Pack Order	None	None
		Step 7: System Updates Order Status to “Ready for Collection” [ALT] Order Status updated to “Ready for Delivery” (Skip to Step 9)	Sale	None	Update Order Status()	None
		Step 8: System Notifies customer that Order is ready for collection	None	None	None	None
		Step 9: Display success modal [ALT] Pack Order process failed	None	Success modal [ALT] Error modal	None	None

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Do Stock-Take	5.3	Step 1: An admin requests to do stock-take	None	None	None	None
		Step 2: The Admin clicks Administration Back Office Menu item and clicks Do Stock-Take dropdown menu item	None	Homescreen	None	None
		Step 3: Invoke Use Case 9.10 Search Product	None	None	Search Product()	None
		Step 4: Admin clicks Do-Stock Take Button	None	Products	None	None
		Step 5: Display Do Stock-Take Screen	None	Stock-Take	None	None
		Step 6: Admin provides Stock-Take Details [ALT] Admin clicks cancel button	None	Stock-Take	None	None
		Step 7: Validate Stock Take details [ALT] Validation Failed	None	None [ALT] Validation Error Modal	Validate Stock Take Details()	None
		Step 8: Display confirmation modal	None	Confirmation modal	None	None
		Step 9: Admin clicks confirm button	None	Confirmation modal	None	None
		Step 10: Save Stock Take Details	Stock Take ProductItem_StockTake	None	Save StockTake Details()	None
		Step 11: Update Quantity on hand	ProductItem	None	Update Quantity on Hand()	None
		Step 12: Display success modal [ALT] Stock Take failed	None	Success modal [ALT] Error Modal	None	None

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
View Delivery Shift Schedule	5.4	Step 1: An employee requests to view the delivery shift schedule	None	None	None	None
		Step 2: The employee clicks Administration Back Office Menu item and clicks View Delivery Shift Schedule dropdown menu item	None	Homescreen	None	None
		Step 3: System reads the shift information from the database	Shift EmployeeShift	None	Read Shift Schedule Information ()	None
		Step 4: Display Delivery Shift Schedule [ALT] Error displaying Delivery Shift Schedule	None	Schedule	None	None

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Write-off Stock	5.5	Step 1: An admin requests to Write-Off Stock	None	None	None	None
		Step 2: The admin clicks Administration Back Office Menu item and clicks Write-Off Stock dropdown menu item	None	Homescreen	None	None
		Step 3: Invoke Use Case 9.10 Search Product	None	None	Search Product()	None
		Step 4: Admin clicks Write-Off Stock Button	None	Products	None	None
		Step 5: Display Do Write-Off Stock Screen	None	Write-Off Stock	None	None
		Step 6: Admin provides Write-Off Details [ALT] Admin clicks cancel button	None	Write-Off Stock	None	None
		Step 7: Validate Write-Off details [ALT] Validation Failed	None	None [ALT] Validation Error Modal	Validate Write-Off Details()	None
		Step 8: Display confirmation modal	None	Confirmation modal	None	None
		Step 9: Admin clicks confirm button	None	Confirmation modal	None	None
		Step 10: Save Write-Off Details	WriteOff ProductItem _ WriteOff	None	Save Write-Off Details()	None
		Step 11: Update Quantity on hand	ProductItem	None	Update Quantity on Hand()	None
		Step 12: Display success modal [ALT] Write-Off failed	None	Success modal [ALT] Error Modal	None	None

16.1.1. CUSTOMER SUBSYSTEM

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
View Product	6.1	Step 1: The customer request to browse through the products by category.	None	None	None	None
		Step 2: The customer clicks on the ‘Products’ tab on the menu-strip and selects a product category from the sub-items:	None	Homepage	None	None
		Step 3: The system loads the Products screen with the following elements(showing the selected product category):	Productitem entity	Product	searchProducts()	None

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
View Products Detail	6.2	Step 1: The customer requests to search for a product.	None	Product	None	None
		Step 2: The system invokes Use Case 6.1 Search Products.	None	Product	None	None
		Step 3: The customer enters the product search criteria into the search textbox.	Productitem entity	Product	None	None
		Step 4: The customer clicks on the search icon.	None	None	validateSearch()	None
		Step 5: The system captures and validates the information against the Productitem table	Productitem enity	None	searchCategory()	None

		Step 6: The system uses a read query to read the matching entries from the Product Category entity. The system retrieves	Productitem enity	None	searchCategory()	None
		Step 7: The system displays the retrieved search results.	None	None	searchCategory()	None
		Step 8: The admin clicks on the product they were searching for.	None	None	None	None
		Step 9: The system loads the Product Details view with the following elements	None	Product Details	None	None

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Search Specials and Promotions	6.3	Step 1: The customer request to browse through the specials and promotions.	None	Promotions	None	None
		Step 2: The customer clicks on the 'Promotions' tab on the menu-strip.	None	Promotions	None	None
		Step 3: The system loads the Products screen with the following elements(showing the selected product category):	ProductItem entity, Product_Special entity, Special entity	Promotions	None	None
		Step 4: The customer enters the product search criteria into the search textbox.	None	None		None
		Step 5: The customer clicks on the search icon.	Productitem enity	None	validateSearch()	None
		Step 6: The system captures and	Productitem enity	None	searchProductItem()	None

		validates the information against the Productitem table				
		Step 7: The system uses a read query to read the matching entries from the Product Category entity. The system retrieves:	None	None	searchProductItem()	None
		Step 8: The system displays the retrieved search results.	None	None	searchProductItem()	None
		Step 9: The admin clicks on the product special they were searching for	None	Promotions	None	None
		Step 10: The system loads the SpecialProduct Details view with the following elements(showing the selected product category):	ProductItem entity, Product_Special entity, Special entity	SpecialProduct Details	None	None

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Add item to cart	6.4	Step 1: The customer requests to add a product item to cart.	None	Promotions	None	None
		Step 2: The customer clicks the ‘add to cart’ button on either the Product view, Product Details view or Promotions view	None	Promotions	confirmAddtoCart()	None
		Step 3: The system loads the Products screen with the following elements(showing the selected product category):	ProductItem entity, Product_Special entity, Special entity	Promotions	None	None

		Step 4: The customer enters the product search criteria into the search textbox.	None	None	None	None
		Step 5: The customer clicks on the search icon.	CartLine enity	None	addToCart()	None
		Step 6: The system captures and validates the information against the Productitem table	Productitem None	None	successfully Added()	None
		Step 7: The system uses a read query to read the matching entries from the Product Category entity. The system retrieves:	CartLine entity	None	updateCartIcon()	None

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Search Cart	6.5	Step 1: The customer request to see the cart.	None	None	None	
		Step 2: The customer clicks on the ‘Cart’ icon on the menu-strip.	None	None	None	
		Step 3: The system loads the Cart view with the following elements	ProductItem entity, Product_Special entity, Special entity	Cart	None	

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Remove Cart Item	6.6	Step 1: The customer request to remove an item from the cart.	None	None	None	
		Step 2: The customer clicks on the ‘Remove icon	None	None	None	

		found next to the item in the Cart view, cart table.				
		Step 3: The system confirm that the customer would like to remove the item from cart by displaying a message box:	None	Cart	ConfirmCartItemRemove()	
		Step 4: The customer clicks the Remove button	None	Cart	None	None
		Step 5: The system retrieves and removes the data to from CartLine table	CartLine entity	Cart	None	None
		Step 6: The system displays a successfully removed from cart message box	None	Cart	SuccessRemoveCartItem()	None
		Step 7: The system updates the cart icon in the page header. Using the CartLine_Quantity attribute in the CartLine entity to show the amount of items in the cart.	CartLine entity	Cart	None	None

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Checkout Order	6.7	Step 1: The customer request to go to checkout.	None	None	None	
		Step 2: The customer clicks the 'Checkout' button located in the Cart view.	None	None	None	
		Step 3: The system checks to see if the customer is logged in	None	Cart	ConfirmLogin()	
		Step 4: The system prompts the	None	Cart	None	None

		customer to enter delivery details.				
		Step 5: The system retrieves and removes the data to from CartLine table	None	Cart	None	None
		Step 6: The system displays a successfully removed from cart message box	Address entity, Province entity, Suburb entity and City entity	Cart	AddAddress ()	None
		Step 7: The system updates the cart icon in the page header. Using the CartLine_Quantity attribute in the CartLine entity to show the amount of items in the cart.	None	Cart	PaymentMessage()	None
		Step 8: The user select a Virtual Card Service method of choice.	None	Cart	None	None
		Step 9: The system loads the payment gateway with the amount due.	None	Cart	None	None
		Step 10: Virtual card services: Their system displays the payment successful message	None	Cart	None	None
		Step 11: The system records the transaction	None	Cart	None	None
		Step 12: The system displays a successfully added to cart message box and generates the invoice.	None	Cart	None	None

16.1.2. REPORTS SUBSYSTEM

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Generate fast-selling product list	7.1	Step 1: The requests to generate a report	None	None	None	None
		Step 2: The admin clicks on the Reports menu option	None	None	None	None
		Step 3: The system displays the “Reports” screen	None	Reports	None	None
		Step 4: The admin selects the “Fast-selling product list”	None	Reports	None	None
		Step 5: The admin selects the required period.	None	Reports	None	None
		Step 6: The admin clicks the “Generate” button	None	Reports	None	None
		Step 7: The system retrieves the required information from the database tables using a SQL Read.	ProductItem Sale SaleLine	None	None	None
		Step 8: The system compiles and generates the retrieved information into the Fast-selling product list.	None	None	None	None
		Step 9: The system displays the Fast-selling product list in the report panel	None	Reports	None	Fast-selling product list

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Generate Slow-selling product list	7.2	Step 1: The requests to generate a report	None	None	None	None
		Step 2: The admin clicks on the Reports menu option	None	None	None	None
		Step 3: The system displays the “Reports” screen	None	Reports	None	None

		Step 4: The admin selects the “Slow-selling product list”	None	Reports	None	None
		Step 5: The admin selects the required period.	None	Reports	None	None
		Step 6: The admin clicks the “Generate” button	None	Reports	None	None
		Step 7: The system retrieves the required information from the database tables using a SQL Read.	ProductItem Sale SaleLine	None	None	None
		Step 8: The system compiles and generates the retrieved information into the Slow-selling product list.	None	None	None	None
		Step 9: The system displays the Slow-selling product list in the report panel	None	Reports	None	Slow-selling product list

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Generate most frequent buyers report	7.3	Step 1: The requests to generate a report	None	None	None	None
		Step 2: The admin clicks on the Reports menu option	None	None	None	None
		Step 3: The system displays the “Reports” screen	None	Reports	None	None
		Step 4: The admin selects the “Most frequent buyer report”	None	Reports	None	None
		Step 5: The admin clicks the “Generate” button	None	None	None	None
		Step 6: The system retrieves the required information from the database entities	Sale Customer	None	None	None

		Step 7: The system sorts through the retrieved information and filters for the top results by using a count	None	None	None	None
		Step 8: The system displays the most frequent buyers report	None	None Reports	None	Most frequent buyers report

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Generate most popular location report	7.4	Step 1: The requests to generate a report	None	None	None	None
		Step 2: The admin clicks on the Reports menu option	None	None	None	None
		Step 3: The system displays the “Reports” screen	None	Reports	None	None
		Step 4: The admin selects the “most popular location report”	None	Reports	None	None
		Step 5: The admin clicks the “Generate” button	None	None	None	None
		Step 6: The system retrieves the required information from the database entities	Customer Sale Address Province City Suburb	None	None	None
		Step 7: The system sorts through the retrieved information and filters for the areas which had the most sales in the past month	None	None	None	None
		Step 8: The system displays the most popular location report	None	Reports	None	Most popular location report

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Generate Weekly Sale Order List	7.5	Step 1: The requests to generate a report	None	None	None	None
		Step 2: The admin clicks on the Reports menu option	None	None	None	None
		Step 3: The system displays the “Reports” screen	None	Reports	None	None
		Step 4: The admin selects the “Weekly Sale Order List”	None	Reports	None	None
		Step 5: The admin clicks the “Generate” button	None	None	None	None
		Step 6: The system retrieves the required information from the database entities	Customer Sale	None	None	None
		Step 7: The system then selects the Sales which have occurred in the past week and puts them in a list with the corresponding customers who made those sale orders also in the list.	None	None	None	None
		Step 8: The system then displays the weekly sale order list to the admin	None	Reports	None	Weekly sale order list

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Generate Monthly Sale Order List	7.6	Step 1: The requests to generate a report	None	None	None	None
		Step 2: The admin clicks on the Reports menu option	None	None	None	None
		Step 3: The system displays the “Reports” screen	None	Reports	None	None

		Step 4: The admin selects the “Monthly Sale Order List”	None	Reports	None	None
		Step 5: The admin clicks the “Generate” button	None	None	None	None
		Step 6: The system retrieves the required information from the database entities	Customer Sale	None	None	None
		Step 7: The system then selects the Sales which have occurred in the past month and puts them in a list with the corresponding customers who made those sale orders also in the list.	None	None	None	None
		Step 8: The system then displays the monthly sale order list to the admin	None	Reports	None	Monthly sale order list

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Generate Stock-Level Report	7.7	Step 1: The requests to generate a report	None	None	None	None
		Step 2: The admin clicks on the Reports menu option	None	None	None	None
		Step 3: The system displays the “Reports” screen	None	Reports	None	None
		Step 4: The admin selects the “Stock-Level Report”	None	Reports	None	None
		Step 5: The admin clicks the “Generate” button	None	None	None	None
		Step 6: The system retrieves the required information from the database entities	ProductCategory CategoryType	None	None	None

			ProductIte m			
		Step 7: The system categorizes each ProductItem into its Category type and each category type into its Product Category.	None	None	None	None
		Step 8: The system then displays the Stock-level report with the necessary columns.	None	Reports	None	Stock-Level Report

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Packing Report	7.8	Step 1: The requests to generate a report	None	None	None	None
		Step 2: The admin clicks on the Reports menu option	None	None	None	None
		Step 3: The system displays the “Reports” screen	None	Reports	None	None
		Step 4: The admin selects the Packing Report”	None	Reports	None	None
		Step 5: The admin clicks the “Generate” button	None	None	None	None
		Step 6: The system retrieves the required information from the database entities	Sale OrderStatus	None	None	None
		Step 7: The system sorts the retrieved sales from the first made to the most recent	None	None	None	None
		Step 8: The system displays the list of Sales to the admin as the Packing Report	None	Reports	None	Packing Report

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Delivery Report	7.9	Step 1: The requests to generate a report	None	None	None	None
		Step 2: The admin clicks on the Reports menu option	None	None	None	None
		Step 3: The system displays the "Reports" screen	None	Reports	None	None
		Step 4: The admin selects the Delivery Report"	None	Reports	None	None
		Step 5: The admin clicks the "Generate" button	None	None	None	None
		Step 6: The system retrieves the required information from the database entities, which have the order status of "ready for delivery" using a SQL Read.	Sale OrderStatus Delivery Address Province City Suburb Customer	None	None	None
		Step 7: The system then filters through the retrieved sales and sorts them according to SaleOrder_Date.	None	None	None	None
		Step 8: The system then displays the list of retrieved Sales to the admin in the correct order as the Delivery Report	None	Reports	None	Delivery Report

16.1.3. DELIVERY SUBSYSTEM

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Search Delivery Status	8.1	Step 1: The employee request to view deliveries.	None	None	None	None
		Step 2: The employee clicks on the 'Delivery' tab from the menu-strip.	None	None	None	None
		Step 3: The system loads the Delivery screen with the following elements:	Delivery entity, Courier entity, Address entity, Province entity, City entity, Suburb entity, OrderStatu s	Delivery	None	None

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Accept delivery	8.2	Step 1: The employee request to view deliveries.	None	None	None	None
		Step 2: The employee clicks on the 'Delivery' tab from the menu-strip.	None	Delivery	None	None
		Step 3: The system prompts the customer to accept or update the delivery:	None	Delivery	deliverystat us()	None
		Step 4: The employee clicks the confirm button.	None	Delivery	None	None
		Step 5: The system updates all the relevant information, including the OrderStatus icon in the DeliveryStatus table.	None	Delivery	None	None

		Step 6: The system displays a message box that informs the employee that the delivery has been accepted.	None	None	acceptedMessage()	None

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Add Courier	8.3	Step 1: Request to add courier	None	None	None	None
		Step 2: Click on Add Courier	None	Home Screen	None	None
		Step 3: Display Add Courier modal	None	Add Courier Modal	None	None
		Step 4: Enter courier details	None	Add Courier Modal	None	None
		Step 5: Click on Add button. [ALT: Admin clicks on Cancel button]	None	Add Courier Modal [ALT: Add Courier Modal to Home screen]	None	None
		Step 6: Display confirmation modal	None	Confirmation modal	None	None
		Step 7: Select Yes button. [ALT: Admin selects No button]	None	Confirmation modal [ALT: Confirmation Modal to Courier Modal]	None	None
		Step 8: Validate courier details [ALT: Validation was unsuccessful.]	Courier	None [ALT: Validation Unsuccessful Modal]	Validate Courier Details()	None
		Step 9: Create new courier in the system.	Courier	None	AddCourier()	None
		Step 10: Courier successfully added.	None	Success modal	None	None

		[ALT: Courier not successfully added]		[ALT: Unsuccessful Modal]		

Functional Requirement	**Use Case No.**	**Narrative Steps**	**ERD Entities**	**Screens**	**Methods**	**Reports**
Search Courier	8.4	Step 1: Request to search for a courier.	None	None	None	None
		Step 2: Click on the search courier option.	None	Home Screen	None	None
		Step 3: Display Search courier	None	Search Courier Screen	None	None
		Step 4: Enter the search criteria.	None	Search Courier Screen	None	None
		Step 5: Validate the information. [ALT: Validation failed]	Courier	None [ALT: Error Modal]	Validate Courier Name()	None
		Step 6: Read the couriers that match the search criteria	Courier	None	Search Courier()	None
		Step 7: Show the search results.	None	Search Courier Screen	None	None

Functional Requirement	**Use Case No.**	**Narrative Steps**	**ERD Entities**	**Screens**	**Methods**	**Reports**
Update courier	8.5	Step 1: Request to update courier details.	None	None	None	None
		Step 2: Invoke Use Case 8.4 Search Courier & select Update option	None	Search Courier Screen	Search Courier()	None
		Step 3: Display Update Courier Modal	None	Update Courier Modal	None	None
		Step 4: Enter new information.	None	Update Courier Modal	None	None

		[ALT: Admin clicks on the Cancel button]				
		Step 5: Validate information on the courier. [ALT: Validation failed]	Courier	None [ALT: Validation Unsuccessful Modal]	Validate Courier Details()	None
		Step 6: Save the updated information.	Courier	None	UpdateCourier()	None
		Step 7: Display success modal. [ALT: Information could not be added to the system.]	None	Success Modal [ALT: Unsuccessful Modal	None	None

16.1.4. PRODUCTS SUBSYSTEM

17. Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Add Product Category	9.1	Step 1: The admin requests to add a new product category	None	Products Screen	None	None
		Step 2: The admin clicks on the "Add" button in the Products Screen.	None	Products Screen	None	None
		Step 3: The system then displays the Add Product category modal	None	Add Product category modal	None	None
		Step 4: The admin enters the specified product category's details.	None	Add Product category modal	None	None
		Step 5: The admin clicks on the "Add" button [ALT] The user clicks on "cancel" which terminates the use case	None	Add Product category modal	None	None
		Step 6: The system captures the product category's details and validates it [ALT] The provided product category details were not in the right format; the system will outline the place where the validation error occurred.	ProductCategory	None	None	None
		Step 7: The system saves the captured information in the ProductCategory Table [ALT] The system cannot save the	ProductCategory	None [ALT] Add Product Category	None	None

		new product category; the system then displays a modal telling the admin that the addition was unsuccessful.		Error MODAL		

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Search Product Category	9.2	Step1: The admin requests to search for a product category	None	None	None	None
		Step 2: The admin clicks on the “Products” screen.	None	None	None	None
		Step 3: The system responds by loading the “Products” screen	None	Products	None	None
		Step 4: The admin selects the “Product Category” radio button.	None	Products	None	None
		Step 5: The admin enters the product category search criteria	None	Products	None	None
		Step 6: The admin clicks on the search icon.	None	Products	None	None
		Step 7: The system captures and validates the information against the Product Category table. [ALT]: The validation fails, and the admin is prompted to enter a new search query	ProductCategory	None	None	None
		Step 8: The system retrieves the	ProductCategory	None	None	None

		matching entries in the ProductCategory Table				
		Step 9: The system displays the retrieved search results in a table.	None	Products	None	None
		Step 10: The admin selects the product category	None	Products	None	None

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Update Product Category	9.3	Step 1: The admin requests to update a Product Category	None	None	None	None
		Step 2: The system invokes UC 9.2 “Search Product Category”	ProductCategory	Products	None	None
		Step 3: The admin clicks on the update button	None	Products	None	None
		Step 4: The system responds by loading the “Update Product Category” modal	None	Update Product Category modal	None	None
		Step 5: The admin edits the Product Category	None	Update Product Category modal	None	None
		Step 6: The admin clicks the “Update” button [ALT] The admin clicks on the “Cancel” button which terminates the use case	None	Update Product Category modal	None	None
		Step 7: The system captures and validates the changes made to the product category entered by the admin against the	ProductCategory	None	None	None

		product category table. [ALT] The validation of the entered information fails; the system will outline the information that does not meet the validation requirements				
		Step 8: The system updates the ProductCategory table [ALT] The system will display a modal stating the update was unsuccessful	ProductCategory	None [ALT] Update Product Category Error Modal	None	None

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Delete Product Category	9.4	Step 1: The admin requests to delete a Product Category	None	None	None	None
		Step 2: The system invokes UC 9.2 “Search Product Category”	ProductCategory	Products	None	None
		Step 3: The admin clicks on the delete button	None	Products	None	None
		Step 4: The system displays the delete product category modal	None	Delete Product Category modal	None	None
		Step 5: The admin clicks on the “Yes” button [ALT] The admin clicks on the “No” button and the use case is terminated.	None	Delete Product Category modal	None	None
		Step 6: The system removes the selected product category from the	ProductCategory	None	None	None

		product category entity. [ALT] The system will display a modal informing the admin that the deletion of the product category was unsuccessful.				
		Step 7: The system removes the deleted product category from the table in the products screen.	None	None	None	None

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Add Category Type	9.5	Step 1: The admin requests to add a new category type	None	None	None	None
		Step 2: The admin clicks on the “Add” button in the Products Screen.	None	Products	None	None
		Step 3: The system retrieves the product categories	ProductCategory	None	None	None
		Step 4: The system then displays the Add category type modal	None	Add category type modal	None	None
		Step 5: The admin selects the Product Category that he wants to add the category type to.	None	Add category type modal	None	None
		Step 6: The admin enters the category type’s details	None	Add category type modal	None	None
		Step 7: The admin clicks on the add button [ALT]: The user clicks on “cancel” which terminates the use case	None	Add category type modal	None	None
		Step 8: The system captures the	CategoryType	None	None	None

		category type's details and validates it against the CategoryType Table [ALT]: The provided category type's details were not in the right format; the system will outline the place where the validation error occurred.				
		Step 9: The system saves the captured information in the CategoryType Table [ALT] The system cannot save the new category type. The system displays a modal telling the admin that the addition was unsuccessful.	CategoryType	None [ALT] Add Category type Error modal	None	None

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Search Category Type	9.6	Step1: The admin requests to search for a category type	None	None	None	None
		Step 2: The admin clicks on the "Products" screen.	None	None	None	None
		Step 3: The system responds by loading the "Products" screen	None	Products	None	None
		Step 4: The admin selects the "Category Type" radio button.	None	Products	None	None
		Step 5: The admin enters the category type search criteria	None	Products	None	None
		Step 6: The admin clicks on the search icon.	None	Products	None	None

		Step 7: The system captures and validates the information against the category type table. [ALT]: The validation fails, and the admin is prompted to enter a new search query	CategoryType	None	None	None
		Step 8: The system retrieves the matching entries in the CategoryType Table	CategoryType	None	None	None
		Step 9: The system displays the retrieved search results in a table.	None	Products	None	None
		Step 10: The admin selects the category type	None	Products	None	None

Functional Requirement	**Use Case No.**	**Narrative Steps**	**ERD Entities**	**Screens**	**Methods**	**Reports**
Update Category Type	9.7	Step 1: The admin requests to update a Category Type	None	None	None	None
		Step 2: The system invokes UC 9.6 “Search Category Type”	CategoryType	Products	None	None
		Step 3: The admin clicks on the update button	None	Products	None	None
		Step 4: The system responds by loading the “Update Category Type” modal	None	Update Category Type modal	None	None
		Step 5: The admin edits the Category Type	None	Update Category Type modal	None	None
		Step 6: The admin clicks the “Save” button	None	Update Category Type modal	None	None

		[ALT] The admin clicks on the “Cancel” button which terminates the use case				
		Step 7: The system captures and validates the changes made to the Category Type entered by the admin against the CategoryType table. [ALT] The validation of the entered information fails; the system will outline the information that does not meet the validation requirements	CategoryType	None	None	None
		Step 8: The system updates the CategoryType table [ALT] The system will display a modal stating the update was unsuccessful	CategoryType	None [ALT] Update Category Type Error Modal	None	None

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Delete Category Type	9.8	Step 1: The admin requests to delete a Category Type	None	None	None	None
		Step 2: The system invokes UC 9.6 “Search Category Type”	CategoryType	Products	None	None
		Step 3: The admin clicks on the delete button	None	Products	None	None
		Step 4: The system displays the delete Category Type modal	None	Delete Category Type modal	None	None

		Step 5: The admin clicks on the “Yes” button [ALT] The admin clicks on the “No” button and the use case is terminated.	None	Delete Product Category modal	None	None
		Step 6: The system removes the selected Category Type from the CategoryType entity. [ALT] The system will display a modal informing the admin that the deletion of the Category Type was unsuccessful.	CategoryType	None	None	None
		Step 7: The system removes the deleted Category Type from the table in the products screen.	None	None	None	None

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Add Product Item	9.9	Step 1: The admin requests to add a new Product Item	None	None	None	None
		Step 2: The admin selects Product Item radio button	None	Products	None	None
		Step 3: The admin clicks on the “Add” button in the Products Screen.	None	Products	None	None
		Step 4: The system retrieves the Category Types	CategoryType	None	None	None
		Step 5: The system then displays the Add Product Item modal	None	Add Product Item modal	None	None
		Step 6: The admin selects the Category Type that he wants	None	Add Product Item modal	None	None

		to add the category type to.				
		Step 7: The admin enters the Product Item details	None	Add Product Item modal	None	None
		Step 8: The admin clicks on the add button [ALT]: The user clicks on “cancel” which terminates the use case	None	Add Product Item modal	None	None
		Step 9: The system captures the Product Item details and validates it against the ProductItem Table [ALT]: The provided category type’s details were not in the right format; the system will outline the place where the validation error occurred.	ProductIte m	None	None	None
		Step 10: The system saves the captured information in the ProductItem Table [ALT] The system cannot save the new product item. The system displays a modal telling the admin that the addition was unsuccessful.	ProductIte m	None [ALT] Add Product Item Error modal	None	None

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Search Product Item	9.10	Step1: The admin requests to search for a Product Item	None	None	None	None

		Step 2: The admin clicks on the “Products” screen.	None	None	None	None
		Step 3: The system responds by loading the “Products” screen	None	Products	None	None
		Step 4: The admin selects the “Product Item” radio button.	None	Products	None	None
		Step 5: The admin enters the Product Item search criteria	None	Products	None	None
		Step 6: The admin clicks on the search icon.	None	Products	None	None
		Step 7: The system captures and validates the information against the ProductItem table. [ALT]: The validation fails, and the admin is prompted to enter a new search query	ProductIte m	None	None	None
		Step 8: The system retrieves the matching entries in the ProductItem Table	ProductIte m	None	None	None
		Step 9: The system displays the retrieved search results in a table.	None	Products	None	None
		Step 10: The admin selects the Product Item	None	Products	None	None

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Update Product Item	9.11	Step 1: The admin requests to update a Product Item	None	None	None	None
		Step 2: The system invokes UC 9.10	ProductIte m	Products	None	None

		“Search Product Item”				
		Step 3: The admin clicks on the update button	None	Products	None	None
		Step 4: The system responds by loading the “Update Product Item” modal	None	Update Product Item modal	None	None
		Step 5: The admin edits the Product Item	None	Update Product Item modal	None	None
		Step 6: The admin clicks the “Save” button [ALT] The admin clicks on the “Cancel” button which terminates the use case	None	Update Product Item modal	None	None
		Step 7: The system captures and validates the changes made to the Product Item entered by the admin against the ProductItem table. [ALT] The validation of the entered information fails; the system will outline the information that does not meet the validation requirements	ProductIte m	None	None	None
		Step 8: The system updates the ProductItem table [ALT] The system will display a modal stating the update was unsuccessful	ProductIte m	None [ALT] Update Product Item Error Modal	None	None

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Delete Product Item	9.12	Step 1: The admin requests to delete a Product Item	None	None	None	None
		Step 2: The system invokes UC 9.10 “Search Product Item”	ProductIte m	Products	None	None
		Step 3: The admin clicks on the delete button	None	Products	None	None
		Step 4: The system displays the delete Product Item modal	None	Delete Product Item modal	None	None
		Step 5: The admin clicks on the “Yes” button [ALT] The admin clicks on the “No” button and the use case is terminated.	None	Delete Product Item modal	None	None
		Step 6: The system removes the selected Product Item from the ProductItem entity. [ALT] The system will display a modal informing the admin that the deletion of the Product Item was unsuccessful.	ProductIte m	None	None	None
		Step 7: The system removes the deleted Product Item from the table in the products screen.	None	None	None	None

17.1.1. SPECIALS SUBSYSTEM

18. Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Add Special	10.1 Add Special	Step 1: Request to add new special.	None	Special	None	None
		Step 2: Select add special button.	None	Special	None	None
		Step 3: Display Add Special screen.	None	Add Special	None	None
		Step 4: Enter special details.	None	Add Special	None	None
		Step 5: Admin clicks on the add button. [ALT: Admin clicks on the Cancel button]	None	Add Special	None	None
		Step 6: The system validates the information	Special	None	None	None
		Step 7: System saves the information	Special	None	AddSpecial( )	None

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Search Special	10.2	Step 1: The admin requests to search a special.	None	None	None	None
		Step 2: Enter search criteria	None	Special	None	None
		Step 3: Validate the search criteria	Special	None	None	None
		Step 4: Read the specials matching the search criteria	Special	None	None	None
		Step 5: Display the search results.	None	Special	SearchSpecial()	None

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Update Special	10.3.	Step 1: Request to update a special.	None	None	None	None

		Step 2: Invoke Use Case 10.10 Search Special	None	None	None	None
		Step 3: Click on the update button.	None	Special	None	None
		Step 4: Display Update Special screen	Special	Update Special	None	None
		Step 5: Enter new special information. [Alt: Admin clicks on the Cancel button]	None	Update Special	None	None
		Step 6: Validate the new information [ALT: Validation unsuccessful]	Special	None	None	None
		Step 7: System saves updated information. [ALT: Couldn't save updated special information]	Special	None	UpdateSpecial()	None

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Delete Special	10.4.	Step 1: Admin requests to delete a special.	None	None	None	None
		Step 2: Invoke Use Case 10.10 Search Special	None	None	None	None
		Step 3: The admin clicks on the Delete button.	None	Special	None	None
		Step 4: Display the delete modal.	None	Delete Modal	None	None
		Step 5: Admin confirms deletion. [ALT: Admin clicks on the cancel button]	None	Delete modal	None	None

		Step 6: The system deletes the special from the system.	Special	None	DeleteSpecial()	None
		Step 7: The rows get updated in the table.	None	Special	None	None

Functional Requirement	Use Case No.	Narrative Steps	ERD Entities	Screens	Methods	Reports
Send Promotional Emails	10.5	Step 1: Request to send promotional email.	None	None	None	None
		Step 2: Admin clicks on the Send Promotions button.	None	Supplier	None	None
		Step 3: Display Send Promotions screen.	None	Send Promotions	None	None
		Step 4: Select the specials.	None	Send Promotions	None	None
		Step 5: Read all the specials in the table.	Special	None	None	None
		Step 6: Click on send email. [ALT: Admin clicks on the Cancel button]	None	Send Promotions	None	None
		Step 7: System generates email.	None	None	None	None
		Step 8: Read the customer email and send email.	Customer	None	None	None

CONCLUSION

This section has the details of the validation of each procedure for each of the requirements in the NKAP Bolting system.

12. Sign-off by Client

INTRODUCTION

This section contains our Client Sign-Off

FORMALLY SIGNED PAGE OF CLIENTS AGREEMENT TO CONTENT

INTRODUCTION

Deliverable Name: Technical Specification

Deliverable Number: 4 of 12

Delivered on Date:

Client Name: NKAP Bolting

Approving Client Representative: Mr Anandh Badrinath

Project Manager: Divya Bagratee

Group Leader: Jerome Amenigy

Deliverable Sign-Off

I ______________________________, as the approving client representative on behalf of NKAP Bolting, declare that I have read this document in its entirety and hereby state that I understand and fully agree with its content.

____

Anandh Badrinath

____

Signed on This Date

CONCLUSION

This section concludes. It contained the Client Sign-Off.

13. Complexity

INTRODUCTION

The complexity matrix has been filled out according to the capabilities of the DS3 Solutions members.

INF370 Complexity Matrix 2021

Topic	Level		Marks	MAX
1. Special GUI	For online applications: Responsive web design For desktop applications: Form design according to design principles (Schneiderman's golden rule on navigation applies here)	*	3	**42**
	Appropriate use of grids/tables	▯	3	
	Appropriate use of tabs/links	▯	3	
	Use of graphs in an appropriate business context	▯	4	
	The storage and display of graphical information, like photos with a good business reason	▯	3	
	Working e-mail automatically generated from the database in an appropriate business context	▯	2	
	SMS messages automatically generated from the system in an appropriate business context	▯	2	
	Extensive user-friendly search facility	▯	3	
	At least one use of a tree to display data from the database	▯	3	
	Able to dynamically modify a data tree structure and in doing so adjusting the data in the database		4	
	At least one use of a calendar view of data (not a date/time picker; not a plug-in such as Google calendar)		3	
	Uploading a file into the system with appropriate business reason	▯	3	

	The use of audio/video in an appropriate business context		3	
	At least one use of an administrator configurable timer in an appropriate business context		3	
2. Database access	At least 30 tables used (4 member groups) or 40 tables used (5 member groups)	*	6	**15**
	Full referential integrity on all tables	*	6	
	At least one use of master-detail table relationships (Schneiderman's golden rule on system status applies here)	*	3	
3. Reports	At least 3 simple list reports in a reporting tool (no control breaks, no graphs, single table)	*	3	**15**
	At least 2 transactional report with 2 or more control breaks (with heading and calculated values/totals, multiple tables)	*	6	
	At least 1 report with adjustable criteria		3	
	At least 1 management report using a graph		3	
4. Flexibility	All data that can change in future should not be hard coded but maintained in a sub-module of the system (e.g. Lookup tables)		6	**12**
	Some business rules are not hard coded but maintained in a sub-module of the system.		6	
5. Error handling	**All** system-generated errors are trapped, and consistent, user-friendly error messages are displayed		6	**12**
	Appropriate data validation on **all** input fields		6	
6. Help	At least one menu item or other control that opens up a complete help document (HTML, PDF, Help-file)		3	**15**
	Extensive context-sensitive help. E.g. calling Help on a specific screen/function will automatically open the specific help for that screen/function.		6	

Topic	Level		Marks	MAX
	Search Facility on Help		3	
	Extensive use of hints		3	

7. Security	Logon screen with user ID and password and fixed user profiles		3	**1 3**
	Applying two factor authentication with applicable business reason.		3	
	Encrypted passwords in database	*	1	
	Flexible user profiles (i.e. you can dynamically add user profiles that will enable/disable access to certain parts of the system)		6	
8. Audit Trail	An audit trail of all transactions in the system showing at least date, time, user, transaction type, critical data (such as amount and quantity of transaction)		6	**9**
	Able to search the audit trail on any of the following: date, user, transaction type		3	
9. Deployment	For a desktop application: Fully functional installation disks that take care of application installation requirements (install and uninstall)		3	**1 5**
	For an online application: Deployment of application to a publicly accessible web server		3	
	For a mobile application: Deployment to an App Market place (such as the PlayStore or the AppStore)		6	
	Deployment of the database to a remote database server		3	
10. Backup and Restore	A backup and restore subsystem exists that backup/restore all data (system may exit during restore)		3	**3**
11. Import/Export Data	Able to open Word or Excel and automatically place data in it based on the parameters provided (with a good business reason)		6	**9**
	XML or JSON: At least 1 XMLor JSON file for Importing or Exporting of data (with good business reason)		3	
	Simple Link to an external INPUT device using plug- and-play technology, such as a swipe card reader, bar code reader, etc. or a native component such as a QR reader, a GPS component, etc.		3	
	Loose Link to an external INPUT device using device specific software. Data or images must seamlessly be stored in the database, but device specific software is visible to the user. (This could include a digital camera, scanner, voice recording device, thump print reader, etc.)		6	

12. External INPUT device	Tight Link to an external INPUT device using device specific software. Data or images must seamlessly be stored in the database, but device specific software is **not** visible to the user. (This could include a digital camera, scanner, voice recording device, thump print reader, etc.)		9	**18**
13. External APPLICATION / Services	Integrate an existing web service into your application (with good business reason)		3	**9**
	A fully functional link to an installed external application system exists and the interface must be shown to work on the external system. Note that this excludes Microsoft Office Applications		6	

Topic	Le vel		Marks	MAX
14. Multiplatform processing for an appropriate business reason	Appropriate business use of static views on an alternative platform.	▯	3	27
	Appropriate use of dynamic views on an alternative platform (i.e. data is displayed from the system's database)	▯	3	
	Appropriate use of substantial dynamic views on an alternative platform (i.e. both reading and writing data from the system's database)	▯	9	
	Uploading a file through an alternative platform onto the system's database.		3	
	Substantial processing on a third platform (i.e. both reading and writing data from the system's database)		9	
15. Programming Principles	The use of a data layer to facilitate interaction between your database and your business layer	▯	3	12
	The use of an API to facilitate interaction between your business layer and your presentation layer	▯	6	
	Comprehensive use of stored procedures and/or triggers and/or jobs.	▯	3	
16. Innovative addition to the system	Any very advanced innovative addition to the system (e.g. machine learning, AI, block chain, text mining, IOT, etc.)	#	1-9	9

Maximum Complexity Marks	**222**
Complexity Marks Required (5 members in team)	150
Selected Complexity Marks	**165**

CONCLUSION

DS3Solutions has selected 165 complexity matrix marks out of the 222 marks.

14. General

INTRODUCTION

This section contains the team sign-off.

14.1. TEAM SIGN-OFF

We, Group 1 – DS^3 Solutions, hereby declare that this work is our own and we are fully satisfied with the work contained within this document. We declare that all members of this group have contributed equally and fairly to this deliverable and submit this document as our final and complete Functional Specification.

Divya Bagratee – Project Manager

Jerome Amenigy – Team Leader

S'nethemba Xulu

Chardé Vries

Sonali Badrinath

CONCLUSION

CONCLUSION

This section concludes. It contained the Team Sign-Of

CONCLUSION

CONCLUSION

This document contained our technical specification for the NKAP Bolting System. We looked at the system architecture design which included a Full set of Use Case Diagrams and Design Use Case Documentation of each Use Case. We also looked at Object-Oriented Design and System Database Design which included the Entity Relationship model and the size estimation of the proposed database. We then looked at the System Interface Design which included the inputs, outputs as well as other interfaces used in our system and lastly the Package Design Specifications which included the test specifications, hardware and software requirements, network/web layout specifications, and validations.