



Software Requirements Specification

for

AppXilon Online Dynamic Ordering System

Version 2.0 approved

Prepared by Team AppXilon

Faculty of Computer Science and Information Technology, UPM

August 13, 2021

GROUP MEMBERS:

Name	Student ID
Nurul Lydia binti Abd Latif	197994
Nur Hazimah binti Hamdan	198919
Syazwani binti Rahimin	196359
Nur Adila Aminah binti Ramli	198067
Muhammad Ashraf Hafizi bin Ros Mahadi	196186
Ammar Hakim bin Redzuwan	198394

DOCUMENT APPROVAL

Review	Name	Date
Reviewed By: _____ (Supervisor)	Mohd Hafeez Bin Osman	August 13, 2021
Approved By: _____ (Client)	Mohammad Shahzar Bin Mohammad Idris	August 13, 2021

Table of Contents

1	Introduction	7
1.1	Purpose	7
1.2	Scope	7
1.3	Definitions and Acronyms	8
1.4	References	8
1.5	Overview.....	8
2	Overall Description.....	9
2.1	Product Perspective.....	9
2.2	Product Functions.....	9
2.3	User Characteristics.....	11
2.4	Constraints	12
2.5	Assumptions and Dependencies.....	12
2.6	Apportioning of Requirements.....	12
3	Specific Requirements	14
3.1	External Interfaces Requirements	14
3.1.1	User Interface Requirements	14
3.1.2	Hardware Interface Requirements	15
3.1.3	Software Interface Requirements.....	15
3.1.4	Communication Interface Requirements	15
3.2	Functional Requirement.....	15
3.2.1	Customer Use Case Diagram	16
3.2.2	Manager Use Case Diagram.....	17
3.2.3	Admin Use Case Diagram.....	18
3.3	Use Case Description	19
3.3.1	UC_01 Register	19

3.3.2	UC_02 Login.....	20
3.3.3	UC_03 Manage Profile.....	22
3.3.4	UC_04 Chat with Chatbot	23
3.3.5	UC_05 View Catalogue.....	24
3.3.6	UC_06 View Personalized Recommendations	26
3.3.7	UC_07 Make Table Bookings	27
3.3.8	UC_08 Make Purchase.....	28
3.3.9	UC_09 Update Cart	29
3.3.10	UC_10 Checkout	30
3.3.11	UC_11 Give Review.....	32
3.3.12	UC_12 View FAQ	33
3.3.13	UC_13 View Booking Details	34
3.3.14	UC_14 Manage Catalogue.....	35
3.3.15	UC_15 Manage Table Criteria	42
3.3.16	UC_16 Manage Order Status.....	43
3.3.17	UC_17 Manage Promotion.....	44
3.3.18	UC_18 Manage Review	45
3.3.19	UC_19 Manage FAQ	46
3.3.20	UC_20 Generate Report	48
3.3.21	UC_21 View Analytics.....	50
3.3.22	UC_22 Track Customer Sentiment	50
3.3.23	UC_23 View Customer Analytics	51
3.3.24	UC_24 View Sales Analytics.....	52
3.3.25	UC_25 View Product Analytics	53
3.3.26	UC_26 Manage Payment.....	54
3.3.27	UC_27 Manage Business Hours.....	55
3.3.28	UC_28 Manage Term Condition.....	56

3.3.29	UC_29 Manage Reminder Reservations.....	57
3.3.30	UC_30 Manage Reservation Form.....	58
3.3.31	UC_31 Manage Captcha and Form	59
3.3.32	UC_32 Manage Backup.....	60
3.3.33	UC_33 Manage User	62
3.3.34	UC_34 Database Cleanup	65
3.3.35	UC_35 View Logs	66
3.4	Performance Requirements	67
3.5	Design Constraints.....	67
3.6	Software System Attributes.....	68
3.6.1	Reliability	68
3.6.2	Availability.....	68
3.6.3	Security.....	68
3.6.4	Maintainability	69
3.7	Business Rules	69
4	Appendix A: Graphical User Interfaces (GUI).....	70
5	Appendix B: Index.....	81

Revision History

Name	Date	Reason for Changes	Version
SRS for AppXilon Online Dynamic Ordering System	30.06.2021	First draft	1.0
SRS for AppXilon Online Dynamic Ordering System	13.08.2021	Additional requirement & subsection details updated	2.0

1 Introduction

The Software Requirements Specification (SRS) is designed to document and describe the agreement between the stakeholder and the developer regarding the specification of the software product requested. This document is broken into several sections used to logically separate the software requirements into easily referenced parts.

1.1 Purpose

The primary purpose of this SRS is to define and describe the functions and specification of the AppXilon Online Dynamic Ordering System (AppXilon). This document illustrates, in clear terms, the system's primary uses and required functionality as specified by our customer.

The intended audience of this document is our primary AppXilon stakeholder: Mr. Shahzar, the Scrum Master Mr. Hafeez Osman, as well as Scrum team that will require access to such documentation.

1.2 Scope

AppXilon is an online ordering system for Taf Bistro Restaurant. This includes both customer and manager of the restaurant. The system will be designed to ease business activities by migrating from brick-and-mortar retail to online retail. The customer able to browse and purchase product online while manager able to manage the restaurant online. The system will be run on a web application with each user interact through a web browser. This system is largely cross-platform and is available to anyone using the web browser.

More specifically, the AppXilon is designed to allow any user to create an account to become a customer. The system will allow customers to browse, search, select ordering time and add products to a shopping cart. Then, provided they have products in their shopping cart, check out products in shopping cart and decrement the stock that the inventory the system maintains. The ordering process can be either booking a table for a dine-in, take-away, and shipping online. The AppXilon also allows a manager to manage the inventory and report with full create, read, update and delete (CRUD) functionality with regards to products and sales in the system. The product review and report retrieved from the system has an analytics capability for the manager to view. It will also allow the customer and managers to interact via a chatbot for support and help desk. The AppXilon has fa personalized recommendation functionality; to recommend product that relevant to the customer's interest. The system will be maintained and enhanced by the admin who has access to all functions in the system.

1.3 Definitions and Acronyms

Customer	A person that is a user of the system but has created an account
Admin	Administrator who maintains/enhances AppXilon
Manager	A single person that has the ability to create, retrieve, update and delete products in the restaurant
Stakeholder	People who has business and other interest in AppXilon including ..
SRS	Software Requirement Specification
IEEE	Institute of Electrical and Electronic Engineers
API	Application Processing Interface
OS	Operating System
GUI	Graphical User Interface
CRUD	Create, Read, Update, Delete
Inventory	An object that holds products available for purchase by the Customer
Product	An individual entity in the inventory which has several descriptive attributes
Chatbot	A computer program designed to simulate conversation with users.

1.4 References

IEEE. IEEE Std. 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.

1.5 Overview

This document is prepared to explain all detailed information about overall system description, functional, non-functional, and specific requirements, data, and behavioral model description of the system. This document basically consists of three parts:

Part 1: Gives brief description of the project and system. This section also explains the relationship of the project with another project management plan if applicable.

Part 2: Gives specific requirements, data, and behavioral model description of the system which are section 3, 4 and 5 in the document.

Part 3: Gives planning, conclusion and supporting information about the system

2 Overall Description

This section includes details about what is and is not expected of the AppXilon system in addition to which cases are intentionally unsupported and assumptions that will be used in the creation of the system.

2.1 Product Perspective

AppXilon is an online dynamic ordering system which supports several functions for both the customer and restaurant's management.

The website must be available to anyone using any web browsers and as much work correctly in both the Google Chrome and Safari. As stated by the customer, there are no hardware or software requirements beyond these including, but not limited to, memory or specific software packages that need to be utilized nor software packages that need not be utilized.

2.2 Product Functions

AppXilon will provide a number of functions; each is listed on the table below.

Use Case ID.	Use Case Name	Brief Description
UC_01	Register	Users need to register to have an account on the AppXilon
UC_02	Login	Users need to log in to get full authorized access of the AppXilon
UC_03	Manage Profile	User can update the information on their profile
UC_04	Chat with Chatbot	Users can communicate with the chatbot if there's any inquiries
UC_05	View Catalogue	Users can see all the menu in the catalogue
UC_06	View Personalised Recommendations	Users can browse the product recommendation that relevant to the user preferences.
UC_07	Make Table Booking	Users can book a table first before they dine in
UC_08	Make Purchase	Users can make purchase of their order after adding to cart and make payment

UC_09	Update Cart	Users can confirm the menu to purchase before proceeding to payment
UC_10	Checkout	Users make payment for the order
UC_11	Give Review	Users can give review and their rating for order
UC_12	View FAQ	Users view a set of FAQs and refer to them as a form of help
UC_13	View Booking Details	Users can view customer's booking details
UC_14	Manage Catalogue	Users can perform CRUD on the catalogue, product availability, and categories.
UC_15	Manage Table Criteria	Users can update table criteria
UC_16	Manage Order Status	Users can track and update order progress
UC_17	Manage Promotion	Users can change the promotion or create the promotion of the restaurant
UC_18	Manage Review	Users can configure the reviews to be displayed on the review page
UC_19	Manage FAQ	Users can change the question or create a question in the FAQ
UC_20	Generate Report	Users can generate a result report
UC_21	View Analytics	Users can view analytics from the dashboard
UC_22	Track Customer Sentiments	Users can view customer sentiment towards the restaurant from three polarity, positive, negative, and neutral
UC_23	View Customer Analytics	Users can analyse customer background data for business decision making
UC_24	View Sales Analytics	Users can view the analysis of current, past, and future of sales performance
UC_25	View Product Analytics	Users can view the analysis of product performance such as popularity of products.
UC_26	Manage Payment	Users can manage the type of payment user can be used
UC_27	Manage Business Hour	Users can update the store business hours
UC_28	Manage Term Condition	Users can update the terms and conditions

UC_29	Manage Reminder Reservation	Users can update the reminder reservation
UC_30	Manage Reservation Form	Users can manage the reservation forms that customer needs to fill in to make a reservation
UC_31	Manage Captcha and Form	Users can manage the captcha to ensure security
UC_32	Manage Backup	Users can manage the backup such as the table
UC_33	Manage User	Users can manage the user's access control on the website
UC_34	Database Cleanup	Users can manage the database cleanup
UC_35	View Logs	Users can view other user's login logs and error logs

2.3 User Characteristics

There are mainly four kinds of users of AppXilon.

Actor ID	Actor/Roll Name	Role Description and Objectives
A001	Administrator	Person who controls the overall of AppXilon. Administrator will mainly use AppXilon to control the assignment of role, system configuration and to manage the website. Administrator has the primary duty of overseeing and maintaining the AppXilon operation are working effectively and efficiently.
A002	Manager	Person who manages the operation of the ordering and booking in AppXilon. Manager will mainly use AppXilon to add, delete, edit and manage the catalogue and manage the booking schedule. Manager has the primary duty of ensuring that the catalogue display is the same with the availability of food with the restaurant and ensure that the booking details is correct.
A003	Registered Customer	Person who can use the system to make ordering and booking at the restaurant with full access. AppXilon will allow registered customer to make

		transaction, view catalogue, and navigate through the website.
A004	Guest Customer	People who can use the system only with limited access. Guest Customer can only view and browse the catalogue in the AppXilon. Guest Customer need to login to the AppXilon to make ordering or table booking.

2.4 Constraints

The following are the main constraints,

- The system will not have full credit-card processing capabilities.
- The system will not allow multiple promotions to be added to a single shopping cart.

2.5 Assumptions and Dependencies

The following are the main assumptions and dependencies,

- The manager cannot be a customer.
- The manager and admin account's username and password maybe hard coded

2.6 Apportioning of Requirements

This subsection identifies the requirements that may be delayed until future versions of the system with priority assigned to each of the requirements.

Use Case ID.	Use Case Name	Primary Actor	Priority
UC_01	Register	Admin, Manager, Customer	1
UC_02	Login	Admin, Manager, Customer	1
UC_03	Manage Profile	Customer	1
UC_05	View Catalogue	Customer	1
UC_07	Make Table Booking	Customer	1
UC_08	Make Purchase	Customer	1
UC_09	Update Cart	Customer	1

UC_10	Checkout	Customer	1
UC_13	View Booking Details	Customer	1
UC_14	Manage Catalogue	Manager	1
UC_15	Manage Table Criteria	Manager	1
UC_16	Manage Order Status	Manager	1
UC_27	Manage Business Hour	Admin	1
UC_28	Manage Term Condition	Admin	1
UC_30	Manage Reservation Form	Admin	1
UC_35	View Logs	Admin	1
UC_11	Give Review	Customer	2
UC_12	View FAQ	Customer	2
UC_18	Manage Review	Manager	2
UC_19	Manage FAQ	Manager	2
UC_21	View Analytics	Manager	2
UC_26	Manage Payment	Admin	2
UC_29	Manage Reminder Reservation	Admin	2
UC_31	Manage Captcha and Form	Admin	2
UC_32	Manage Backup	Admin	2
UC_33	Manage User	Admin	2
UC_06	View Personalised Recommendations	Customer	3
UC_17	Manage Promotion	Manager	3
UC_20	Generate Report	Manager	3
UC_22	Track Customer Sentiments	Manager	3

UC_23	View Customer Analytics	Manager	3
UC_24	View Sales Analytics	Manager	3
UC_25	View Product Analytics	Manager	3
UC_34	Database Cleanup	Admin	3
UC_04	Chat with Chatbot	Customer	4

3 Specific Requirements

This section provides all the detailed functional and non-functional requirements.

3.1 External Interfaces Requirements

This section provides details of all inputs and outputs including user interface, hardware, software, communication.

3.1.1 User Interface Requirements

AppXilon application should contain following user interfaces,

- Login page for authenticating registered users. This screen should accept user id, password and authenticate against corporate authentication system. It also provides features for 'Create New Account' and 'Forgot Password'.
- Product search page where registered user can search product based on product attributes.
- Search results page displays the result of search operation
- Shopping cart page displays the existing items in the shopping cart along with total amount and allows the user to check out.
- Checkout page allows the user to purchase the products using e-wallet or using PayPal account. It should be integrated with payment gateway and display the order details after successful payment.

3.1.2 Hardware Interface Requirements

There is no direct hardware interface specifically for AppXilon application. The web application is a client/server system runs on an application server hosted in-house on enterprise hardware.

3.1.3 Software Interface Requirements

- User Side: Google Chrome, Safari, Microsoft Edge, Mozilla Firefox
- System Side: Web-based application and database information storage system

3.1.4 Communication Interface Requirements

There are no AppXilon specific communication interface requirements. Existing OS and network infrastructure will be leveraged for communication.

3.2 Functional Requirement

This section provides details of all major functionalities supported by AppXilon system.

3.2.1 Customer Use Case Diagram



Figure 1 Customer Use Case Diagram

3.2.2 Manager Use Case Diagram

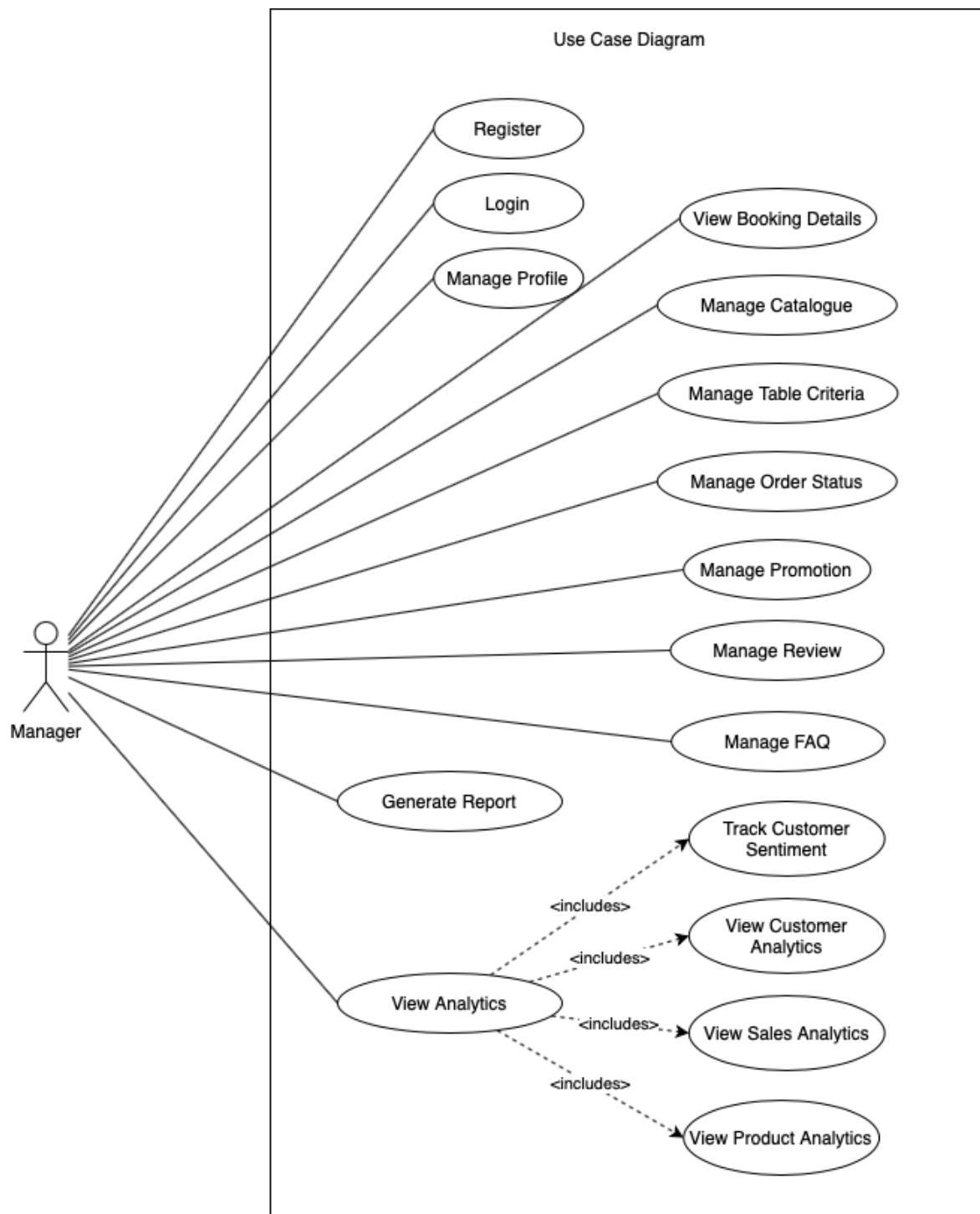


Figure 2 Manager Use Case Diagram

3.2.3 Admin Use Case Diagram

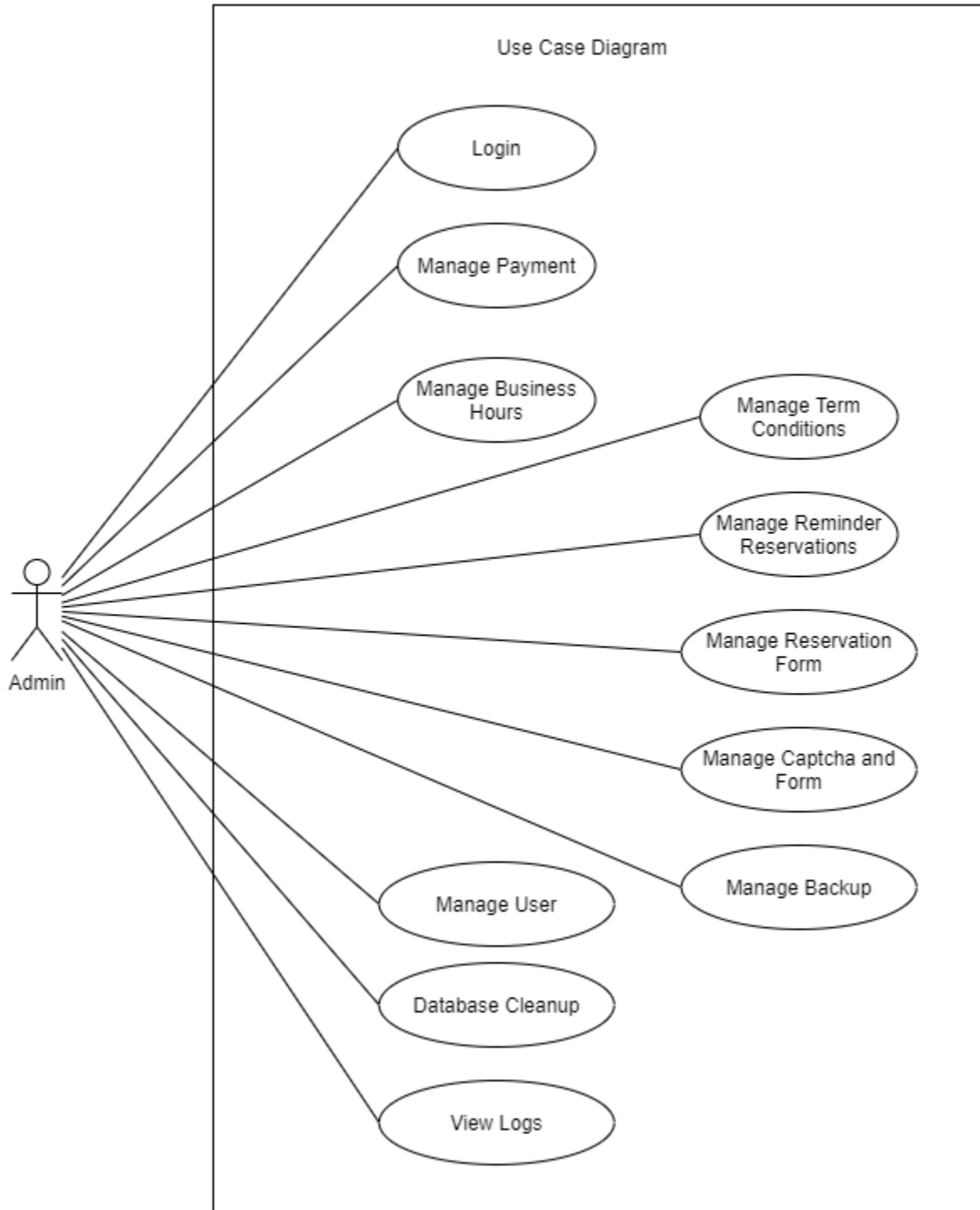


Figure 3 Admin Use Case Diagram

3.3 Use Case Description

3.3.1 UC_01 Register

Description	The user needs to register to have an account on the AppXilon
Actors	Customer, Manager, Administrator, Email verification API
Preconditions	None
Basic Flow	<ol style="list-style-type: none"> 1. The user clicks register 2. The user enters email 3. AppXilon checks whether the email exists or not in the database 4. AppXilon displays “email is available” (A1: Email already exist) 5. The user enters the first name 6. The user enters the last name 7. The user enters password 8. The user re-enters password 9. The user clicks submit button 10. Email verification API sends verification code to user’s email 11. The user enters verification code (E1: Users do not receive verification code) 12. The user clicks submit button 13. The user successfully registers
Alternative Flow	A1: Email already exists <ol style="list-style-type: none"> 3.1 AppXilon displays message that email already exists and ask to use a different email 3.2 Return to step 2
Post-condition	The user automatically logs in to the AppXilon
Exception Flow	E1: The user does not receive verification code <ol style="list-style-type: none"> 9.1 The user requests another verification code. 9.2 Return to step 11.

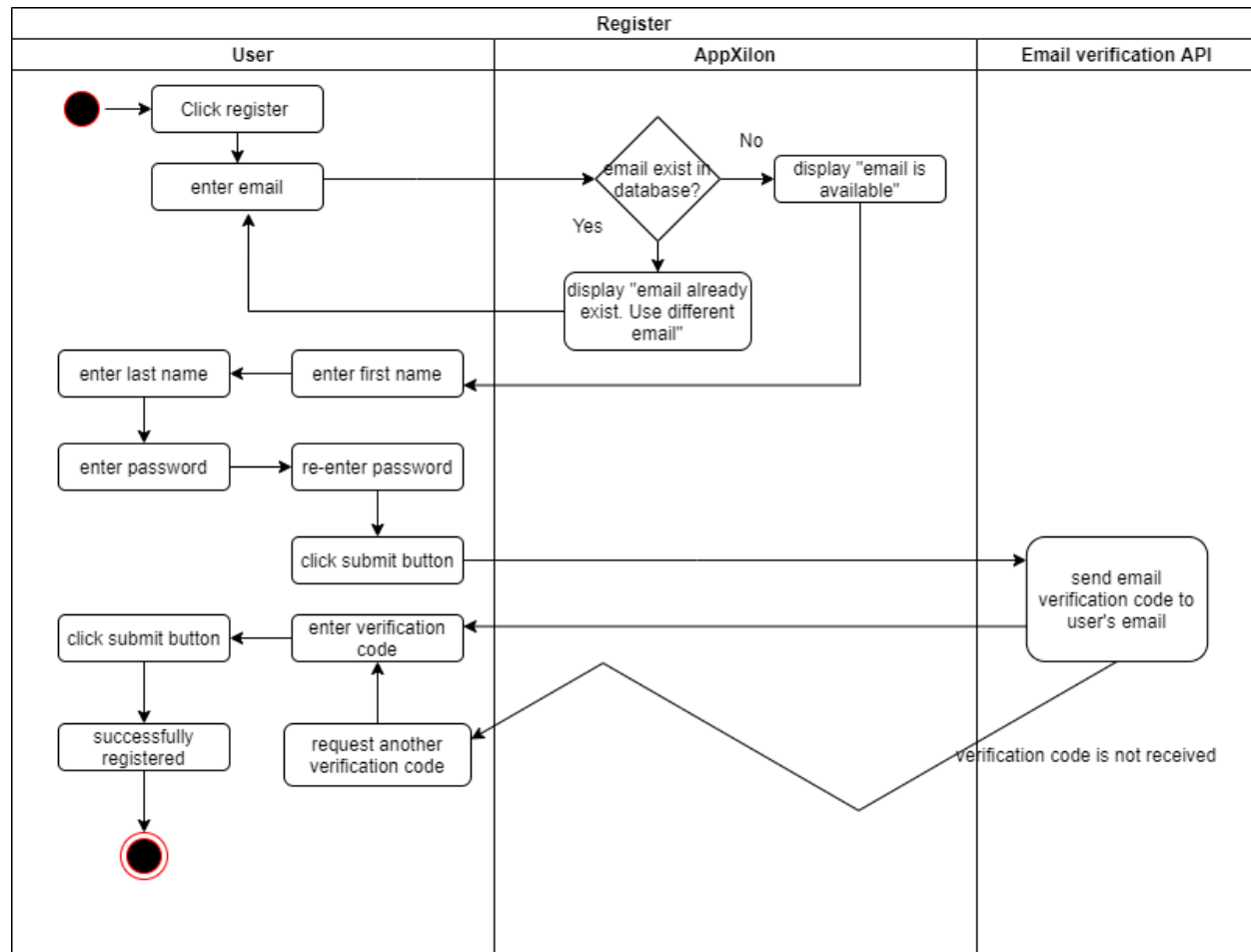


Figure 4 Activity Diagram: Register

3.3.2 UC_02 Login

Description	The user needs to log in to get full authorized access of the AppXilon
Actors	Customer, Manager, Administrator, Facebook API, Google API
Preconditions	The user needs to be registered into the AppXilon
Basic Flow	<ol style="list-style-type: none"> 1. The user chooses login method 2. The user logs in with username (A1: Users log in with Facebook, A2: User login with Google) 3. The user enters username 4. The user enters password (A4: Users click forgot password button) 5. The user clicks the login button 6. The user successfully logs in to the system
Alternative Flow	A1: Users log in with Facebook

	<p>1.1.1 AppXilon redirects user to Facebook authentication API</p> <p>1.1.2 The user enters sign in credentials into Facebook sign-in interface and submit</p> <p>1.1.3 Facebook authenticate user</p> <p>1.1.4 Facebook redirects back to AppXilon</p> <p>1.1.5 AppXilon checks if the Facebook account exists or not in database</p> <p>1.1.6 The user successfully logs in to AppXilon (A1.1 System create new account)</p> <p>A1.1 AppXilon create new account</p> <p>1.1.8.1 AppXilon creates new user with the account</p> <p>1.1.8.2 The user successfully logs in to AppXilon</p> <p>A2: User log in with Google</p> <p>1.2.1 AppXilon redirects user to Google authentication API</p> <p>1.2.2 The user enters sign in credentials into Google signin interface and submit</p> <p>1.2.3 Google authenticate user</p> <p>1.2.4 Google redirects back to AppXilon</p> <p>1.2.5 AppXilon checks if that Google account exists or not in the database</p> <p>1.2.6 AppXilon successfully logs in into AppXilon with Google account (A.1.1: System creates new account)</p> <p>A4: Users click forgot password button</p> <p>2.1 AppXilon send link to user's email to reset the password</p> <p>2.2 The user clicks the link</p> <p>2.3 The user resets the password</p>
Post-condition	The user enters the homepage
Exception Flow	N/A

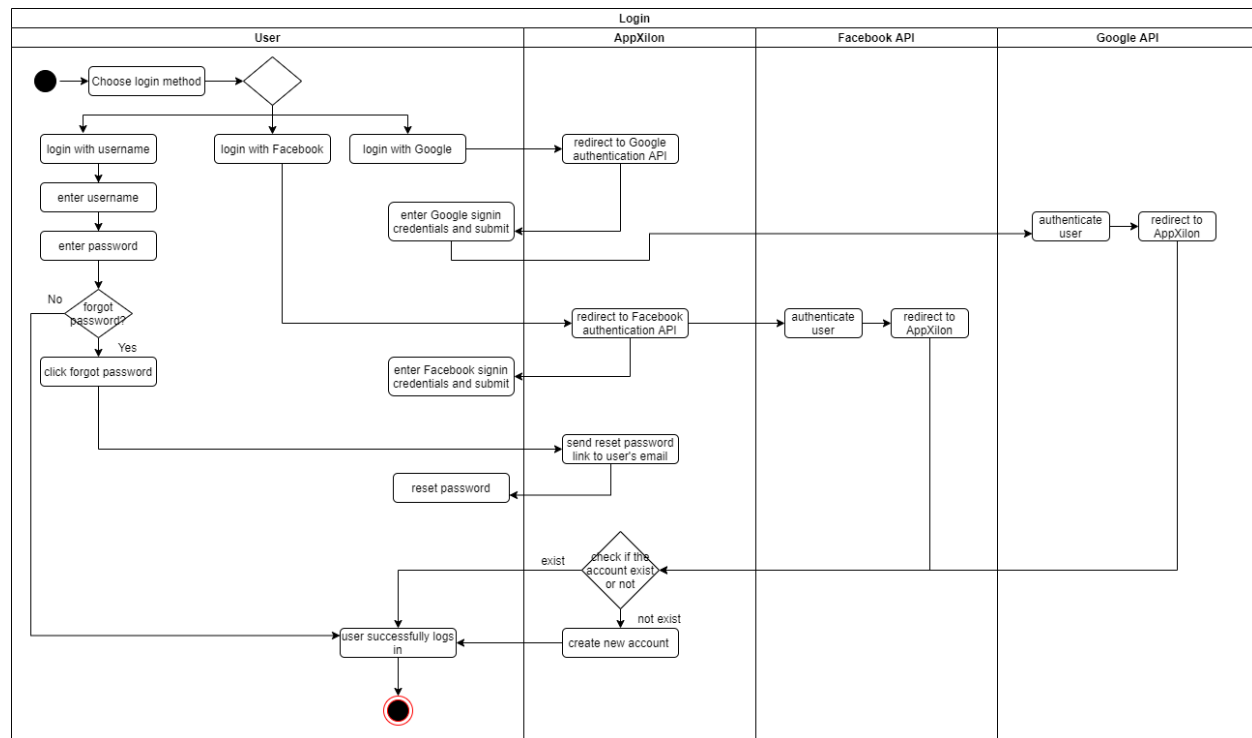


Figure 5 Activity Diagram: Login

3.3.3 UC_03 Manage Profile

Description	The user can update the information on their profile
Actors	Admin, Manager, Customer
Preconditions	The user needs to be registered into the AppXilon
Basic Flow	<ol style="list-style-type: none"> 1. The user clicks profile button 2. AppXilon displays profile details 3. The user views profile (A1: User edits profile, A2: User deletes profile)
Alternative Flow	<p>A1: The user edits profile</p> <ol style="list-style-type: none"> 3.1.1 The user edits data fields 3.1.2 The user clicks update button 3.1.3 AppXilon Display “Profile successfully updated” <p>A2: User deletes profile</p> <ol style="list-style-type: none"> 3.2.1 The user click delete profile button 3.2.2 The user confirms delete 3.2.3 AppXilon displays “Account is successfully deleted”
Post-condition	N/A

Exception Flow	N/A
-----------------------	-----

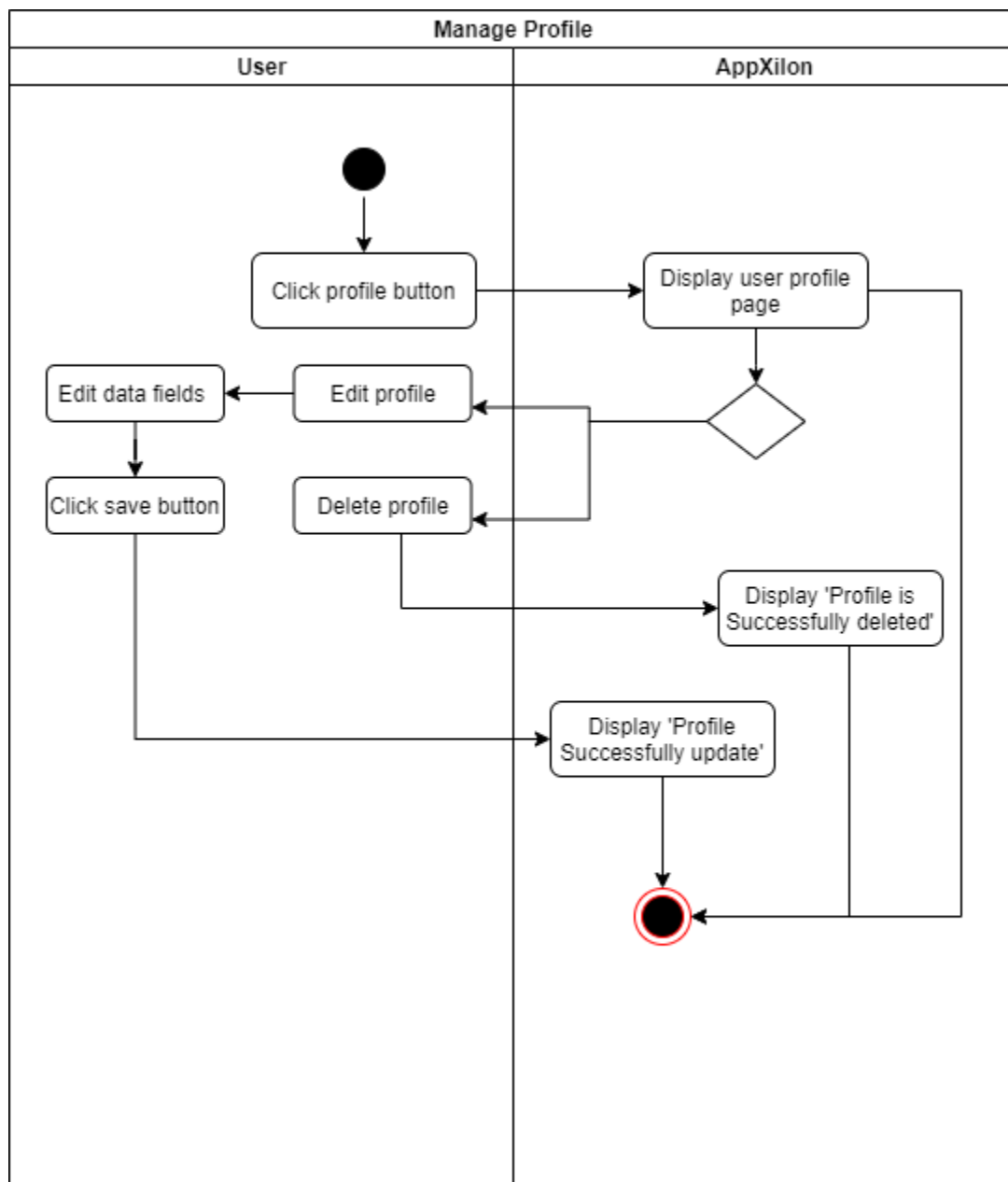


Figure 6 Activity Diagram: Manage Profile

3.3.4 UC_04 Chat with Chatbot

Description	The user can communicate with the chatbot if there's any inquiries
Actors	Customer
Preconditions	None

Basic Flow	<ol style="list-style-type: none"> 1. Chatbot chat widget displays welcome message and menu 2. The user chooses question from the menu 3. Chatbot response by giving the answer 4. The user ends the chatbot (A1: User has another question)
Alternative Flow	A1: User has another question 4.1.1. Return to step no.2
Post-condition	None
Exception Flow	N/A

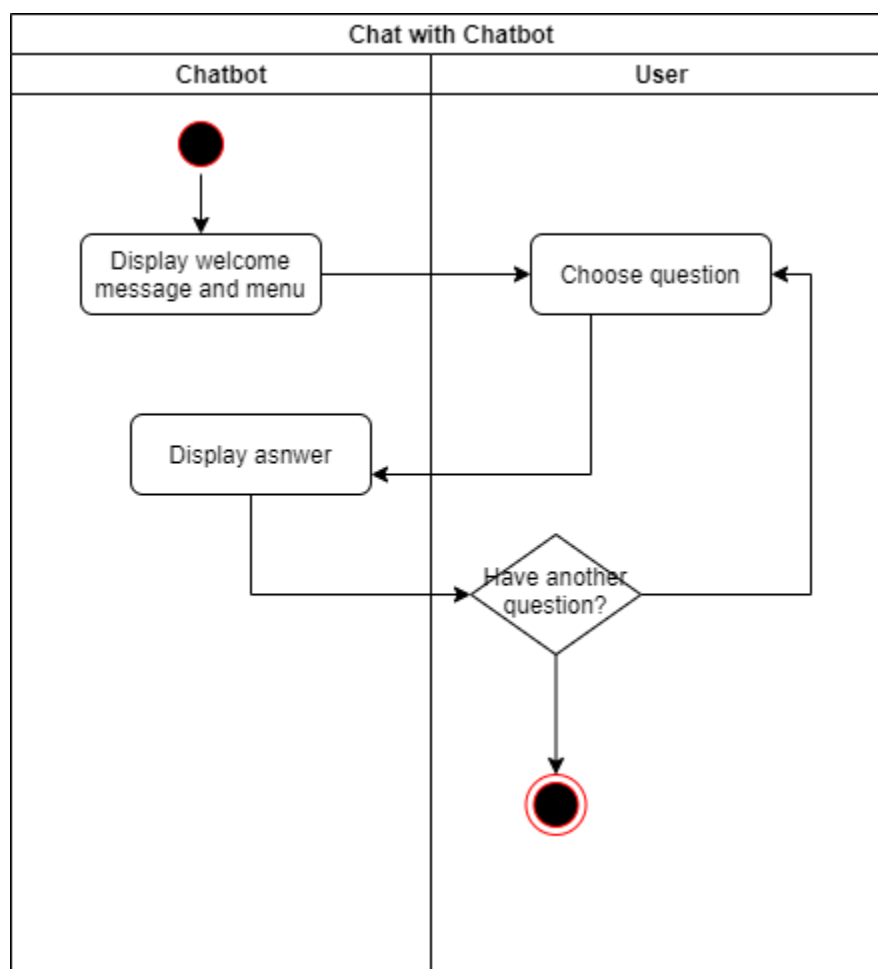


Figure 7 Activity Diagram: Chat with Chatbot

3.3.5 UC_05 View Catalogue

Description	The user can view the catalogue of the restaurant with its details to make an order
--------------------	---

Actors	Customer
Preconditions	The user searches or scans QR code (if in the restaurant) to open the AppXilon
Basic Flow	<ol style="list-style-type: none"> 1. AppXilon displays the catalogue 2. The catalogue will display all the categories of the menu such as main food, western, dessert, etc. 3. The user selects menu (E1: menu not available) 4. AppXilon displays the menu details such as the price and the availability of the status 5. The user selects add to cart button (A1: Customer select back button)
Alternative Flow	A1: Customer selects back button <ol style="list-style-type: none"> 5.1 If the user selects back button, the customers will be redirected to catalogue page 5.2 The flow returns to the basic flow
Post-condition	The menu selected will be displayed at cart
Exception Flow	E1: Menu not available <ol style="list-style-type: none"> 3.1 If the user selects unavailable menu, the menu will display “Not available” message on the menu 3.2 The flow returns to the basic flow

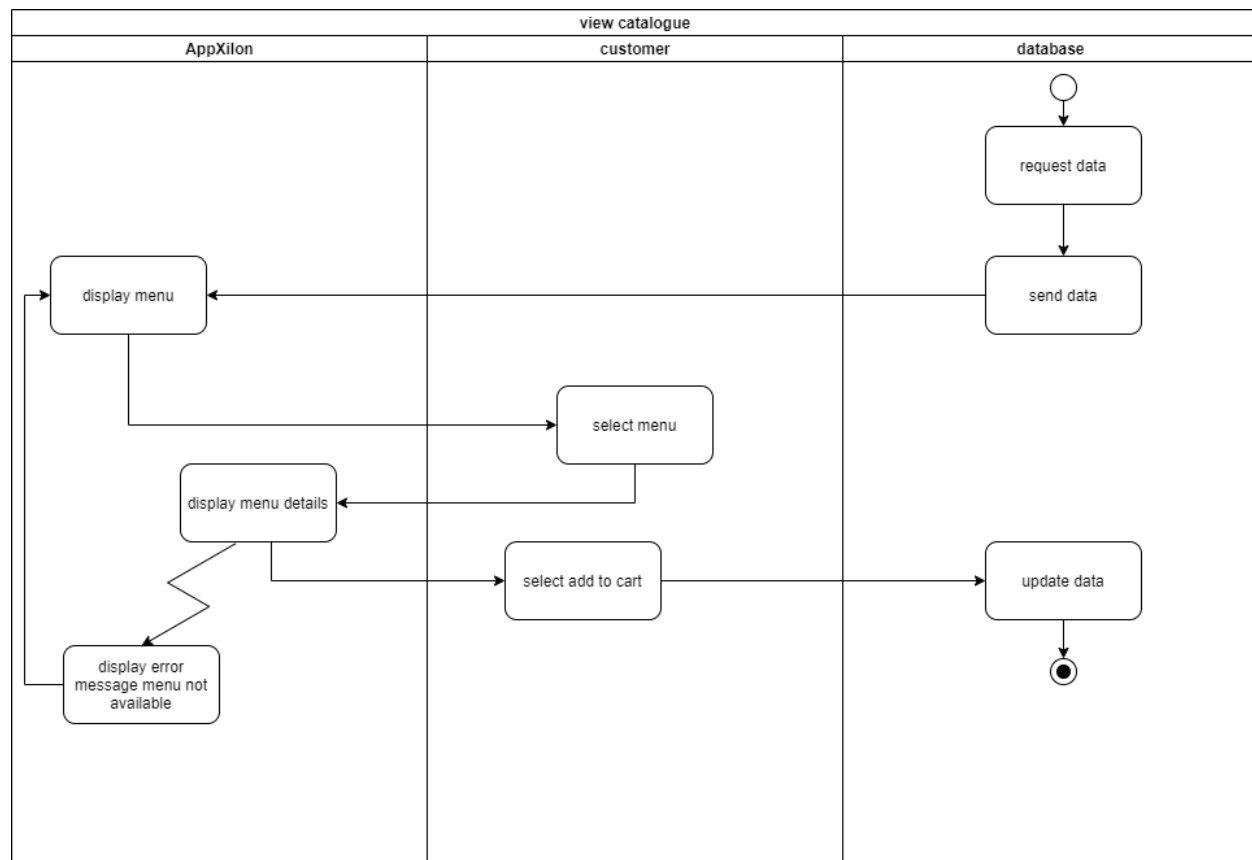


Figure 8 Activity Diagram: View Catalogue

3.3.6 UC_06 View Personalized Recommendations

Description	Users can browse the product recommendation that relevant to the user preferences.
Actors	Customer
Preconditions	Users need to be registered into the AppXilon
Basic Flow	<ol style="list-style-type: none"> 1. AppXilon displays the catalogue. 2. The user selects menu 3. AppXilon displays the menu details 4. AppXilon displays the similar menu with similar product attributes 5. AppXilon displays the menu that has bought with similar user.
Alternative Flow	N/A
Post-condition	N/A
Exception Flow	N/A

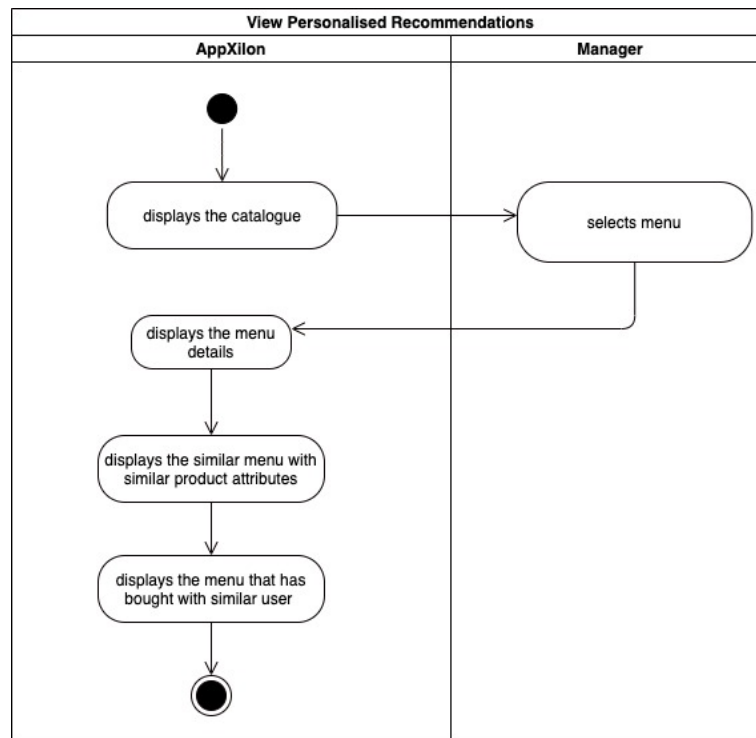


Figure 9 Activity Diagram: View Personalised Recommendations

3.3.7 UC_07 Make Table Bookings

Description	Users can make table booking before arriving at the restaurant
Actors	Customer
Preconditions	The user selects booking table button
Basic Flow	<ol style="list-style-type: none"> 1. AppXilon will ask the user to enter the date, time, and number of pax to make table booking 2. AppXilon displays the table layout of the restaurant 3. The user selects the table which are only clickable/chosen according to the pax (E1: Unavailable Seats) 4. The user will be directed to reservation detail's page 5. The user fills in the details provided in the page 6. The user selects next button to proceed to catalogue page
Alternative Flow	A1: Customer select back button 3.2.1 The flow returns to the basic flow
Post-condition	Customer will be directed to catalogue page
Exception Flow	E1: Unavailable Seats

	<p>3.1 AppXilon displays “the seat is currently unavailable for this date / time / pax” message</p> <p>3.2 AppXilon display “select new date / time / pax” (A1: Customer select back button)</p> <p>3.3 The flow returns to the basic flow</p>
--	---

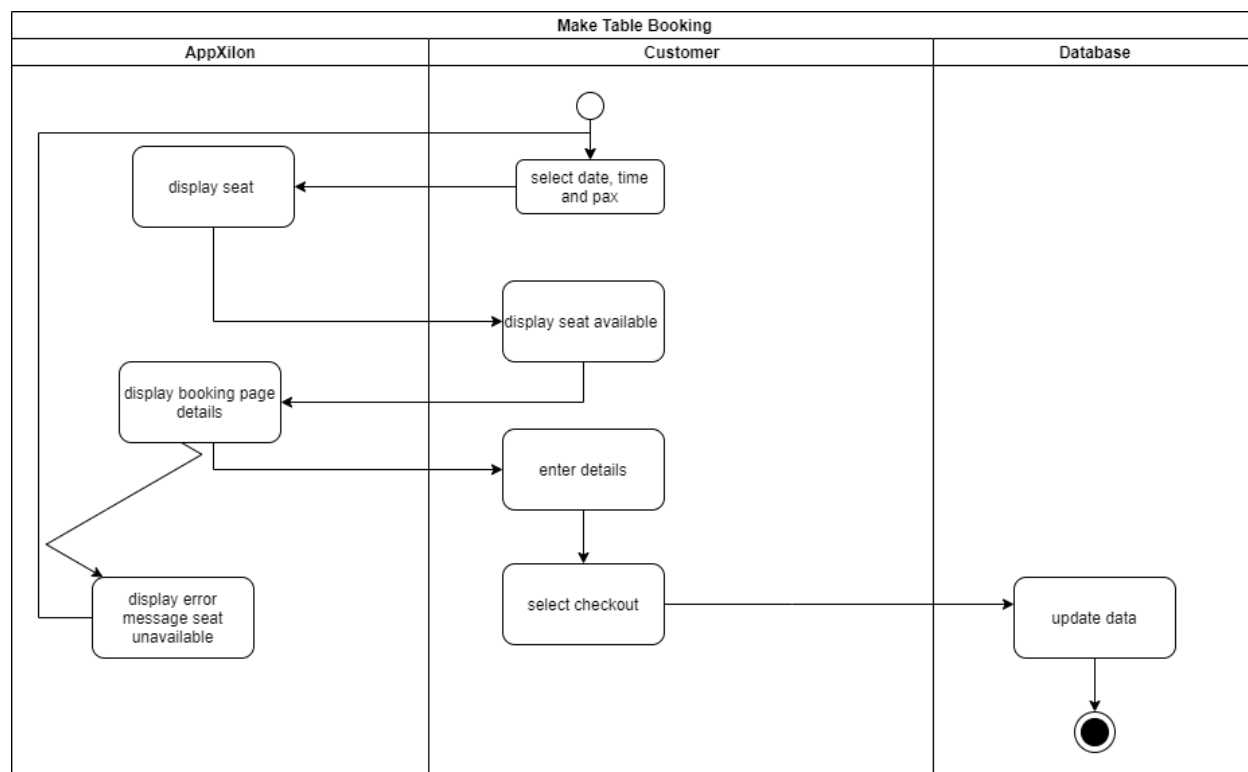


Figure 10 Activity Diagram: Make Table Booking

3.3.8 UC_08 Make Purchase

Description	The user can make purchase of their order after adding to cart and make payment
Actors	Customer
Preconditions	The user selects menu from catalogue
Basic Flow	<ol style="list-style-type: none"> 1. The user selects cart button (UC_09 Update Cart) 2. The user selects checkout button 3. The user chooses payment options to make payment (UC_10 Checkout) 4. The user makes payment

	5. The user proceeds to order status page
Alternative Flow	N/A
Post-condition	The user proceeds to order status page
Exception Flow	N/A

3.3.9 UC_09 Update Cart

Description	The user can confirm the menu to purchase before proceeding to payment
Actors	Customer
Preconditions	The user selects cart button
Basic Flow	<ol style="list-style-type: none"> 1. AppXilon displays all the orders that have been selected by the user 2. The user selects add quantity button; therefore, the quantity of the food is added. (A1: The user select reduce quantity button) 3. The user selects checkout button. (A2: The user selects back button, A3: The user selects delete button)
Alternative Flow	<p>A1: The user selects reduce quantity button</p> <ol style="list-style-type: none"> 2.1 If the user selects reduce quantity button, therefore the quantity of the food will be reduced. 2.2 The flow returns to the basic flow. <p>A2: The user selects back button</p> <ol style="list-style-type: none"> 3.1.1 If the user selects back button, the customers will be redirected to catalogue page. 3.1.2 The flow returns to the basic flow. <p>A3: The user selects delete button</p> <ol style="list-style-type: none"> 3.2.2 If the user selects delete button, therefore the menu list will be deleted from cart 3.2.3 The flow returns to the basic flow.
Post-condition	The user will be directed to the checkout page
Exception Flow	N/A

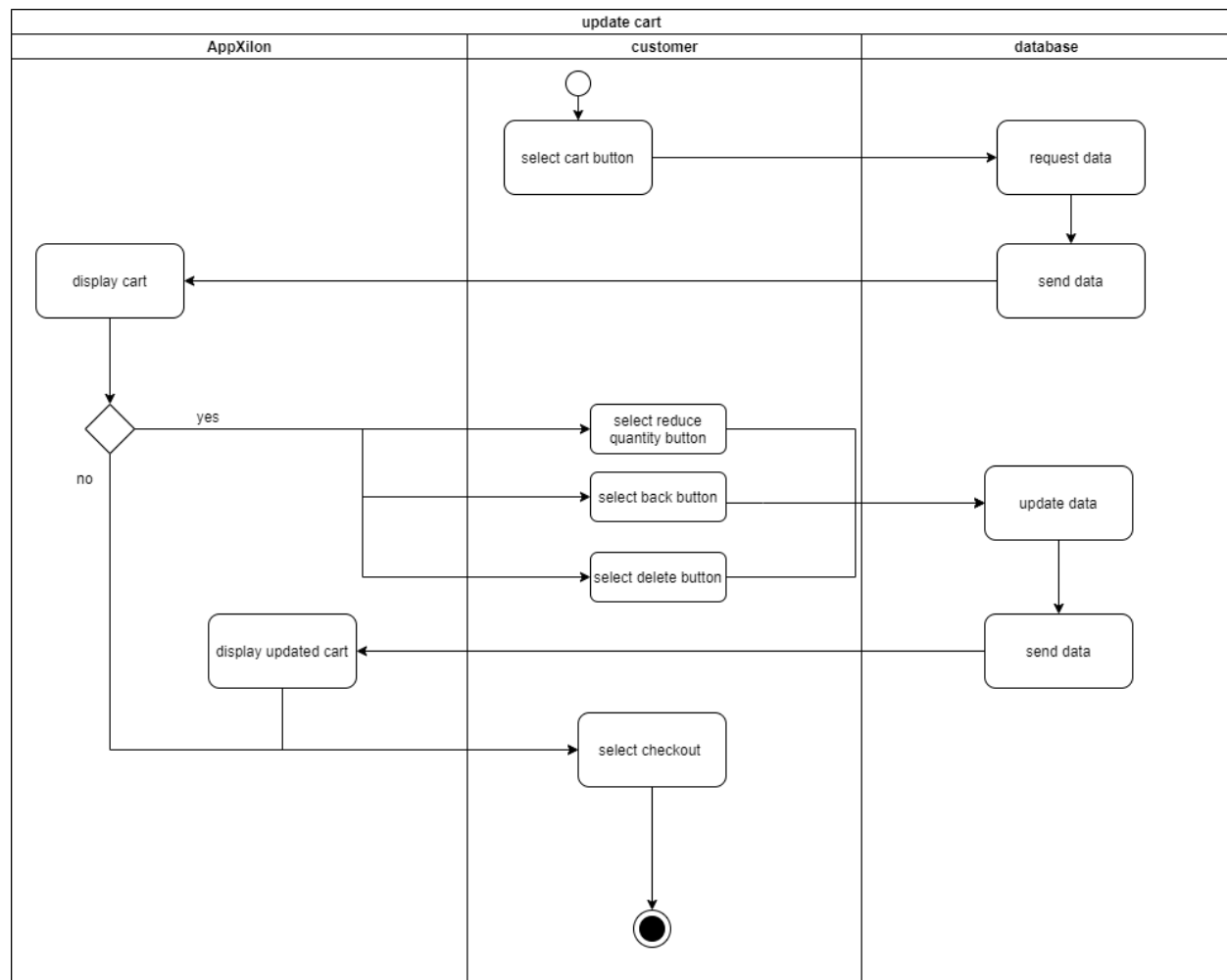


Figure 11 Activity Diagram: Update Cart

3.3.10 UC_10 Checkout

Description	The user make payment using various of payment options
Actors	Customer
Preconditions	The user selects checkout button at cart page
Basic Flow	<ol style="list-style-type: none"> 1. AppXilon displays the order summary 2. The user selects payment method (A1: Select paypal, A2: Select Credit Card) (A3: Select Online Banking) (A4: Select E-Wallet) (E1: API display error message) 3. The user proceeds to order status page (A5: The user selects back button)
Alternative Flow	A1: Select PayPal 2.1.1 The user will be directed to PayPal login page

	<p>2.1.2 AppXilon displays message “Payment is successful”</p> <p>2.1.3 The user will be redirected to the website back</p> <p>2.1.4 The flow returns to the basic flow</p> <p>A2: Select Credit Card</p> <p>2.2.1 The user will be directed to credit card details page</p> <p>2.2.2 The user fills in the field with the information that the page request</p> <p>2.2.3 AppXilon displays message “Payment is successful”</p> <p>2.2.4 The user will be redirected to the website back.</p> <p>2.2.5 The flow returns to the basic flow.</p> <p>A3: Select Online Banking</p> <p>2.3.1 The user will select the type of bank they want to proceed with payment</p> <p>2.3.2 The user will be directed to bank’s API</p> <p>2.3.3 The flow return to basic flow</p> <p>A4: Select E-Wallet</p> <p>2.4.1 The user will select the type of e-wallet they want to proceed with payment</p> <p>2.4.2 The user will be directed to e-wallet’s API</p> <p>2.4.3 The flow returns to basic flow</p> <p>A3: The user selects back button</p> <p>3.1 If the user selects back button, the customer the user will be redirected to catalogue page.</p> <p>3.2 The flow returns to the basic flow.</p>
Post-condition	The user will be directed to order status page
Exception Flow	<p>E1: API display error message</p> <p>2.3.1 After the user selects the payment method, they will be directed to their respective API.</p> <p>2.3.2 The API display error message</p> <p>2.3.3 The user will be redirected to checkout page.</p> <p>2.3.4 The flow returns to the basic flow</p>

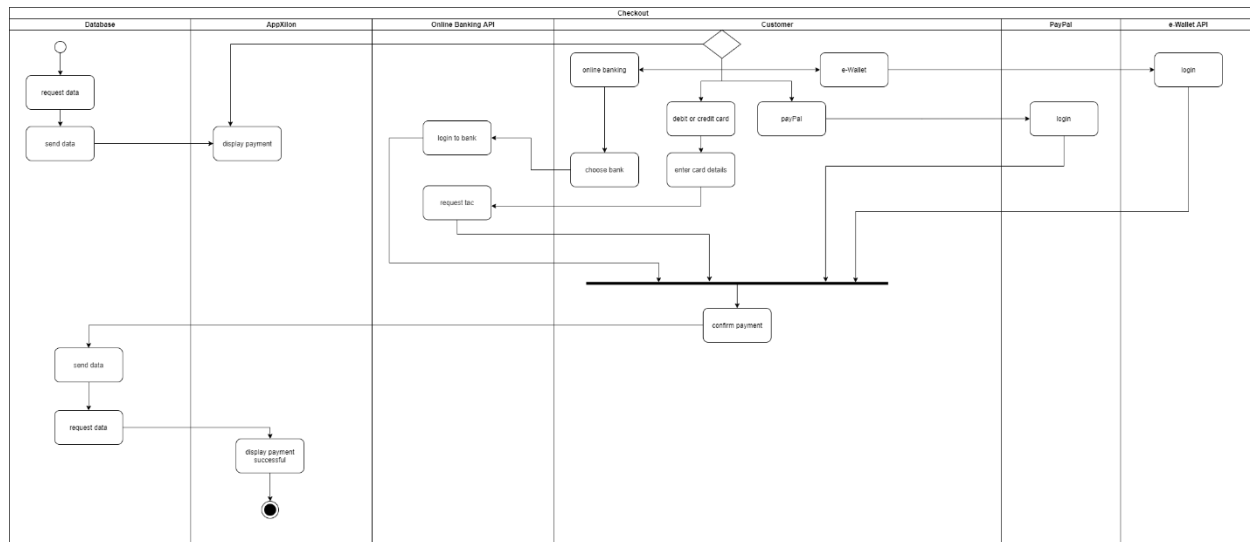


Figure 12 Activity Diagram: Checkout

3.3.11 UC_11 Give Review

Description	The user can give review and their rating for the order
Actors	Customer
Preconditions	The user has to complete an order first
Basic Flow	<ol style="list-style-type: none"> 1. The user clicks add new feedback button in feedback page (A1: The user views order history first) 2. AppXilon displays order history 3. The user chooses an order to give feedback 4. The user fills in the feedback form 5. The user clicks submit button
Alternative Flow	A1: The user views order history first 1.1 The user goes to order history page 1.2 Flow returns to step 2
Post-condition	N/A
Exception Flow	N/A

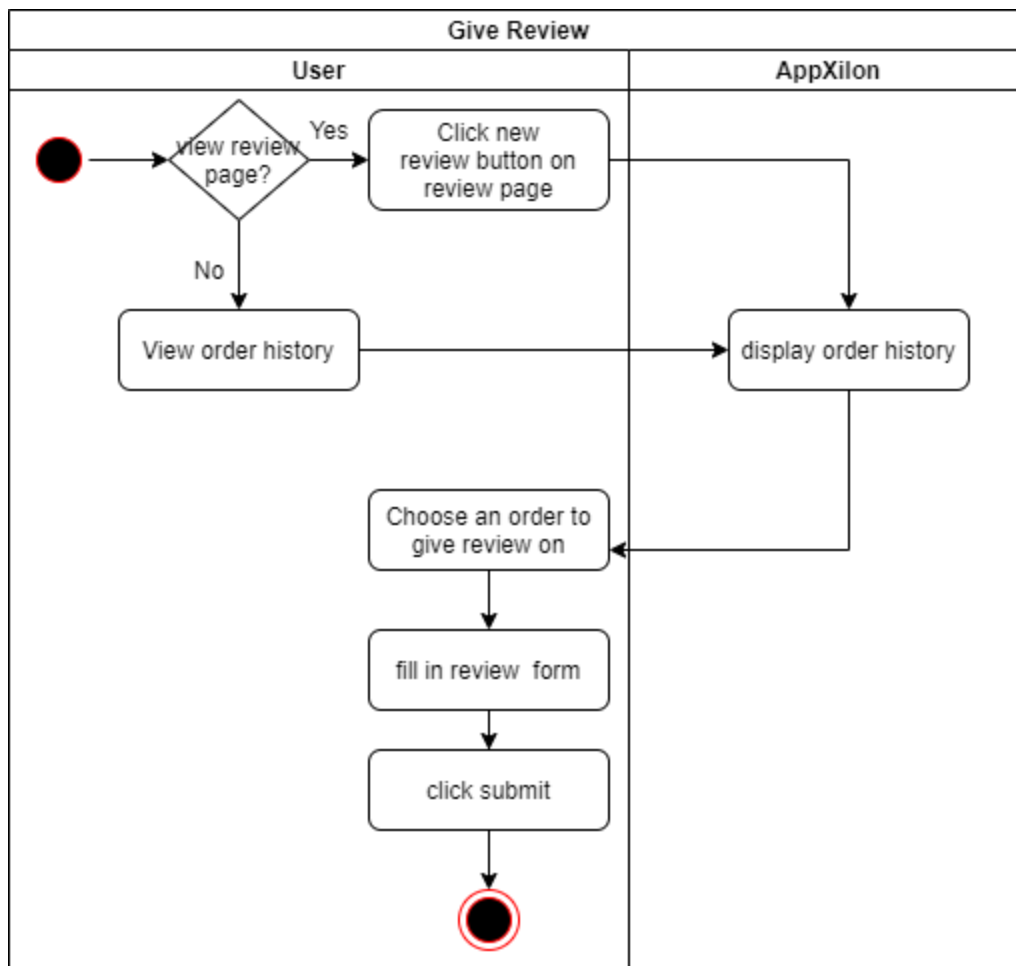


Figure 13 Activity Diagram: Give Review

3.3.12 UC_12 View FAQ

Description	The user views a set of FAQs and refer them as help
Actors	Customer
Preconditions	None
Basic Flow	<ol style="list-style-type: none"> 1. The user clicks help button 2. AppXilon displays a set of FAQs 3. The user clicks any one of the FAQ 4. AppXilon displays the answer of the FAQ 5. End (A1: The User clicks back to see the list of FAQ again)
Alternative Flow	A1: The user clicks back to see the list of FAQ again <ol style="list-style-type: none"> 4.1 The user clicks back button 4.2 Return to step 2

Post-condition	N/A
Exception Flow	N/A

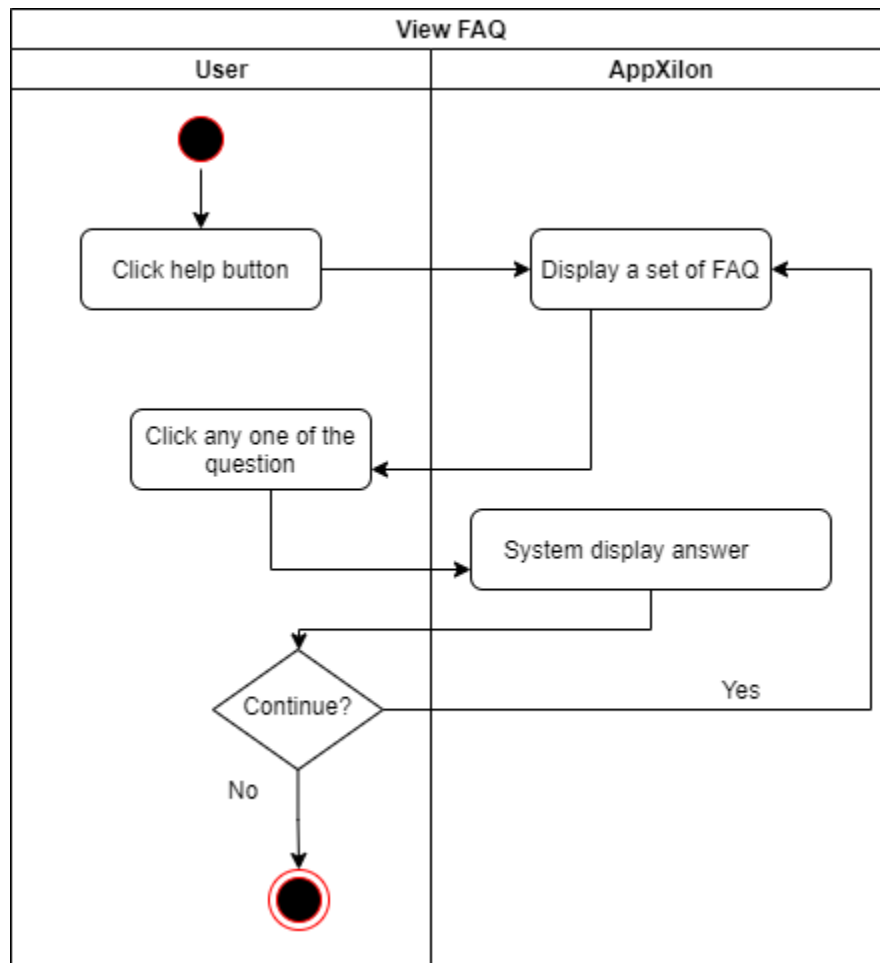


Figure 14 Activity Diagram: View FAQ

3.3.13 UC_13 View Booking Details

Description	The user can view customer's booking details
Actors	Manager
Preconditions	The user must log in into AppXilon.
Basic Flow	<ol style="list-style-type: none"> 1. AppXilon displays manager dashboard. 2. The user clicks on booking tab. 3. AppXilon displays booking page. 4. The user selects date and time. 5. AppXilon displays list of booking order. (A1: No booking order)

	6. The user selects a booking order.
Alternative Flow	A1: No booking order 5.1 AppXilon displays no booking order. 5.2 Flows return to step 4
Post-condition	AppXilon displays selected booking details.
Exception Flow	N/A

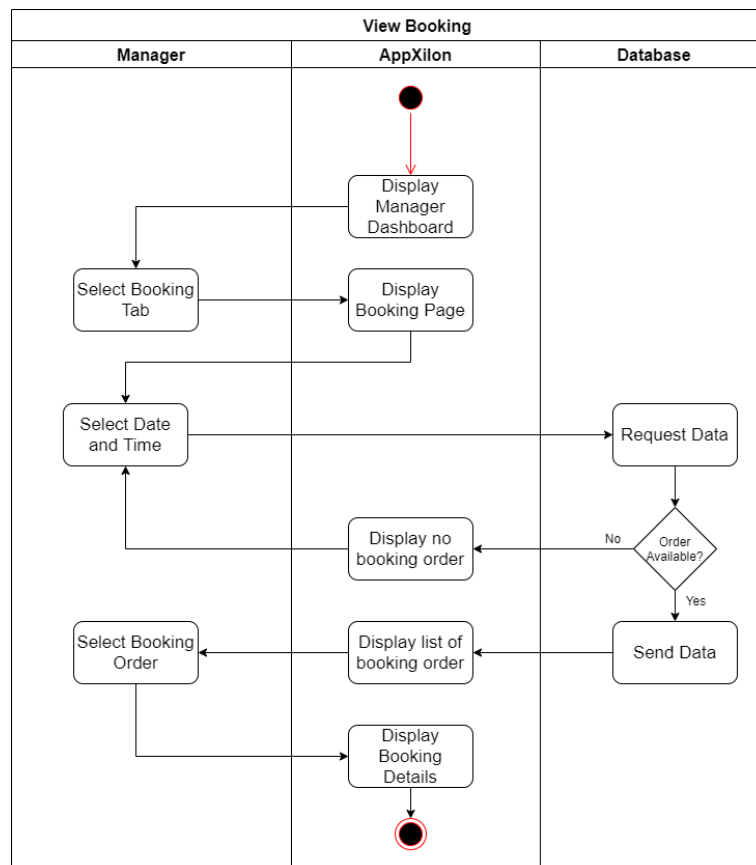


Figure 15 Activity Diagram: View Booking Details

3.3.14 UC_14 Manage Catalogue

Description	The user can perform CRUD on the catalogue and categories, and update product availability.
Actors	Manager
Preconditions	The user must log in into AppXilon.

Basic Flow	<ol style="list-style-type: none"> 1. AppXilon displays manager dashboard. 2. The user clicks on catalogue tab. 3. AppXilon displays catalogue page. 4. The user can choose to add, delete, or update product or, update product availability or, create, remove, or update categories in the catalogue. (A1: The user adds product) (A2: The user removes product) (A3: The user updates menu) (A4: The user updates product availability) (A5: The user adds new categories) (A6: The user removes categories) (A7: The user updates categories)
Alternative Flow	<p>A1: The user adds product.</p> <ol style="list-style-type: none"> 4.1.1 The user clicks add product button. 4.1.2 The user fills in product details. 4.1.3 The user clicks submit button. (A1.1: Manager click cancel button) 4.1.4 AppXilon displays catalogue updated message. 4.1.5 Flow returns to step 3 <p>A1.1: The user clicks cancel button.</p> <ol style="list-style-type: none"> 4.1.3.1 The user clicks cancel button. 4.1.3.2 Flow returns to step 3 <p>A2: The user removes product.</p> <ol style="list-style-type: none"> 4.2.1 The user clicks remove button on a product. 4.2.2 AppXilon displays confirmation message. 4.2.3 The user clicks confirm button. (A2.1: Manager click cancel button) 4.2.4 AppXilon displays catalogue updated message. 4.2.5 Flow returns to step 3. <p>A2.1: The user clicks cancel button.</p> <ol style="list-style-type: none"> 4.2.3.1 The user click cancel button. 4.2.3.2 Flow returns to step 3. <p>A3: The user updates product.</p> <ol style="list-style-type: none"> 4.3.1 The user click update button on a product. 4.3.2 AppXilon displays selected product details. 4.3.3 The user updates the details.

	<p>4.3.4 The user clicks submit button. (A3.1: Manager click cancel button)</p> <p>4.3.5 AppXilon display catalogue updated message.</p> <p>4.3.6 Flow returns to step 3.</p> <p>A3.1: The user clicks cancel button.</p> <p>4.3.4.1 The user clicks cancel button.</p> <p>4.3.4.2 Flow returns to step 3.</p> <p>A4: The user updates product availability.</p> <p>4.4.1 The user turns off availability button on a product. (A4.1: The user turns on availability button)</p> <p>4.4.2 AppXilon changes the product status to not available.</p> <p>4.4.3 Flow returns to step 3.</p> <p>A4.1: The user turns on availability button.</p> <p>4.4.1.1 The user turns on availability button.</p> <p>4.4.1.2 AppXilon changes the product status to available.</p> <p>4.4.1.3 Flow returns to step 3.</p> <p>A5: The user adds new categories.</p> <p>4.5.1 The user clicks update categories button.</p> <p>4.5.2 AppXilon displays updateable categories page.</p> <p>4.5.3 The user clicks add categories button.</p> <p>4.5.4 The user fills in categories details.</p> <p>4.5.5 The user clicks submit button. (A5.1: The user cancels changes)</p> <p>4.5.6 AppXilon display categories inserted message.</p> <p>4.5.7 The user click save button.</p> <p>4.5.8 Flow returns to step 3</p> <p>A5.1: The user cancels changes.</p> <p>4.5.5.1 The user clicks cancel button.</p> <p>4.5.5.2 The user clicks save button.</p> <p>4.5.5.3 Flow returns to step 3.</p> <p>A6: The user removes categories.</p> <p>4.6.1 The user clicks update categories button.</p> <p>4.6.2 AppXilon displays updateable categories page.</p> <p>4.6.3 The user clicks remove on a category tab.</p> <p>4.6.4 AppXilon displays confirmation message.</p>
--	--

	<p>4.6.5 The user clicks confirm button. (A6.1: The user cancels changes)</p> <p>4.6.6 AppXilon display category removed message.</p> <p>4.6.7 The user clicks save button.</p> <p>4.6.8 Flow returns to step 3.</p> <p>A6.1: The user cancels changes.</p> <p>4.6.5.1 The user clicks cancel button.</p> <p>4.6.5.2 The user clicks save button</p> <p>4.6.5.3 AppXilon return to step 3.</p> <p>A7: The user updates categories.</p> <p>4.7.1 The user clicks update categories button.</p> <p>4.7.2 AppXilon displays updateable categories page.</p> <p>4.7.3 The user double clicks on a category tab.</p> <p>4.7.4 The user update categories details.</p> <p>4.7.5 AppXilon displays updated categories.</p> <p>4.7.6 The user clicks save button</p> <p>4.7.7 Flow returns to step 3</p>
Post-condition	AppXilon display updated catalogue page.
Exception Flow	N/A

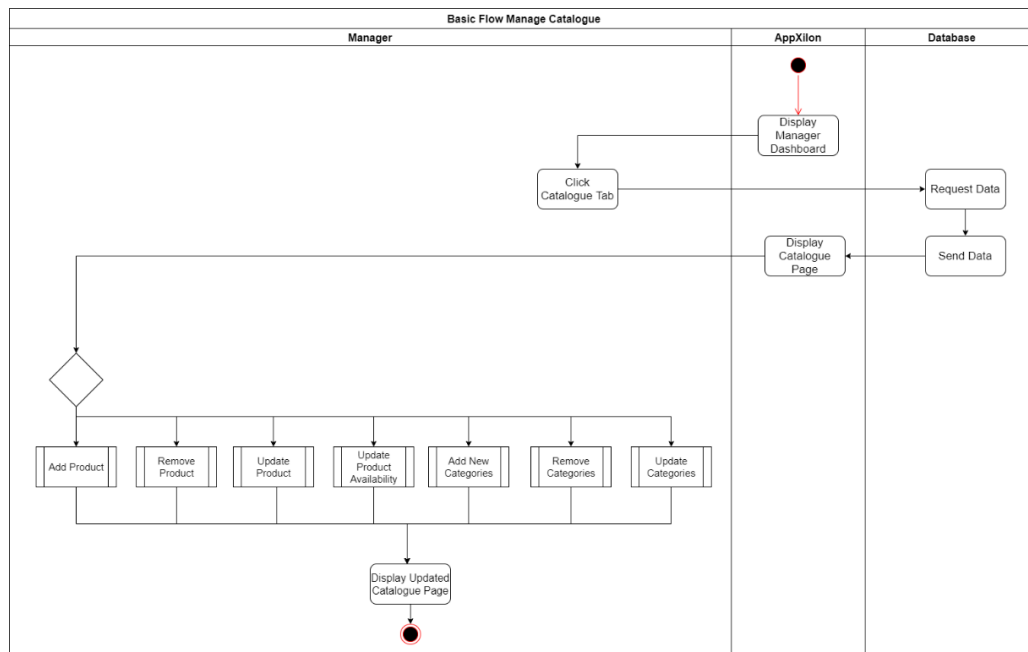


Figure 16 Activity Diagram: Manage Catalogue

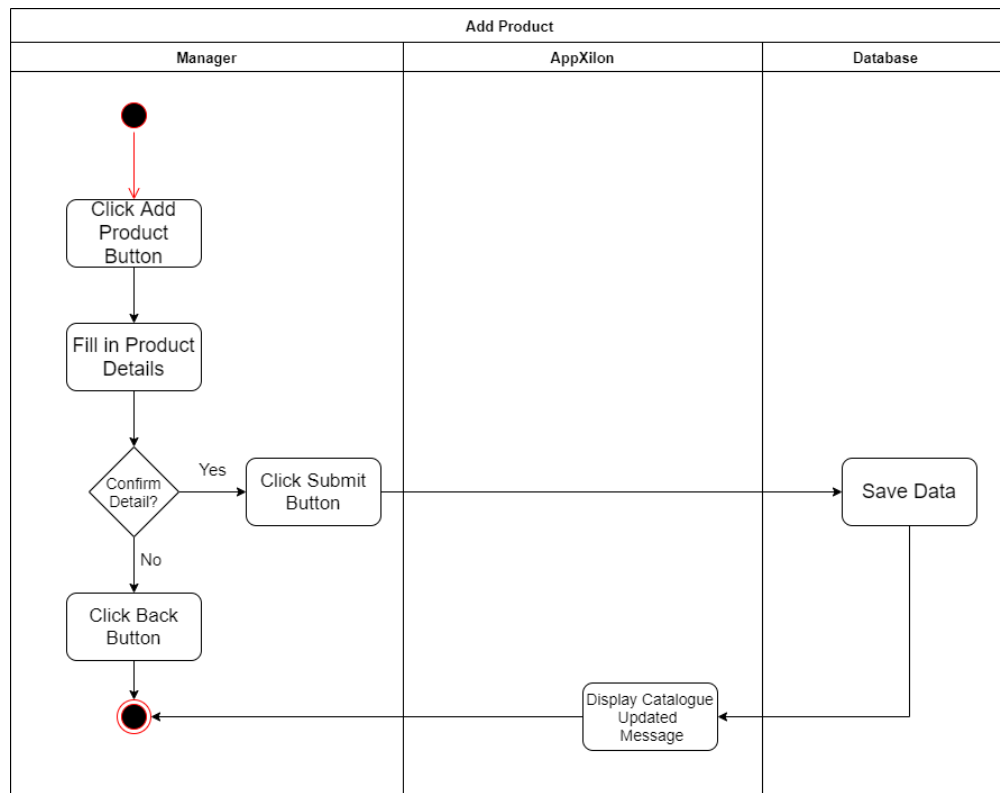


Figure 17 Activity Diagram: Add product

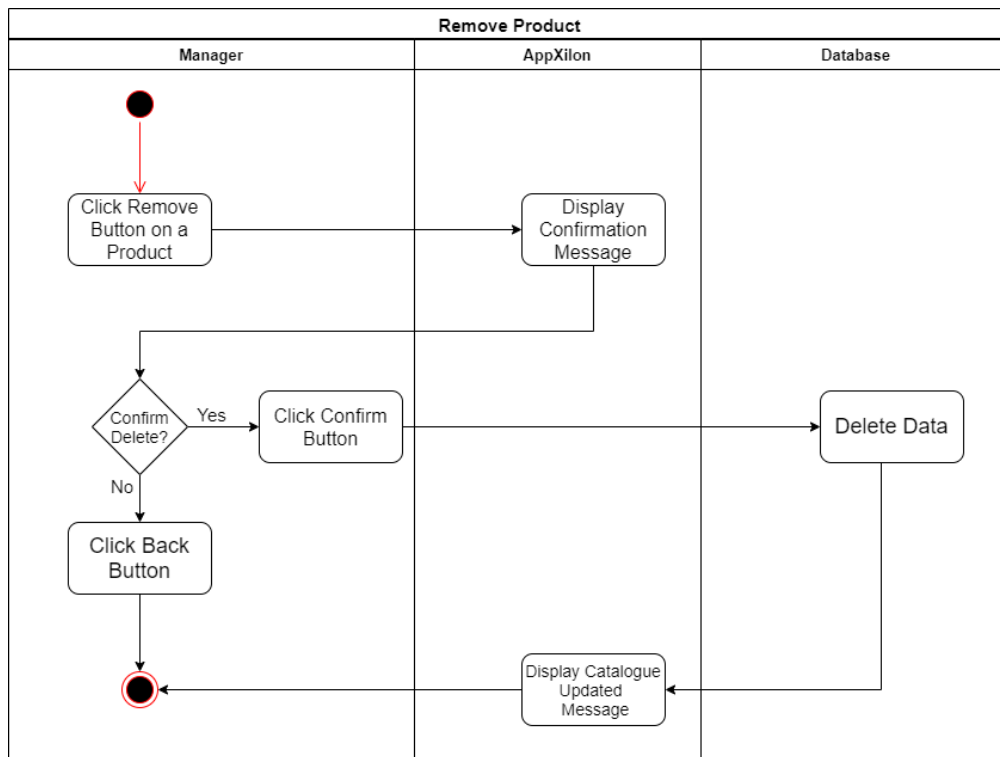


Figure 18 Activity Diagram: Remove Product

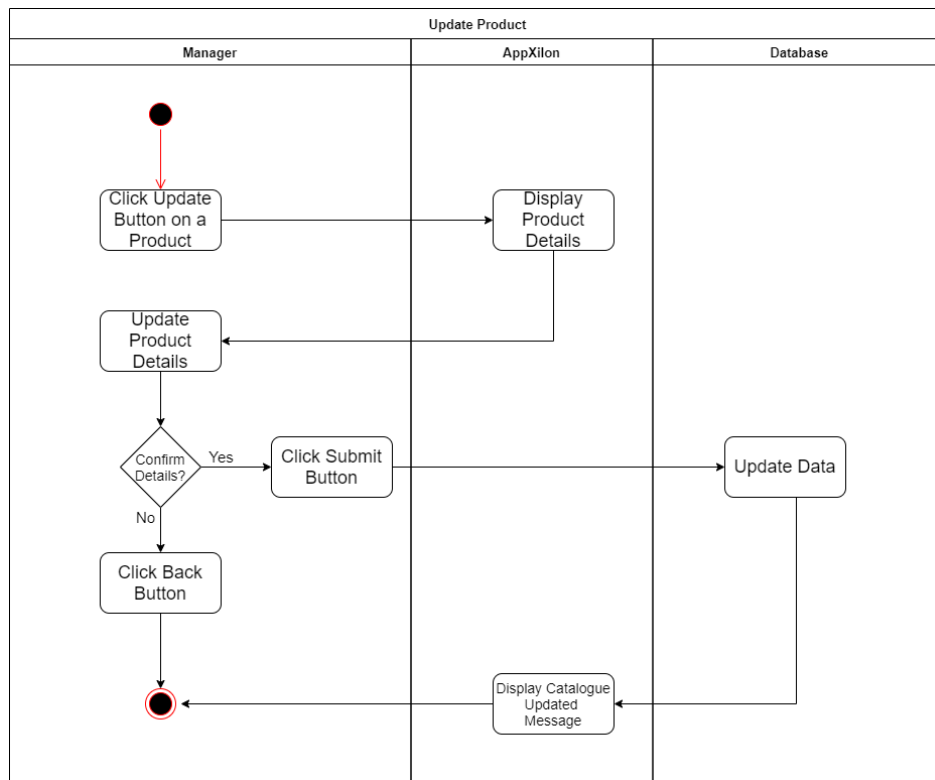


Figure 19 Activity Diagram: Update Product

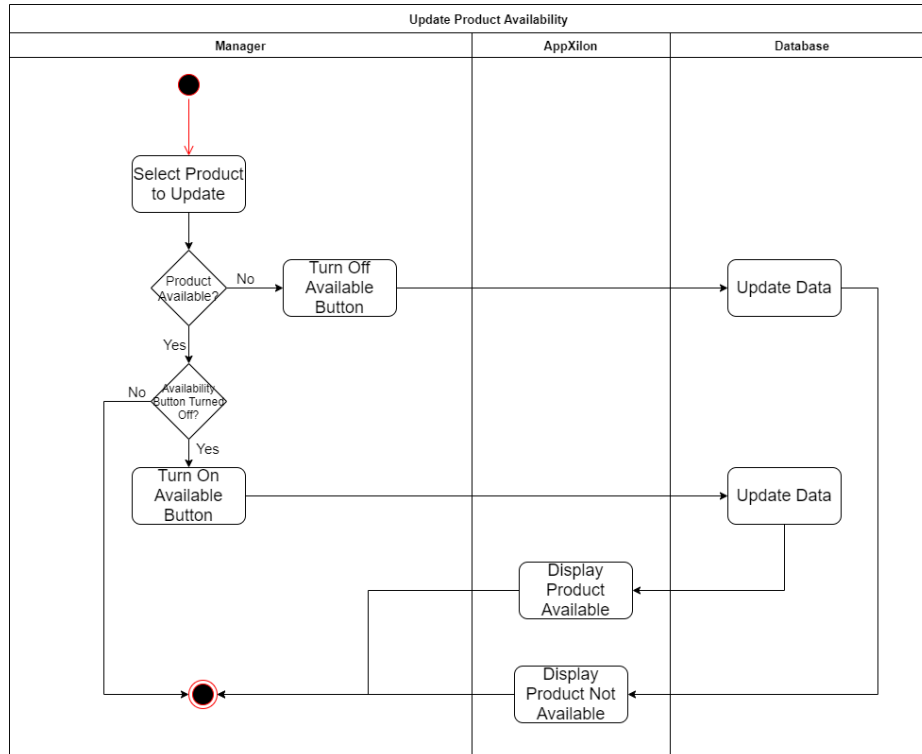


Figure 20 Activity Diagram: Update Product Availability

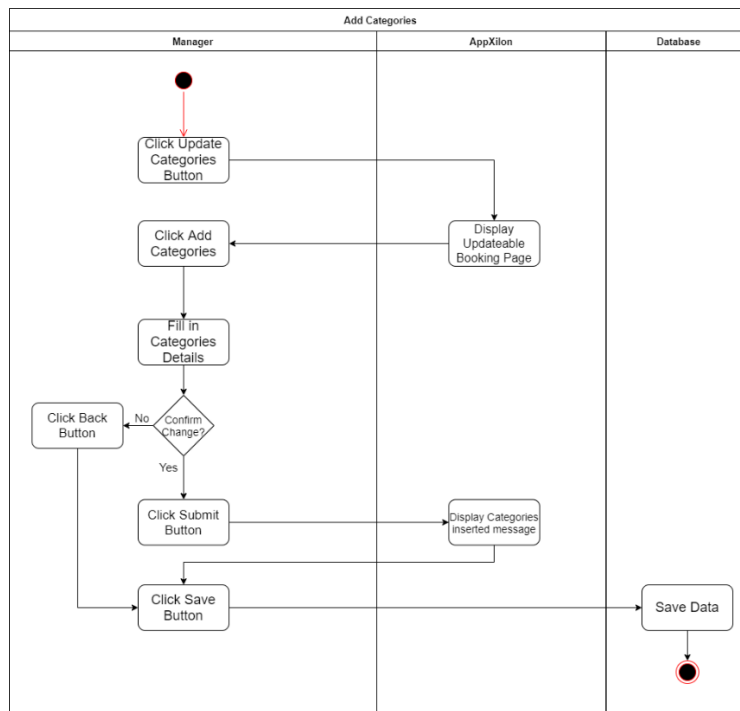


Figure 21 Activity Diagram: Add Categories

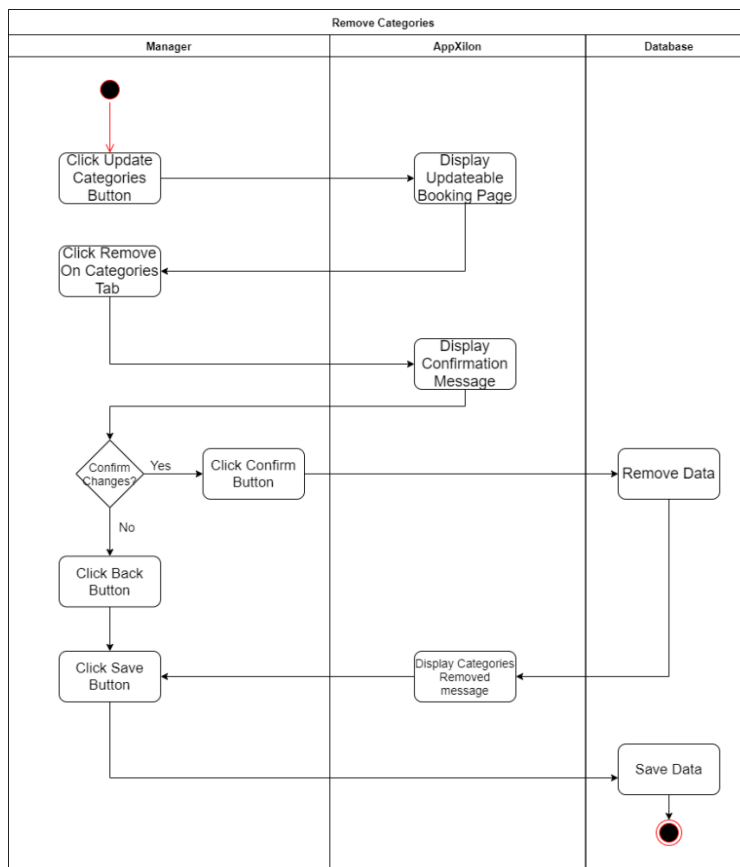


Figure 22 Activity Diagram: Remove Categories

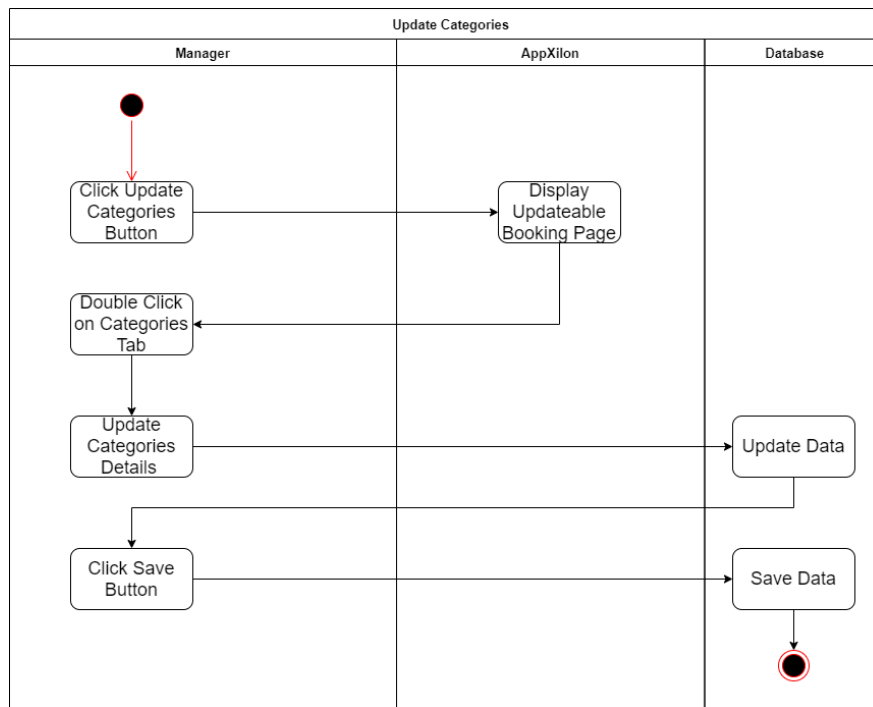


Figure 23 Activity Diagram: Update Categories

3.3.15 UC_15 Manage Table Criteria

Description	The user can update table criteria
Actors	Manager
Preconditions	The user must log in into AppXilon and clicks booking tab on manager dashboard
Basic Flow	<ol style="list-style-type: none"> 1. The user clicks on table map tab. 2. AppXilon displays restaurant table map 3. The user selects a seat. 4. AppXilon displays seat details. 5. The user updates table details. 6. The user clicks submit button. (A1: The user cancels changes) 7. Display seat details updated
Alternative Flow	A1: The user clicks cancel button. <ol style="list-style-type: none"> 6.1 The user clicks cancel button. 6.2 Flow returns to step 2.
Post-condition	AppXilon update table details
Exception Flow	N/A

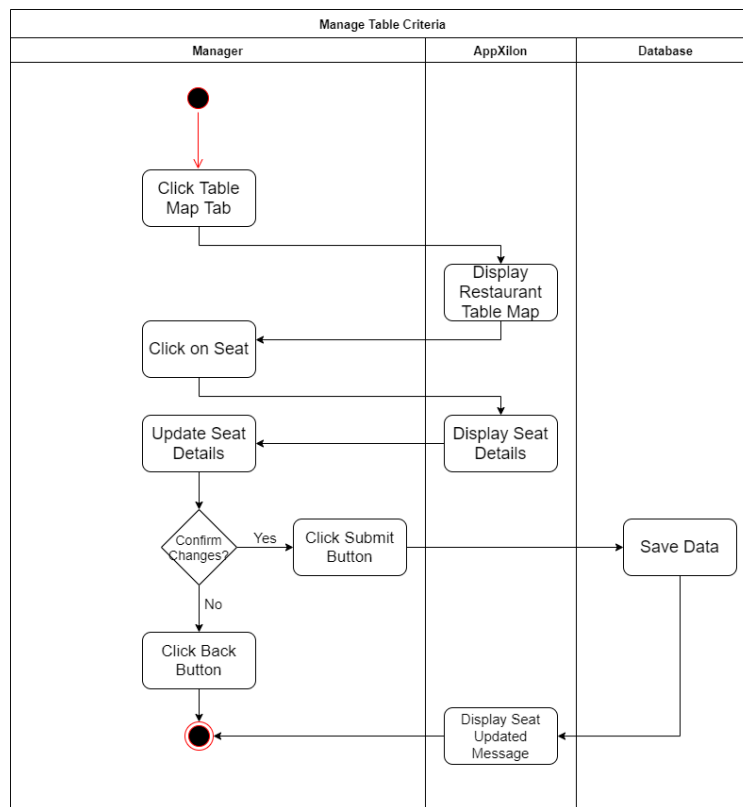


Figure 24 Activity Diagram: Manage Table Criteria

3.3.16 UC_16 Manage Order Status

Description	The user can track and update order progress
Actors	The user must login into the system
Preconditions	The user must login into the system and clicks order tab on manager dashboard
Basic Flow	<ol style="list-style-type: none"> 1. AppXilon displays current ordered product. 2. The user selects an order. 3. The user clicks order is preparing button. 4. AppXilon displays order is preparing message. 5. The user clicks order is ready button. 6. AppXilon displays order is order ready message.
Alternative Flow	N/A
Post-condition	AppXilon update order quantity.
Exception Flow	N/A

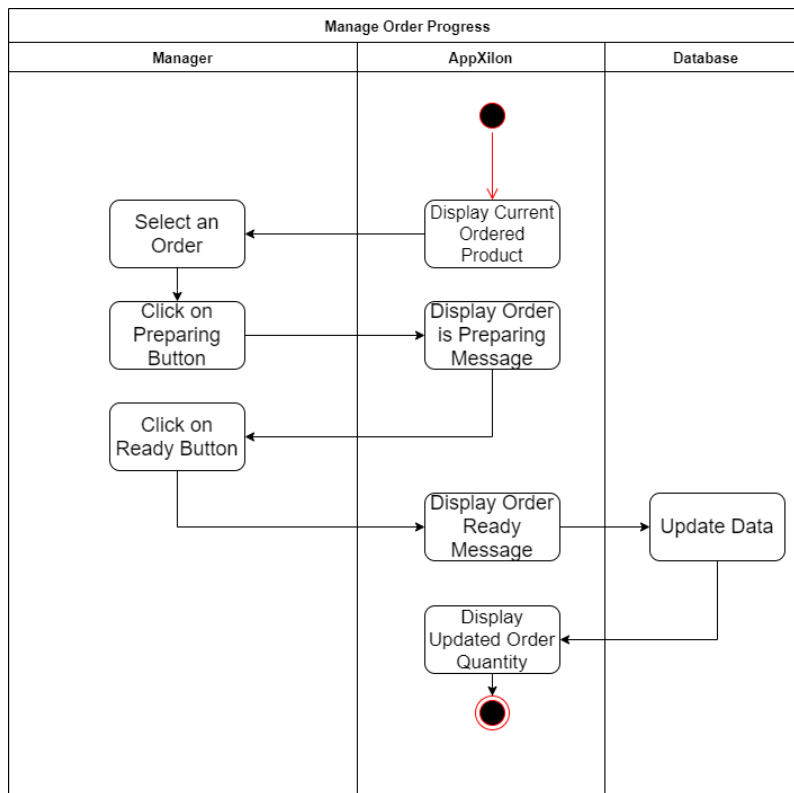


Figure 25 Activity Diagram: Manage Order Progress

3.3.17 UC_17 Manage Promotion

Description	The user can change the promotion or create the promotion of the restaurant
Actors	Manager
Preconditions	The user has to log in as manager
Basic Flow	<ol style="list-style-type: none"> 1. The user clicks promotion configuration menu 2. The user fills in product promotion form 3. The user clicks the submit button
Alternative Flow	N/A
Post-condition	AppXilon displays the promotion
Exception Flow	N/A

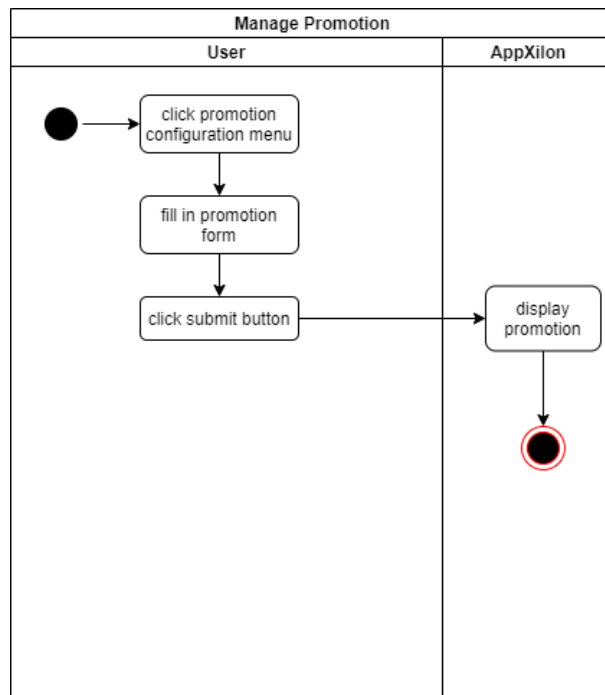


Figure 26 Activity Diagram: Manage Promotion

3.3.18 UC_18 Manage Review

Description	The user can configure the reviews to be displayed on the review page
Actors	Manager
Preconditions	The user must login into the AppXilon
Basic Flow	<ol style="list-style-type: none"> 1. The user clicks manage review page button. 2. The user clicks change review display setting (A1: Click delete review) 3. The user fills in setting form 4. The user clicks save button
Alternative Flow	A1: Delete review <ol style="list-style-type: none"> 2.1 AppXilon displays all the reviews 2.2 The user selects a review to delete 2.3 The user click delete
Post-condition	Review display is updated
Exception Flow	N/A

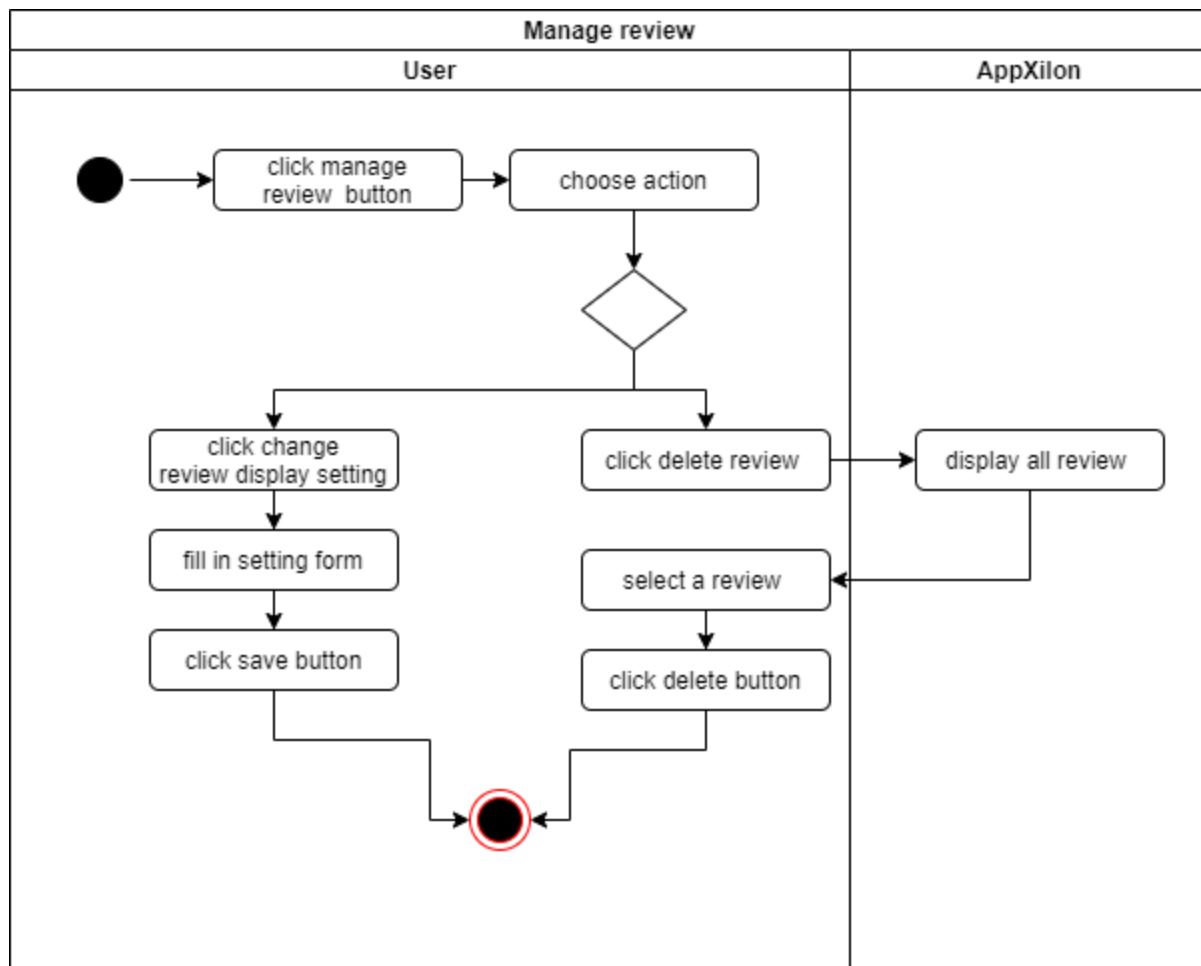


Figure 27 Activity Diagram: Manage Preview

3.3.19 UC_19 Manage FAQ

Description	The user can change the question or add a question in the FAQ
Actors	Manager
Preconditions	The user must login into the AppXilon
Basic Flow	<ol style="list-style-type: none"> 1. The user clicks manage help page button 2. AppXilon displays a list of question 3. The user click add new question (A1: Update question, A2: Delete question) 4. The user fills in question form 5. The user clicks the submit button
Alternative Flow	A1: Update question

	<p>3.1.1 The user clicks a question</p> <p>3.1.2 The user clicks update button</p> <p>3.1.3 Return to step 4</p> <p>A2: Delete question</p> <p>3.2.1 The user clicks a question</p> <p>3.2.2 The user clicks delete button</p> <p>3.2.3 The user confirms delete</p>
Post-condition	AppXilon displays the new or updated question on FAQ page
Exception Flow	N/A

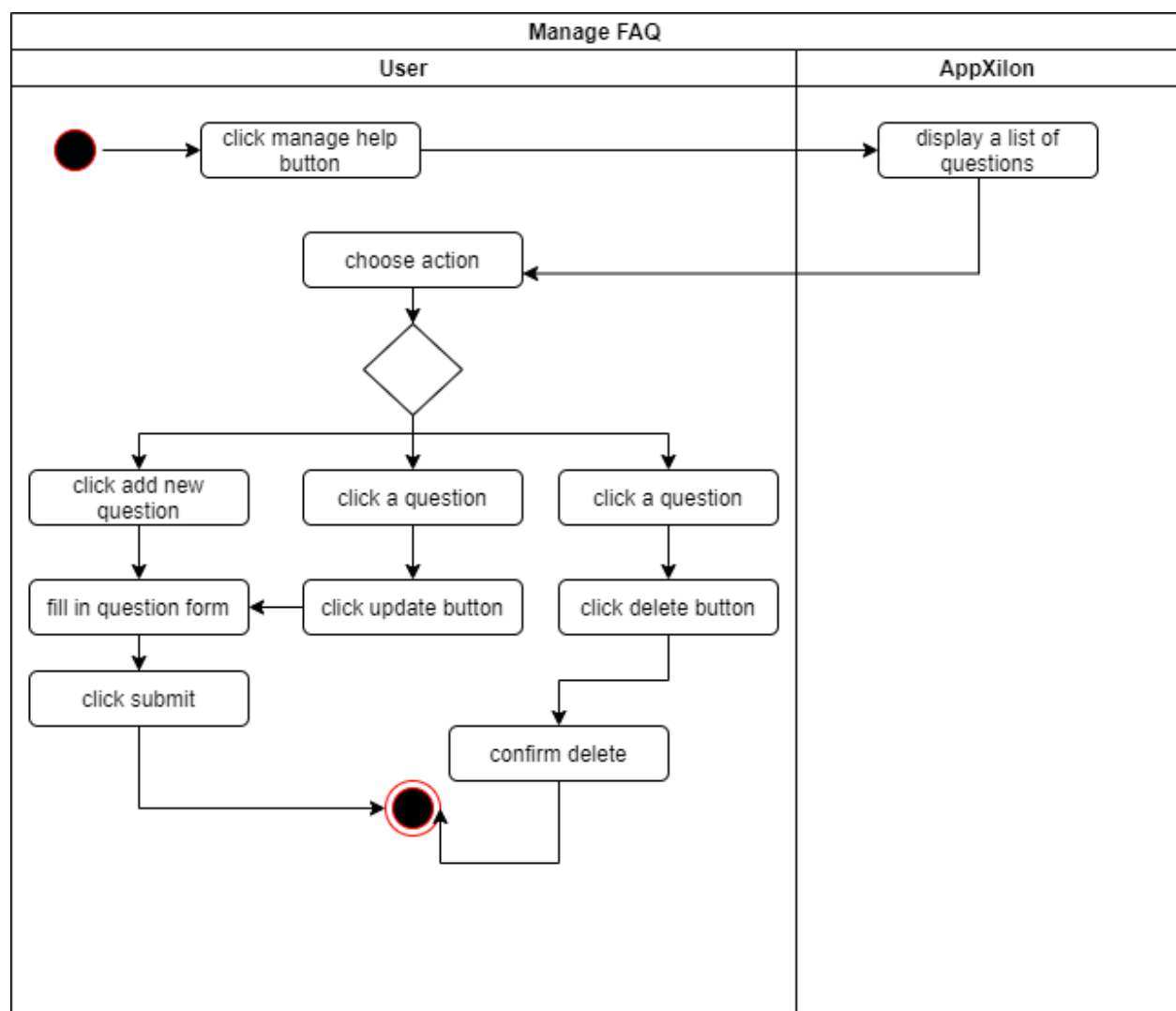


Figure 28 Activity Diagram: Manage Preview

3.3.20 UC_20 Generate Report

Description	The user can generate a result report.
Actors	Manager
Preconditions	The user clicks 'Report' tab
Basic Flow	<ol style="list-style-type: none"> 1. AppXilon retrieves the information associated with the report. 2. AppXilon displays the column names, and other report characteristics to be specified in the report. 3. The user specifies the columns to include in the report. 4. The user specifies how the report is delivered. 5. The user specifies which result pages to include in the report. 6. The user specifies the format of the generated format. 7. The user requests to generate the report. 8. AppXilon generate the report.
Alternative Flow	<ol style="list-style-type: none"> 3.1. The user does not specify the columns to include in the report. All columns are included in the report. 4.1. The user does not specify how the report is delivered. AppXilon use the default report delivery mechanism. 5.1. The user does not specify which query pages to include in the report. AppXilon includes all pages in the report 6.1 The user does not specify the format of the generated report. AppXilon uses the default report format. 7.1 The user cancels the report generation option.
Post-condition	The report is generated.
Exception Flow	N/A

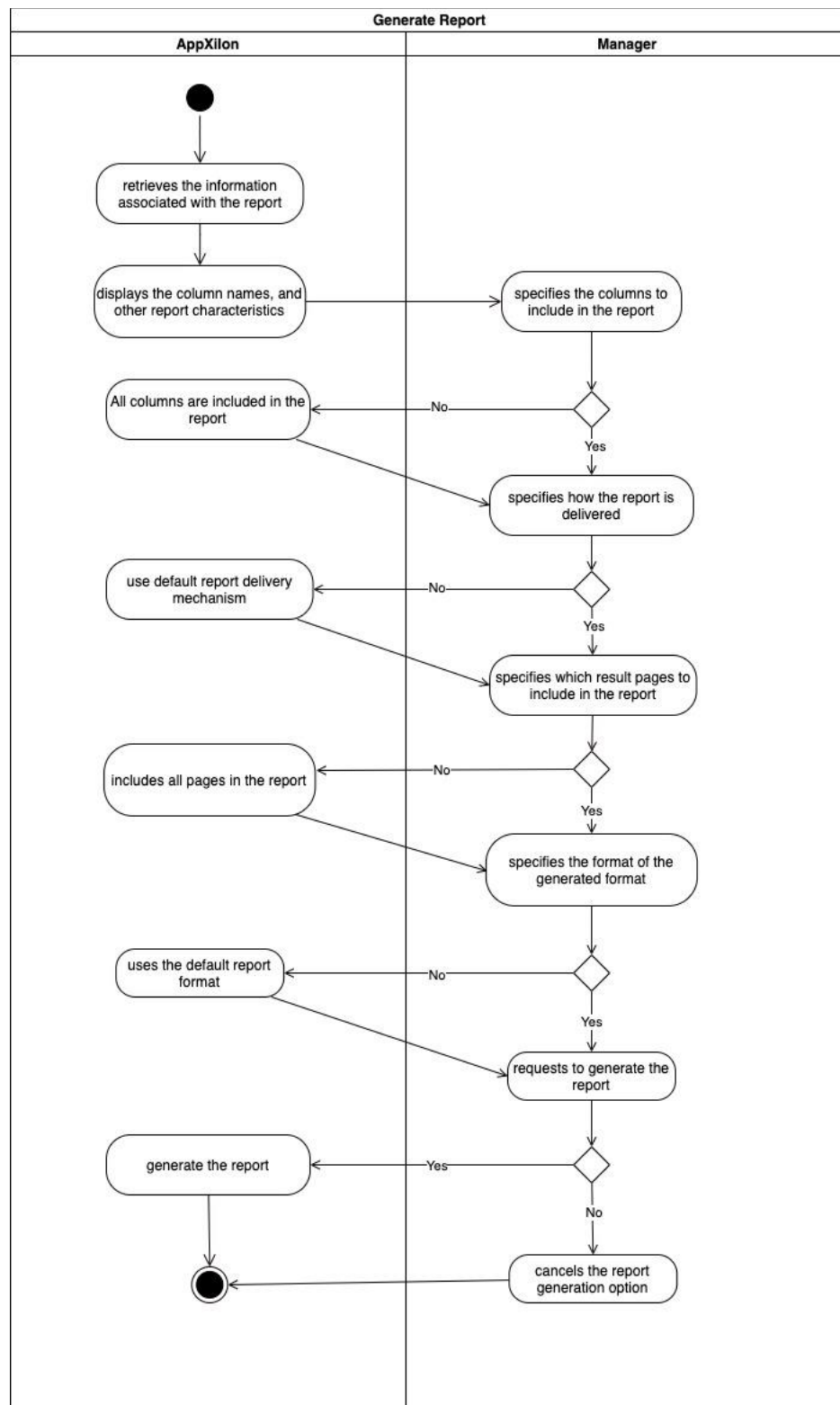


Figure 29 Activity Diagram: Generate Report

3.3.21 UC_21 View Analytics

Description	The user can view analytics from the dashboard
Actors	Manager
Preconditions	The user clicks on 'Analytics' tab
Basic Flow	<ol style="list-style-type: none"> 1. The user clicks on 'Analytics' tab. 2. AppXilon displays analytics dashboard. 3. Display Analytics as <ul style="list-style-type: none"> - Customer Sentiment (UC_22 Track Customer Sentiment) - Customer Analytics (UC_23 View Customer Analytics) - Sales Analytics (UC_24 View Sales Analytics) - Product Analytics (UC_25 View Product Analytics)
Alternative Flow	N/A
Post-condition	AppXilon visualizes the analytics in the dashboard.
Exception Flow	N/A

3.3.22 UC_22 Track Customer Sentiment

Description	The user can view customer sentiment towards the restaurant from three polarity, positive, negative, and neutral
Actors	Manager
Preconditions	The user clicks on 'Analytics' tab
Basic Flow	<ol style="list-style-type: none"> 1. AppXilon analyse reviews from customer with machine learning tool 2. AppXilon request data from database for the selected periodicity 3. AppXilon send data and present back to the end-user. 4. AppXilon displays the customer sentiment 5. The user can view the customer reviews from positive, negative, and neutral (A1: Manage Review)
Alternative Flow	A1: Manage Review <ol style="list-style-type: none"> 3.1 The user clicks on selected review 3.2 Go to UC_18 Manage Review
Post-condition	The customer sentiment is display on AppXilon dashboard.
Exception Flow	N/A

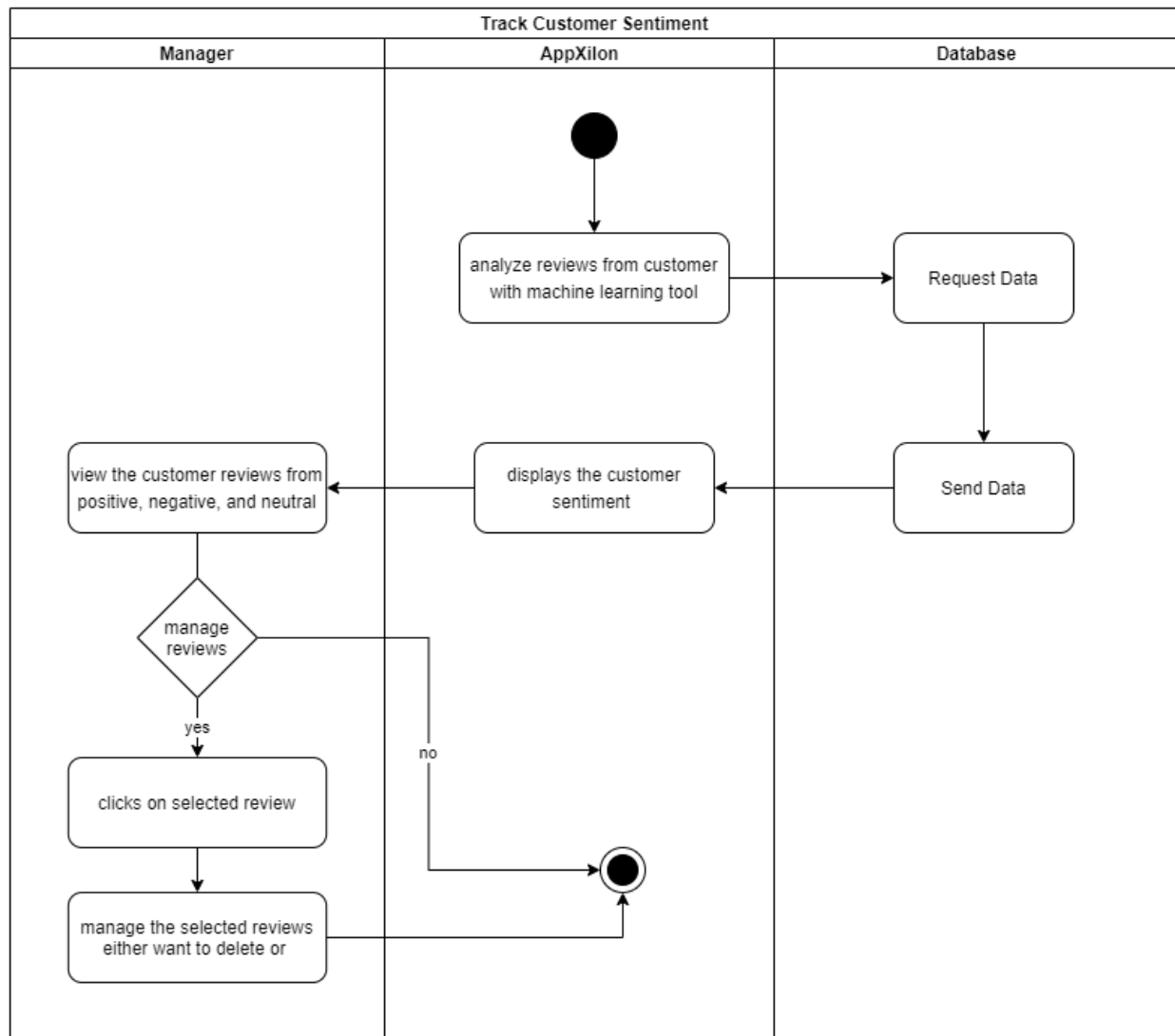


Figure 30 Activity Diagram: Track Customer Sentiment

3.3.23 UC_23 View Customer Analytics

Description	The user can analyse customer background data for business decision making
Actors	Manager
Preconditions	The user clicks on 'Analytics' tab
Basic Flow	<ol style="list-style-type: none"> 1. AppXilon retrieves the analytics the associated with customer performance. 2. AppXilon displays the number of customers accumulated, today's customer and customer demographic breakdown.

	3. The user specifies a period for the customer performance 4. AppXilon displays the customer analytics for that period.
Alternative Flow	N/A
Post-condition	The customer analytics is display on AppXilon dashboard.
Exception Flow	N/A

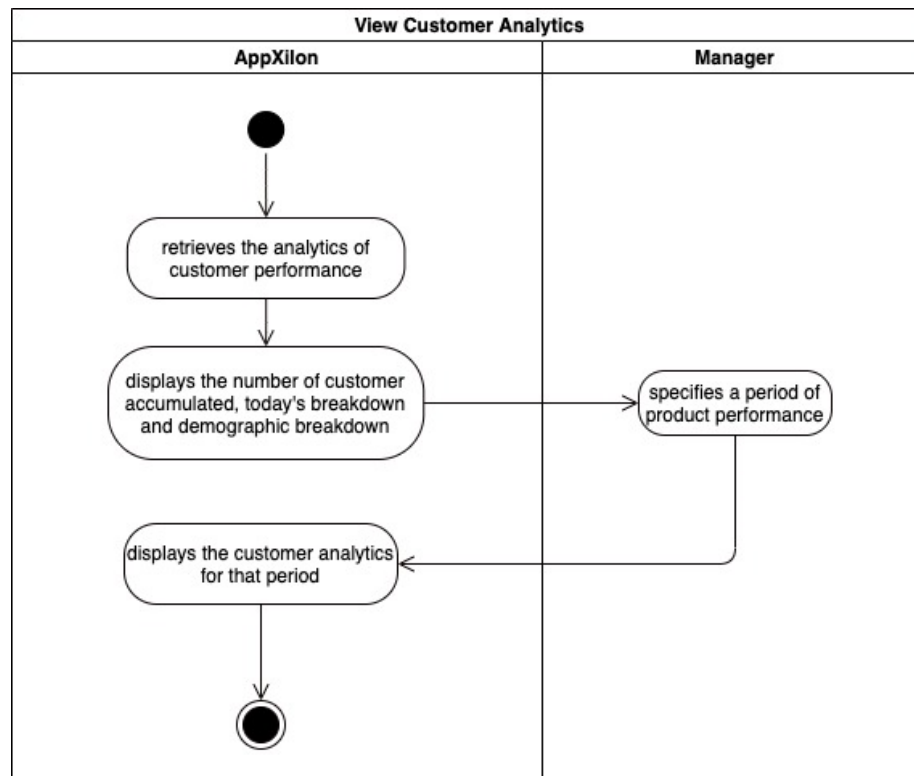


Figure 31 Activity Diagram: View Customer Analytics

3.3.24 UC_24 View Sales Analytics

Description	The user can view the analysis of current, past, and future of sales performance
Actors	Manager
Preconditions	The user clicks on 'Analytics' tab
Basic Flow	1. AppXilon retrieves the analytics that associated with sales performance. 2. AppXilon displays the comparison between current, past, future of sales performance.

	3. The user specifies a period for sales. 4. AppXilon displays the sales analytics for that period.
Alternative Flow	N/A
Post-condition	The sales analytics is display on AppXilon dashboard.
Exception Flow	N/A

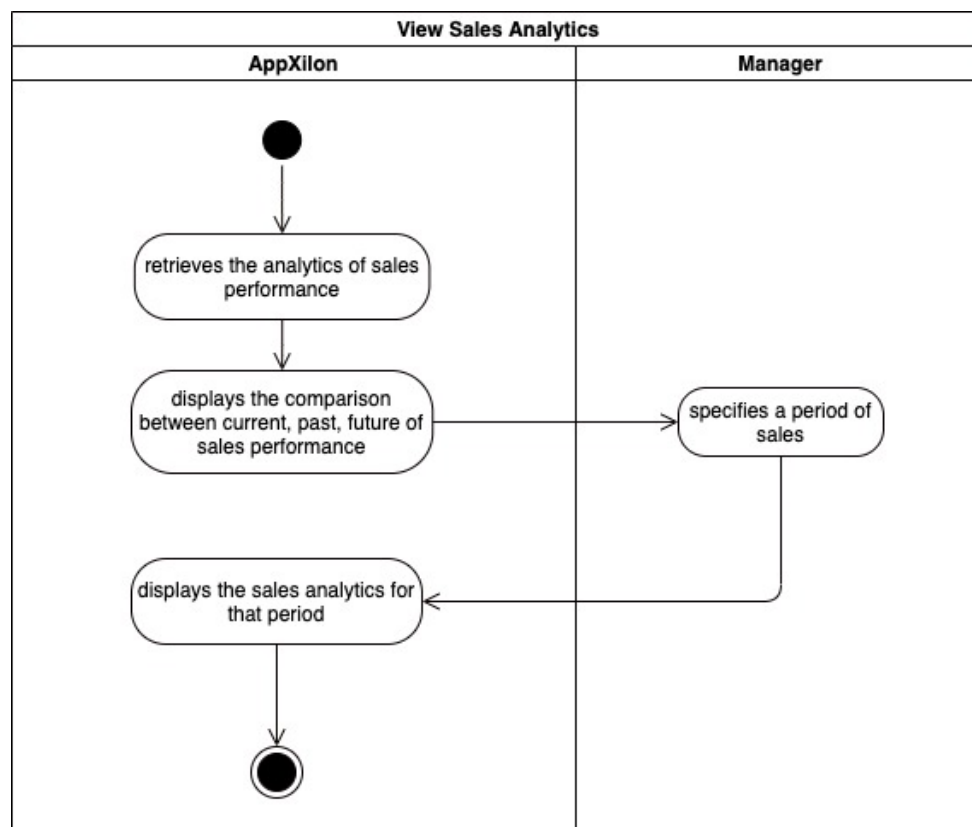


Figure 32 Activity Diagram: View Sales Analytics

3.3.25 UC_25 View Product Analytics

Description	The user can view the analysis of product performance such as popularity of products.
Actors	Manager
Preconditions	The user must login into AppXilon
Basic Flow	1. AppXilon retrieves the analytics the associated with product performance.

	<ol style="list-style-type: none"> AppXilon displays the top product popularity The user specifies a period for the product performance AppXilon displays the product analytics for that period.
Alternative Flow	N/A
Post-condition	The product analytics is display on AppXilon dashboard.
Exception Flow	N/A

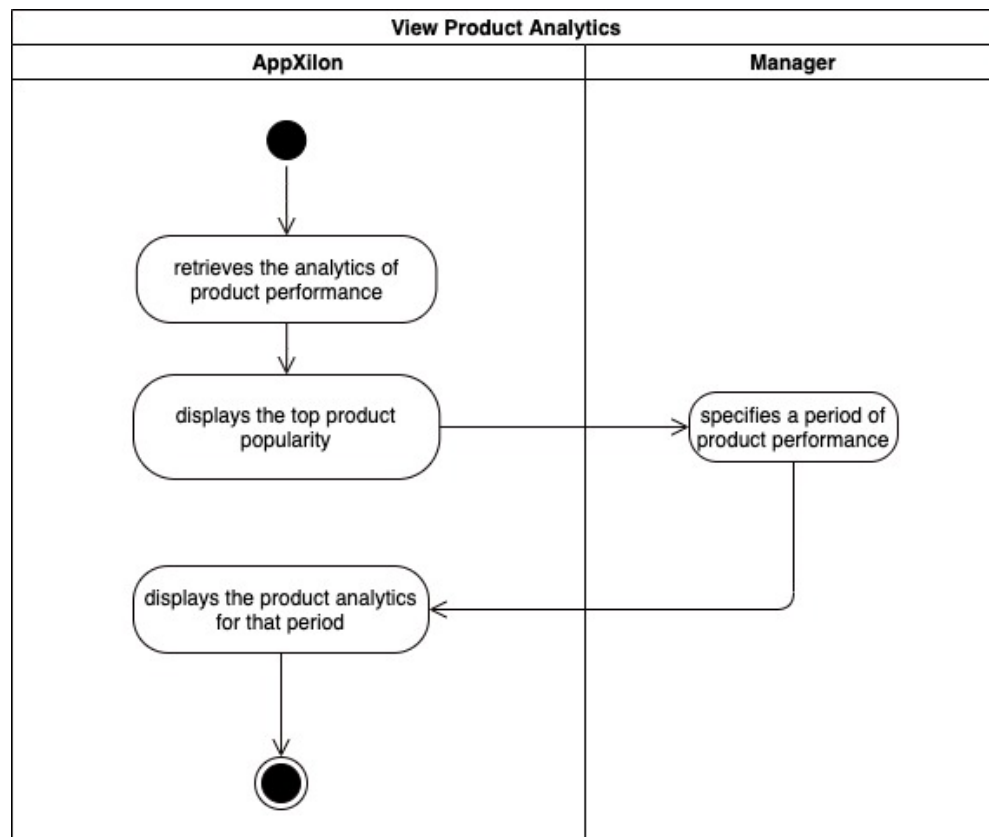


Figure 33 Activity Diagram: View Product Analytics

3.3.26 UC_26 Manage Payment

Description	The user can manage the type of payment user can be used
Actors	Administrator
Preconditions	The user must login into AppXilon
Basic Flow	<ol style="list-style-type: none"> The user clicks settings tab. The user chooses payments. The user chooses the payment type.

	<ol style="list-style-type: none"> 4. The user clicks submit. 5. The system will update the payment type in AppXilon
Alternative Flow	N/A
Post-condition	AppXilon displays the updated payment types.
Exception Flow	N/A

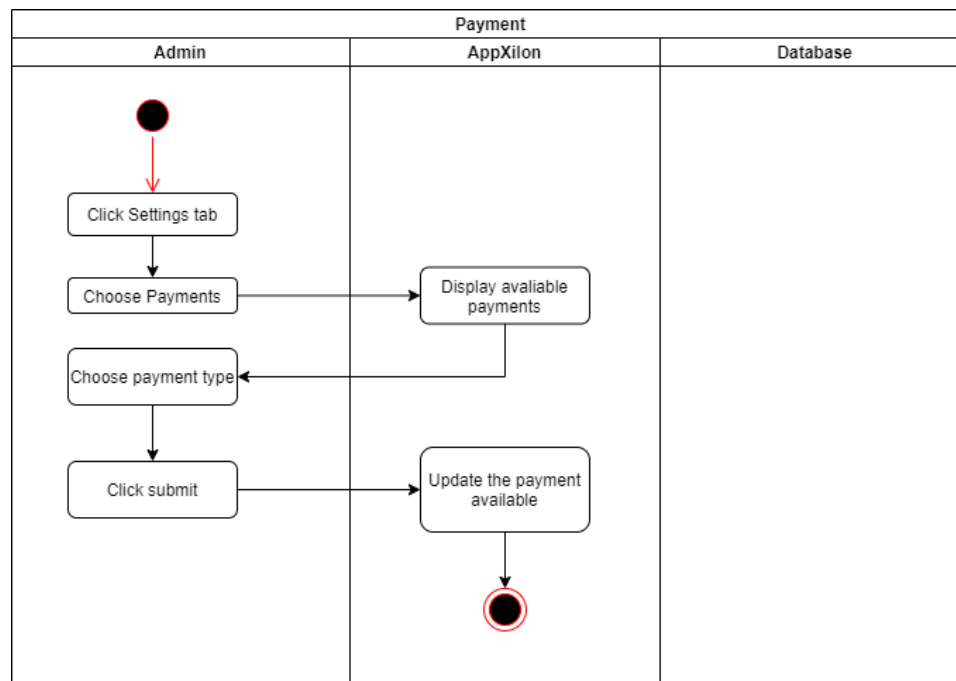


Figure 34 Activity Diagram: Payment

3.3.27 UC_27 Manage Business Hours

Description	The user can update the store business hours
Actors	Administrator
Preconditions	The user must login into the system
Basic Flow	<ol style="list-style-type: none"> 1. The user clicks Restaurant tab. 2. The user chooses Business Hour 3. The user chooses the start time and end time for each day. 4. The user clicks save button. 5. The business hour will be updated in AppXilon.
Alternative Flow	N/A
Post-condition	The updated business hour will be displayed in AppXilon

Exception Flow	N/A
-----------------------	-----

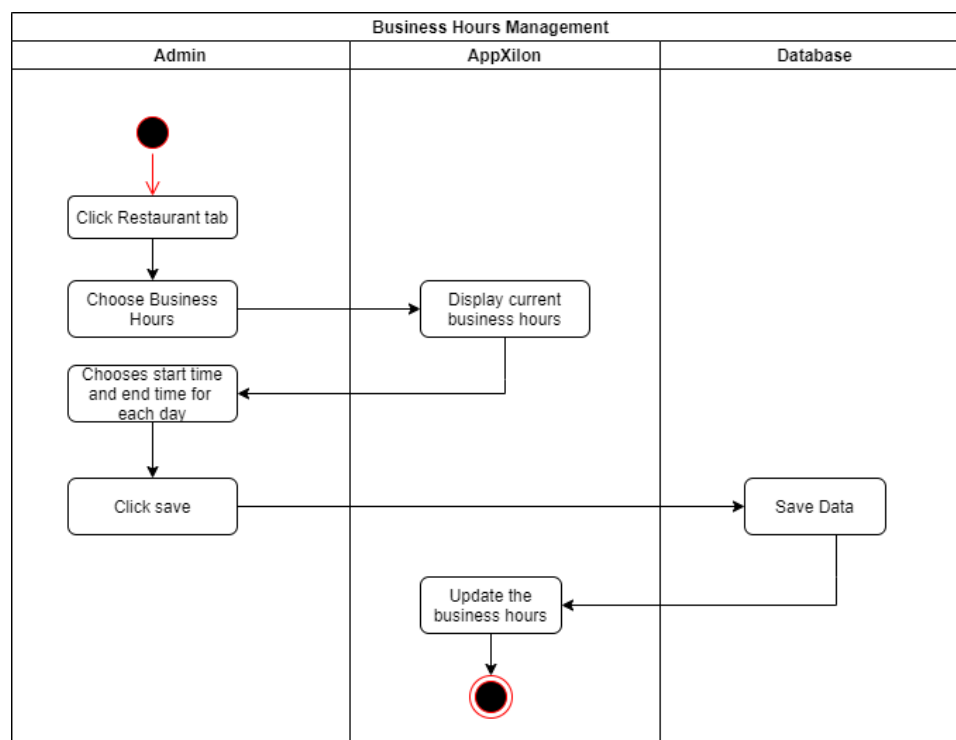


Figure 35 Activity Diagram: Business Hours Management

3.3.28 UC_28 Manage Term Condition

Description	The user can update the terms and conditions
Actors	Administrator
Preconditions	The user must login into AppXilon
Basic Flow	<ol style="list-style-type: none"> 1. The user clicks the settings tab. 2. The user chooses Terms & Conditions 3. The user inserts the store's terms & conditions and enquiry. 4. The user clicks save button. 5. AppXilon will save the terms & conditions and enquiry into the database.
Alternative Flow	N/A
Post-condition	AppXilon displays the updated terms and conditions
Exception Flow	N/A

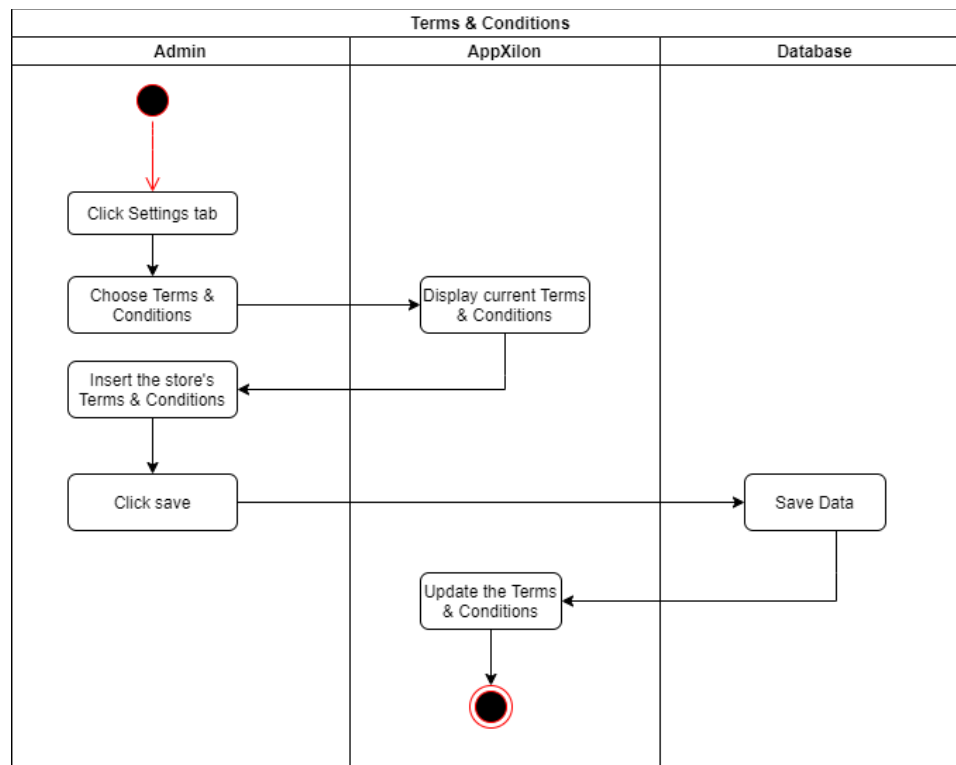


Figure 36 Activity Diagram: Terms and Conditions

3.3.29 UC_29 Manage Reminder Reservations

Description	The user can update the reminder reservation
Actors	Administrator
Preconditions	The user must login into AppXilon
Basic Flow	<ol style="list-style-type: none"> 1. The user clicks the Settings tab. 2. The user chooses reminder. 3. AppXilon will display enable emails notification and enable SMS notification. 4. The user can choose between email notifications and SMS notifications (A1: Email Notifications, A2: SMS Notifications)
Alternative Flow	A1: Email Notifications <ol style="list-style-type: none"> 4.1 The user checks the Email notifications. 4.2 The user inserts the details for email notifications. 4.3 The user clicks save button.

	A2: SMS Notifications 4.1 The user checks the SMS notifications. 4.2 The user inserts the details for SMS notifications. 4.3 The user clicks save button.
Post-condition	AppXilon will save the notifications type and details into AppXilon.
Exception Flow	N/A

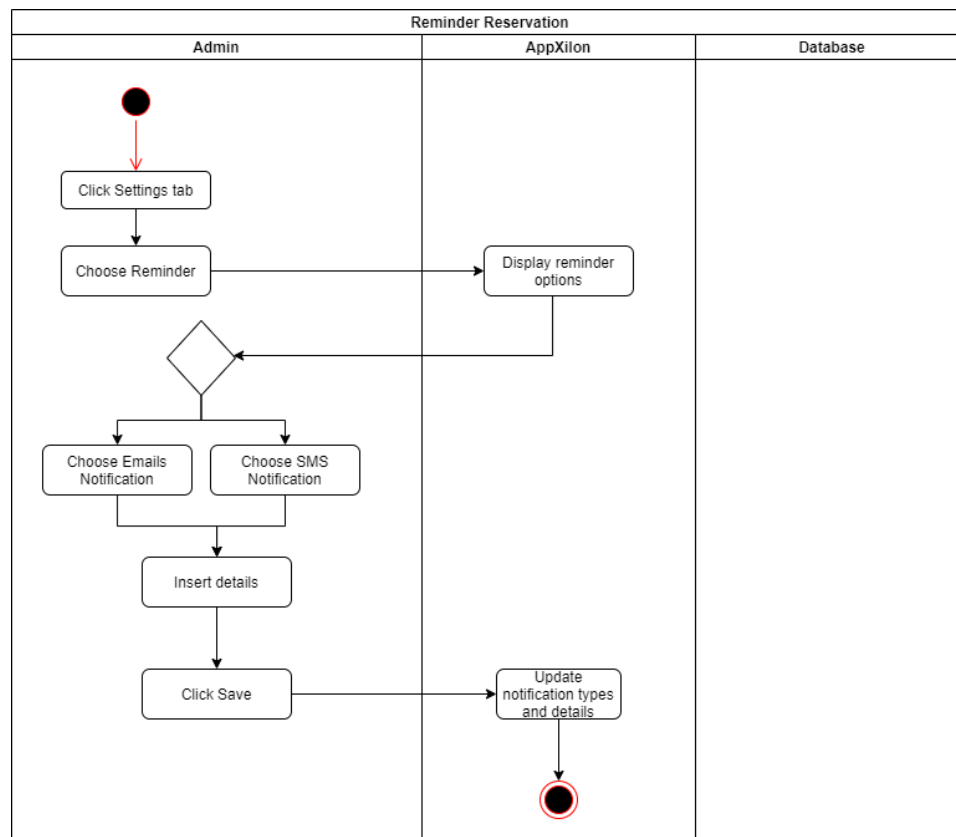


Figure 37 Activity Diagram: Reminder Reservations

3.3.30 UC_30 Manage Reservation Form

Description	The user can manage the reservation forms that customer needs to fill in to make a reservation
Actors	Administrator
Preconditions	The user must login into AppXilon
Basic Flow	<ol style="list-style-type: none"> 1. The user clicks Settings tab. 2. The user chooses Reservation form.

	<ol style="list-style-type: none"> AppXilon will display reservation form options. The user inserts the fields that should be available on the reservation form. The user clicks save button. The reservation form will be updated into AppXilon
Alternative Flow	N/A
Post-condition	AppXilon displays the updated reservation form.
Exception Flow	N/A

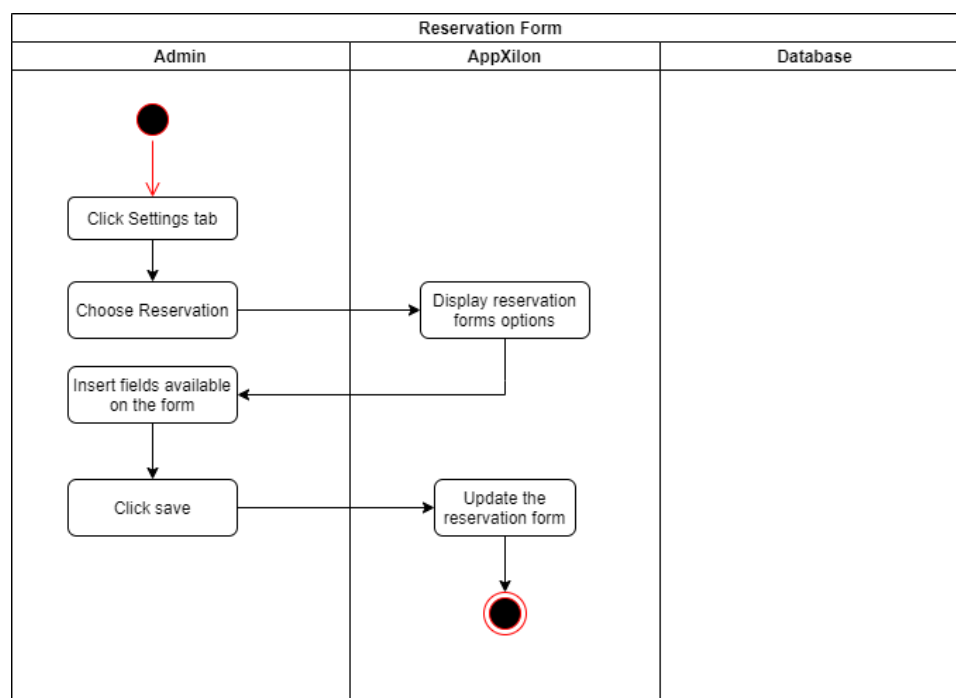


Figure 38 Activity Diagram: Reservation Form

3.3.31 UC_31 Manage Captcha and Form

Description	The user can manage the captcha to ensure security
Actors	Administrator
Preconditions	The user must login into AppXilon
Basic Flow	<ol style="list-style-type: none"> The user clicks System Options tab. The user chooses Captcha & Spam The user set the captcha and spam protection settings. The user clicks save button.

	5. The captcha and spam protection settings will be updated into AppXilon.
Alternative Flow	N/A
Post-condition	AppXilon displays the updated captcha when login.
Exception Flow	N/A

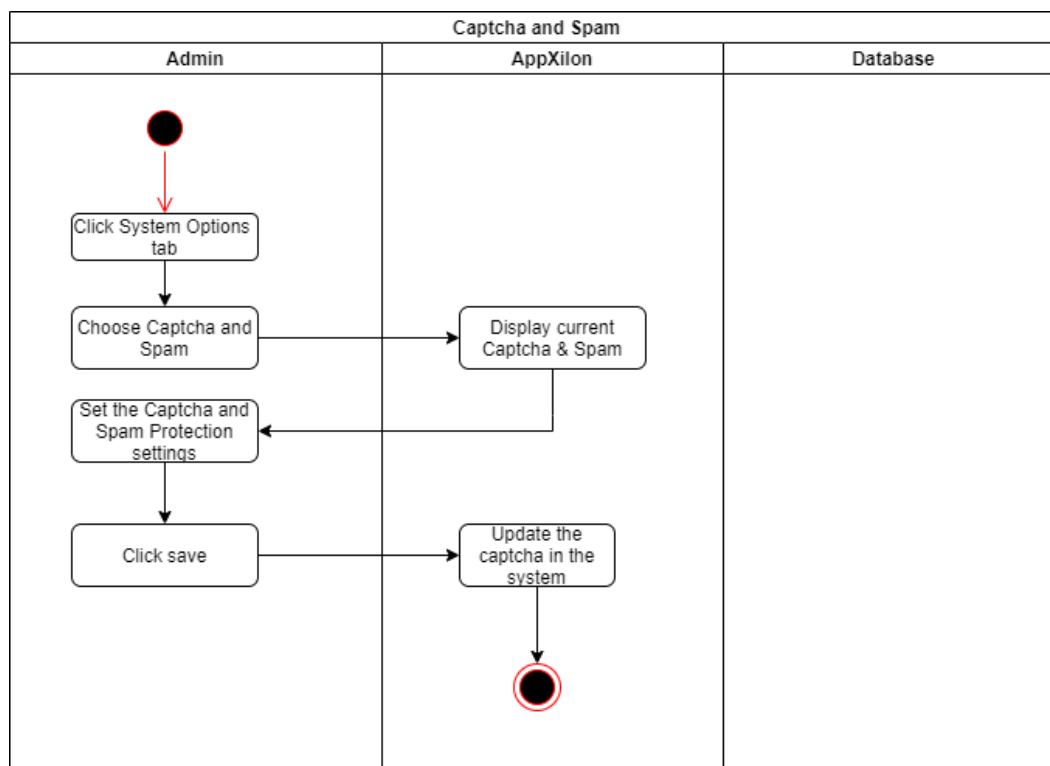


Figure 39 Activity Diagram: Captcha and Spam

3.3.32 UC_32 Manage Backup

Description	The user can manage the backup such as the table
Actors	Administrator
Preconditions	The user must login into AppXilon
Basic Flow	<ol style="list-style-type: none"> 1. The user clicks on backup tab. 2. The user clicks create backup (A1: The user clicks restore backup) 3. The user chooses tables that want to be backed up (E1: Tables not chosen) 4. AppXilon will create a local backup in the user device.

Alternative Flow	A1: The user clicks restore backup. 2.1 The user chooses files need to be imported. 2.2 The user clicks confirm.
Post-condition	AppXilon will create a backup, or the latest database will be updated
Exception Flow	E1: Tables not chosen. 3.1 AppXilon will display message “Tables not chosen”. 3.2 The flow returns to basic

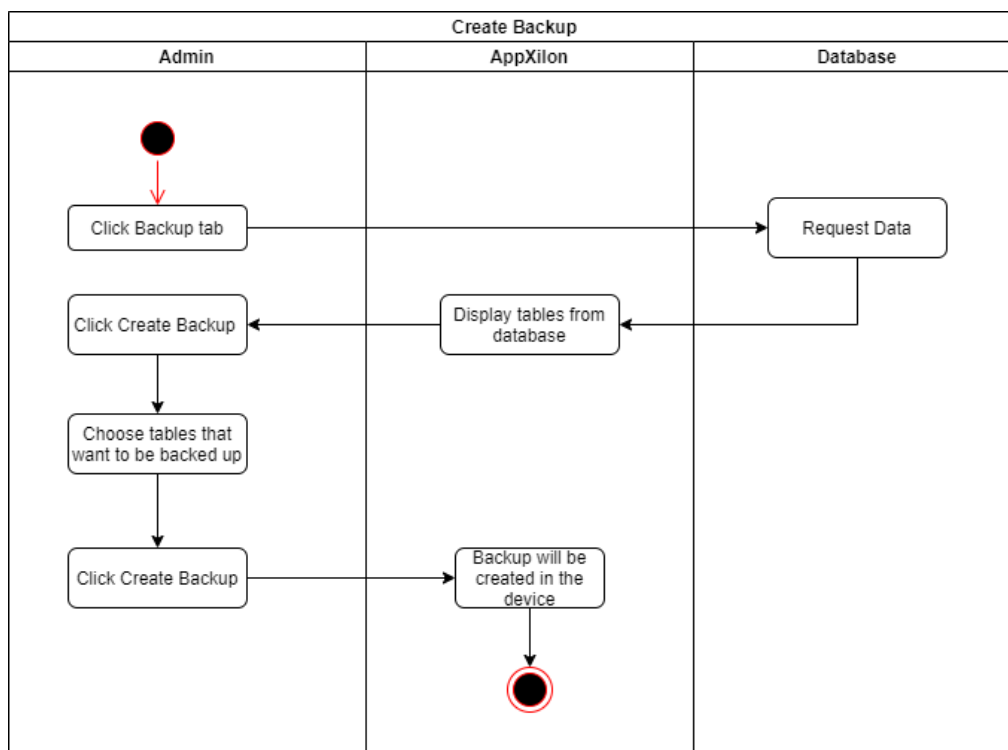


Figure 40 Activity Diagram: Create Backup

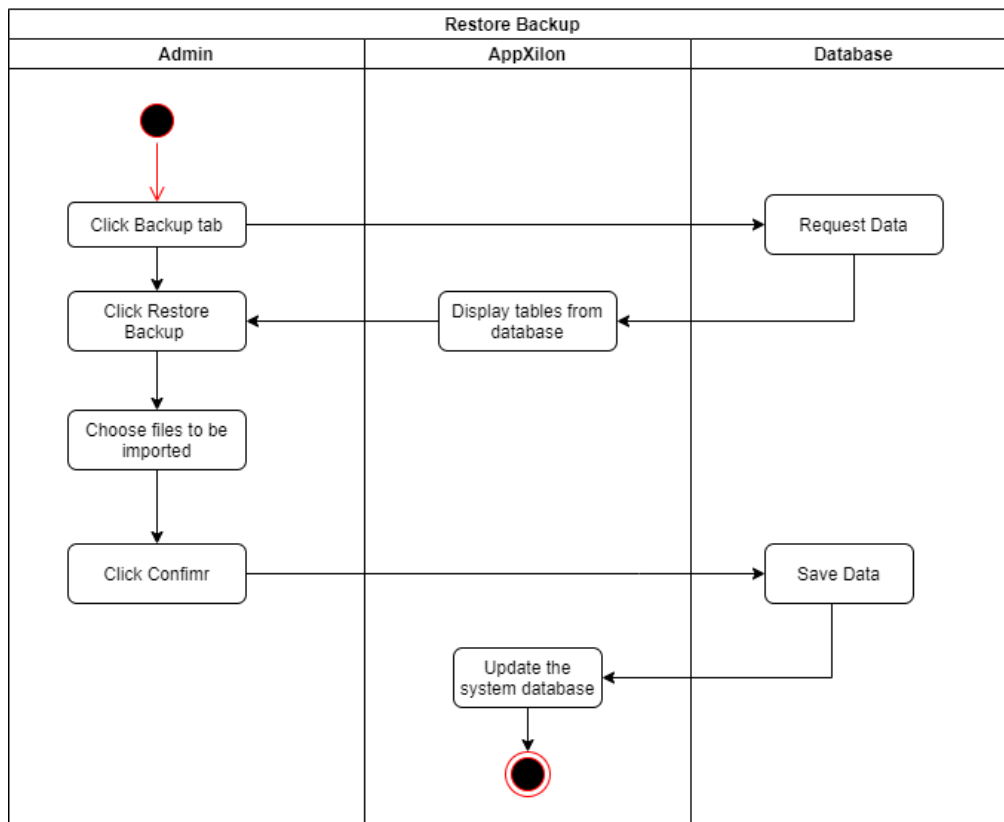


Figure 41 Activity Diagram: Restore Backup

3.3.33 UC_33 Manage User

Description	The user can manage the user's access control on the website
Actors	Administrator
Preconditions	The user must login into AppXilon
Basic Flow	<ol style="list-style-type: none"> 1. The user click manage user tab. 2. The user chooses manager. (A1: The user chooses customer) 3. The user can choose to create, update or delete manager. (A2: Create Manager, A3: Update Manager, A4: Delete Manager)
Alternative Flow	<p>A1: The user choose customer.</p> <ol style="list-style-type: none"> 2.1 The user can choose to create, update or delete customer (A1.1: Create Customer, A1.2: Update Customer, A1.3: Delete Customer) <p>A1.1: Create Customer</p> <ol style="list-style-type: none"> 2.1.1.1 The user clicks create customer button.

	<p>2.1.1.2 The user enters customer details.</p> <p>2.1.1.3 The user clicks submit button.</p> <p>2.1.1.4 AppXilon will add the new customer into the database.</p> <p>A1.2: Update Customer</p> <p>2.1.2.1 The user clicks update customer button.</p> <p>2.1.2.2 The user enters the updated customer details.</p> <p>2.1.2.3 The user clicks submit button.</p> <p>2.1.2.4 AppXilon will update the latest customer details into the database.</p> <p>A1.3: Delete Customer</p> <p>2.1.3.1 The user clicks delete manager.</p> <p>2.1.3.2 AppXilon will prompt a confirmation message.</p> <p>2.1.3.3 The user clicks confirm.</p> <p>2.1.3.4 AppXilon will delete the manager from the database.</p> <p>A2: Create Manager</p> <p>3.1.1 The user clicks create manager button.</p> <p>3.2.1 The user enters manager details.</p> <p>3.2.2 The user clicks submit button.</p> <p>3.2.3 AppXilon will add the new manager into the database.</p> <p>A3: Update Manager</p> <p>3.2.1 The user clicks update manager button.</p> <p>3.2.2 The user enters the updated manager details.</p> <p>3.2.3 The user clicks submit button.</p> <p>3.2.4 AppXilon will update the latest manager details into the database.</p> <p>A4: Delete Manager</p> <p>3.3.1 The user clicks delete manager.</p> <p>3.3.2 AppXilon will prompt a confirmation message.</p> <p>3.3.3 The user clicks confirm.</p> <p>AppXilon will delete the manager from the database.</p>
Post-condition	AppXilon displays the updated user details.
Exception Flow	N/A

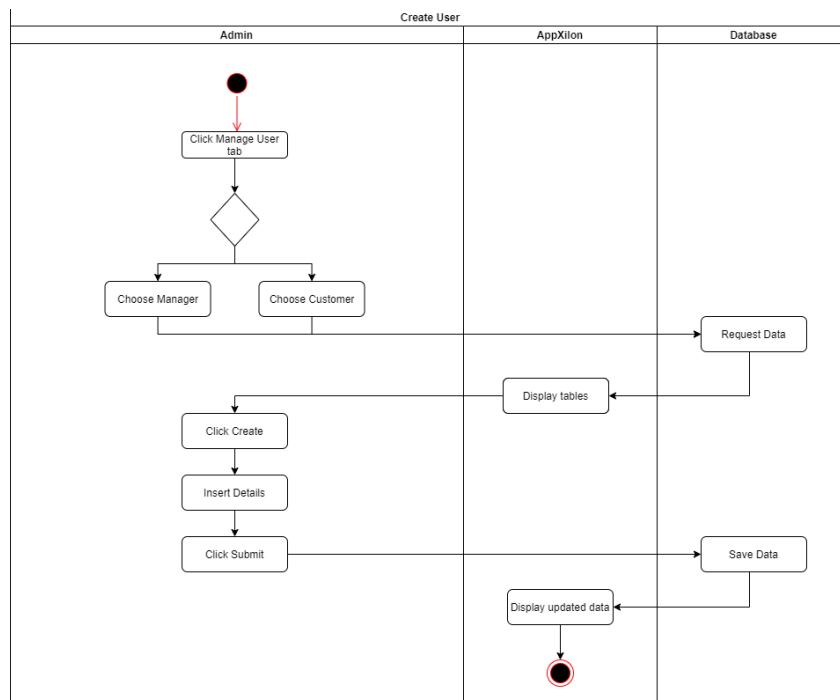


Figure 42 Activity Diagram: Create User

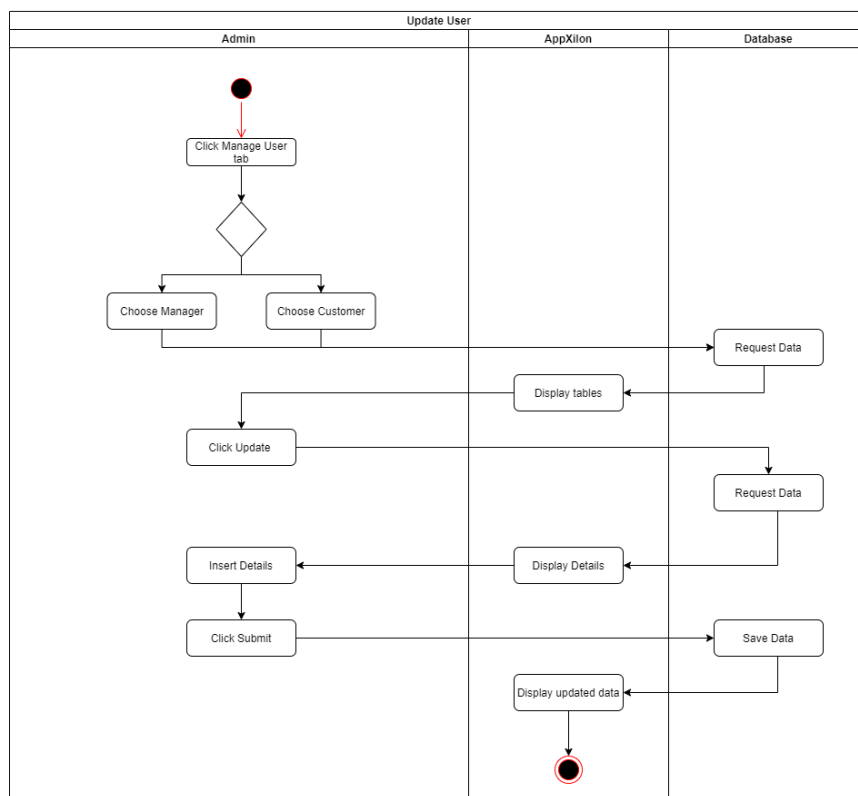


Figure 43 Activity Diagram: Update User

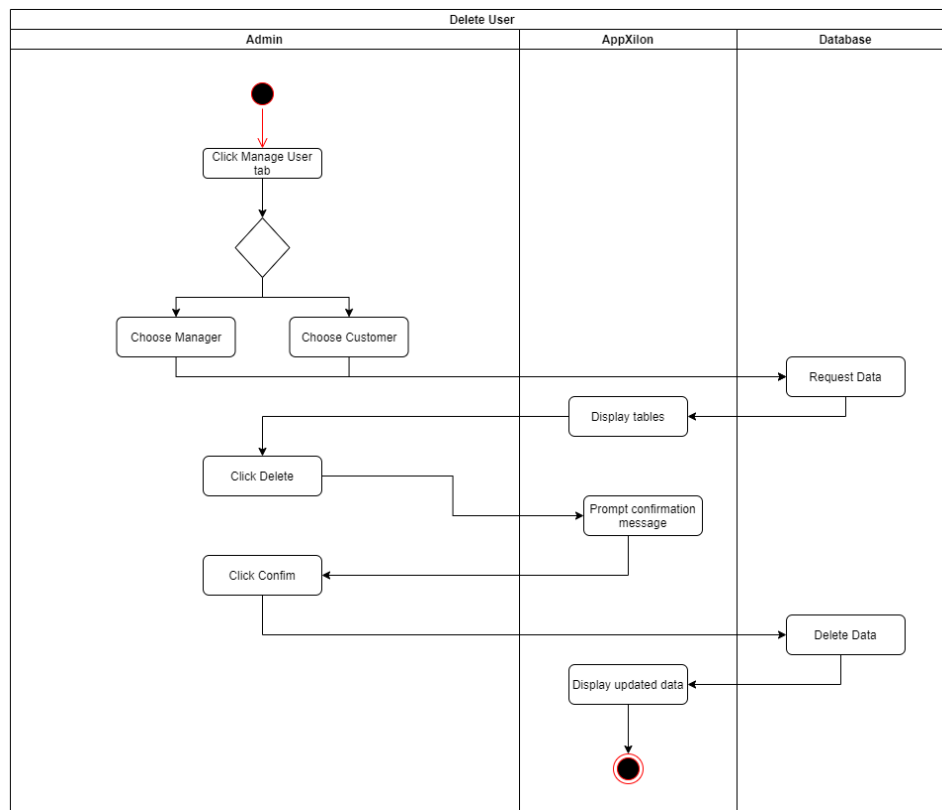


Figure 44 Activity Diagram: Delete User

3.3.34 UC_34 Database Cleanup

Description	The user can manage the database cleanup
Actors	Administrator
Preconditions	The user must login into AppXilon
Basic Flow	<ol style="list-style-type: none"> 1. The user clicks on database cleanup tab. 2. The user chooses date database need to be cleaned. 3. The user clicks clean. (A1: The user clicks cancel) 4. AppXilon will clean database based on the date.
Alternative Flow	A1: The user clicks cancel. <ol style="list-style-type: none"> 3.1 The user clicks cancel. 3.2 AppXilon will return to the user main page.
Post-condition	AppXilon displays the updated database
Exception Flow	N/A

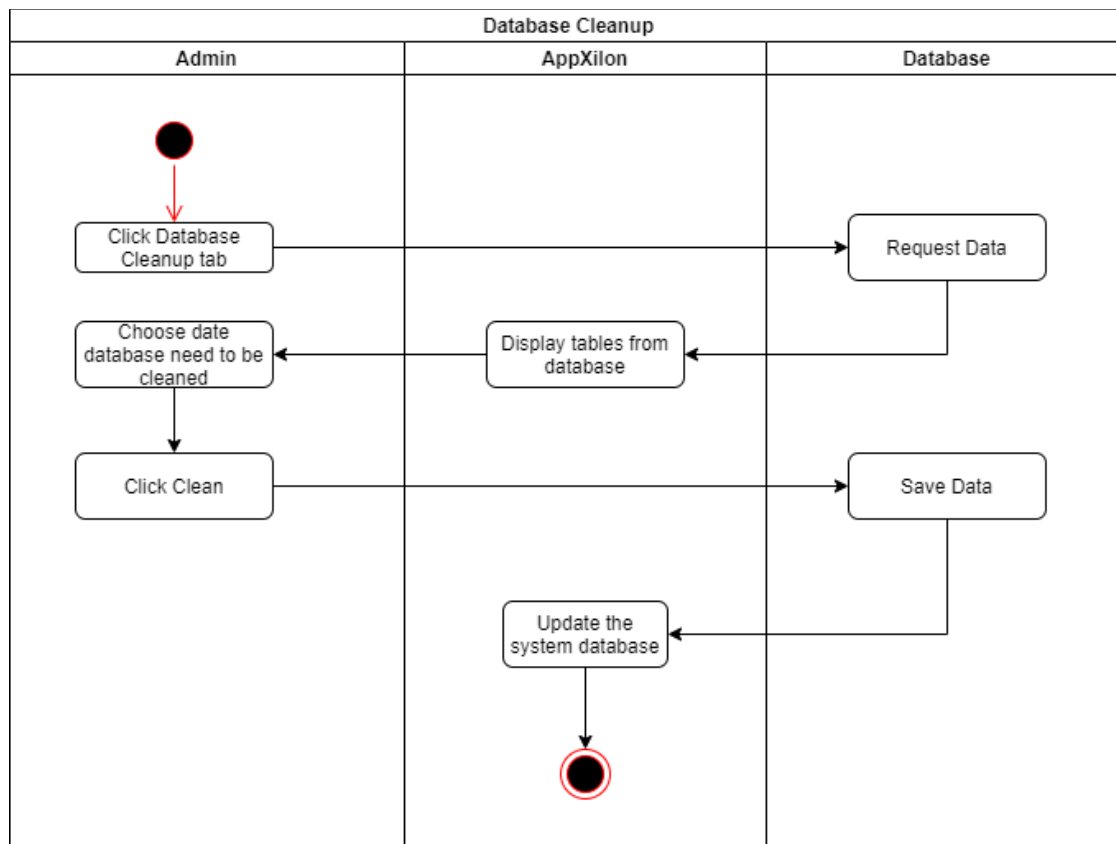


Figure 45 Activity Diagram: Database Cleanup

3.3.35 UC_35 View Logs

Description	The user can view other user's login logs and error logs
Actors	Administrator
Preconditions	The user must login into AppXilon
Basic Flow	<ol style="list-style-type: none"> 1. The user clicks on logs tab 2. The user chooses errors (A1: Choose Login logs) 3. The user choose date 4. AppXilon will display the errors logs based on date
Alternative Flow	A1: Choose Login logs <ol style="list-style-type: none"> 2.1 The user chooses date 2.2 AppXilon will display the login logs based on date
Post-condition	AppXilon displays logs based on date
Exception Flow	N/A

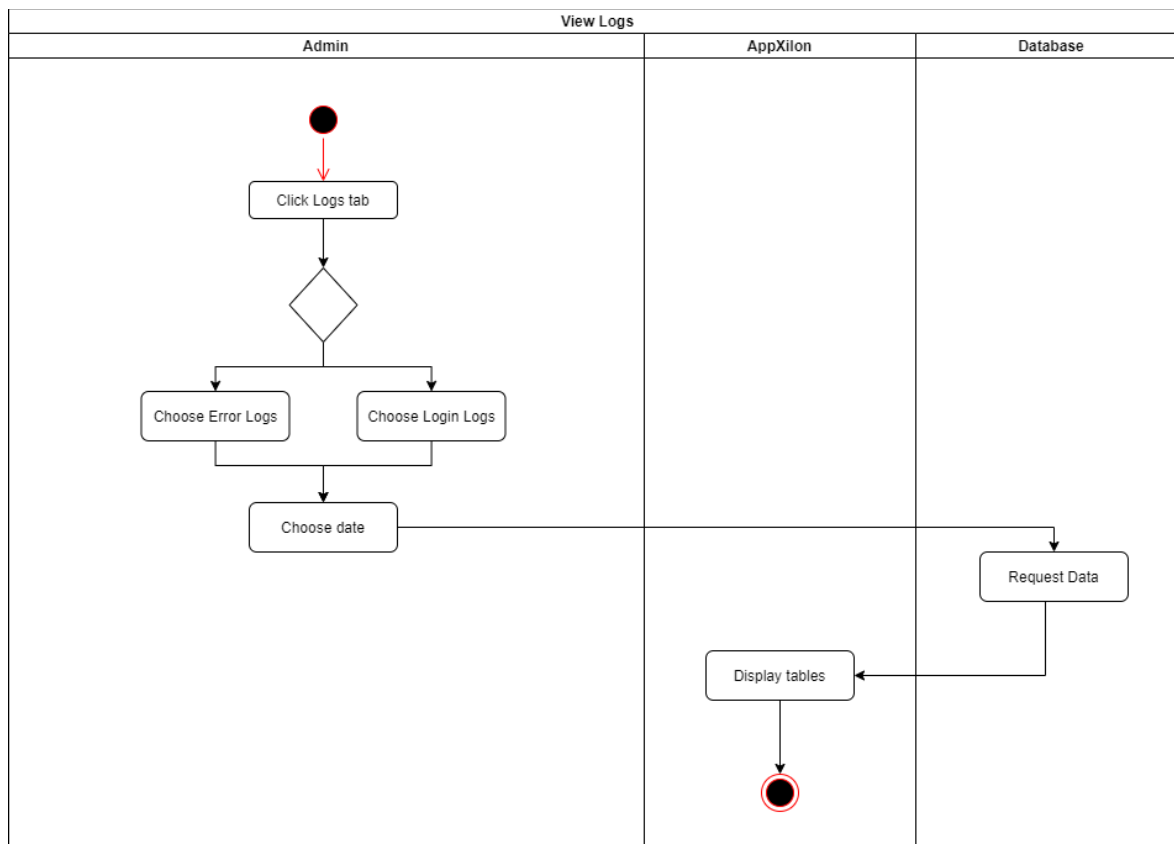


Figure 46 Activity Diagram: View Logs

3.4 Performance Requirements

The following are the key performance requirements,

- All pages should load within 2 seconds
- Search result should display within 1 seconds.
- Checkout should happen within 5 seconds after providing payment information.

3.5 Design Constraints

AppXilon application should adhere to following standards,

- Web pages should be designed using HTML 5.0 transitional standards.
- W3C Web Accessibility standards (Web Content Accessibility Guidelines (WCAG)) should be followed including keyboard navigation, alternate titles for images, etc.

3.6 Software System Attributes

The non-functional requirements are the following:

3.6.1 Reliability

The following reliability standards should be followed,

- The system should keep the database information consistently.
- The application part of the system should never fail. In the database side, failures should be minimal and there should be crash recovery systems in order not to lose information in a potential database failure.
- The system should display informative messages when its component does not work properly.
- The system should provide reliable and relevant search results 100% of times. The checkout operation should end reliably within 5 seconds.

3.6.2 Availability

The following availability features should be supported,

- The system should be available for 7 days and 24 hours.
- All software upgrades, patches and fixes should be done without shutting down the application.
- There should be disaster recovery environment to handle natural disasters.
- In the application side, the system should be tested against probable failures before publishing the first version or updated versions of application. The published version should be error free.
- In database side, in case of a failure, the system should recover any information for user and system.

3.6.3 Security

The following security standards should be followed,

- Login operation should be performed using transport layer security (HTTPS).
- All user id and password information should be encrypted using one-way hash algorithms in the database.
- Registration process should use CAPTCHA to prevent machine or robot brute force attacks.

- All input fields should be validated for SQL injection scenarios and HTML reserved words scenarios. Input should be sanitized before sending them to the upstream systems.
- There should be well-defined password policy covering password change frequency, invalid attempts allowed, etc.
- The system must not request unnecessary permissions from the user in order to prevent unwanted attacks.
- The stored data in the web browser and sent data via internet should be encrypted. Sent and received data should be transferred via HTTPS connection. And also authenticated and encrypted socket-level communication should be implemented.

3.6.4 Maintainability

The following maintainability features should be supported,

- All code artifacts should have proper documentation.
- All code components should be thoroughly tested, and the test coverage should be more than 95%.
- A Git software should be used in development phase in order to reduce complexity, make the system traceable and recover the code from an unwanted crash while more than one developer is dealing with the code.
- The design elements should be documented well.
- Since programming language is object-oriented, program tasks are independent of each other and therefore easier to maintain.
- All parts of the code should have a detailed comment to make the code easier to understand and read.

3.7 Business Rules

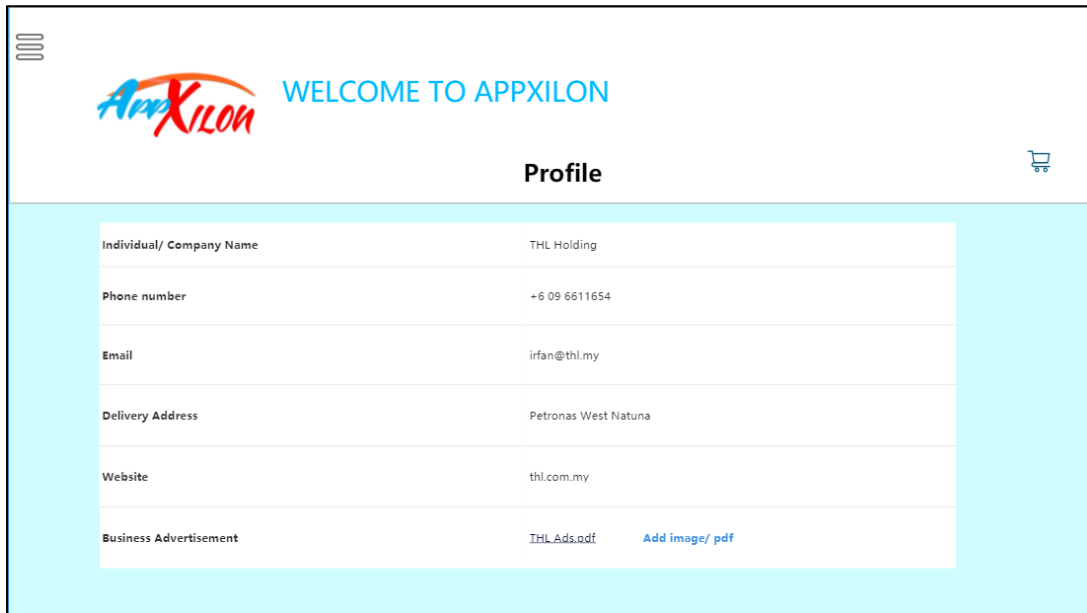
In AppXilon

- A customer can make ordering for many times in a day
- An order must be only in single payment transaction

4 Appendix A: Graphical User Interfaces (GUI)

This section shows several critical GUIs that are expected to be developed for AppXilon system.

1. UC_03 Manage Profile



Individual/ Company Name	THL Holding
Phone number	+6 09 6611654
Email	irfan@thl.my
Delivery Address	Petronas West Natuna
Website	thl.com.my
Business Advertisement	THL Ads.pdf Add image/ pdf

Figure GUI 1: Manage Profile

2. UC_05 View Catalogue

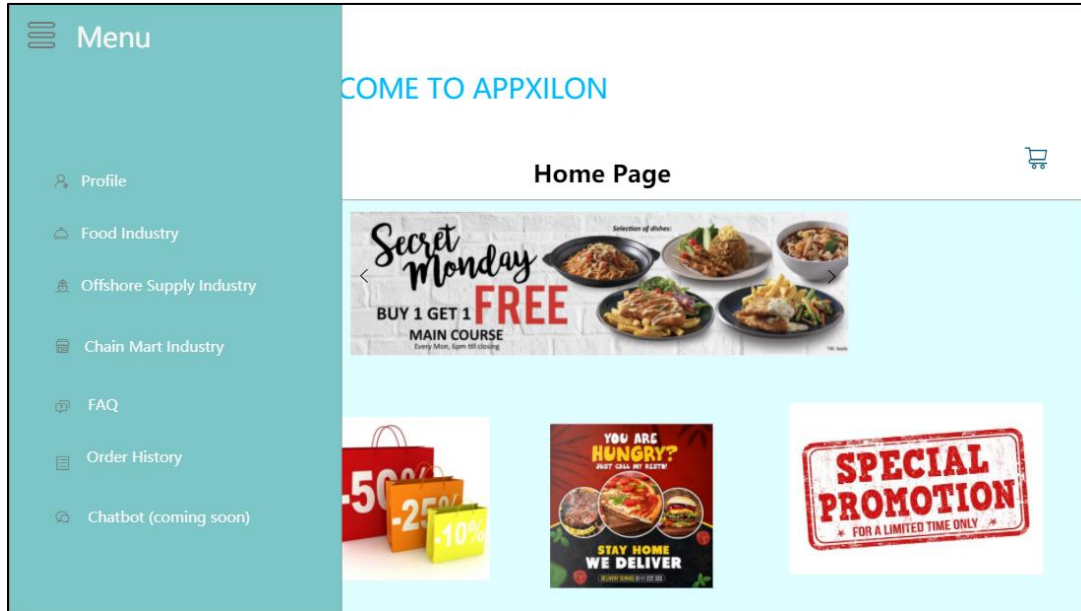


Figure GUI 2: View Catalogue 1

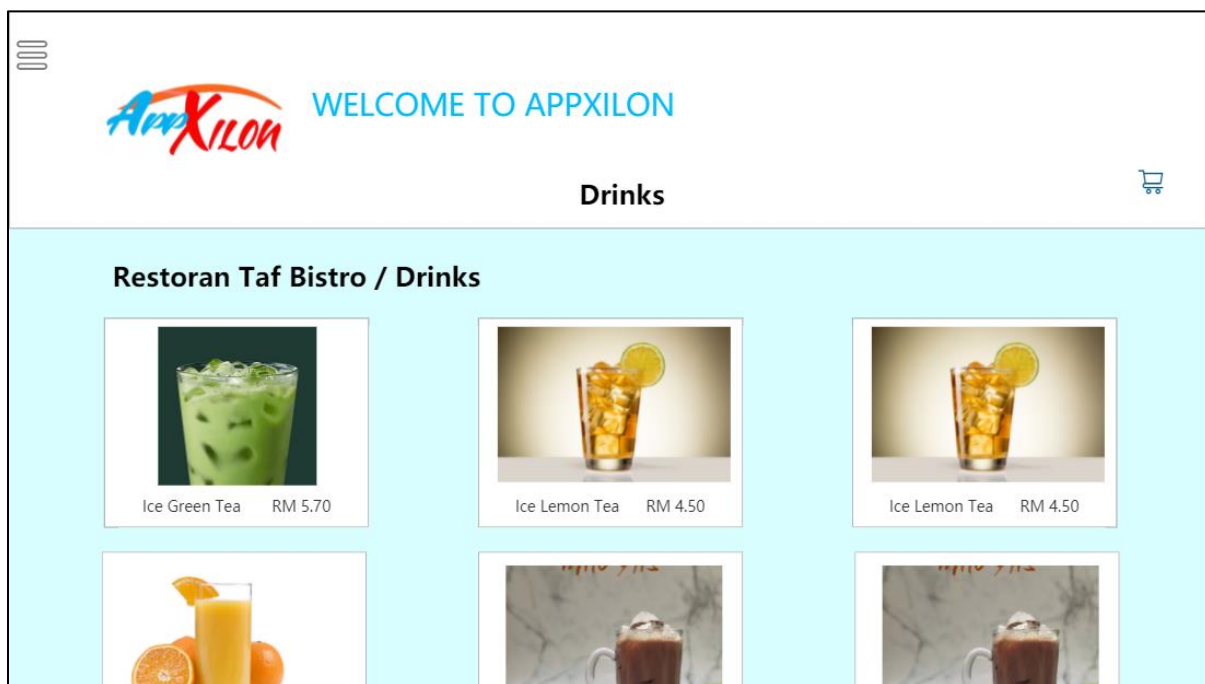


Figure GUI 3: View Catalogue 2

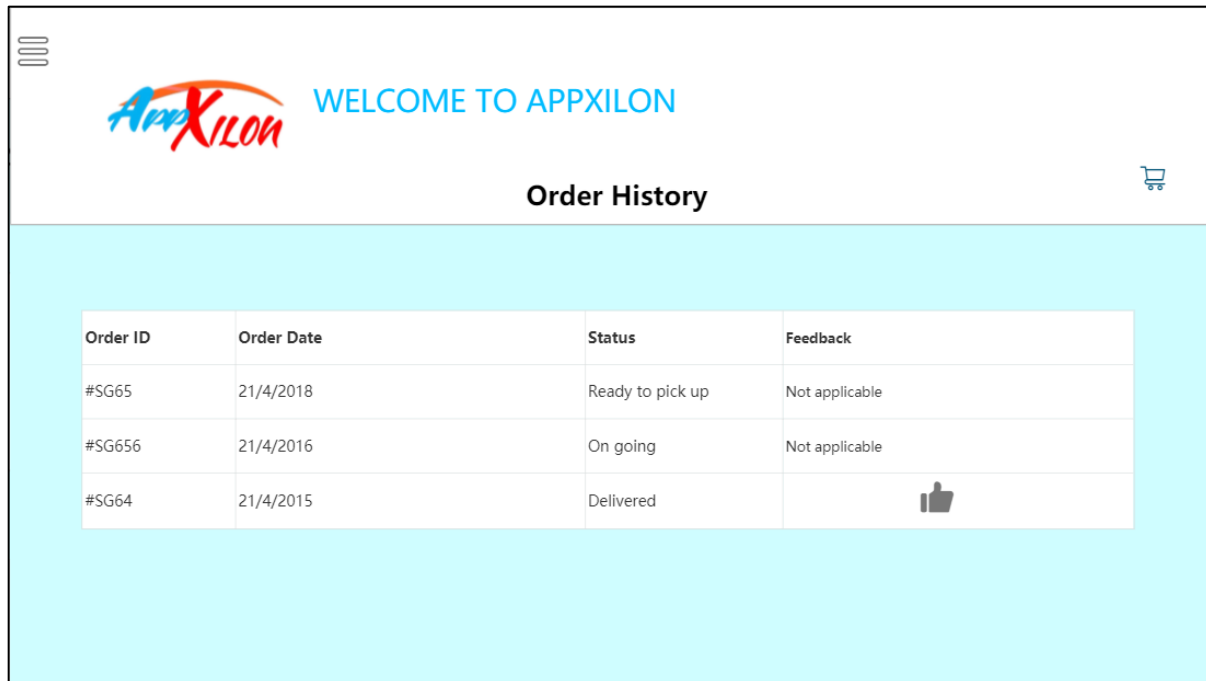


Figure GUI 4: View Catalogue 3

3. UC_07 Make Table Booking

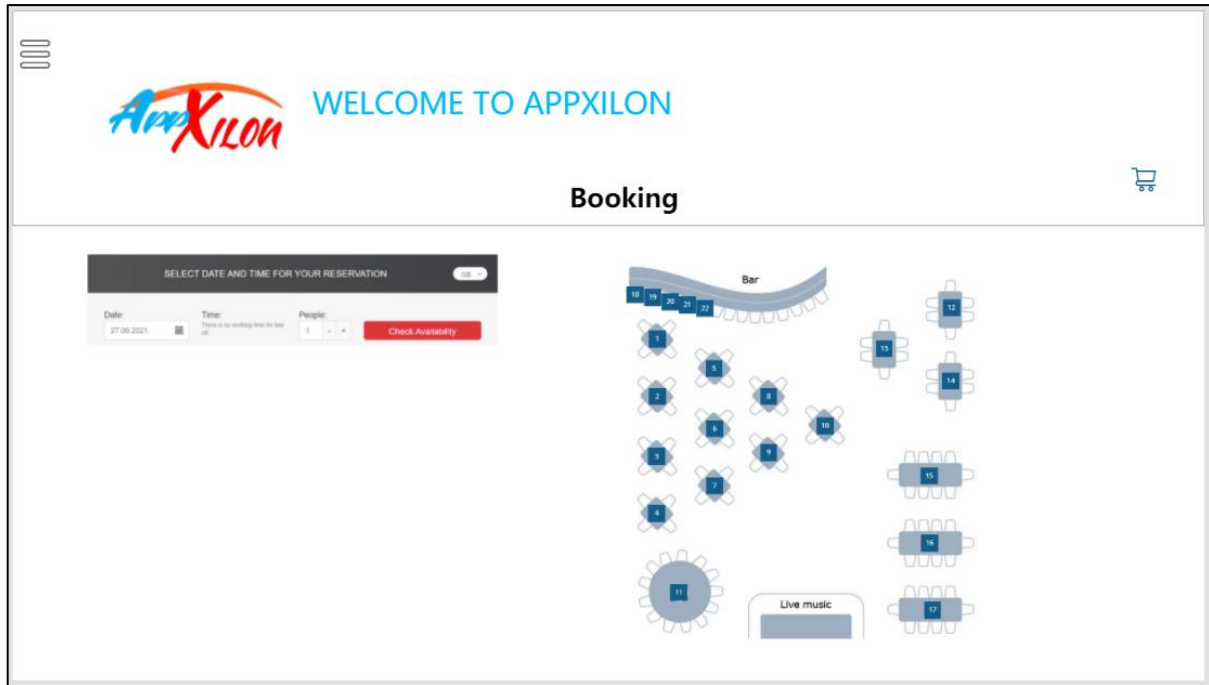


Figure GUI 5: Make Table Booking

4. UC_09 Update Cart

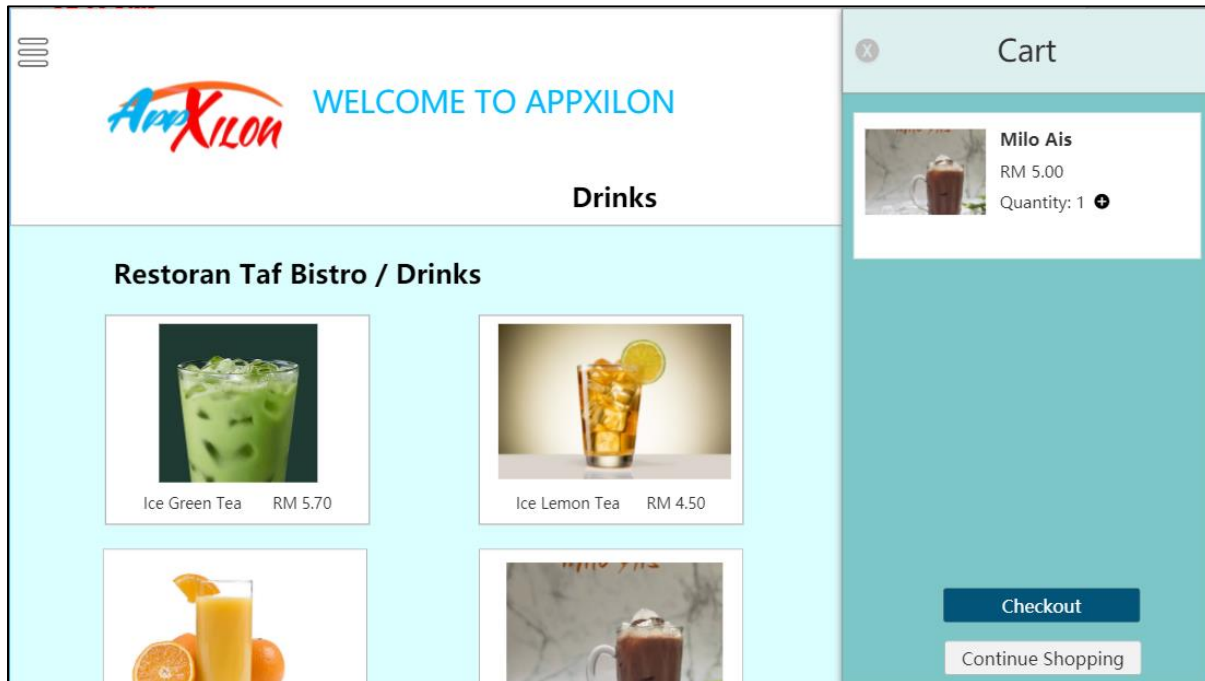


Figure GUI 6: Update Cart

5. UC_10 Checkout

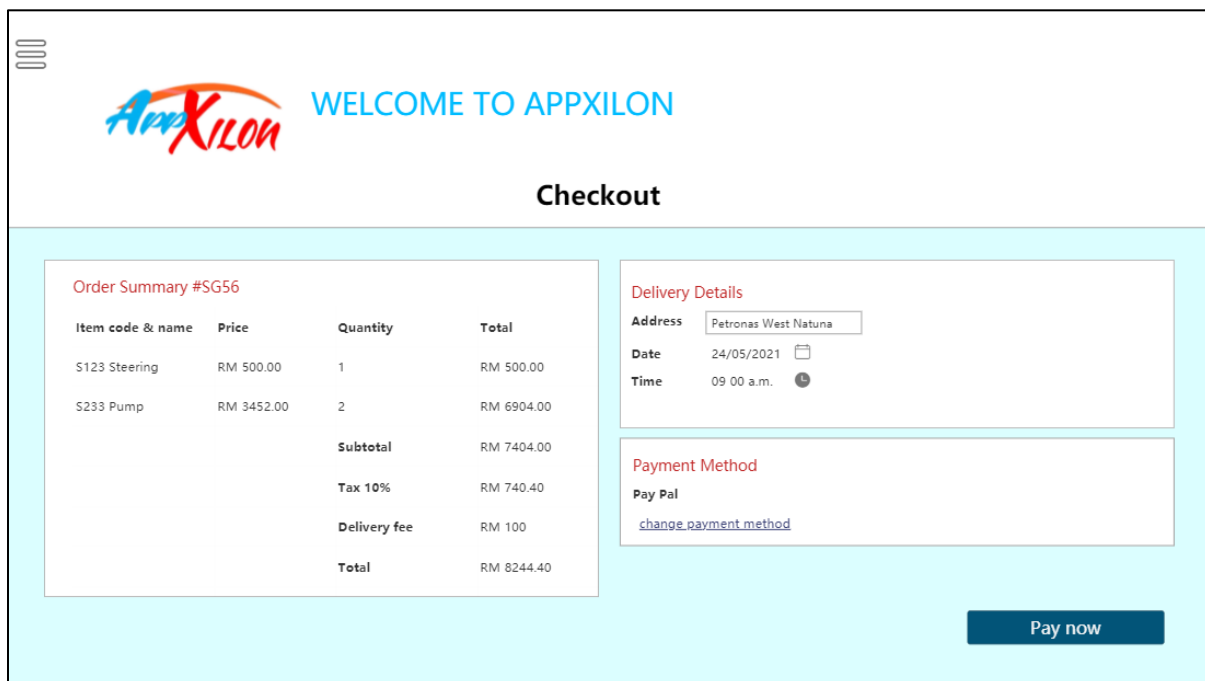


Figure GUI 7: Checkout

6. UC_14 Manage Catalogue

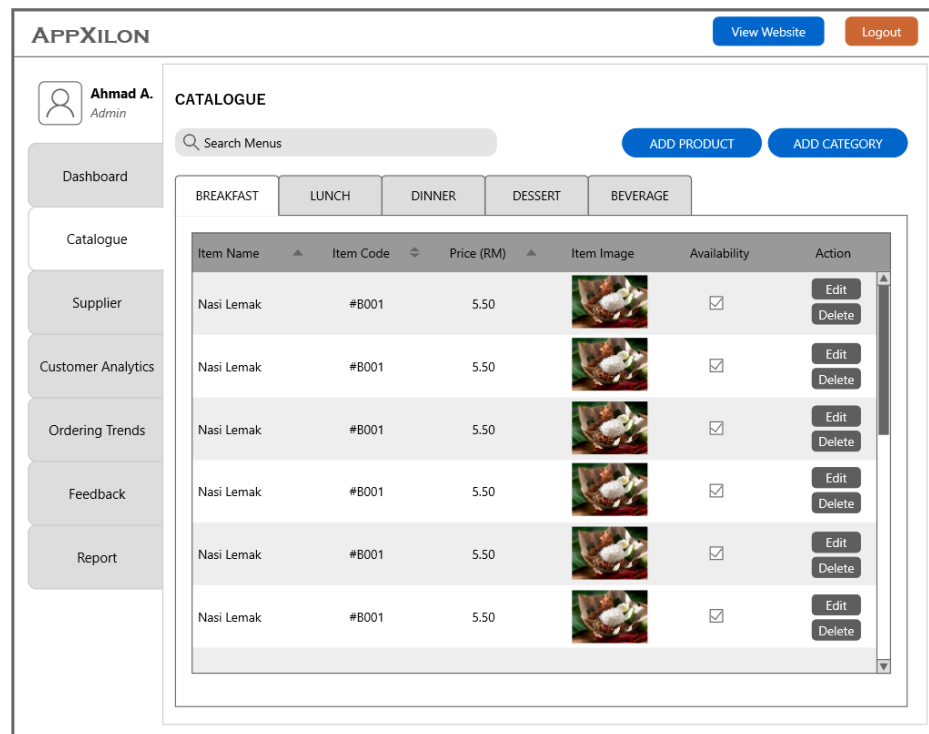


Figure GUI 8: Manage Catalogue

7. UC_17 Manage Promotion

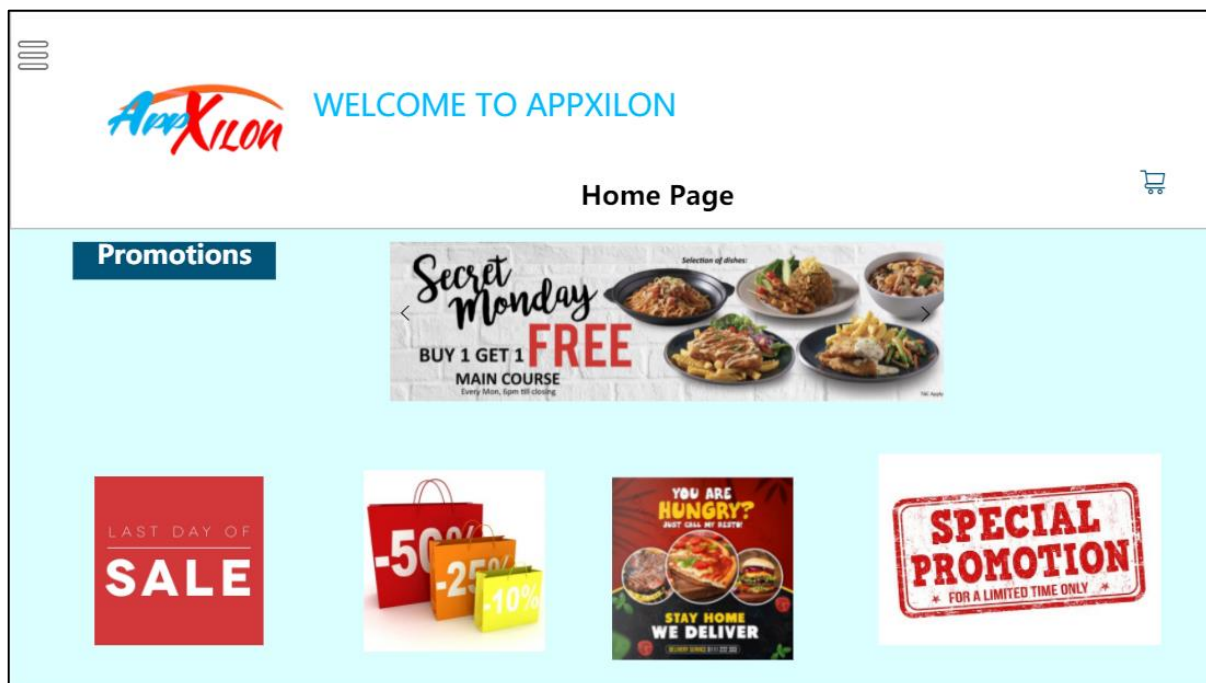


Figure GUI 9: Manage Promotion

8. UC_20 Produce Report

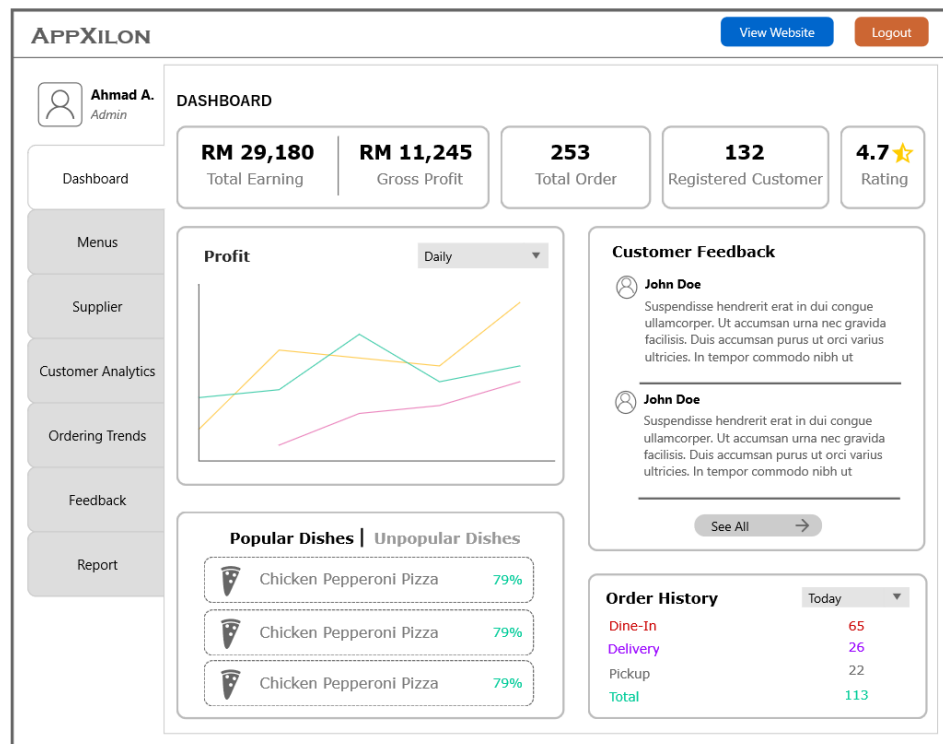


Figure GUI 10: Produce Report 1

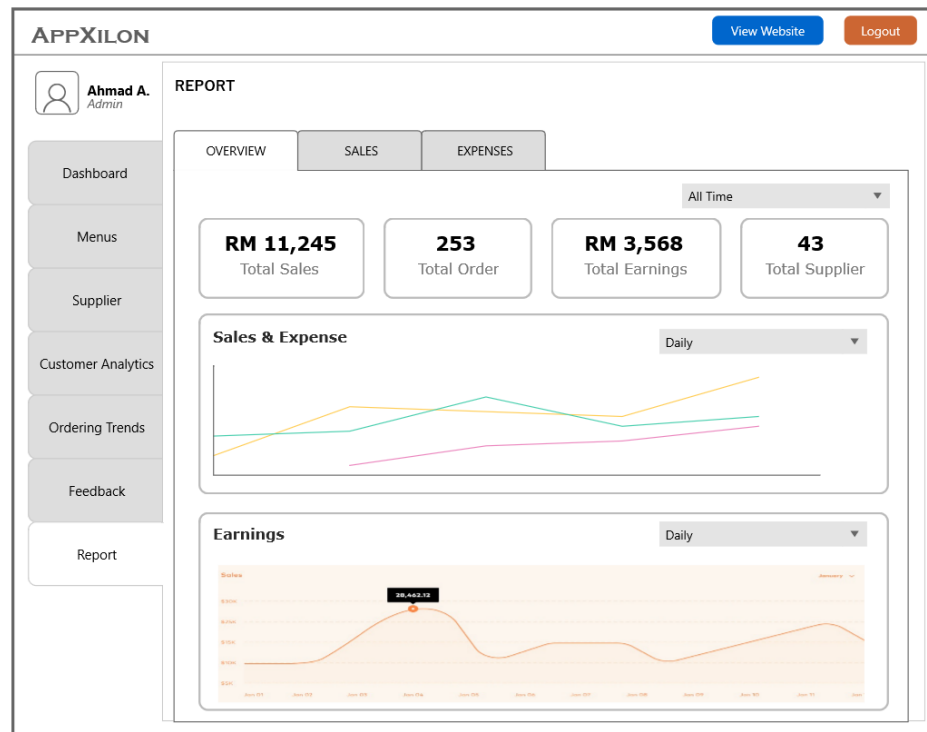


Figure GUI 11: Produce Report 2

9. UC_21 View Analytics

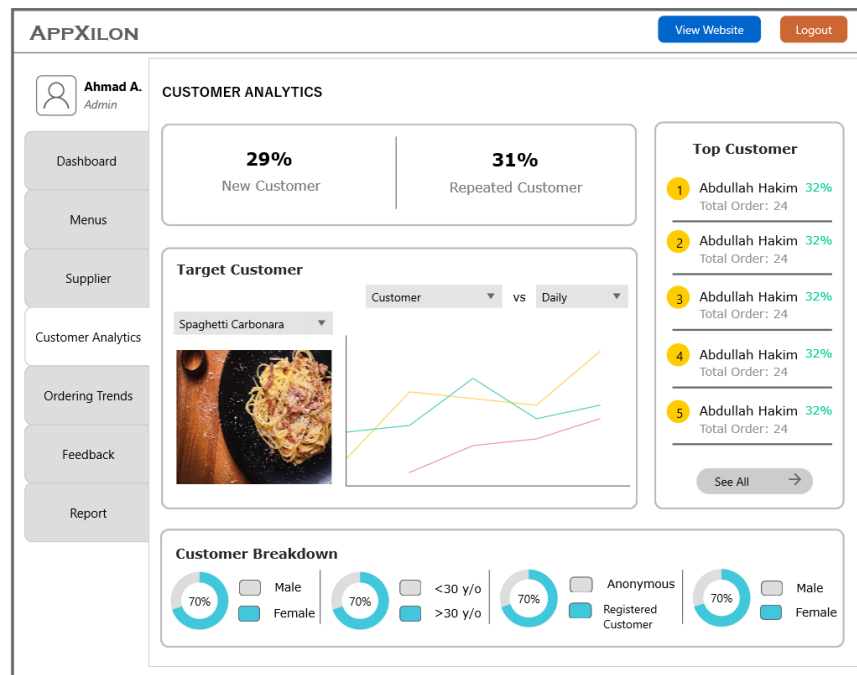


Figure GUI 12: View Analytics

10. UC_22 Track Customer Sentiment

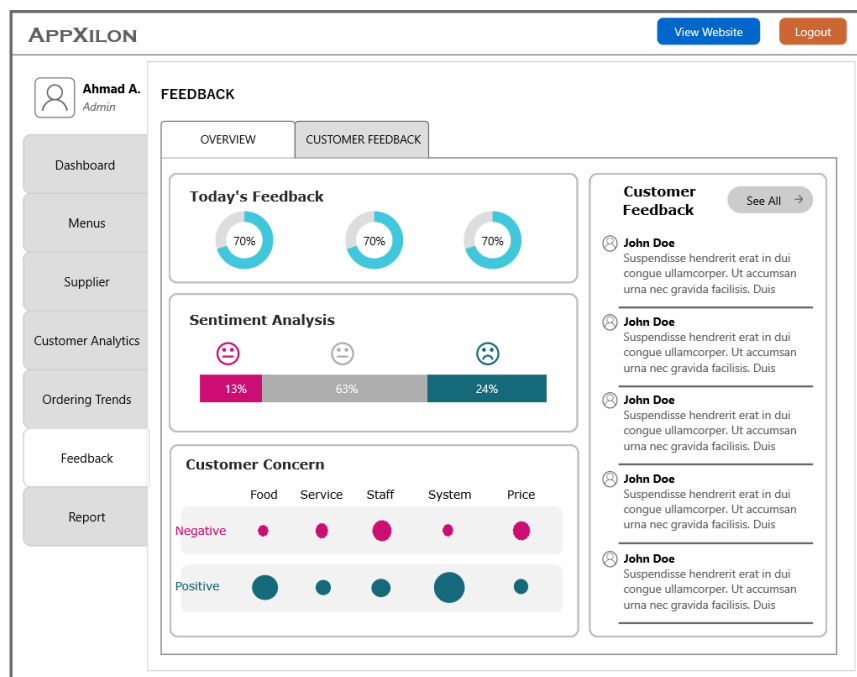


Figure GUI 13: Track Customer Sentiment

11. UC_27 Manage Business Hour

APPXILON [View Website](#) [Logout](#)

Ahmad A. Admin

BUSINESS HOUR

Default Custom

Working Time

Day of Week	Start Time	End Time	Is Day off
Monday	09 00	23 00	<input type="checkbox"/> NO
Tuesday	09 00	23 00	<input type="checkbox"/> NO
Wednesday	09 00	23 00	<input type="checkbox"/> NO
Thursday	09 00	23 00	<input type="checkbox"/> NO
Friday	09 00	23 00	<input type="checkbox"/> NO
Saturday	09 00	23 00	<input type="checkbox"/> NO
Sunday	00 00	00 00	<input checked="" type="checkbox"/> YES

[Save](#)

Figure GUI 14: Manage Business Hour

12. UC_32 Manage Backup

The screenshot displays the 'APPXILON' web application interface. At the top right, there are 'View Website' and 'Logout' buttons. On the left, a sidebar shows the user 'Ahmad A. Admin' and navigation links for 'Dashboard', 'Restaurant', 'Settings', and 'System Options'. The main content area is titled 'SYSTEM OPTIONS' and contains four tabs: 'GENERAL', 'BACKUP', 'DATABASE CLEANUP', and 'PAYMENT'. The 'BACKUP' tab is active, showing a table with columns 'Back-up made', 'Data type', 'File size', and 'File name'. The table is empty, with a message 'No records found'. Below the table are pagination controls: 'Choose Action -', '< Prev', '0', 'Next >', and 'Show 10 of 0 total'. There are two main sections: 'Enable auto-backup' and 'Manually make a back-up'. The 'Enable auto-backup' section has a toggle switch set to 'YES' and a yellow warning box stating: 'Please, make sure CRON jobs are configured. Please, go to System Options - Cron jobs and follow the instructions.' The 'Manually make a back-up' section has two toggle switches, 'Backup database:' and 'Backup files:', both set to 'YES', and a 'Make back-up' button.

Figure GUI 15: Manage Backup

5 Appendix B: Index

Figure 1 Customer Use Case Diagram	16
Figure 2 Manager Use Case Diagram	17
Figure 3 Admin Use Case Diagram	18
Figure 4 Activity Diagram: Register	20
Figure 5 Activity Diagram: Login	22
Figure 6 Activity Diagram: Manage Profile	23
Figure 7 Activity Diagram: Chat with Chatbot	24
Figure 8 Activity Diagram: View Catalogue	26
Figure 9 Activity Diagram: View Personalised Recommendations	27
Figure 10 Activity Diagram: Make Table Booking	28
Figure 11 Activity Diagram: Update Cart	30
Figure 12 Activity Diagram: Checkout	32
Figure 13 Activity Diagram: Give Review	33
Figure 14 Activity Diagram: View FAQ	34
Figure 15 Activity Diagram: View Booking Details	35
Figure 16 Activity Diagram: Manage Catalogue	38
Figure 17 Activity Diagram: Add product	39
Figure 18 Activity Diagram: Remove Product	39
Figure 19 Activity Diagram: Update Product	40
Figure 20 Activity Diagram: Update Product Availability	40
Figure 21 Activity Diagram: Add Categories	41
Figure 22 Activity Diagram: Remove Categories	41
Figure 23 Activity Diagram: Update Categories	42
Figure 24 Activity Diagram: Manage Table Criteria	43
Figure 25 Activity Diagram: Manage Order Progress	44
Figure 26 Activity Diagram: Manage Promotion	45
Figure 27 Activity Diagram: Manage Preview	46
Figure 28 Activity Diagram: Manage Preview	47
Figure 29 Activity Diagram: Generate Report	49
Figure 30 Activity Diagram: Track Customer Sentiment	51
Figure 31 Activity Diagram: View Customer Analytics	52
Figure 32 Activity Diagram: View Sales Analytics	53

Figure 33 Activity Diagram: View Product Analytics	54
Figure 34 Activity Diagram: Payment	55
Figure 35 Activity Diagram: Business Hours Management.....	56
Figure 36 Activity Diagram: Terms and Conditions.....	57
Figure 37 Activity Diagram: Reminder Reservations	58
Figure 38 Activity Diagram: Reservation Form	59
Figure 39 Activity Diagram: Captcha and Spam.....	60
Figure 40 Activity Diagram: Create Backup.....	61
Figure 41 Activity Diagram: Restore Backup	62
Figure 42 Activity Diagram: Create User	64
Figure 43 Activity Diagram: Update User	64
Figure 44 Activity Diagram: Delete User.....	65
Figure 45 Activity Diagram: Database Cleanup.....	66
Figure 46 Activity Diagram: View Logs.....	67