UTM
UNIVERSITI TEKNOLOGI MALAYSIA

Software Requirements Specification

Restaurant Dine In Online Ordering and Management System

Version 1.2

10 June 2023

Faculty of Computing

Revision Page

a. Overview

This is the 3rd version of the SRS. Compare to the previous version, some of the use case flow is modify to match with the latest requirement.

b. Target Audience

Stakeholder of the application, developer of the system.

c. Project Team Members

Ricky A/L Karunakaran

d. Version Control History

Version	Primary Author(s)	Description of Version	Date Completed
1.0	RICKY	SRS Creation	9 June 2022
1.1	Ricky	New Actor admin and its use cases is added.	12 July 2022
1.2	Ricky	Requirement and Use Case updated	10 June 2023

Table of Contents

1. Introduction

1.1 Purpose

This Software Requirement Specification (SRS) describes the result of requirement engineering of the project. It documents the functional requirements (the system features), non-functional requirements (performance attribute) and the system constraint into different perspectives to deal with the different concern of the audience. The intended audience of this document include the stakeholder and the developers.

1.2 Scope

The software product that will be produced is called foodverse. The system is a generic product which design for restaurant to manage their service. It is also a centralized application which allows the user to access any restaurant that registered in the application. For the restaurant, the system will allow them to manage menu which will be displayed to the customer or visitor. Besides, the restaurant will get notification once customer check in to the restaurant. They can either accept the check in or reject the check in. Once the check in is accepted, the customer will be bind to the table they checked in. The order made by the customer will be handler under that table. Moreover, the restaurant could view the feedback submitted by the customer and reply the feedback.

For the restaurant, the system will allow them to view any registered restaurant's menu from the application. They only can access to the service once they check in to the restaurant by scanning the QR code. Once they scan the QR code and their check in is accepted by the restaurant, the services will be unlocked. The customer can add order to the cart. If multiple customer scanning a same QR code, their order will be displayed into the same cart. The customer will then can submit the order in cart to the restaurant and the restaurant can start to prepared it. The system also provide feature for the customer to call waiter when they need. Besides, the user will be checked out automatically from the system once they clear the bill either using cash or using credit/debit card. The customer will be allowed to send feedback to the restaurant they visited within three days. They will also allow to feed the reply of the feedback if the restaurant add reply or update response to the feedback.

1.3 Definitions, Acronyms and Abbreviation

SRS: Software Requirement Specification
Check In: Binding the customer to a table of the restaurant they visit.
Check Out: Clear the customer bill and break the binding of the customer from the restaurant

1.4 References

IEEE 29148-2011 Systems and software engineering — Life cycle processes —Requirements engineering

1.5 Overview

This SRS contains three sections which is Introduction, Overall Description and Specific Requirements. The Overall Description describe the product interfaces, product functions, user characteristic, system constraints and the assumption and dependencies of the system. The specific requirements section describes the system external interface requirement, system features and it performance requirements.

2. Overall Description

The requirement of this system come from 3 sources: brainstorming, existing system review and interviews. Brainstorming is carried out to find the requirement of the system based on the developer knowledge and experience. The brainstorming produce requirement which are the developer expectation. Besides, the requirement also derived from existing system review. The requirement derived from the review is the requirement that already practiced in the current environment. Last but not lease, the requirement is gathered from stakeholder and potential user interview. In the interview, some requirement generated in brainstorming and existing review is validated and more requirement is explored. The Figure 2.1 below show the overall requirement from use case view. The details of the use case diagram will be discussed in later chapter.

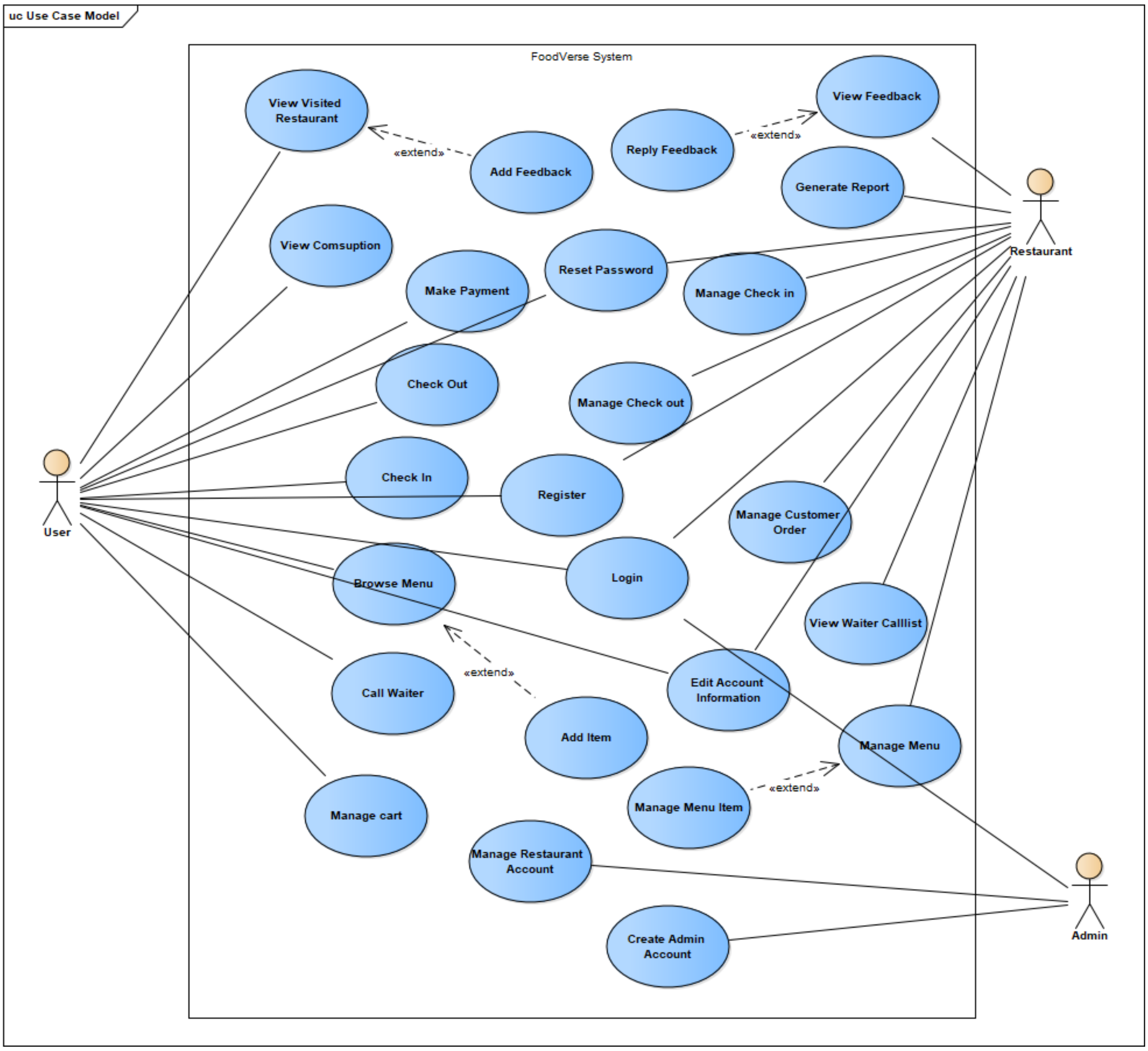


Figure 2.1: Use Case Diagram of Foodverse

2.1 Product Perspective

In this section the interface of the system is discussed. The discussed interface include system interface, user interface, hardware interface, software interface and communication interface.

2.1.1 System Interfaces

Mail Server

This interface is required in the function of reset password.

2.1.2 User Interfaces

The system user interfaces should be flexible, which mean the layout will change depending on the user device size. For restaurant’s application, the layout should be design for wide screen whereas the user’s application layout should be design to fix high screen.

The system user interfaces should be clear and simple. The system feature should be displayed in multiple language. Besides, the system shall show the error happening and show possible solution and cause of problem to the user. The other user interfaces requirement and limitation is listed below:

I. The system shall display menu only to the user who do not check in to restaurant in the restaurant window.

II. The system shall display all the functional button and menu to the user who has check in to the restaurant in the restaurant window.

III. The system shall provide a button which can bring the user to the home window at every window.

IV. The system shall provide a description on the button when user hover or long pressed on the button.

2.1.3 Hardware Interfaces

The user’s application is run on the mobile phone only and no other hardware is required. The restaurant application is run on the desktop and printer is required.

2.1.4 Software Interfaces

Software Interfaces	Specifications
Database Management System	Name: Mysql Version Number: 5.6 Source: https://dev.mysql.com/

2.1.5 Communication Interfaces

The system uses Wireless local network, mobile data to communicate with the sever.

2.1.6 Memory

The user's application should take less than 100MB memory whereas the restaurant's application should take less than 300MB.

2.1.7 Operations

No special operations required in this system.

2.1.8 Site Adaptation Requirements

No site adaption required in this system.

2.2 Product Functions

2.2.1 UC001 Register

The system shall allow the restaurant and user to register an account with email to access the service of the application.

2.2.2 UC002 Login

The system shall allow the restaurant and user to login to the account they registered by email to access the service of the application

2.2.3 UC003 Edit Account Information

The system shall allow the restaurant and user to edit their account information except email and password.

2.2.4 UC004 Reset Password

The system shall allow the restaurant and user to reset the password of their account.

2.2.5 UC005 Check In

The system shall allow the user to check in to the restaurant using camera to scan QR code of the restaurant before accessing the service of the restaurant.

2.2.6 UC006 Check Out

The system shall allow the user to check out from the restaurant after they pay the bill.

2.2.7 UC007 Make Payment

The system shall allow the user to make payment using debit/credit card or using cash.

2.2.8 UC008 Manage Check In

The system shall allow the restaurant to manage the check in request of the user either approve or reject the request.

2.2.9 UC009 Manage Check Out

The system shall allow the restaurant to check out user that has solved the payment.

2.2.10 UC010 Browse Menu

The system shall allow the user to browse the menu of a restaurant.

2.2.11 UC011 Add Item

The system shall allow the user to add food or item to the cart after check in to the restaurant.

2.2.12 UC012 Manage Cart

The system shall allow the user to manage the item in the cart such as change quantity and remove item from cart.

2.2.13 UC013 Call Waiter

The system shall allow the user to call waiter with a small message which indicate the reason of calling.

2.2.14 UC014 View Waiter Call List

The system shall allow the restaurant to view the waiter call list.

2.2.15 UC015 Manage Menu

The system shall allow the restaurant to manage the menu such as create new menu and edit menu information.

2.2.16 UC016 Manage Menu Item

The system shall allow the restaurant to manage menu item such as create new menu item, edit menu item information and remove menu item.

2.2.17 UC017 Manage Order

The system shall allow the restaurant to manage order such as add order, change order quantity and remove order.

2.2.18 UC018 View Visited Restaurant

The system shall allow the user to view the restaurant they visited.

2.2.19 UC019 Add Feedback

The system shall allow the user to add feedback to the restaurant they visited.

2.2.20 UC020 View Consumption

The system shall allow the user to view the consumption or the total amount of bill in the system.

2.2.21 UC021 Generate Report

The system shall allow the restaurant to generate sale report based on the user information and bill information.

2.2.22 UC022 View Feedback

The system shall allow the restaurant to view the feedback added by the user.

2.2.23 UC023 Reply Feedback

The system shall allow the restaurant to reply the feedback of a user.

2.2.24 UC024 Create Admin Account

The system shall allow the admin to create another admin account.

2.2.25 UC025 Manage Restaurant Account

The system shall allow the admin to view or delete the restaurant account registered in this system.

2.3 User Characteristics

The user of the system should be any residents of Malaysia who are able to read and understand English and Malay or Mandarin.

2.4 Constraints

The constraints of the system include the feasibility of connecting the bank gateway interface. The system does not connect to a real bank gateway but with simulated interface.

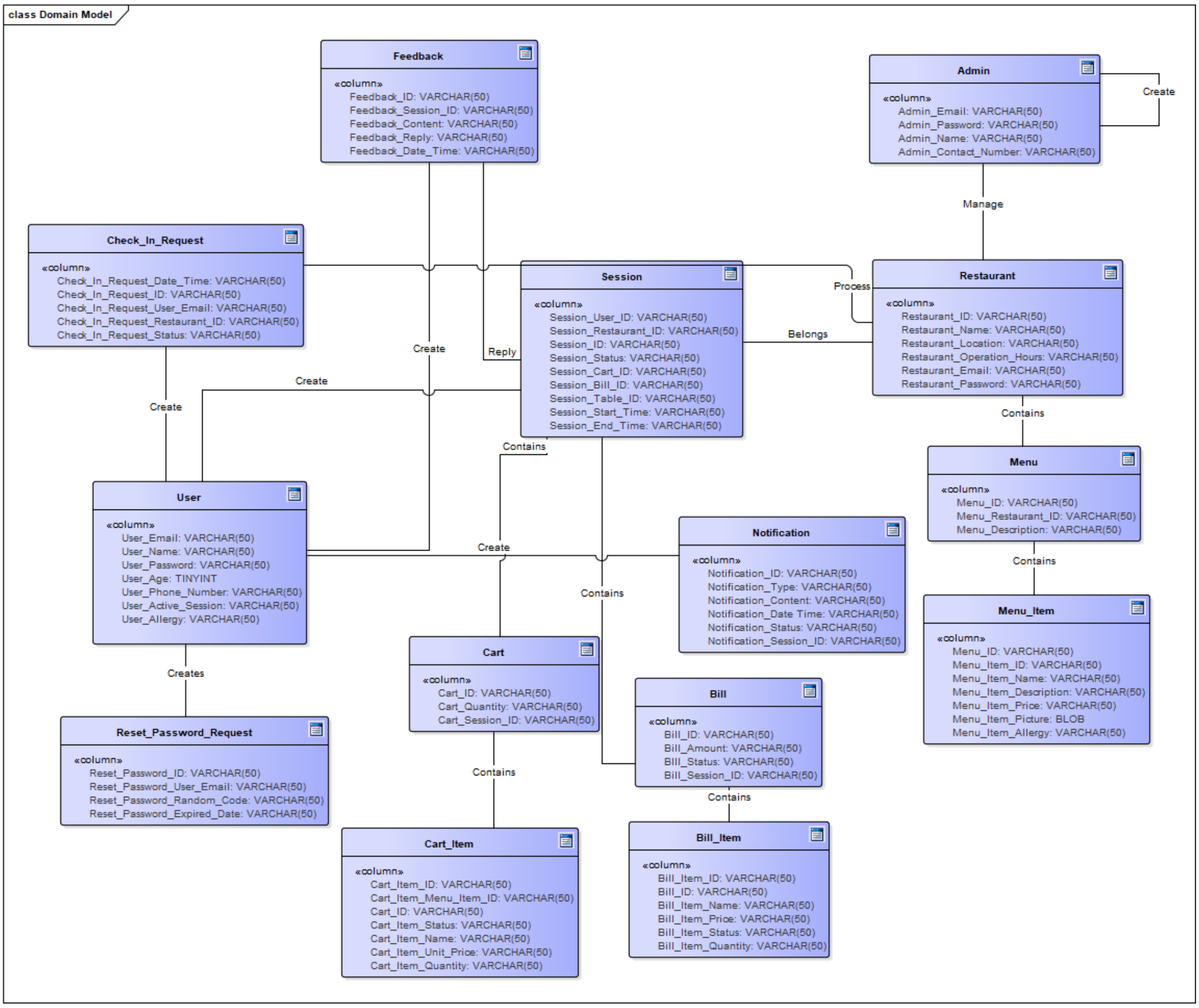
2.5 Assumption and Dependencies

The system is assumed to be able to run on android, IOS environment and Windows, Mac and Linux operating system.

2.6 Apportioning of Requirements

The requirement might be delayed will be the payment feature which might be if simulated bank gateway cannot function or arise issues.

3. Specific Requirements

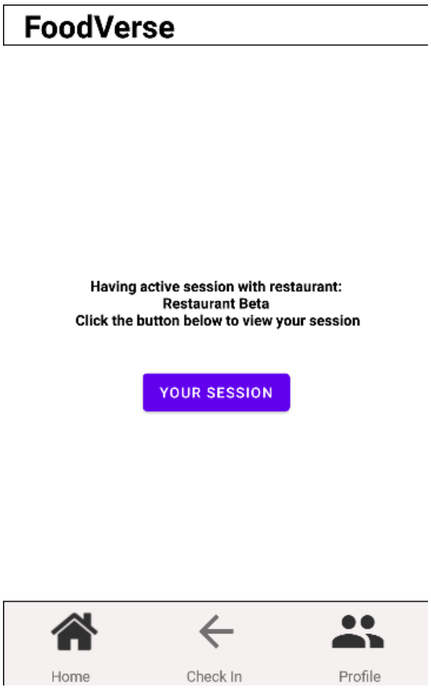

class Domain Model
Feedback
«column»
Feedback_ID: VARCHAR(50)
Feedback_Session_ID: VARCHAR(50)
Feedback_Content: VARCHAR(50)
Feedback_Reply: VARCHAR(50)
Feedback_Date_Time: VARCHAR(50)
Admin
«column»
Admin_Email: VARCHAR(50)
Admin_Password: VARCHAR(50)
Admin_Name: VARCHAR(50)
Admin_Contact_Number: VARCHAR(50)
Create
Manage
Check_In_Request
«column»
Check_In_Request_Date_Time: VARCHAR(50)
Check_In_Request_ID: VARCHAR(50)
Check_In_Request_User_Email: VARCHAR(50)
Check_In_Request_Restaurant_ID: VARCHAR(50)
Check_In_Request_Status: VARCHAR(50)
Session
«column»
Session_User_ID: VARCHAR(50)
Session_Restaurant_ID: VARCHAR(50)
Session_ID: VARCHAR(50)
Session_Status: VARCHAR(50)
Session_Cart_ID: VARCHAR(50)
Session_Bill_ID: VARCHAR(50)
Session_Table_ID: VARCHAR(50)
Session_Start_Time: VARCHAR(50)
Session_End_Time: VARCHAR(50)
Process
Belongs
Restaurant
«column»
Restaurant_ID: VARCHAR(50)
Restaurant_Name: VARCHAR(50)
Restaurant_Location: VARCHAR(50)
Restaurant_Operation_Hours: VARCHAR(50)
Restaurant_Email: VARCHAR(50)
Restaurant_Password: VARCHAR(50)
Create
Reply
Create
Create
Contains
Contains
Menu
«column»
Menu_ID: VARCHAR(50)
Menu_Restaurant_ID: VARCHAR(50)
Menu_Description: VARCHAR(50)
User
«column»
User_Email: VARCHAR(50)
User_Name: VARCHAR(50)
User_Password: VARCHAR(50)
User_Age: TINYINT
User_Phone_Number: VARCHAR(50)
User_Active_Session: VARCHAR(50)
User_Allergy: VARCHAR(50)
Create
Notification
«column»
Notification_ID: VARCHAR(50)
Notification_Type: VARCHAR(50)
Notification_Content: VARCHAR(50)
Notification_Date Time: VARCHAR(50)
Notification_Status: VARCHAR(50)
Notification_Session_ID: VARCHAR(50)
Contains
Contains
Menu_Item
«column»
Menu_ID: VARCHAR(50)
Menu_Item_ID: VARCHAR(50)
Menu_Item_Name: VARCHAR(50)
Menu_Item_Description: VARCHAR(50)
Menu_Item_Price: VARCHAR(50)
Menu_Item_Picture: BLOB
Menu_Item_Allergy: VARCHAR(50)
Creates
Cart
«column»
Cart_ID: VARCHAR(50)
Cart_Quantity: VARCHAR(50)
Cart_Session_ID: VARCHAR(50)
Bill
«column»
Bill_ID: VARCHAR(50)
Bill_Amount: VARCHAR(50)
Bill_Status: VARCHAR(50)
Bill_Session_ID: VARCHAR(50)
Reset_Password_Request
«column»
Reset_Password_ID: VARCHAR(50)
Reset_Password_User_Email: VARCHAR(50)
Reset_Password_Random_Code: VARCHAR(50)
Reset_Password_Expired_Date: VARCHAR(50)
Contains
Contains
Cart_Item
«column»
Cart_Item_ID: VARCHAR(50)
Cart_Item_Menu_Item_ID: VARCHAR(50)
Cart_ID: VARCHAR(50)
Cart_Item_Status: VARCHAR(50)
Cart_Item_Name: VARCHAR(50)
Cart_Item_Unit_Price: VARCHAR(50)
Cart_Item_Quantity: VARCHAR(50)
Bill_Item
«column»
Bill_Item_ID: VARCHAR(50)
Bill_ID: VARCHAR(50)
Bill_Item_Name: VARCHAR(50)
Bill_Item_Price: VARCHAR(50)
Bill_Item_Status: VARCHAR(50)
Bill_Item_Quantity: VARCHAR(50)

DM 01: Domain Model of FoodVerse

Model	Description
User	User is used to record user personal data. The User_Active_Session is set to active if a user checks in to a restaurant and set to inactive if they check out. User with active status are not allowed to check out to other restaurant.
Restaurant	Restaurant is used to record the data of a restaurant.
Admin	Admin is used to record the data of a admin.
Menu	Menu is used to record the menu item added or created by the restaurant. It composited of menu item.

Menu_Item	Menu_Item is the child of the menu. One menu can contain multiple menu item.
Session	User check in create a session with the restaurant and restaurant table code they scan. A session contains bill and cart with cart item. It also includes feedback if user added it.
Cart	Cart use to record the item added to the cart in the session. When it is submitted, all of its child (Cart_Item) will be converted and added to bill as bill_item.
Cart_Item	Record the individual food that added to the cart.
Bill	Bill use to record the bill condition of the session. When the bill status is unsolved, the customer are not allowed to check out from the restaurant. Bill will be refer by restaurant as order of the session.
Bill_Item	Record the individual food that submit to the restaurant.
Feedback	Feedback is used to record the feedback submitted by the restaurant in a session.
Check_In_Request	Record the check in request of a restaurant from a user.
Reset_Password_Request	Record the reset password request of the restaurant and user account

Figure 3.1: State Machine Diagram of Session

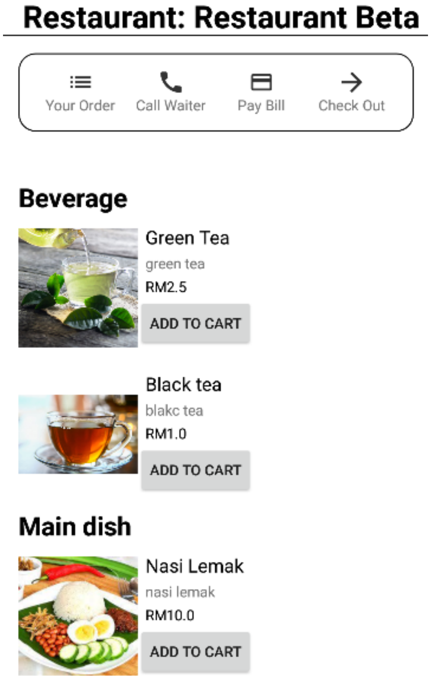


FoodVerse

Having active session with restaurant:
Restaurant Beta
Click the button below to view your session

Figure 3.4: Home Screen of user

Session Page After Check In



Restaurant: Restaurant Beta

Beverage

Green Tea
green tea
RM2.5
ADD TO CART

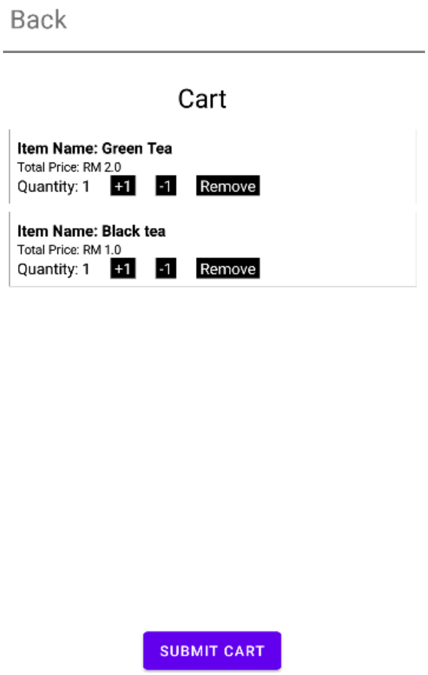
Black tea
blakc tea
RM1.0
ADD TO CART

Main dish

Nasi Lemak
nasi lemak
RM10.0
ADD TO CART

Figure 3.5: Restaurant page after check in

Cart



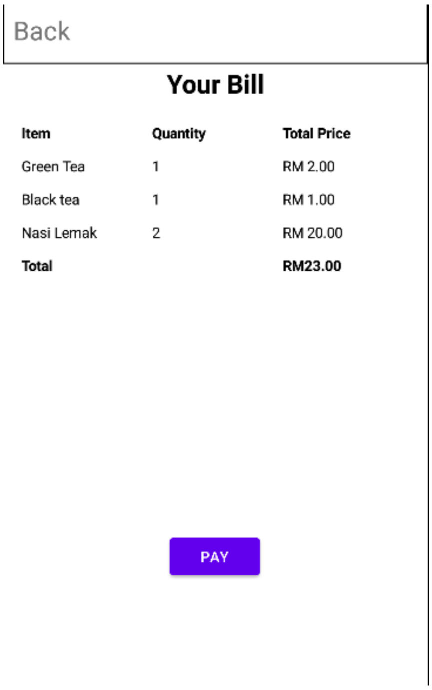
Cart

Item Name: Green Tea
Total Price: RM 2.0
Quantity: 1

Item Name: Black tea
Total Price: RM 1.0
Quantity: 1

Figure 3.6: Cart Page

Bill

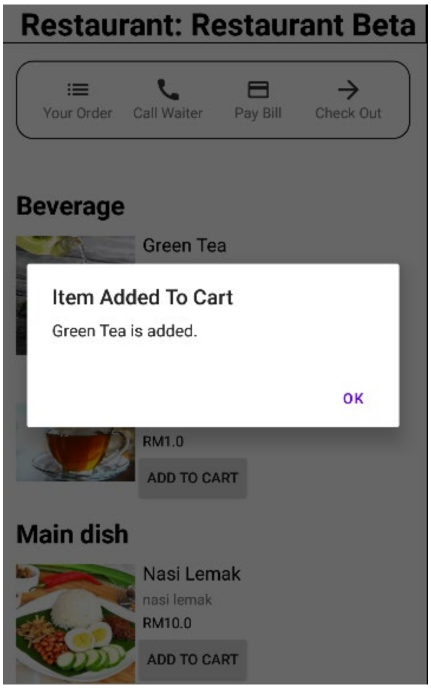


Your Bill

Item	Quantity	Total Price
Green Tea	1	RM 2.00
Black tea	1	RM 1.00
Nasi Lemak	2	RM 20.00
Total		**RM23.00**

Figure 3.7: Bill Page

Add item to cart



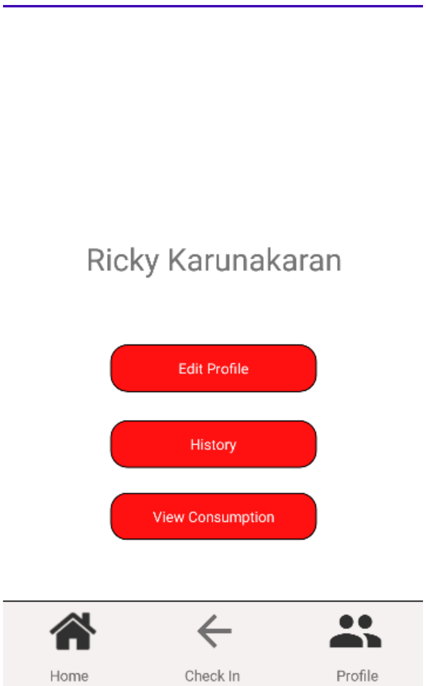
Restaurant: Restaurant Beta

Item Added To Cart

Green Tea is added.

Figure 3.8: Add Item to Cart

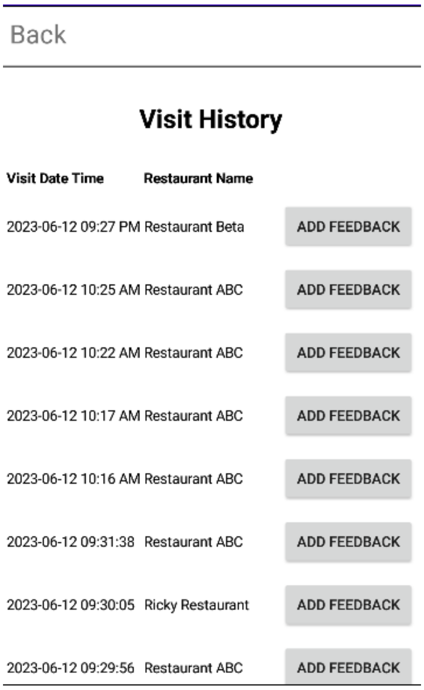
Profile



Ricky Karunakaran

Figure 3.9: Profile Page

View Visited Restaurant

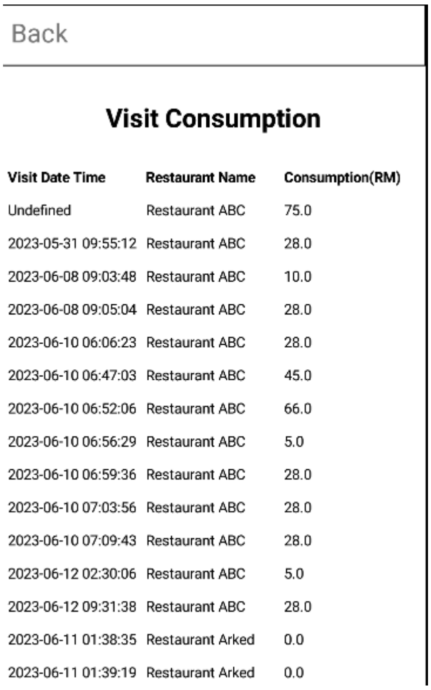


Visit History

Visit Date Time	Restaurant Name	
2023-06-12 09:27 PM	Restaurant Beta	ADD FEEDBACK
2023-06-12 10:25 AM	Restaurant ABC	ADD FEEDBACK
2023-06-12 10:22 AM	Restaurant ABC	ADD FEEDBACK
2023-06-12 10:17 AM	Restaurant ABC	ADD FEEDBACK
2023-06-12 10:16 AM	Restaurant ABC	ADD FEEDBACK
2023-06-12 09:31:38	Restaurant ABC	ADD FEEDBACK
2023-06-12 09:30:05	Ricky Restaurant	ADD FEEDBACK
2023-06-12 09:29:56	Restaurant ABC	ADD FEEDBACK

Figure 3.10: History Visit Page

View Consumption

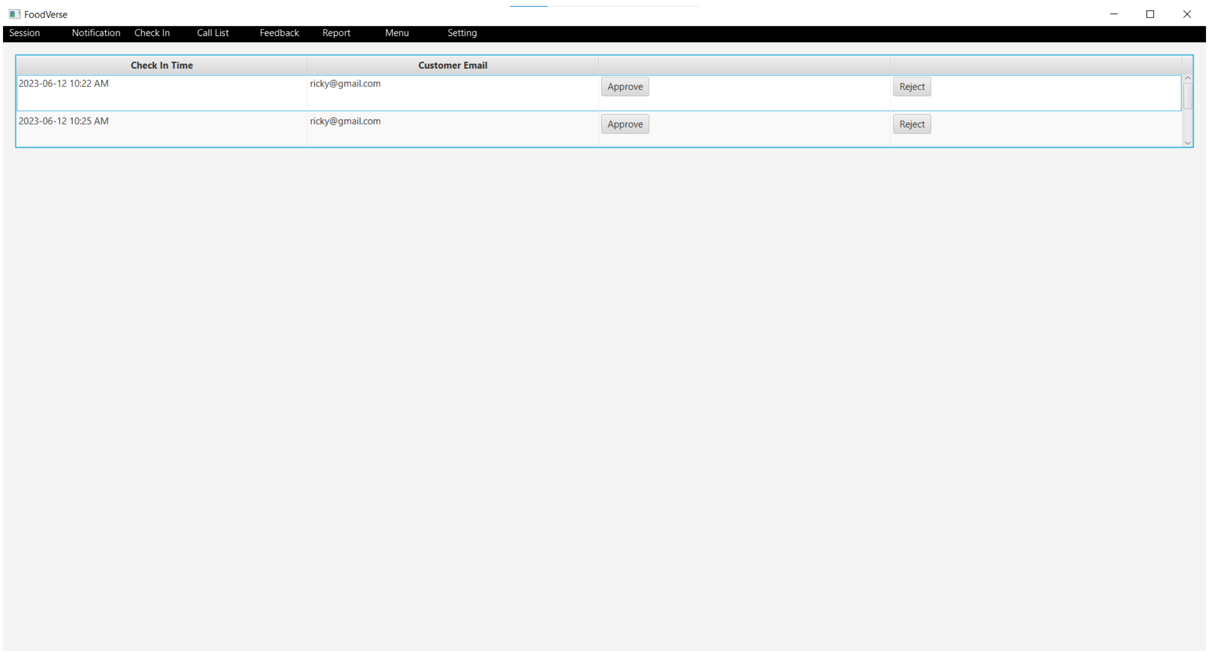


Visit Consumption

Visit Date Time	Restaurant Name	Consumption(RM)
Undefined	Restaurant ABC	75.0
2023-05-31 09:55:12	Restaurant ABC	28.0
2023-06-08 09:03:48	Restaurant ABC	10.0
2023-06-08 09:05:04	Restaurant ABC	28.0
2023-06-10 06:06:23	Restaurant ABC	28.0
2023-06-10 06:47:03	Restaurant ABC	45.0
2023-06-10 06:52:06	Restaurant ABC	66.0
2023-06-10 06:56:29	Restaurant ABC	5.0
2023-06-10 06:59:36	Restaurant ABC	28.0
2023-06-10 07:03:56	Restaurant ABC	28.0
2023-06-10 07:09:43	Restaurant ABC	28.0
2023-06-12 02:30:06	Restaurant ABC	5.0
2023-06-12 09:31:38	Restaurant ABC	28.0
2023-06-11 01:38:35	Restaurant Arked	0.0
2023-06-11 01:39:19	Restaurant Arked	0.0

Figure 3.11: View Consumption Page

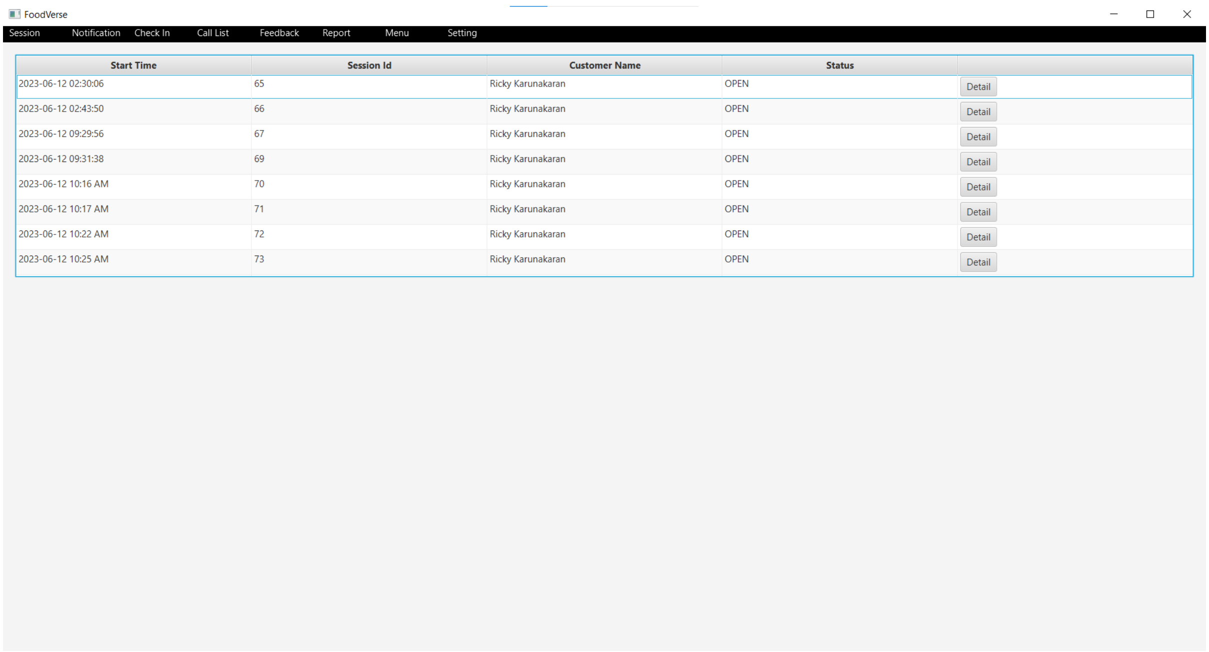
Manage Check In Request



Check In Time	Customer Email		
2023-06-12 10:22 AM	ricky@gmail.com	Approve	Reject
2023-06-12 10:25 AM	ricky@gmail.com	Approve	Reject

Figure 3.12: Manage Check In Request

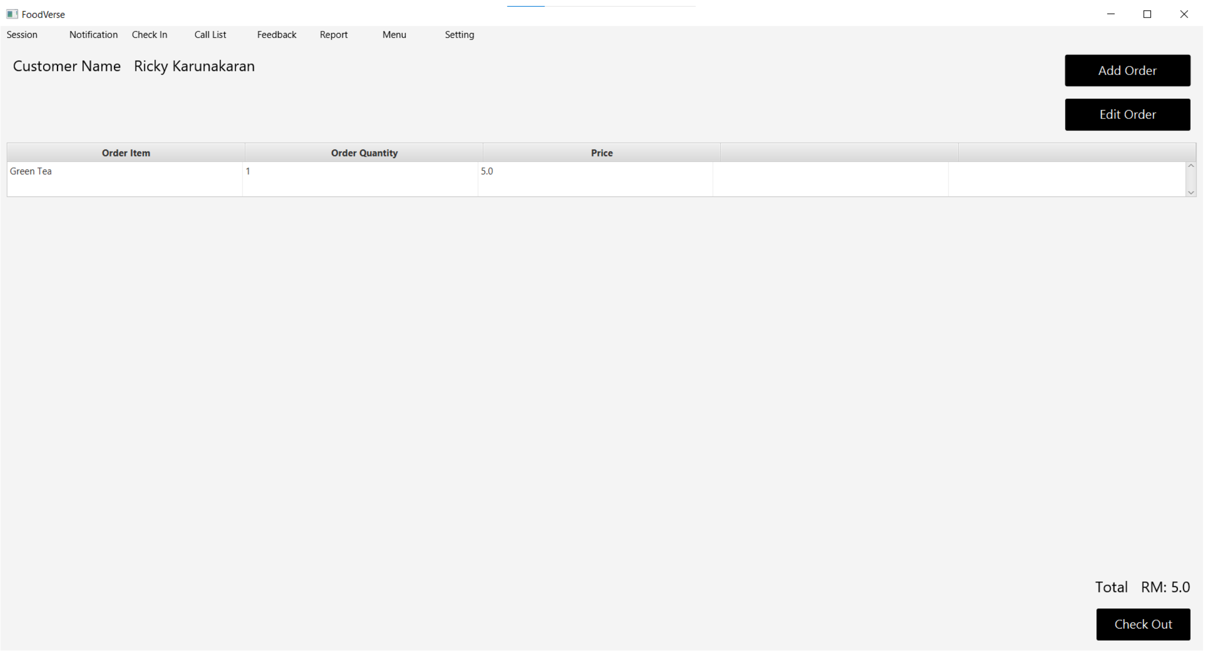
Session



Start Time	Session Id	Customer Name	Status	
2023-06-12 02:30:06	65	Ricky Karunakaran	OPEN	Detail
2023-06-12 02:43:50	66	Ricky Karunakaran	OPEN	Detail
2023-06-12 09:29:56	67	Ricky Karunakaran	OPEN	Detail
2023-06-12 09:31:38	69	Ricky Karunakaran	OPEN	Detail
2023-06-12 10:16 AM	70	Ricky Karunakaran	OPEN	Detail
2023-06-12 10:17 AM	71	Ricky Karunakaran	OPEN	Detail
2023-06-12 10:22 AM	72	Ricky Karunakaran	OPEN	Detail
2023-06-12 10:25 AM	73	Ricky Karunakaran	OPEN	Detail

Figure 3.13: Session Page

Session Details



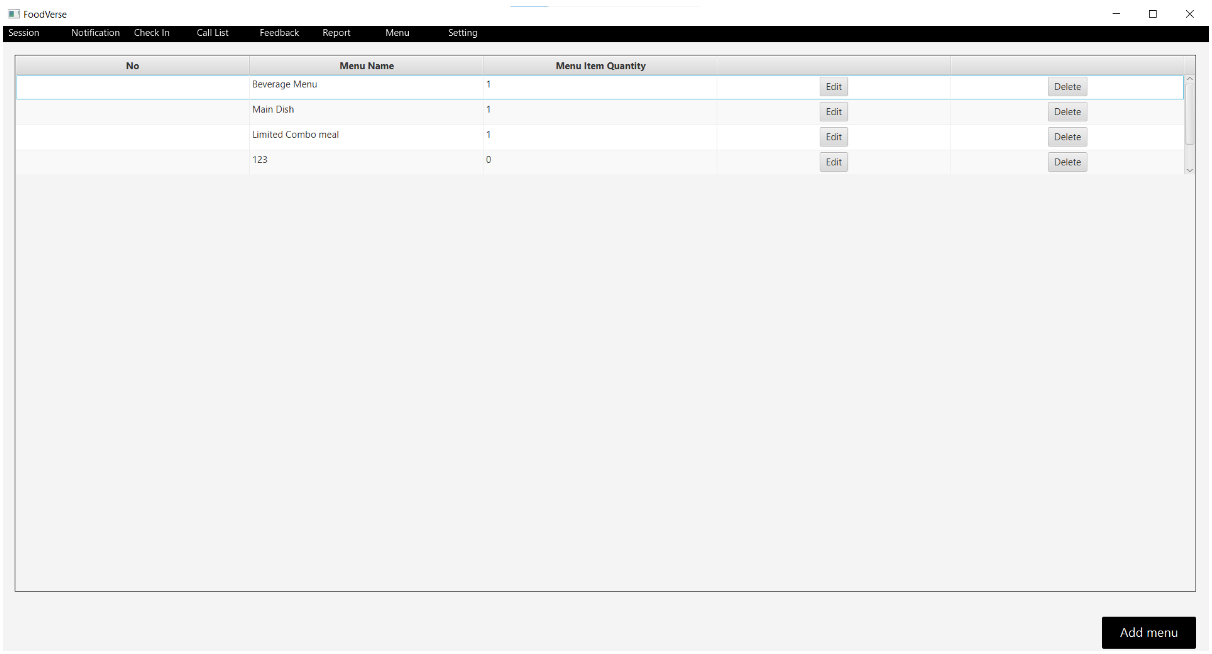
Customer Name Ricky Karunakaran

Order Item	Order Quantity	Price		
Green Tea	1	5.0		

Total RM: 5.0

Figure 3.14: Session Detail Page

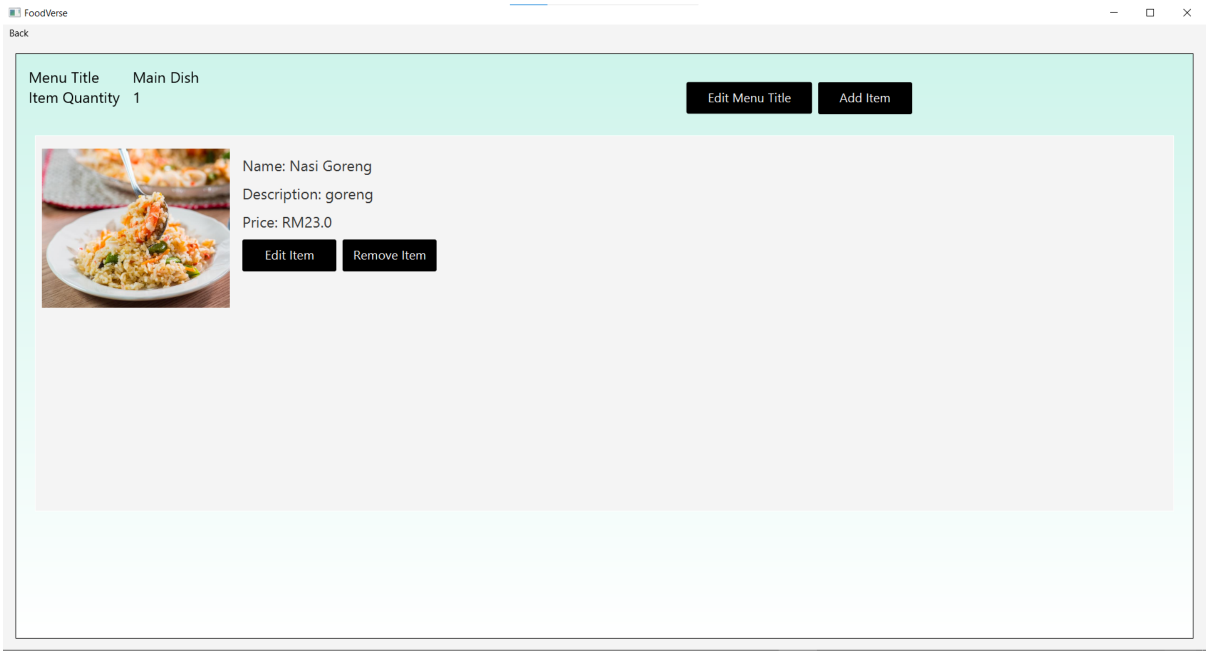
Manage Menu



No	Menu Name	Menu Item Quantity		
	Beverage Menu	1	Edit	Delete
	Main Dish	1	Edit	Delete
	Limited Combo meal	1	Edit	Delete
	123	0	Edit	Delete

Figure 3.15: Manage Menu Page

Manage Menu Item

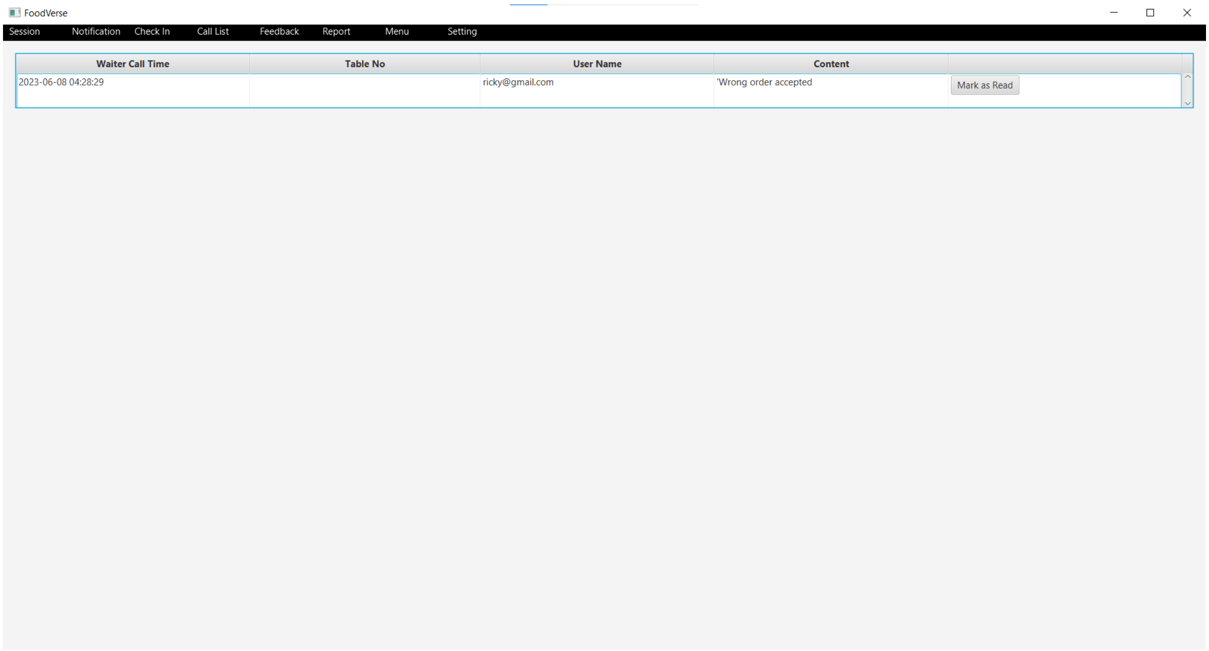


Menu Title Main Dish
Item Quantity 1

Name: Nasi Goreng
Description: goreng
Price: RM23.0

Figure 3.16: Manage Menu Item Page

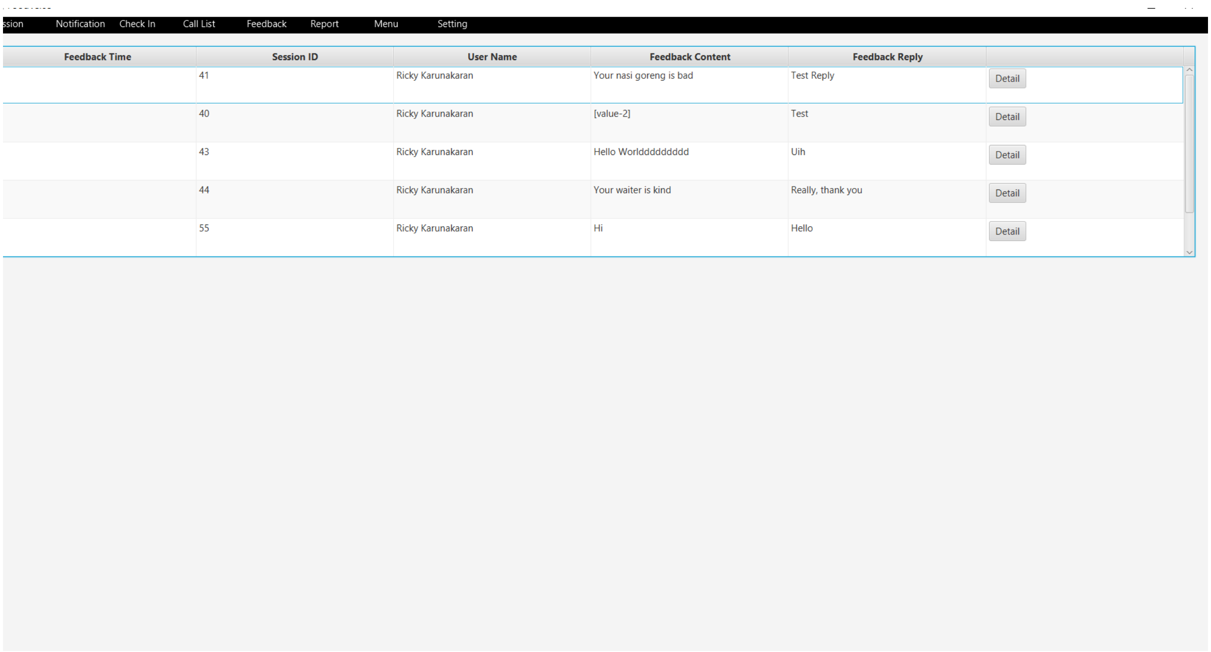
Waiter Call List



Waiter Call Time	Table No	User Name	Content	
2023-06-08 04:28:29		ricky@gmail.com	'Wrong order accepted	Mark as Read

Figure 3.17: View Waiter Call Page

Feedback



Feedback Time	Session ID	User Name	Feedback Content	Feedback Reply	
	41	Ricky Karunakaran	Your nasi goreng is bad	Test Reply	Detail
	40	Ricky Karunakaran	[value-2]	Test	Detail
	43	Ricky Karunakaran	Hello Worlddddddddd	Uih	Detail
	44	Ricky Karunakaran	Your waiter is kind	Really, thank you	Detail
	55	Ricky Karunakaran	Hi	Hello	Detail

Figure 3.18: View Feedback Page

3.1.2 Hardware Interfaces

No external hardware interface requirement.

3.1.3 Software Interfaces

No external software interface requirement.

3.1.4 Communication Interfaces

No external communication interface requirement.

3.2 System Features

3.2.1 System account subsystem

This subsystem includes the functional requirement regarding the account creation and modification. It includes three use cases which are Register, Login, Edit Account Information and Reset Password.

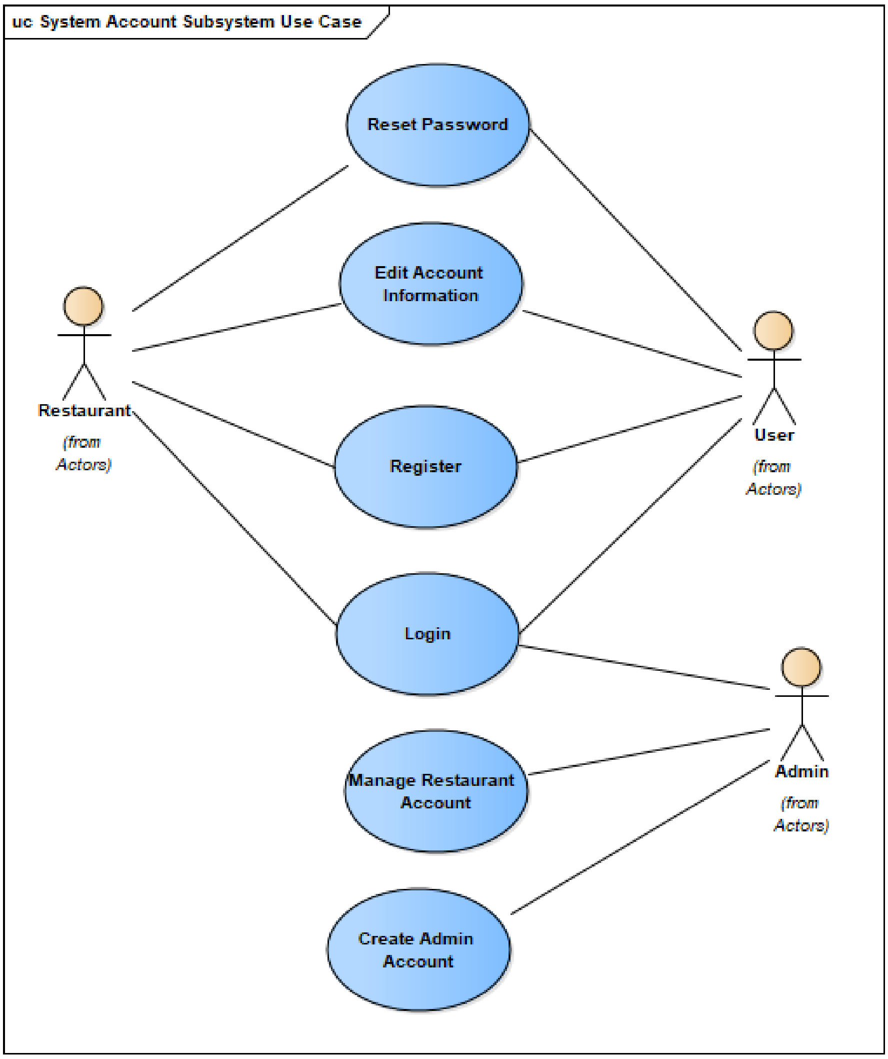


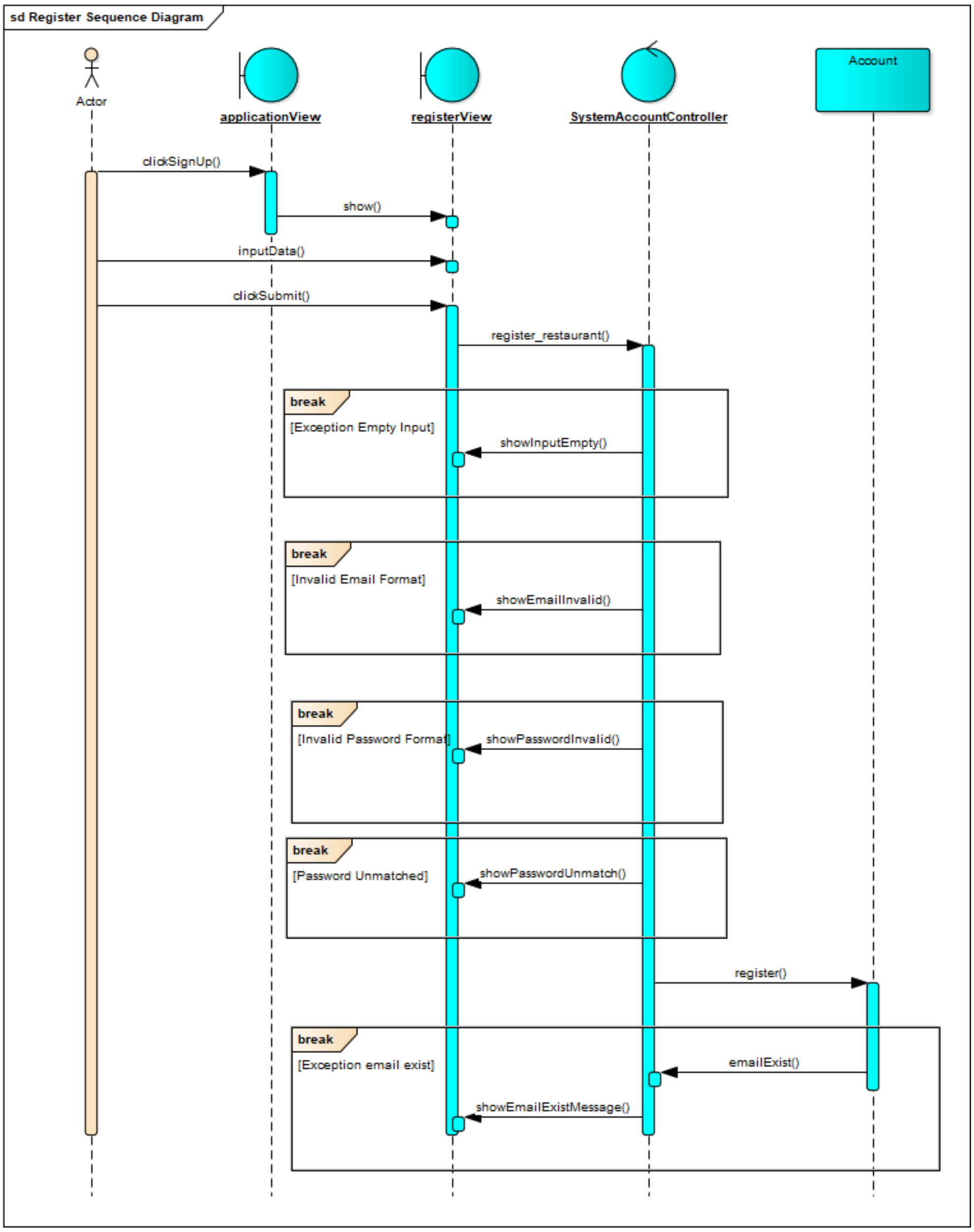
Figure 3.19: Use Case Diagram for System Account Subsystem

3.2.1.1 UC001: Register

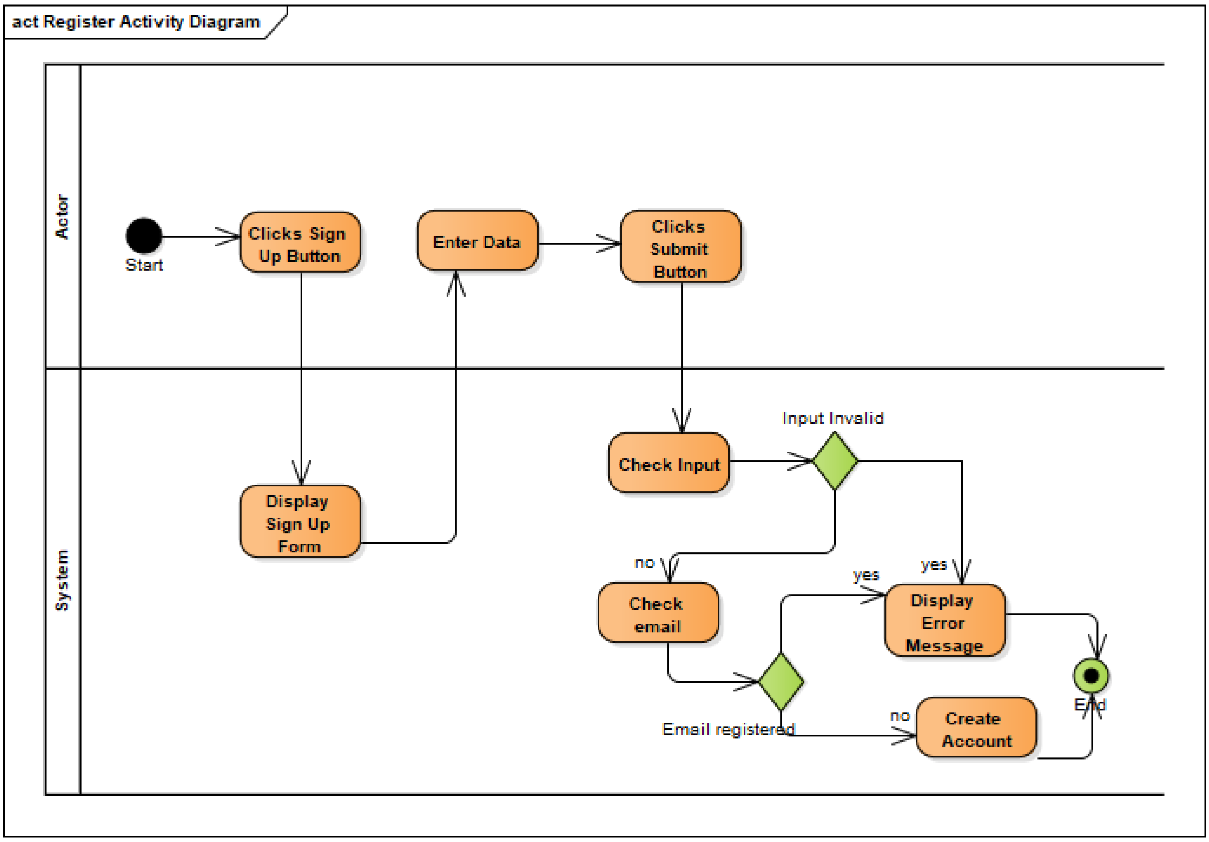
Use Case ID	UC001
Use Case Name	Register
Actors	Restaurant/User
Trigger Event	The restaurant or user wants to register an account
Brief Description	This use case is trigger when scan the QR code of the restaurant.
Pre-Condition	Application has been installed.
Post-Condition	An account is created and store in database.
Flow of events	1. Actor clicks sign up button 2. System display register view 3. Actor fills in information required including email and password. 4. Actor clicks submit button 5. System create restaurant/user account with the information. If email is registered, perform exception flow 5E. 6. System display restaurant/user Home window.
Alternative Flow	-
Exception Flow	4E1: Empty Input 1. System display error message "Input cannot be empty" 2. Use case end. 4E2: Invalid Email Format 1. System display error message "Invalid email format" 2. Use case end. 4E3: Invalid Password 1. System display error message "Invalid password format" 2. Use case end. 4E4: Unmatched Password:

	1. System display error message “Password unmatched” 2. Use case end. 5E: Email Registered 1. System display error message “This email has been used” 2. Use case end.

UCD001: Register



SD001: Sequence Diagram for UC001 Register

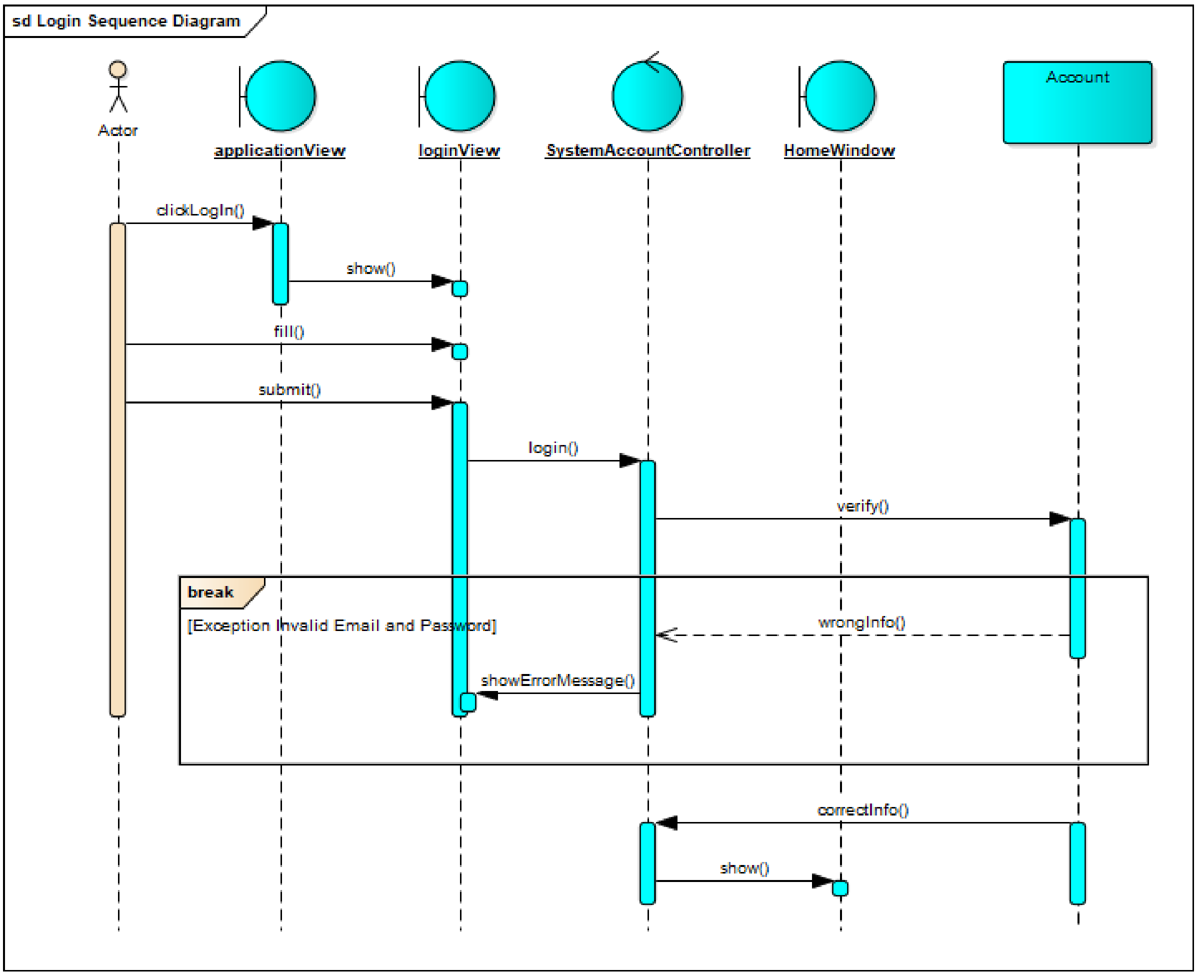


AD001: Activity Diagram for UC001 Register

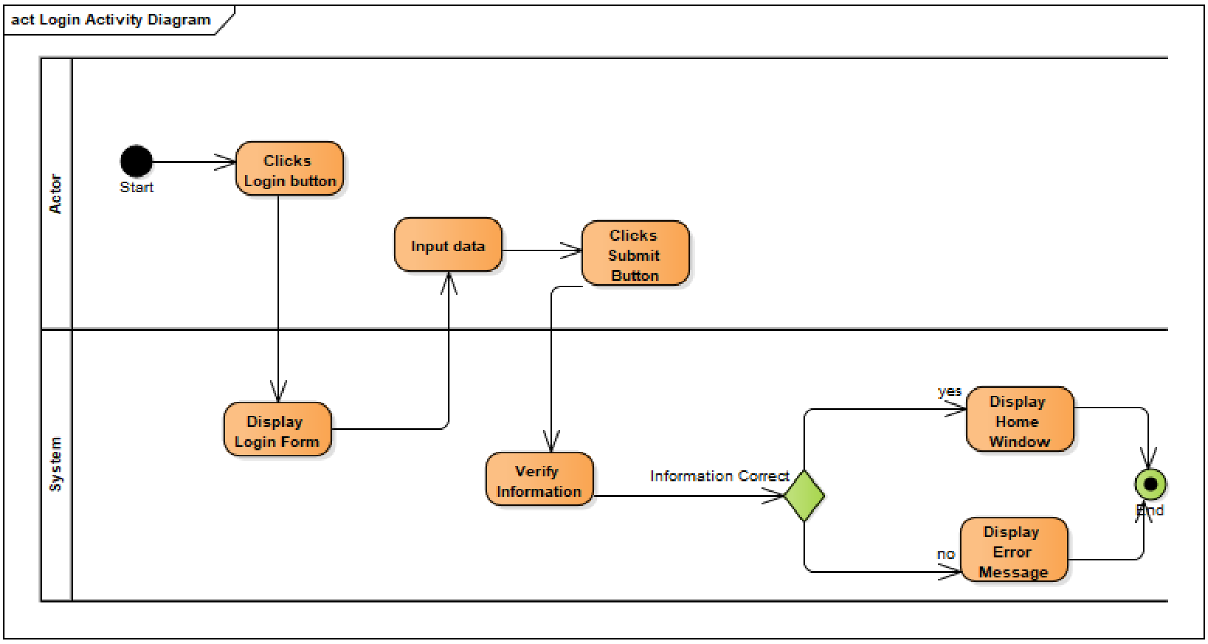
3.2.1.2 UC002:Login

Use Case ID	UC002
Use Case Name	Login
Actors	Restaurant/User/Admin
Trigger Event	Restaurant/User/Admin want to access the services of the application.
Brief Description	This use case is trigger when the user clicks login button in the application page.
Pre-Condition	Restaurant/User/Admin own an account.
Post-Condition	Restaurant/User is redirected to its home window respectively.
Flow of events	1. Actor clicks login button 2. System display login form 3. Actor fill in data 4. Actor click submit. 5. System verify information. If the information is incorrect, perform exception flow 4E. 6. System redirect user to home window.
Alternative Flow	-
Exception Flow	5E: Incorrect data. 1. System display error message “Wrong email and password”. 2. Use case end.

UCD002: Login



SD002: Sequence Diagram for UC002 Login

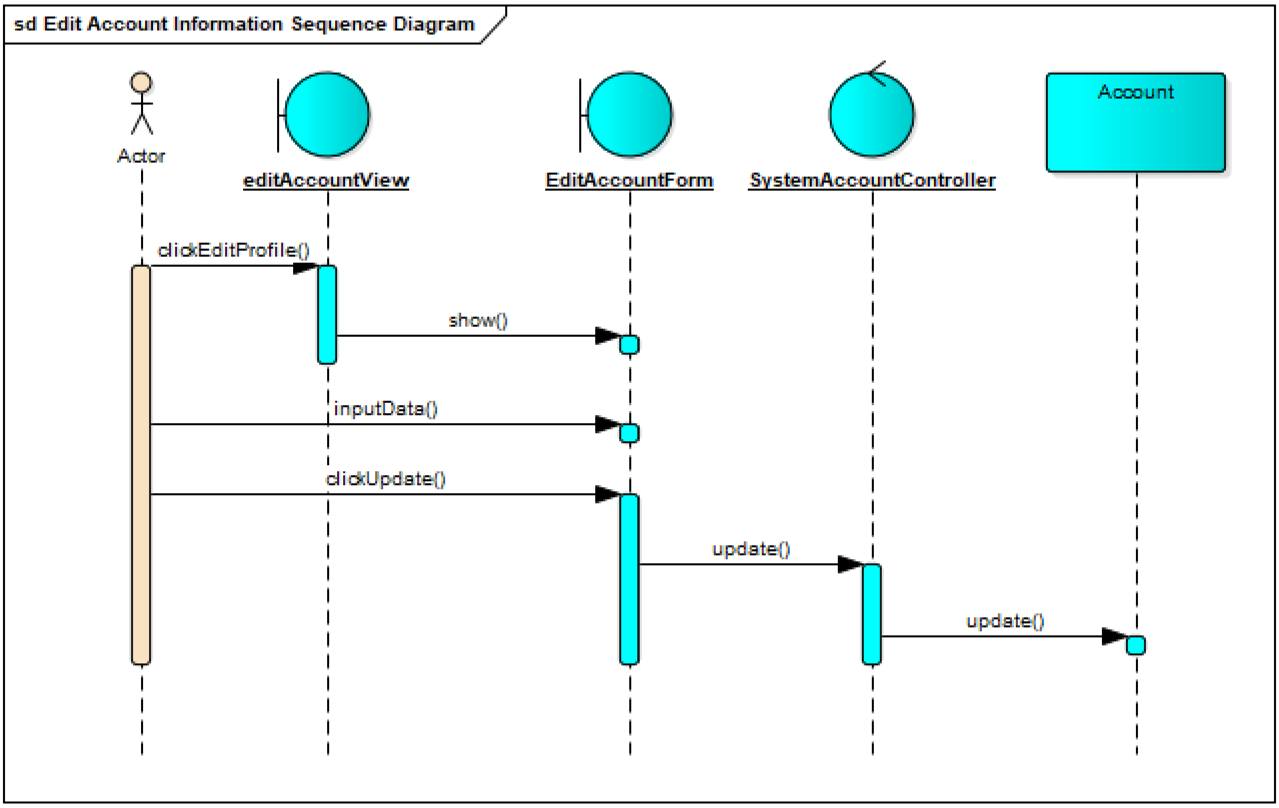


AD002: Activity Diagram for UC002 Login

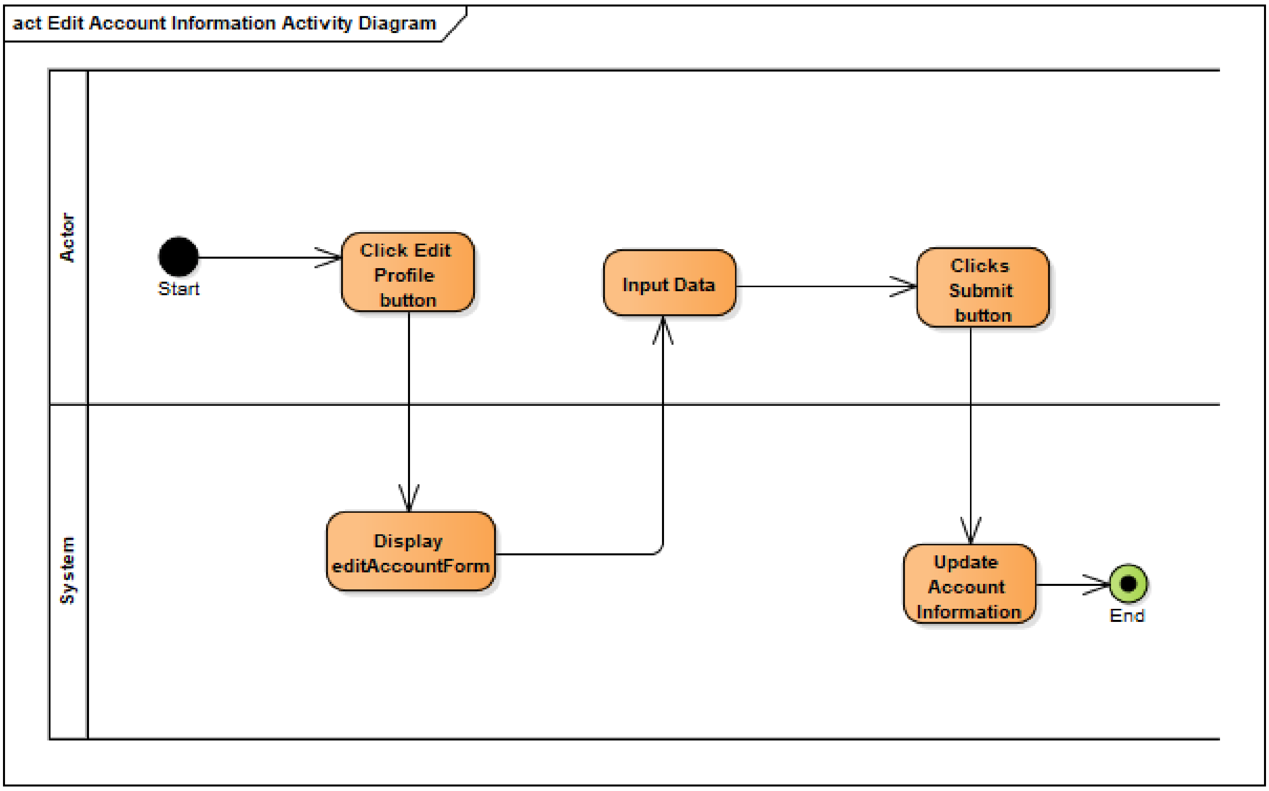
3.2.1.3 UC003:Edit Account Information

Use Case ID	UC003
Use Case Name	Edit Account Information
Actors	Restaurant/User
Trigger Event	Restaurant/User want to edit the registered account information.
Brief Description	This use case is trigger when the user clicks edit profile button in the profile page.
Pre-Condition	-
Post-Condition	The information has been updated.
Flow of events	1. Actor click “Update Profile” button. 2. System display edit account information form. 3. Actor fill information 4. Actor click “Update” button. 5. System updated actor’s account information.
Alternative Flow	-
Exception Flow	-

UCD003: Edit Account Information



SD003: Sequence Diagram for UC003 Edit Account Information

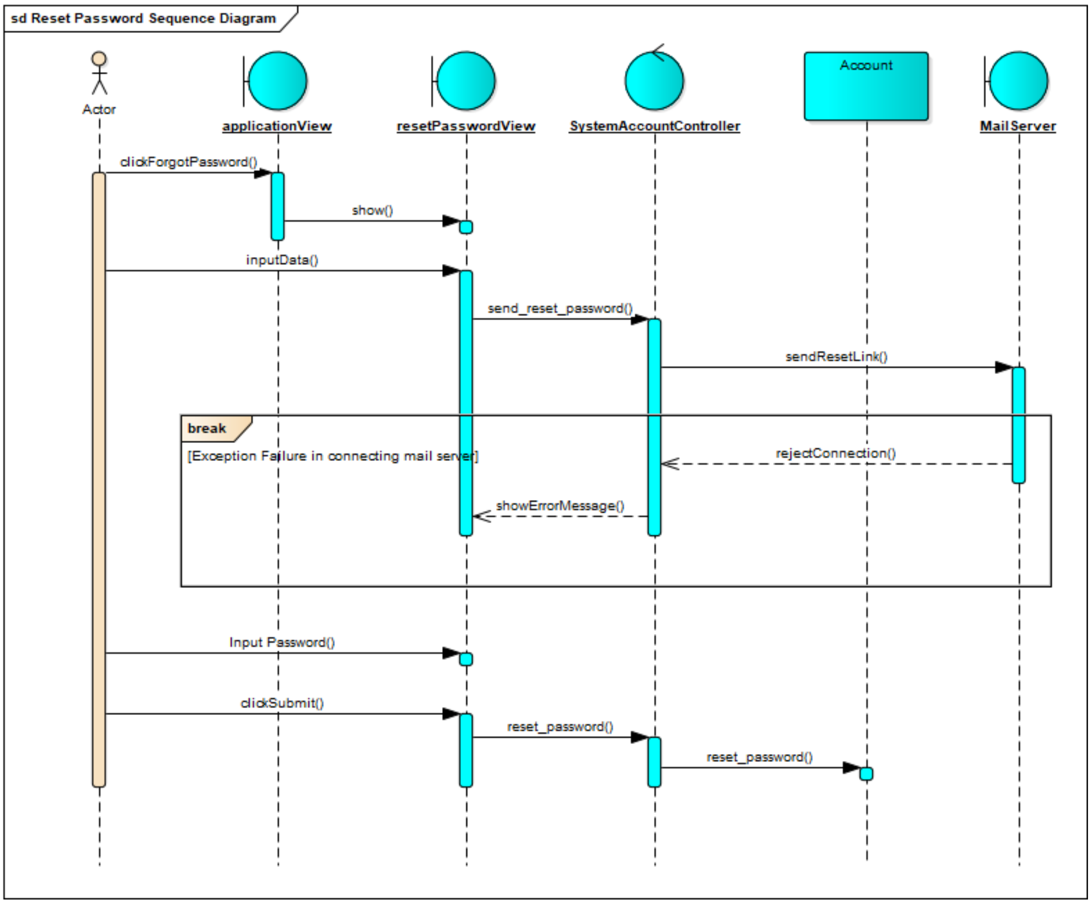

act Edit Account Information Activity Diagram
Actor
System
Start
Click Edit Profile button
Input Data
Clicks Submit button
Display editAccountForm
Update Account Information
End

AD003: Activity Diagram for UC003 Edit Account Information

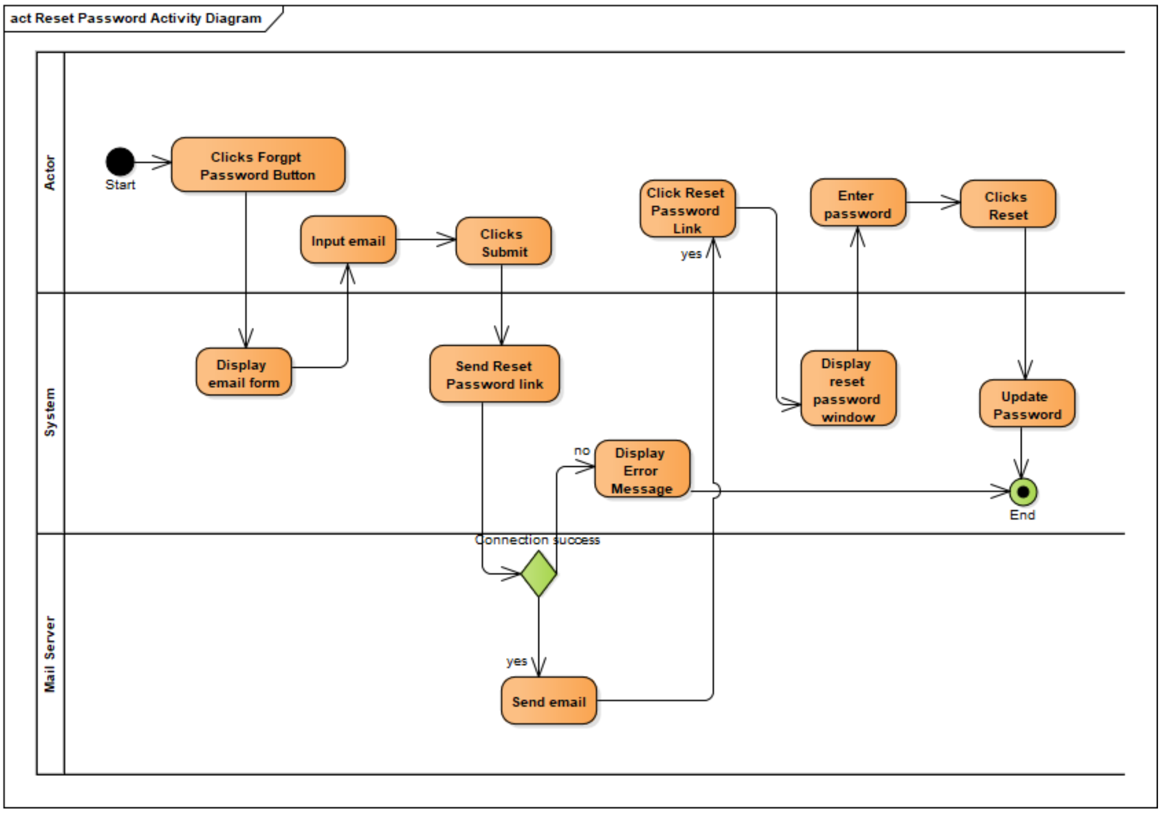
3.2.1.4 UC004:Reset Password

Use Case ID	UC004
Use Case Name	Reset Password
Actors	Restaurant/User
Trigger Event	Restaurant/User want to reset the password without logging in system.
Brief Description	This use case is trigger when the restaurant/user click forget password in the application home window.
Pre-Condition	Restaurant/User own an account.
Post-Condition	A reset password link is sent to the restaurant/user email.
Flow of events	1. Actor clicks forget password. 2. System display email form. 3. Actor fill in email. 4. System send reset password verification code via mail server. If mail server unavailable, perform exception flow 4E. 5. System display reset password window 6. Actor enter verification code and new password. 7. Actor clicks reset button 8. System update account's password.
Alternative Flow	-
Exception Flow	4E: Mail Server Failure 1. System display error message "Failure in sending reset password email." 2. Use case end. 7E: Wrong Verification Code 1. System display error message "verification code do not match"

UDC004: Reset Password



SD004: Sequence Diagram for UC004 Reset Password

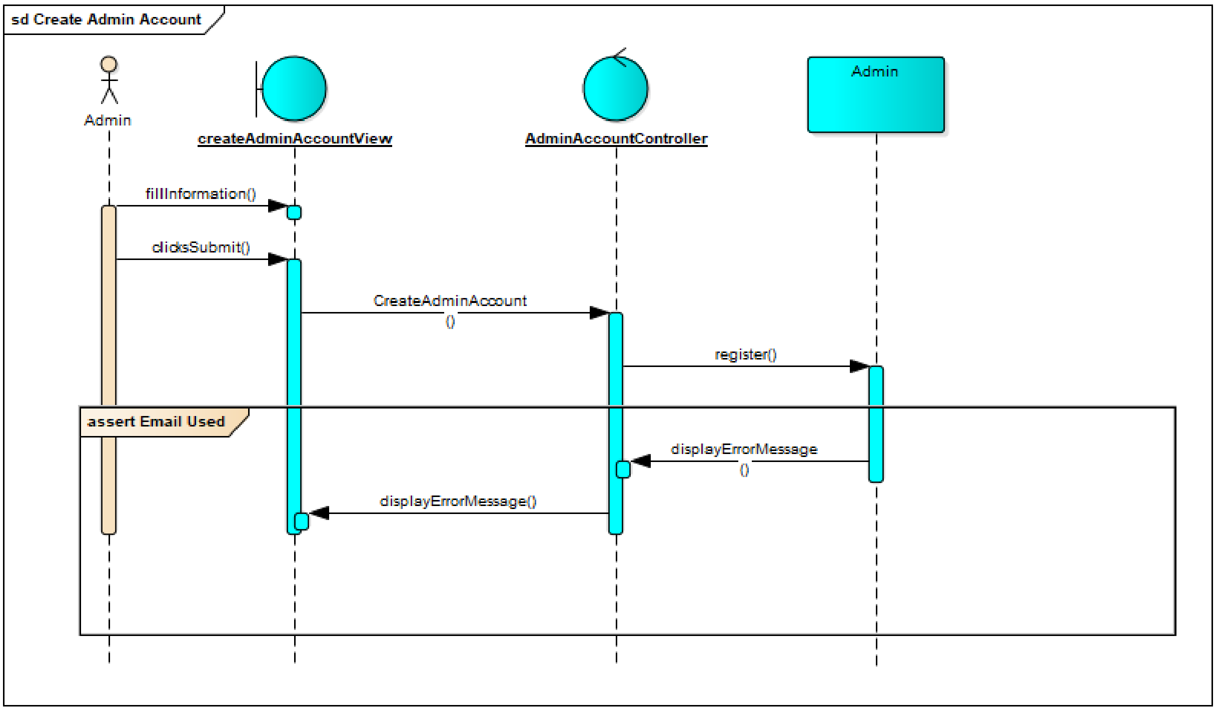


AD004: Activity Diagram for UC004 Reset Password

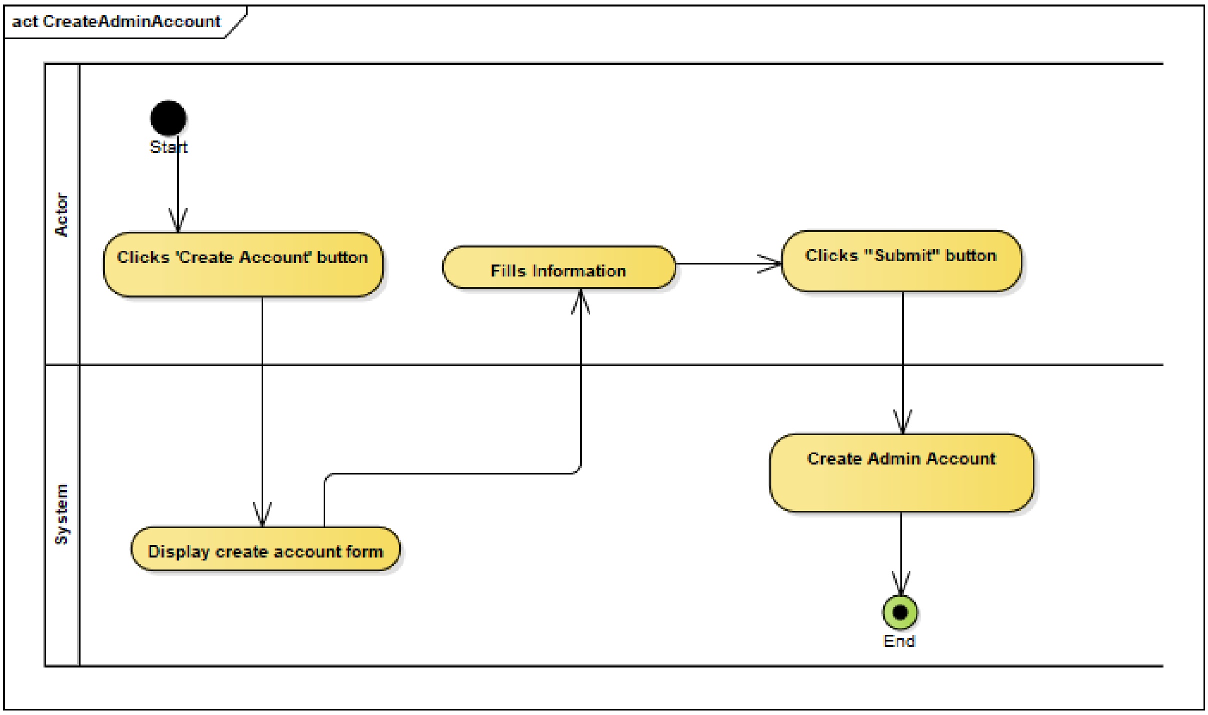
3.2.1.5 UC024: Create Admin Account

Use Case ID	UC024
Use Case Name	Create Admin Account
Actors	Admin
Trigger Event	Admin want to create an admin account
Brief Description	This use case is trigger when admin want to create an admin account with access right to manage the restaurant account
Pre-Condition	Application has been installed
Post-Condition	An admin account is created
Flow of events	1. Admin clicks “Create Account” button 2. System display Account Creation form 3. Admin fills information 4. Admin clicks ‘Submit’ button 5. System create an admin account
Alternative Flow	
Exception Flow	3E: Email used 1. System display error message “Email has been used” 2. Use case end.

UCD024:Create Admin Account



SD024:Sequence Diagram for UC024 Create Admin Account

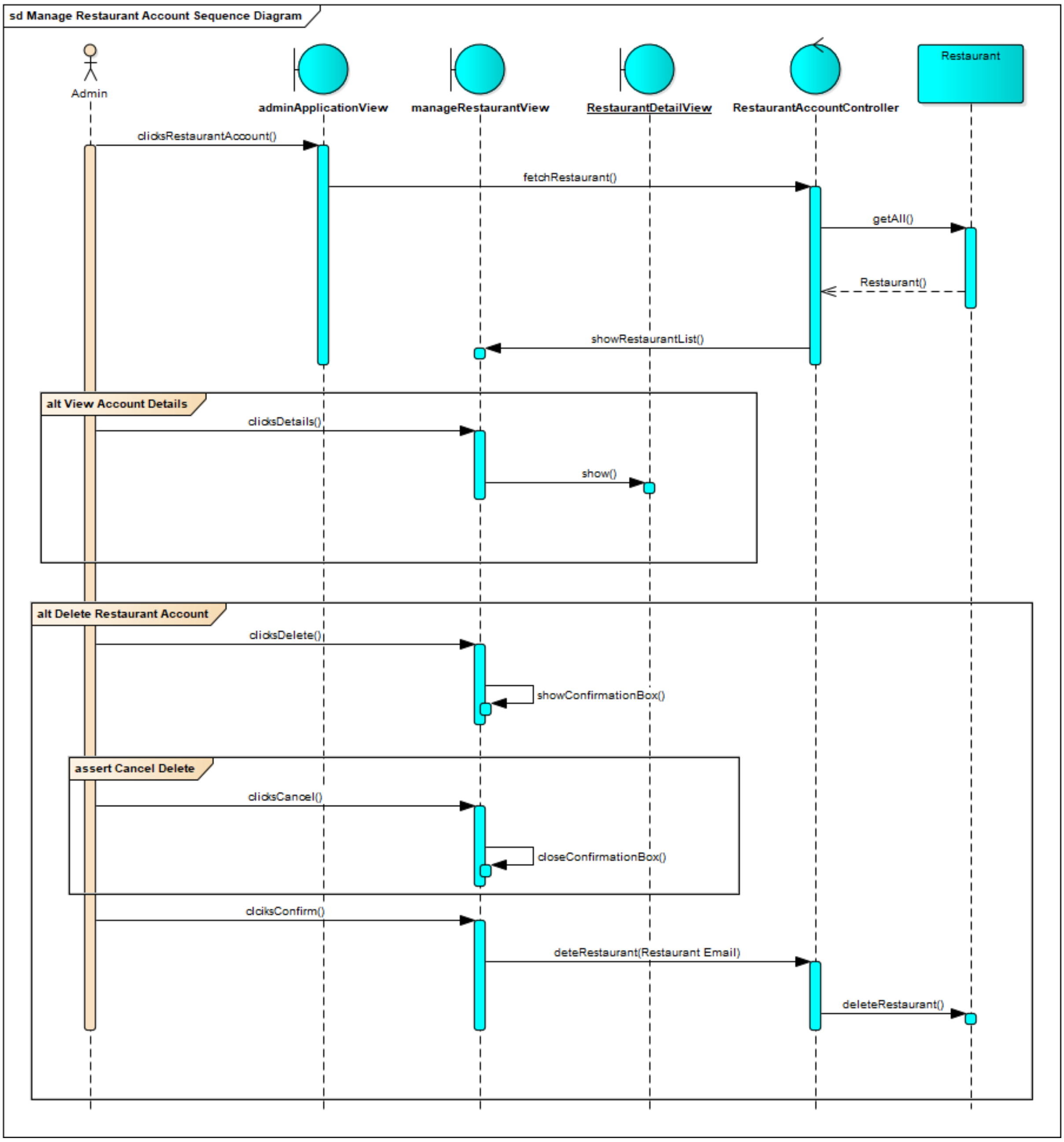


AD024: Activiti Diagram for UC024 Create Admin Account

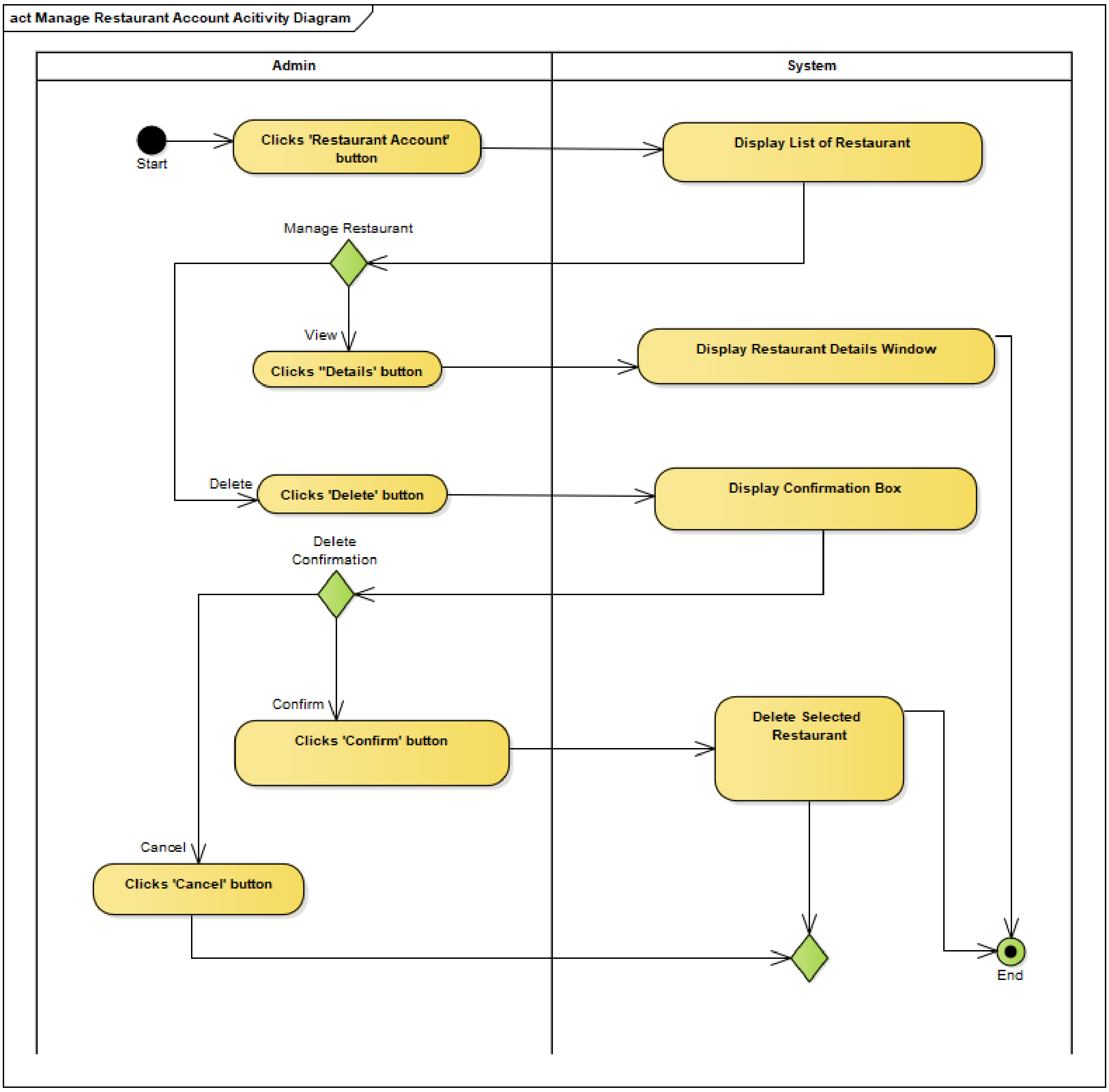
3.2.1.6 UC025: Manage Restaurant Account

Use Case ID	UC025
Use Case Name	Manage Restaurant Account
Actors	Admin
Trigger Event	Admin want to delete or freeze or view the account of restaurant
Brief Description	This use case is trigger when admin clicks account button
Pre-Condition	Admin own an account
Post-Condition	-
Flow of events	1. Admin clicks “Restaurant Account” Button 2. System display list of restaurant account 3. If admin clicks “Detail” button, perform Alternative Flow 3A. If admin clicks “Delete” button, perform Alternative Flow 3B.
Alternative Flow	3A: View Restaurant Account Details 1. System display restaurant details information 3B: Delete Restaurant Account 1. System display restaurant delete confirmation box. 2. If actor clicks ‘Confirm’ button, system delete the selected restaurant.
Exception Flow	3E: Cancel Delete 1. System close delete confirmation box. 2. Use case end.

UCD025: Manage Restaurant Account



SD025: Sequence Diagram for UC025 Manage Restaurant Account



AD025: Activity Diagram for UC025 Manage Restaurant Account

3.2.2 Customer Authentication Subsystem

This subsystem includes the functional requirement of the system regarding the authentication of user as a customer. Unauthenticated user can only browse the menu of the restaurant. To access other services, the user need to get authentication as a customer first. The use cases included in this subsystem are Check In, Check Out, Make Payment, Manage Check In and Manage Check Out.

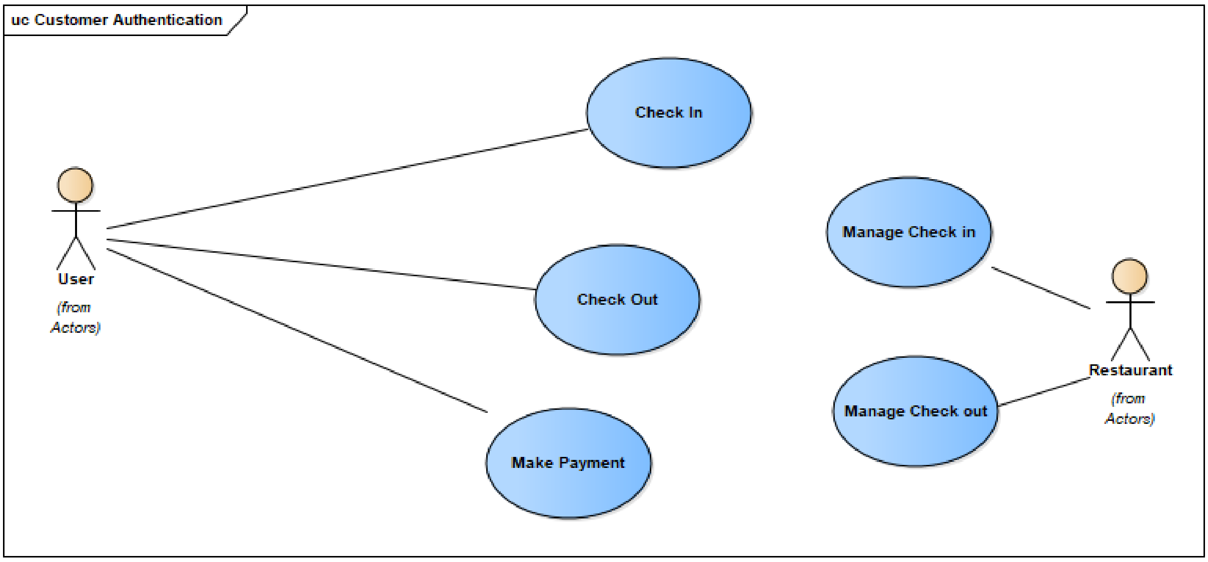


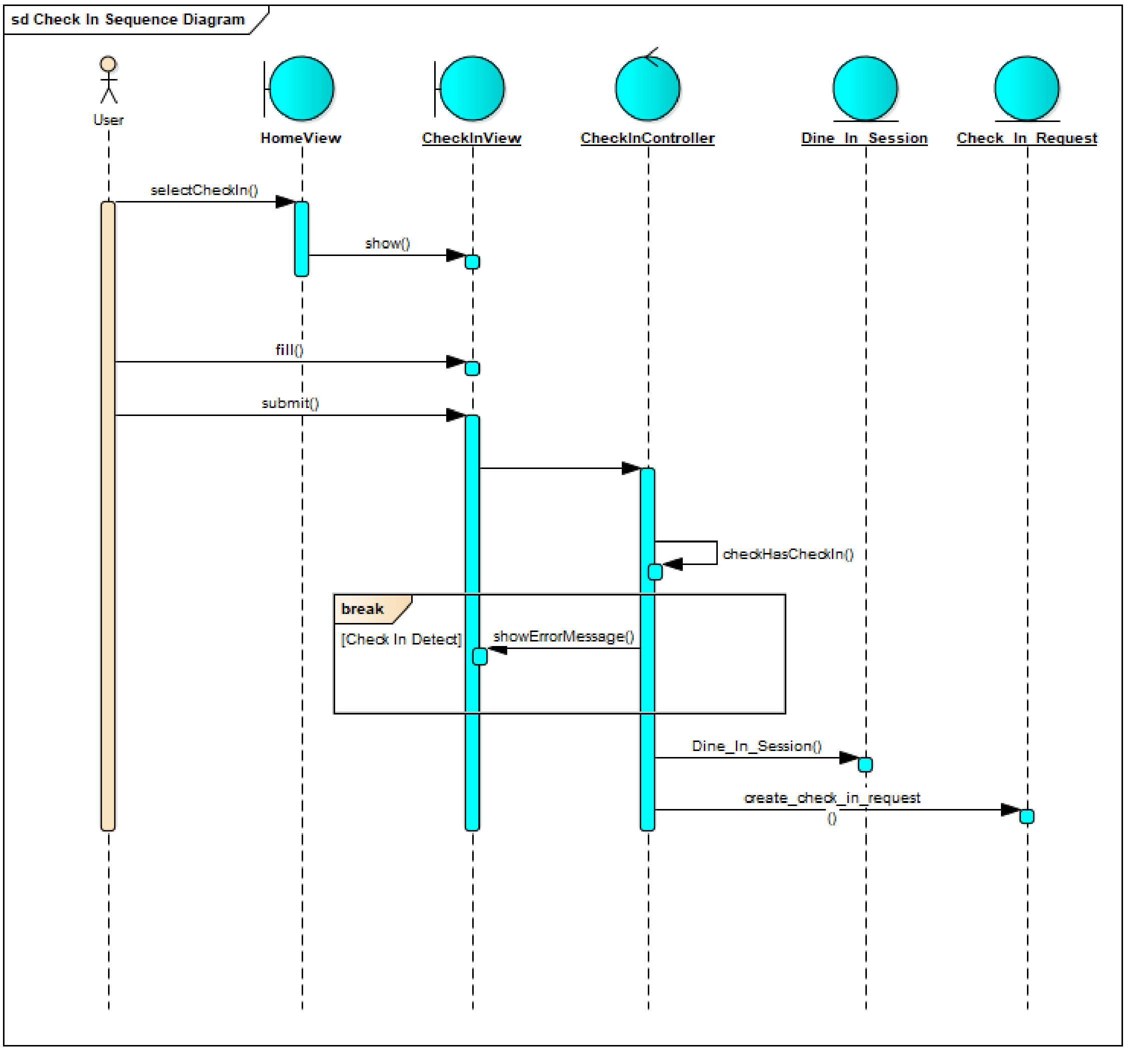
Figure 3.20: Use Case Diagram for Customer Authentication Subsystem

3.2.2.1 UC005: Check In

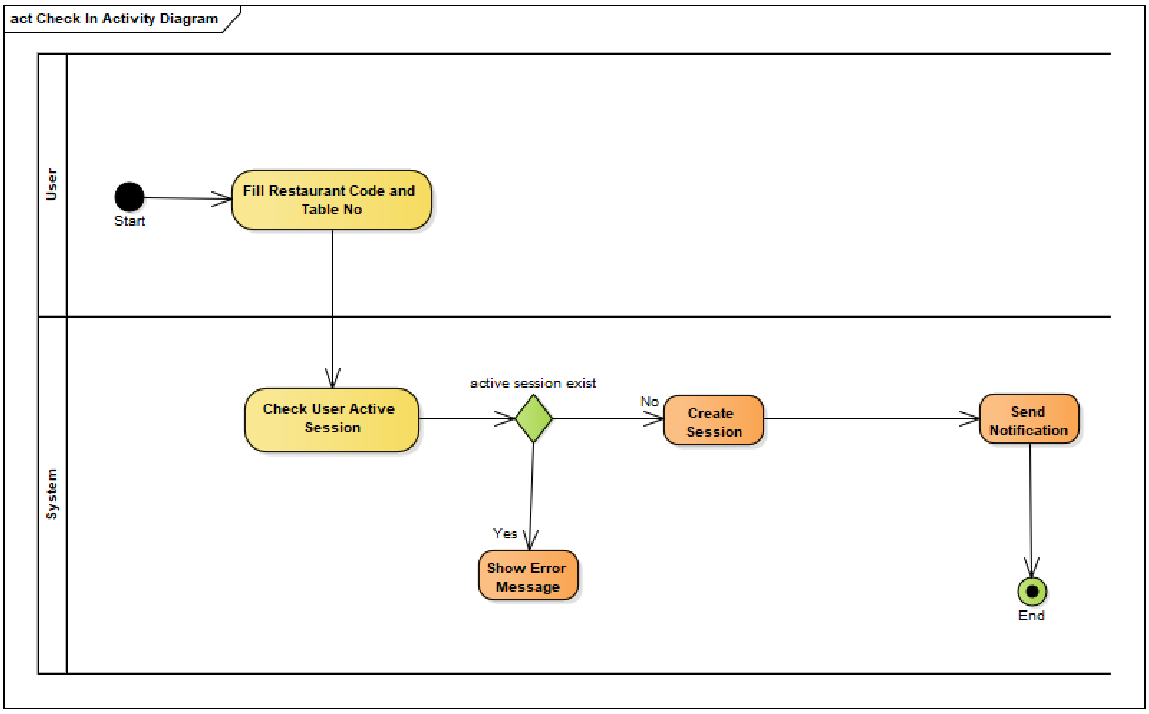
Use Case ID	UC005
Use Case Name	Check In
Actors	Users
Trigger Event	The user wants to check in to a restaurant
Brief Description	This use case is trigger when scan the QR code of the restaurant.
Pre-Condition	The user does not check in to any restaurant.
Post-Condition	A check in notification is send to the restaurant
Flow of events	1. Actor click check in menu button. 2. System display check in form 3. Actor fill in the restaurant check in code and table number

	4. Actor click check in button. If Actor has checked in to another restaurant, perform exception flow 4E. 5. System create Dine In Session 6. System create check in request to restaurant
Alternative Flow	-
Exception Flow	4E: Check in detect The system displays error message “You have check in to a restaurant. Pls check out first before check in to another restaurant” and use case end.

UCD005: Check In



SD005: Sequence Diagram for UC005 Check In

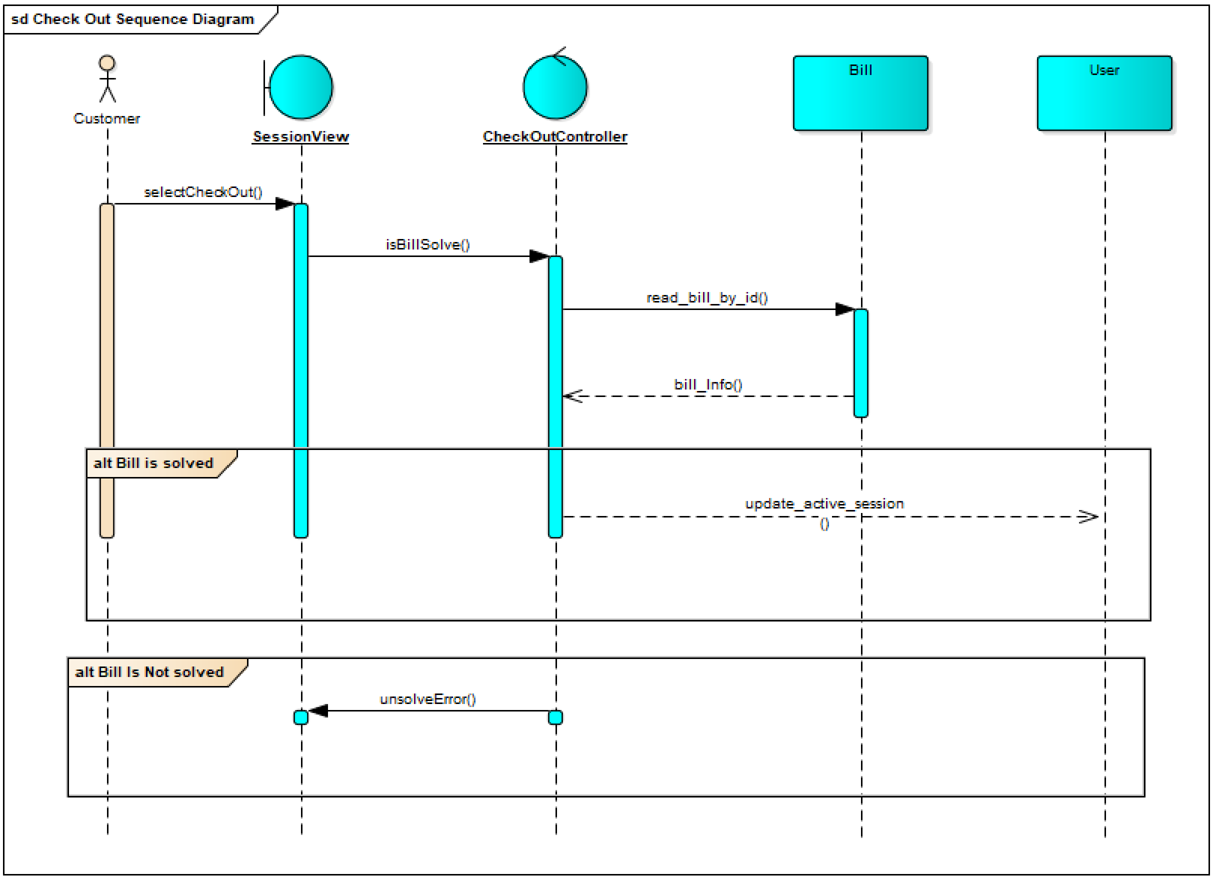


AD005: Activity Diagram for UC005 Check In

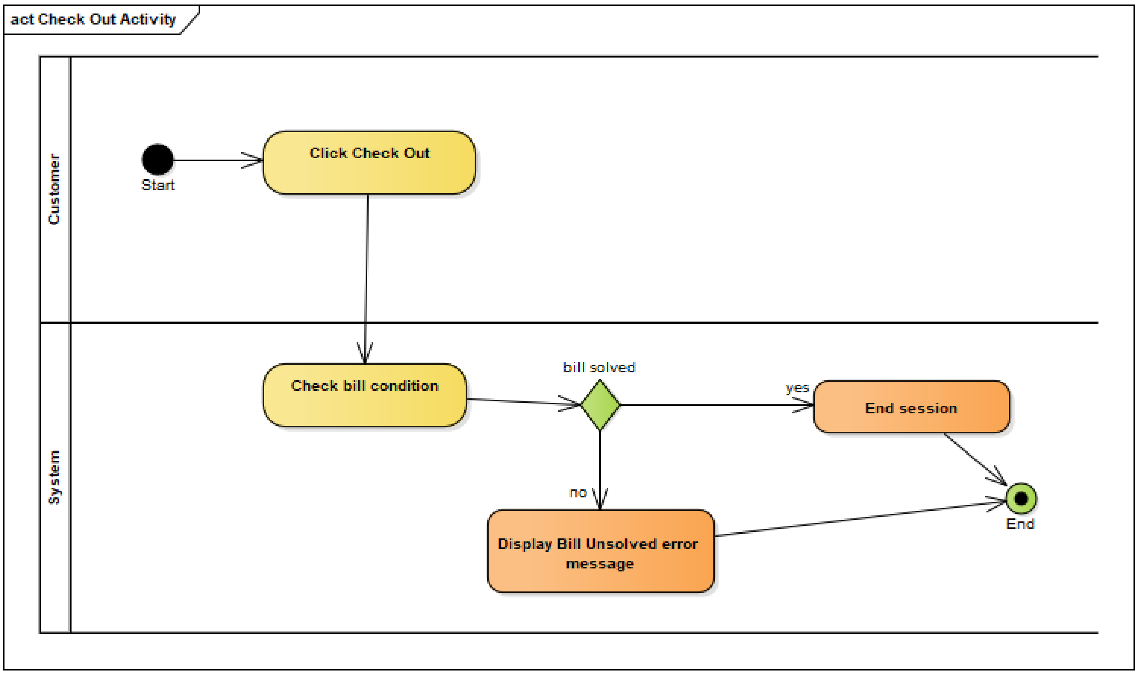
3.2.2.2 UC006: Check Out

Use Case ID	UC006
Use Case Name	Check Out
Actors	User
Trigger Event	The user wants to check out from a restaurant
Brief Description	The customer clicks the check out button. The system will check out the customer from the restaurant if condition meet.
Pre-Condition	The actor is checked in in a restaurant
Post-Condition	The customer is checked out from a restaurant.
Flow of events	1. Actor click check out button. 2. System closes the session and update user active session. If the actor has unsolved bill, perform exception flow 2E.
Alternative Flow	-
Exception Flow	2E1: Unsolved Bill The system displays error message “Please pay your bill before check out” and use case end.

UCD006: Check Out



SD006: Sequence Diagram for UC006 Check Out



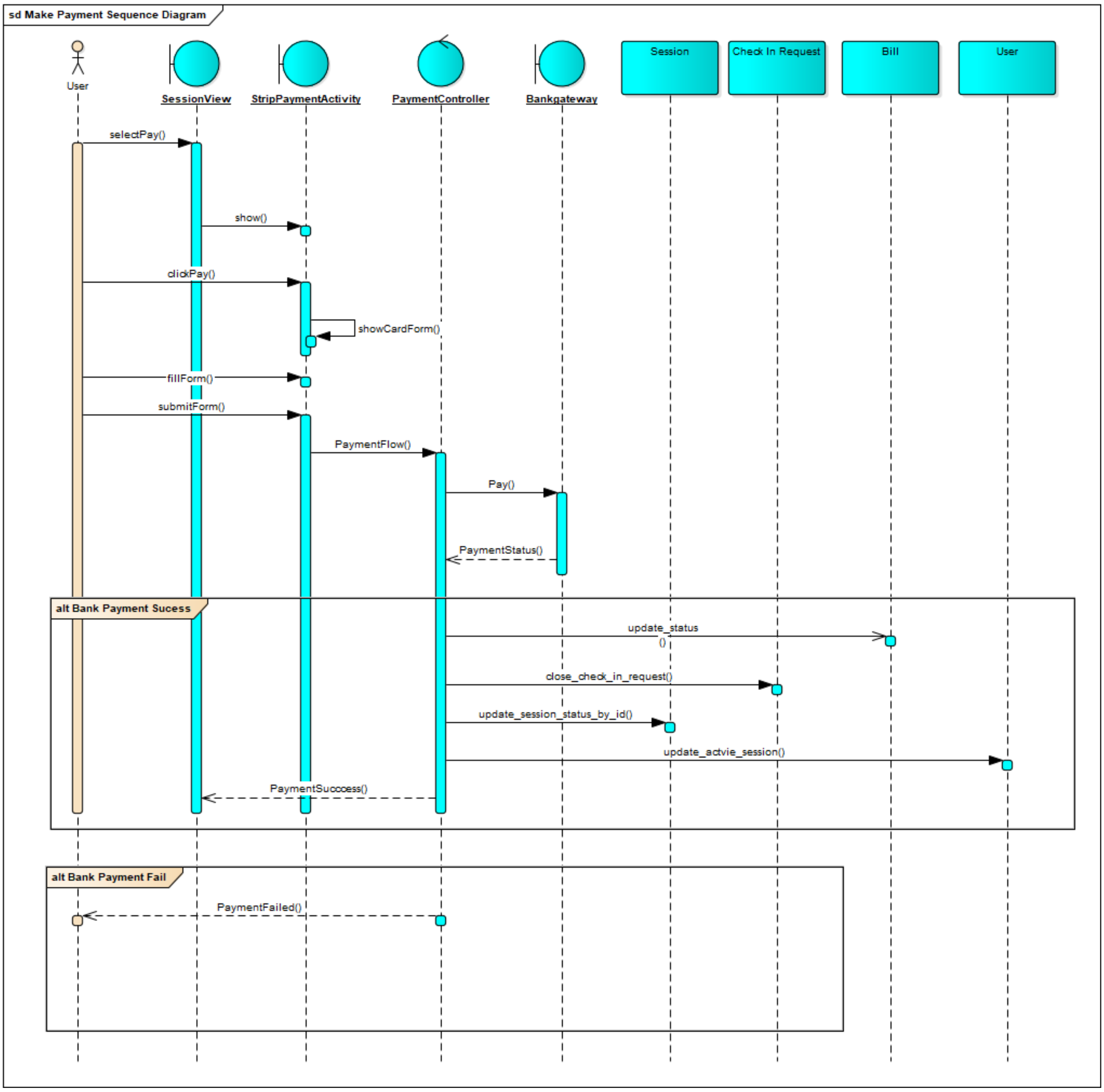
AD006: Activity Diagram for UC006 Check Out

3.2.2.3 UC007: Make Payment

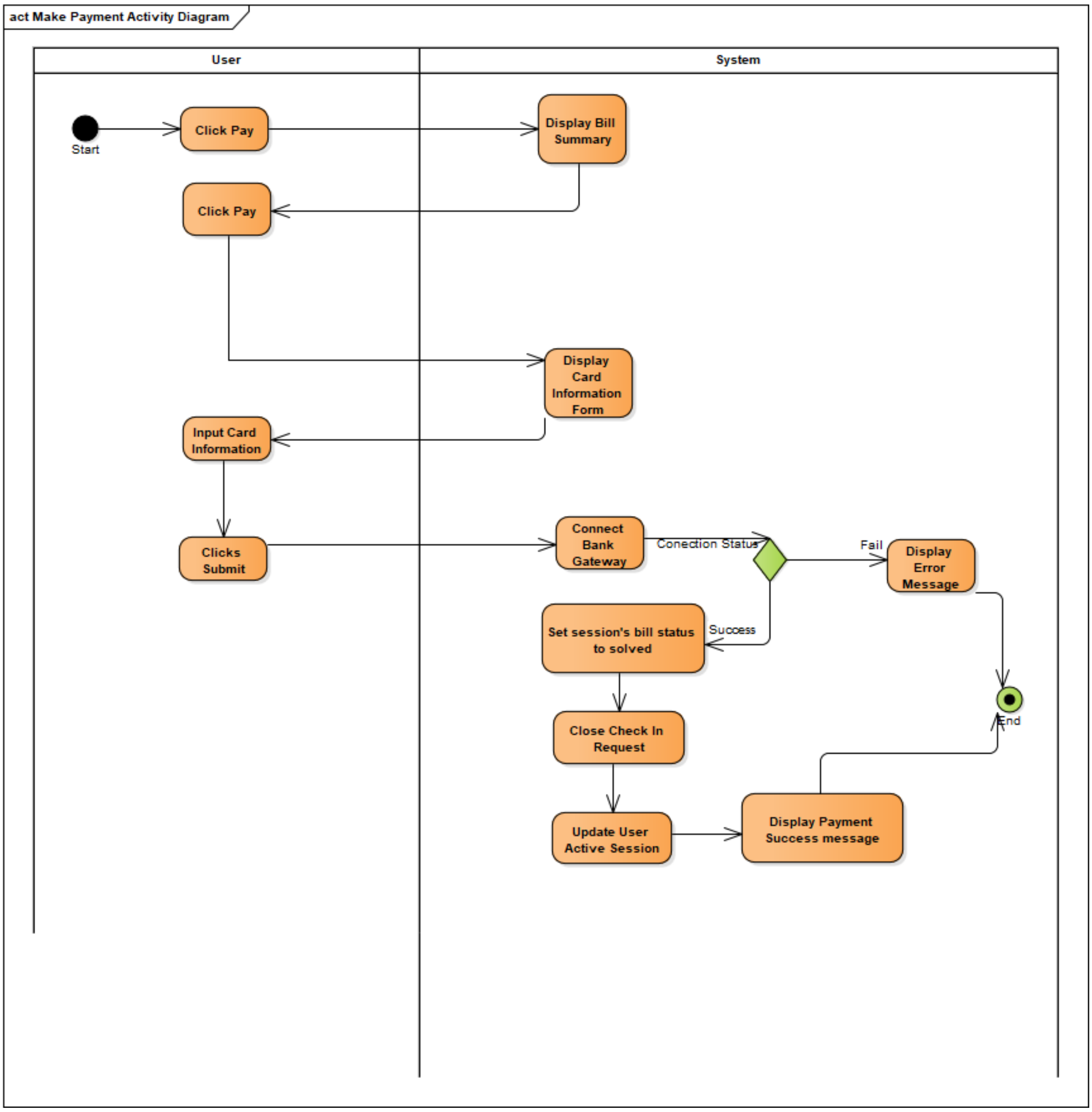
Use Case ID	UC007

Use Case Name	Make Payment
Actors	User
Trigger Event	The user wants to solve their bill
Brief Description	The customer clicks the pay button. The system will check out the customer from the restaurant if condition meet.
Pre-Condition	There is unsolved bill exist.
Post-Condition	Bill updated to “Solved” Status.
Flow of events	1. Actor click ‘Pay Bill’ button 2. The system prompts out bill summary window. 3. Actor click ‘Pay’ button 4. System prompts card information window 5. Actor fill in card information 6. Actor click ‘Pay’ button 7. System connect to the bank gateway. 8. Bank gateway return success to system. If bank gateway return error, perform exception flow 6E. 9. System set the session’s bill status to “Solved” 10. System set the session’s status to ‘Closed’ 11. System update user active session. 12. System display “Payment success” message. 13. Actor click ‘OK’ button 14. System redirect to home window
Alternative Flow	
Exception Flow	6E: Bank gateway return error 1. The system displays error message return by bank gateway and use case end.

UCD007: Make Payment



SD007: Sequence Diagram for UC007 Make Payment

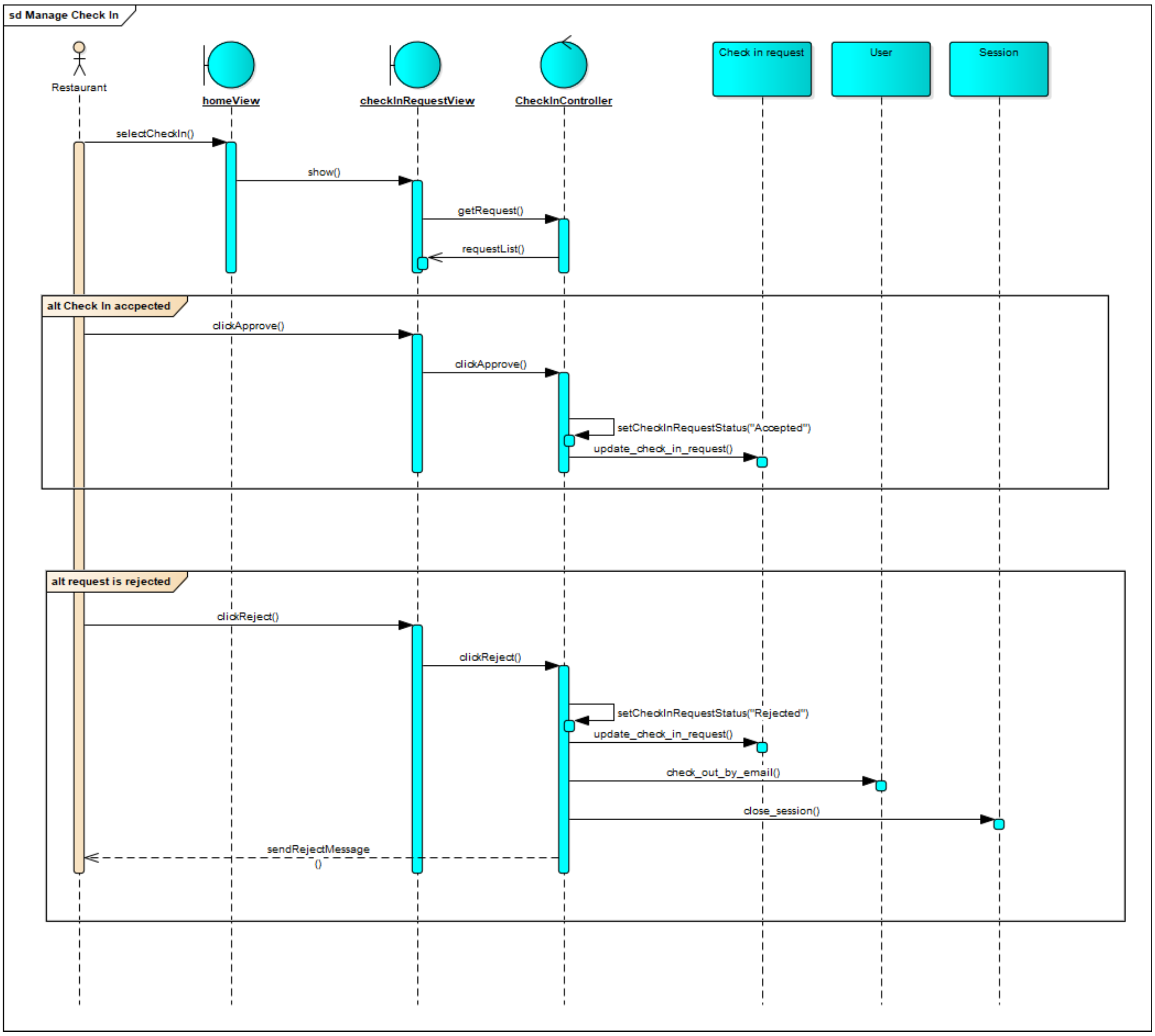


AD007: Activity Diagram for UC007 Make Payment

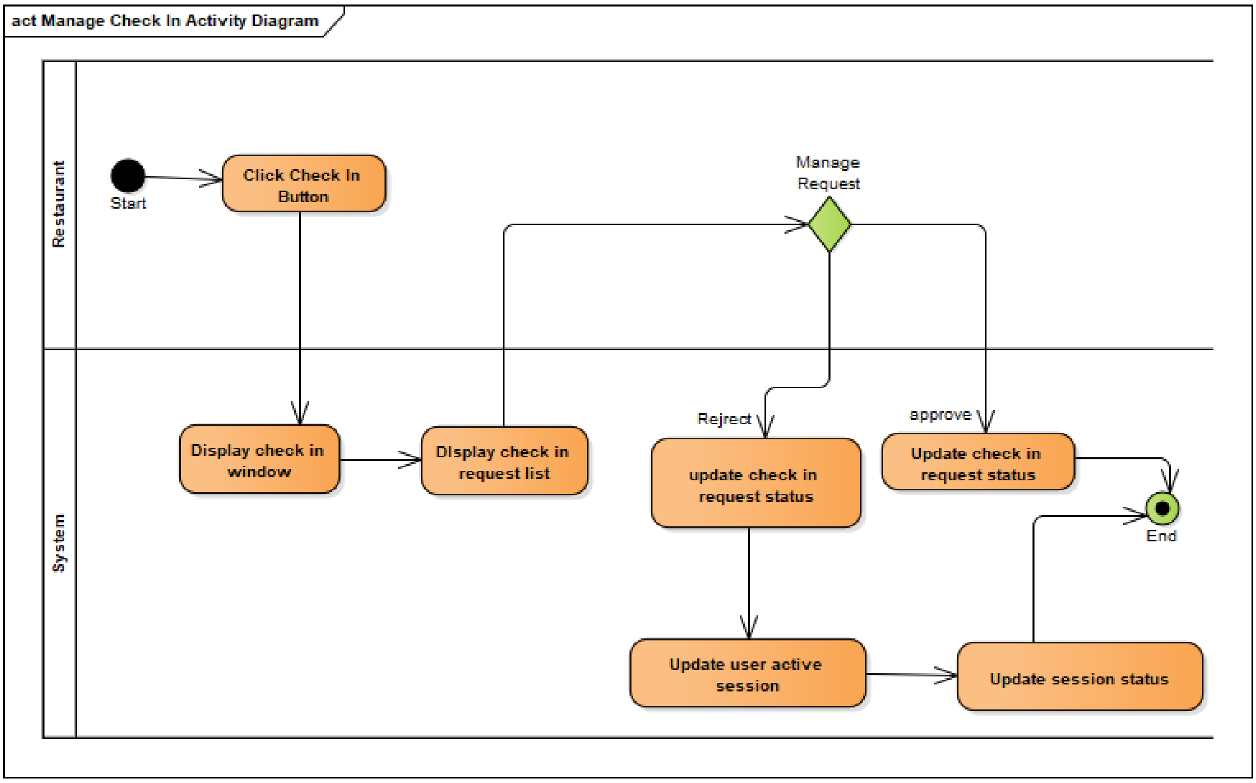
3.2.2.3 UC008 Manage Check In

Use Case ID	UC008
Use Case Name	Manage Check In
Actors	Restaurant
Trigger Event	Restaurant want to manage the user check in.
Brief Description	This use case is performed by restaurant to accept or reject the check in request after the user scan a QR code.
Pre-Condition	There is check in request.
Post-Condition	User check in request is updated.
Flow of events	1. Actor click Check in Button 2. System display check in window with a list of check in request. If no check in request in system, perform exception flow 2E. 3. If actor click “Accept” button, perform alternative flow 3A. If actor click “Reject” button, perform alternative flow 3B
Alternative Flow	3A: Accept check in 1. System set check in request status to ‘Accpeted’ 2. Use case end 3B: Reject Check In 1. System set check in request status to ‘Rejected’ 2. System update session status to ‘Closed’. 3. System update user active session. 4. Use case end.
Exception Flow	2E: No check in request in system 1. System display “No check in request” in check in window.

UCD008: Manage Check In



SD008: Sequence Diagram for UC008 Manage Check In


act Manage Check In Activity Diagram
Restaurant
System
Start
Click Check In Button
Manage Request
Display check in window
Dlsplay check in request list
Rejrect
approve
update check in request status
Update check in request status
End
Update user active session
Update session status

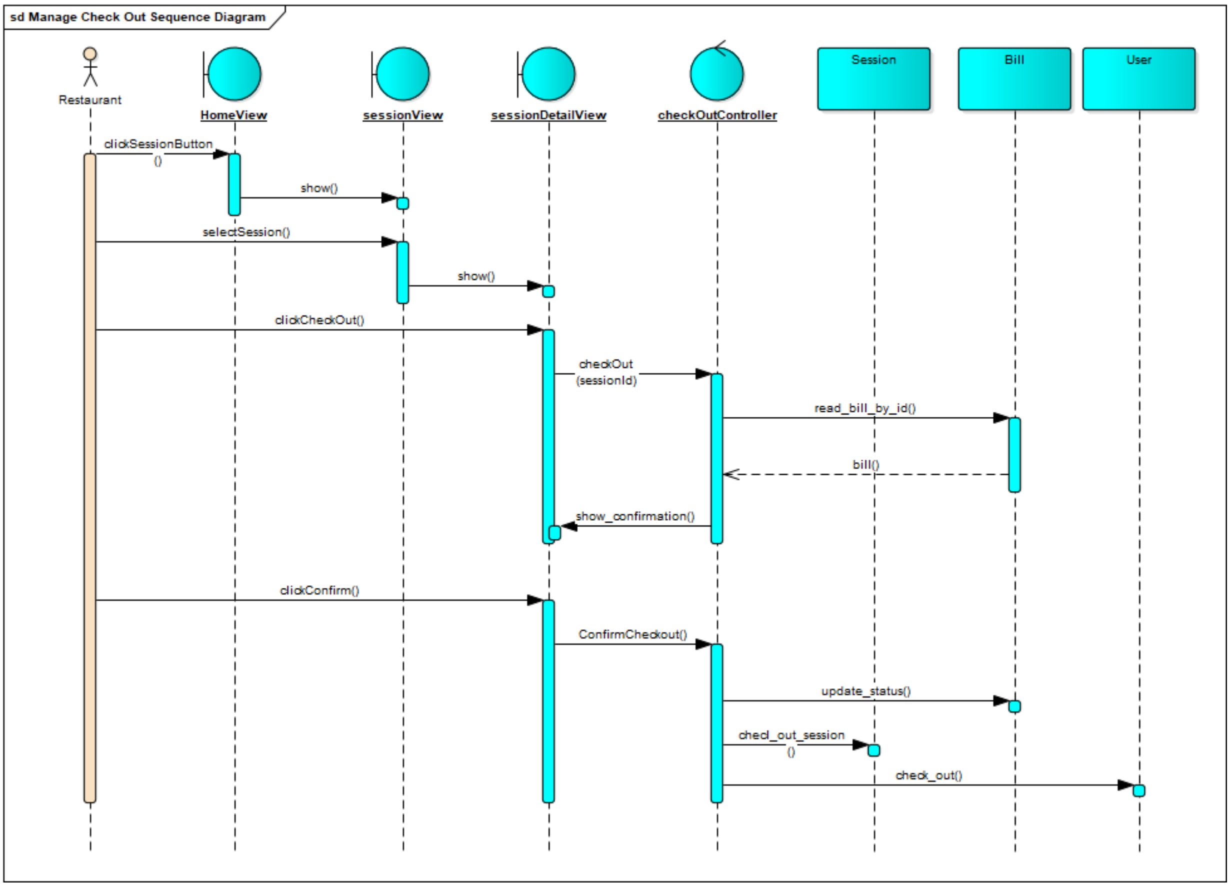
AD008: Activity Diagram for UC008 Manage Check In

3.2.2.4 UC009: Manage Check Out

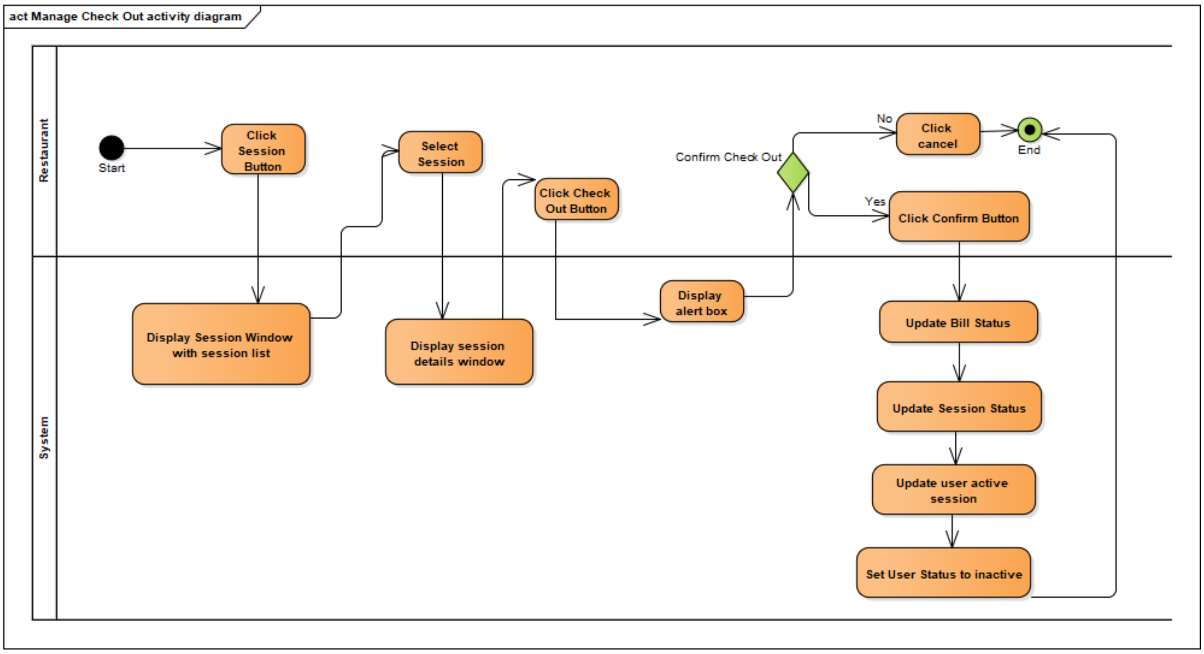
Use Case ID	UC009
Use Case Name	Manage Check Out
Actors	Restaurant
Trigger Event	Restaurant want to handler cash payment
Brief Description	This use case is performed by restaurant to check out the bill of the user who select cash payment. It can also be performed to check out paid but uncheck out user.
Pre-Condition	There session bill status is pending or solved.
Post-Condition	The session is close and user is checked out.
Flow of events	1. Actor select the session button 2. System display session window with a list of session 3. Actor select a session. 4. System display the details session window.

	5. Actor click check out button. 6. System display alert box with message “Pls make sure the customer has solved the payment before closing the session.” 7. Actor click confirm button 8. System update bill status 9. System update session status. 10. System update user active session
Alternative Flow	-
Exception Flow	5E: Check out Cancel 1. System display details session window. 2. Use case end.

UCD009: Manage Check Out



SD009: Sequence Diagram for UC009 Manage Check Out



AD009: Activity Diagram for UC009 Manage Check Out

3.2.3 Order Subsystem

This subsystem includes the functional requirement regarding the order and it related services after the user check in to the restaurant. The use case in this subsystem include Browse Menu, Add Item, Manage Cart, Call Waiter, Manage Menu, Manage Menu Item, Manage Customer Order, View Waiter Call List.

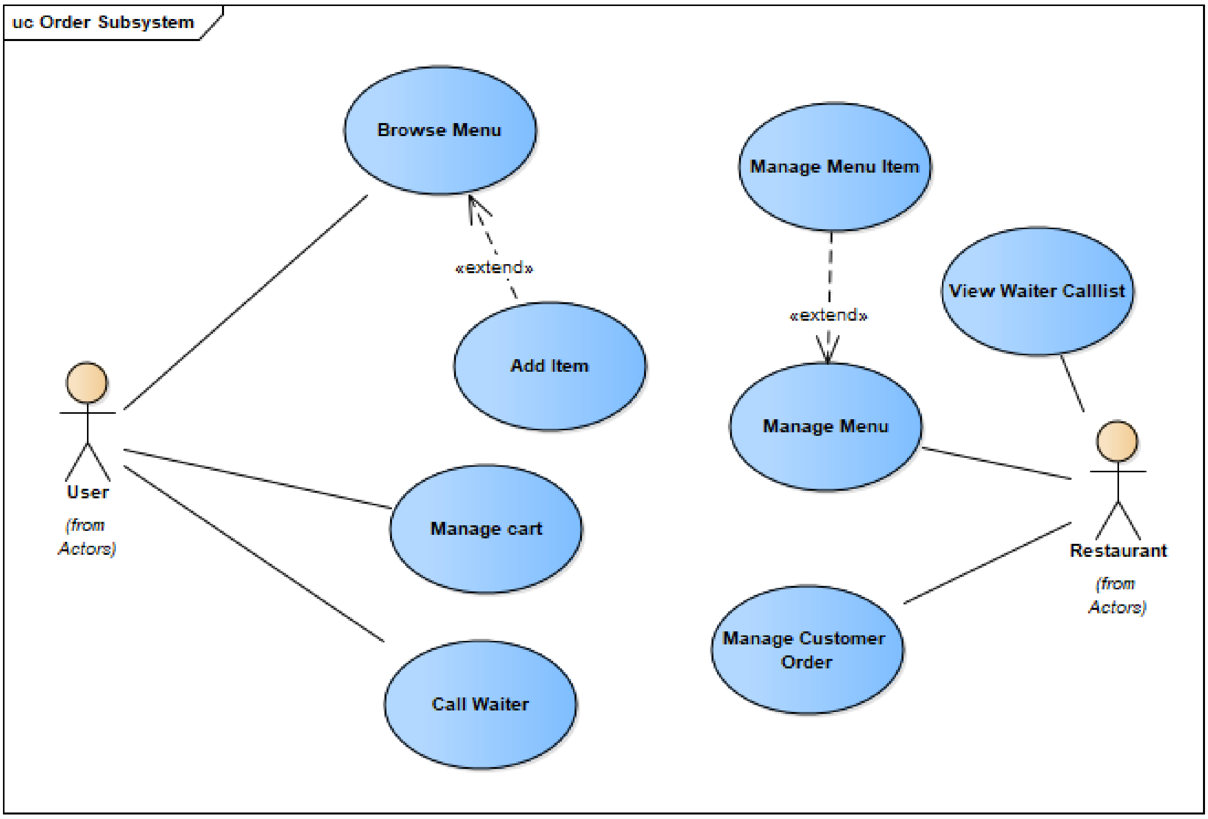


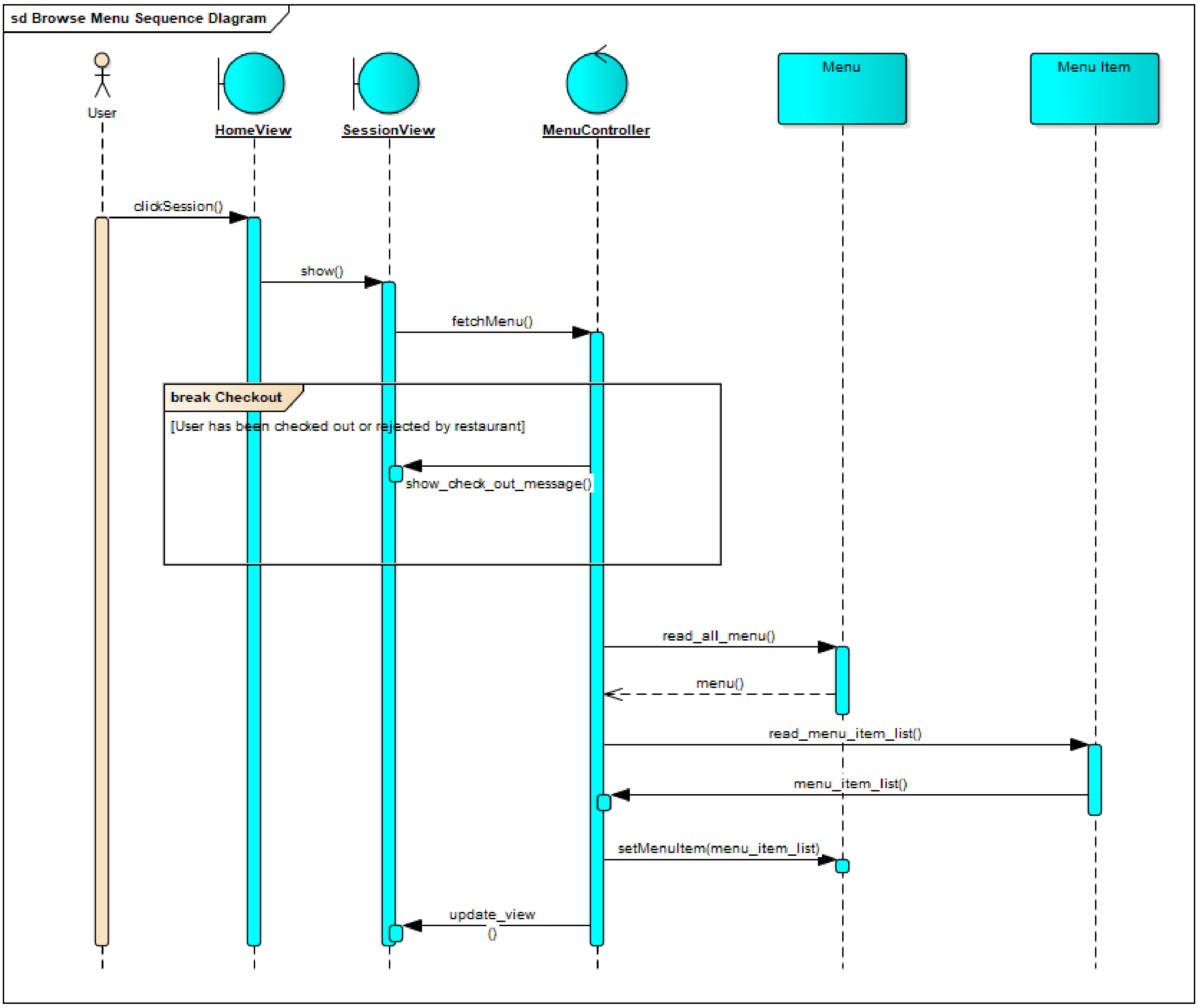
Figure 3.21: Use Case Diagram for Order Subsystem

3.2.3.1 UC0010: Browse Menu

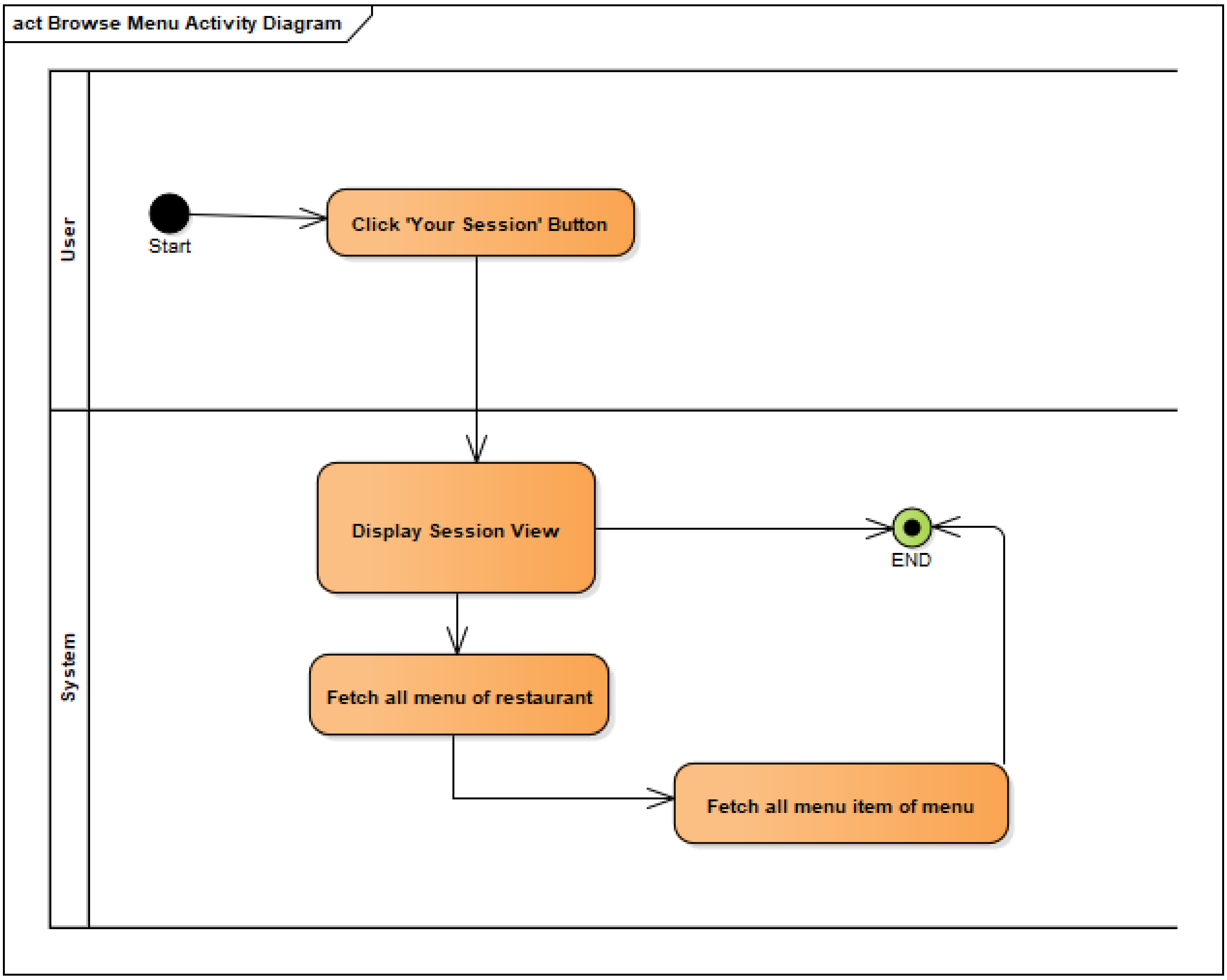
Use Case ID	UC0010
Use Case Name	Browse Menu
Actors	User
Trigger Event	The user wants to view the menu of the restaurant
Brief Description	This use case is performed by user when they want to view the menu of a restaurant. It has an extended use case called “Add Item”
Pre-Condition	Actor has checked in to a restaurant
Post-Condition	-

Flow of events	1. Actor clicks ‘Your Session’ button in the home view 2. System display session view 3. System fetch menu of restaurant 4. System fetch menu item of menu 5. System update session view with menu item list
Alternative Flow	-
Exception Flow	2E: User has been checked out or rejected by restaurant 1. System display error message “Your session has been rejected or checked out by the restaurant” 2. Use case end

UCD010: Use Case Description for Browse Menu



SD010: Sequence Diagram for UC010 Browse Menu


act Browse Menu Activity Diagram
User
System
Start
Click 'Your Session' Button
Display Session View
END
Fetch all menu of restaurant
Fetch all menu item of menu

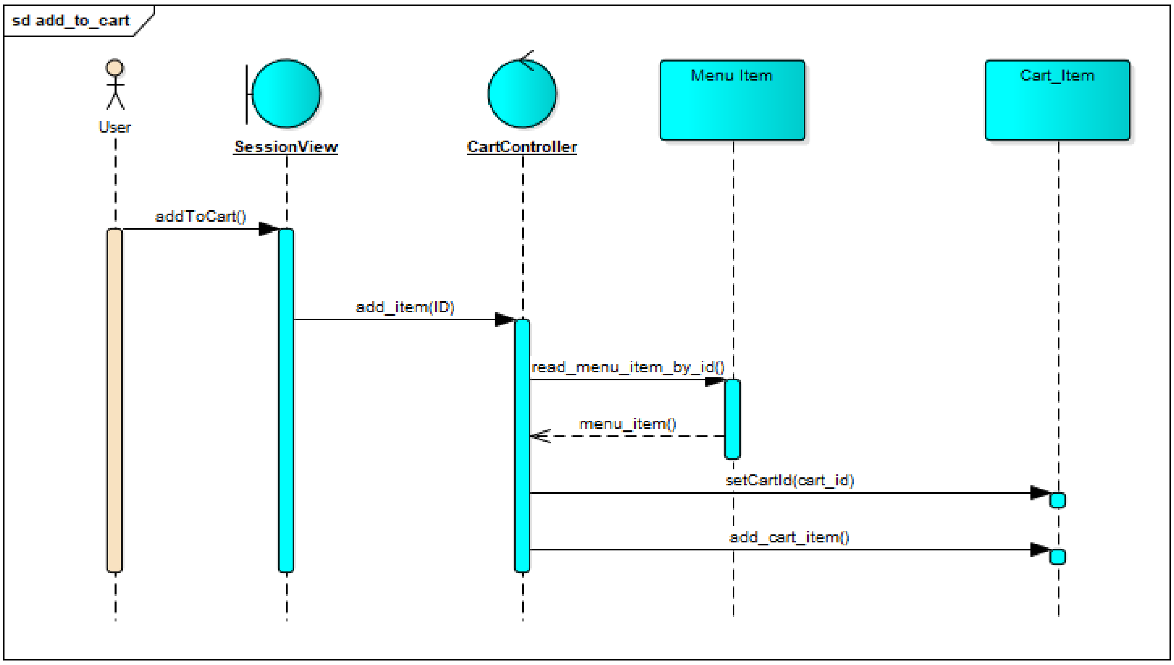
AD010: Activity Diagram for UC011 Browse Menu

3.2.3.2 UC011 Add Item

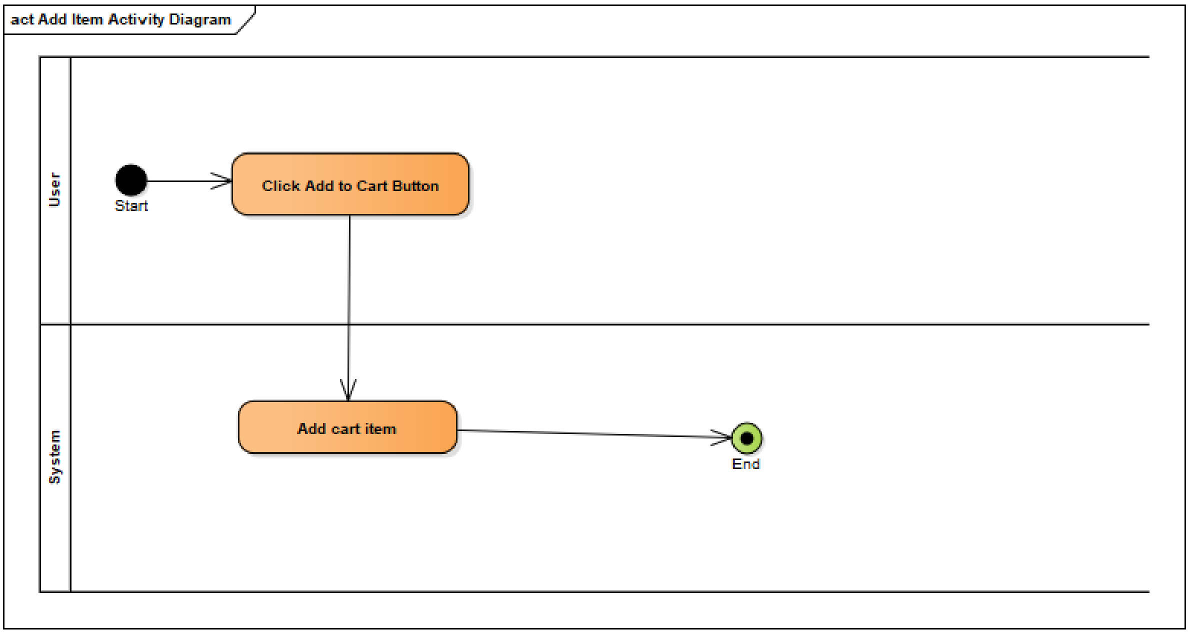
Use Case ID	UC011
Use Case Name	Add Item
Actors	user
Trigger Event	The user wants to add a food item to cart.
Brief Description	This is the extended use case of Browse Menu. It is performed when the user clicks “Add to Cart” button.
Pre-Condition	The user is checked in to a restaurant and use case Browse Menu is performed.
Post-Condition	Item is added to a cart.
Flow of events	1. User click ‘Add To Cart’ button.

	2. System add item to session cart.
Alternative Flow	-
Exception Flow	-

UCD011: Add Item



SD011: Sequence Diagram for UC011 Add Item



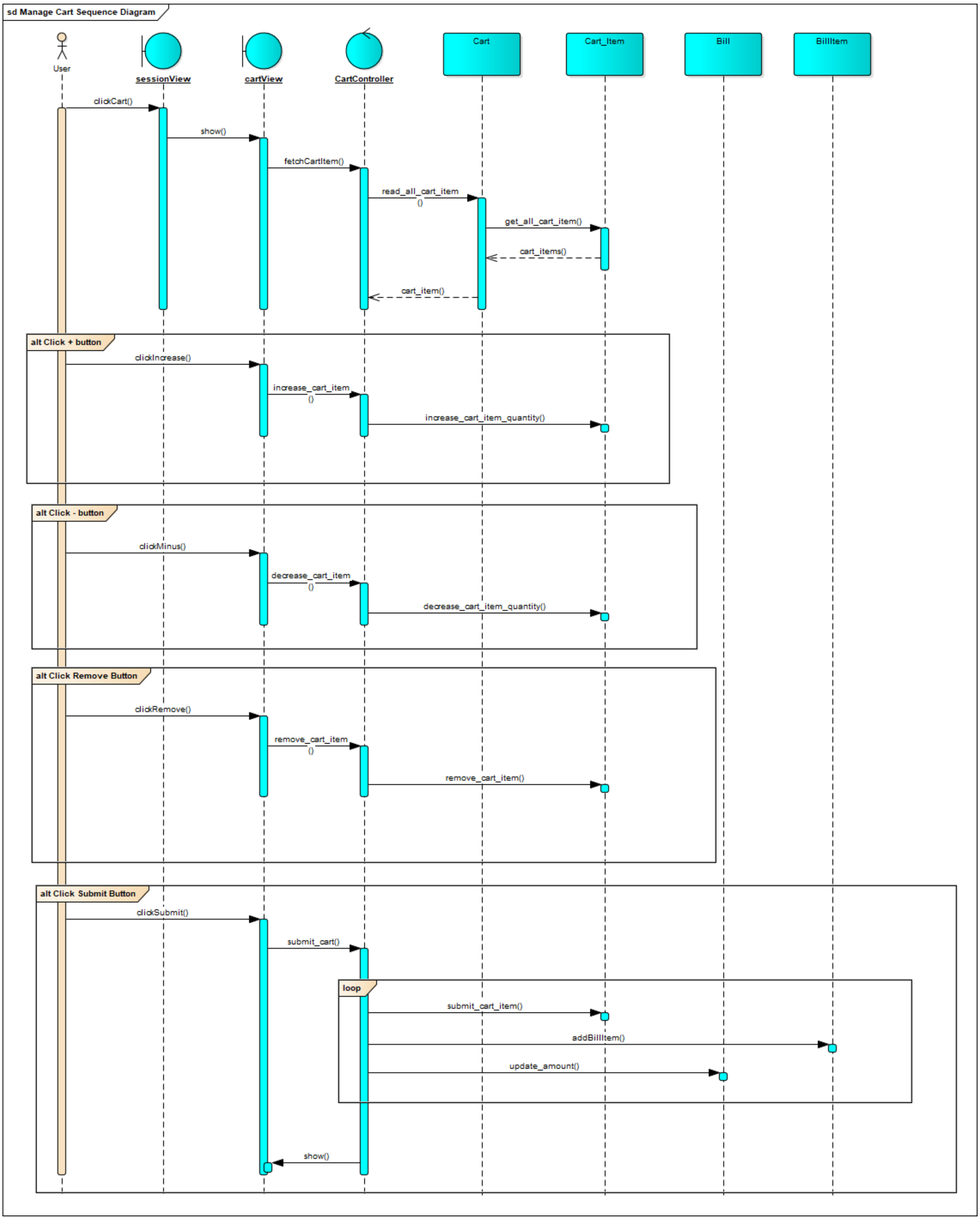
AD011: Activity Diagram for UC011 Add Item

3.2.3.3 UC012 Manage Cart

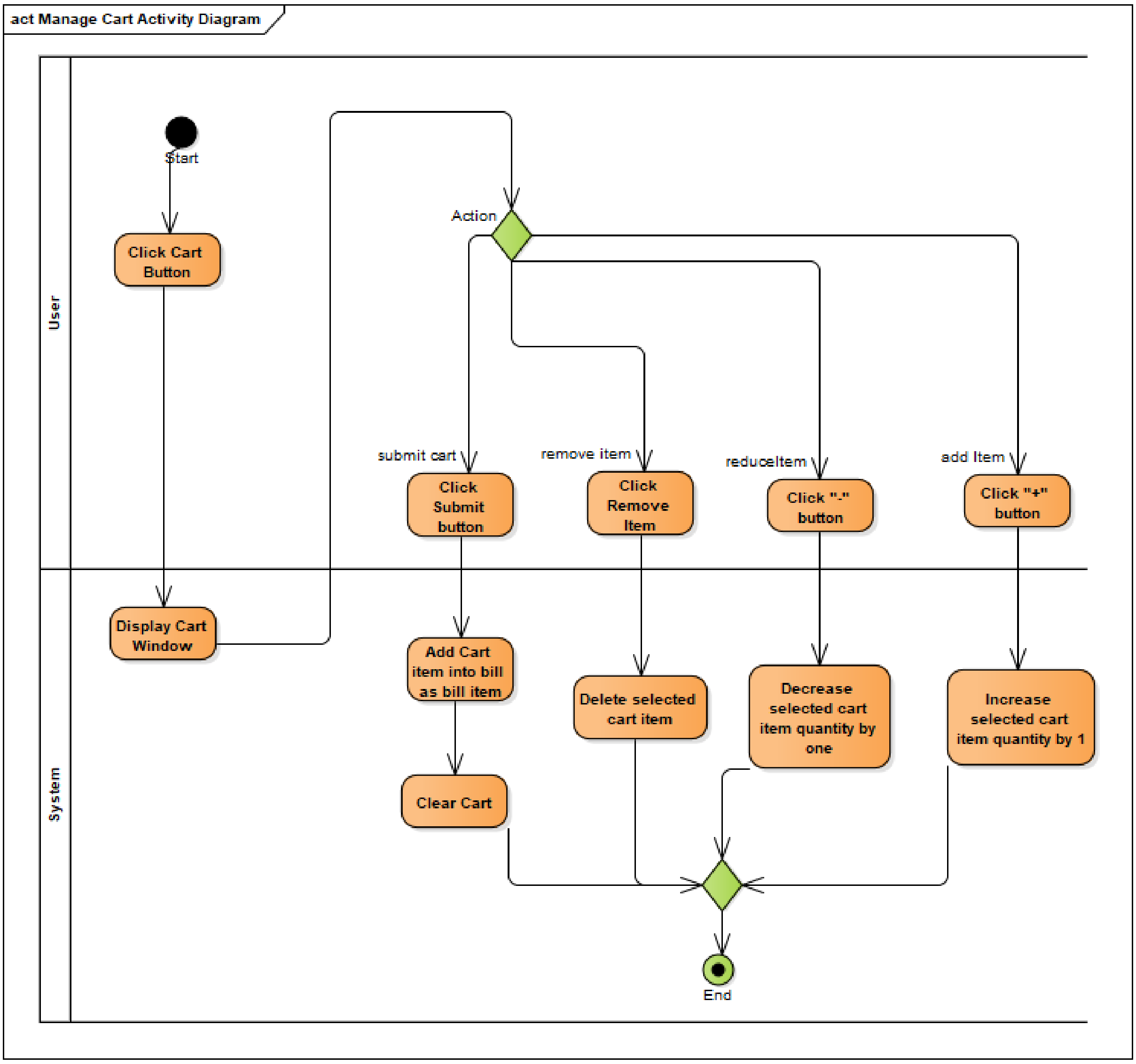
Use Case ID	UC0012
Use Case Name	Manage Cart
Actors	User
Trigger Event	The user wants to manage item in the cart
Brief Description	This use case is performed when the user wants to manage the information of the cart such as change the quantity of the item or remove the item from cart.
Pre-Condition	The user is checked in to a restaurant
Post-Condition	Cart Information is updated.
Flow of events	1. Actor clicks 'Your Order' Button in session window. 2. System display cart window. 3. If actor clicks "+" button of an item, perform alternative flow 3A. If actor clicks "-" button of an item, perform alternative flow 4A. If actor clicks "Remove" button, perform alternative flow 5A 4. If actor clicks "Submit" button, perform alternative flow 6A. If there is no item in cart, perform exception 6E.
Alternative Flow	3A: Add item 1. System update the item quantity to current quantity plus one. 2. Back to main flow. 4A:Reduce item 1. System update the item quantity to current quantity minus one. 2. Back to main flow 5A: Remove Item 1. System delete the item from the cart. 6A: Submit Cart 1. System convert the cart item to bill item. 2. System clear the cart.

	3. System notify restaurant.
Exception Flow	6E: No item in cart 1. System does nothing. 2. Back to main flow step 2.

UCD012: Manage Cart.



SD012: Sequence Diagram for UC012 Manage Cart.

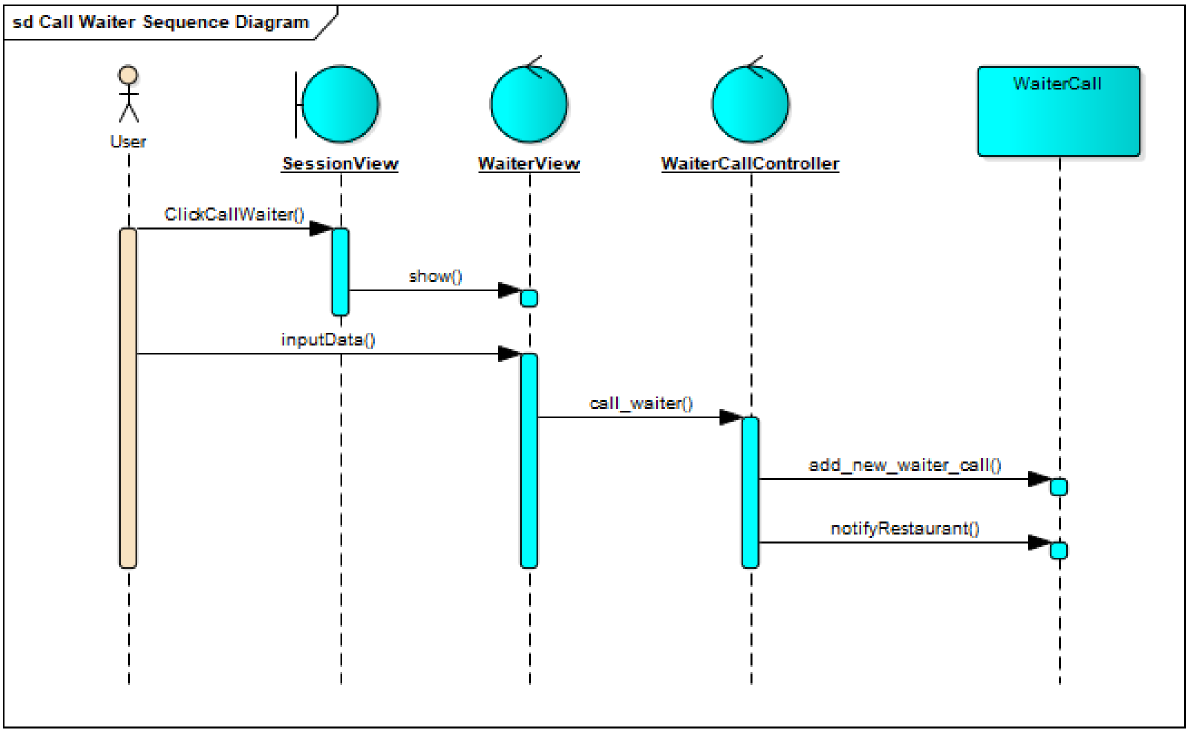


AD012: Activity Diagram for UC012 Manage Cart.

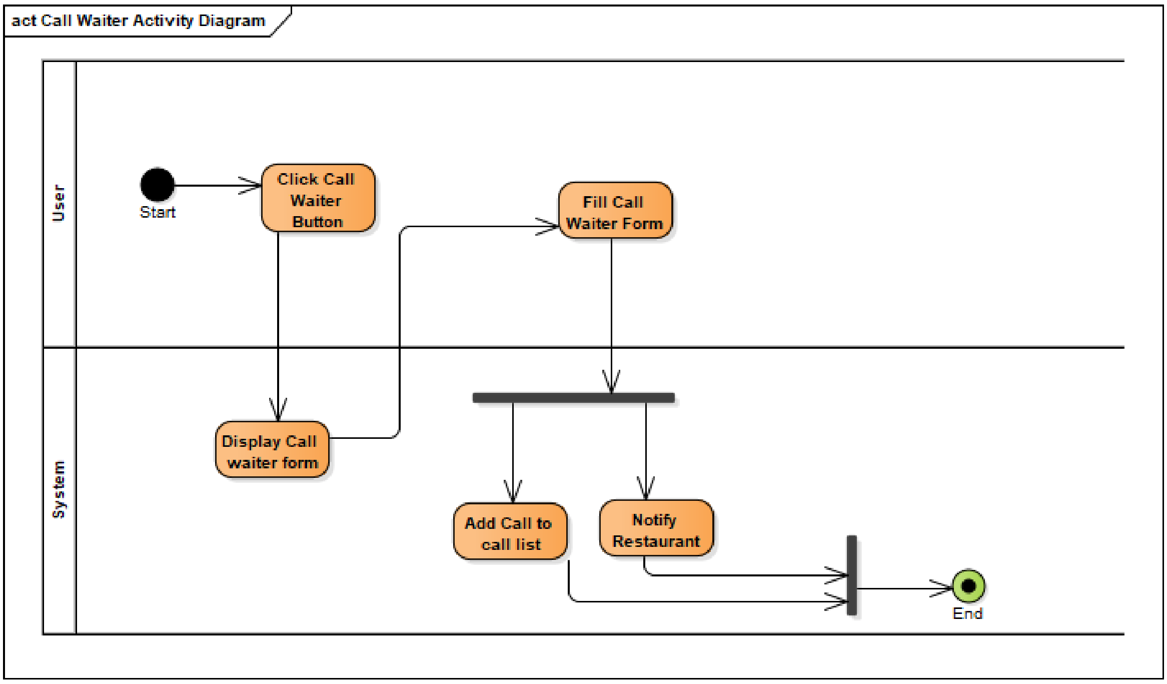
3.2.3.4 UC013 Call Waiter

Use Case ID	UC0013
Use Case Name	Call Waiter
Actors	User
Trigger Event	The User wants to call waiter for further enquiry.
Brief Description	This use case is performed by customer who want to ask for help from waiter.
Pre-Condition	The user is checked in to the restaurant
Post-Condition	Notification is added to the restaurant's waiter call list.
Flow of events	1. Actor click "Call Waiter" button 2. System display call waiter form 3. Actor fill in reason for calling waiter. 4. System add actor's call to the call list 5. System send notification to restaurant.
Alternative Flow	-
Exception Flow	-

UCD013: Call Waiter



SD013: Sequence Diagram for UC013 Call Waiter



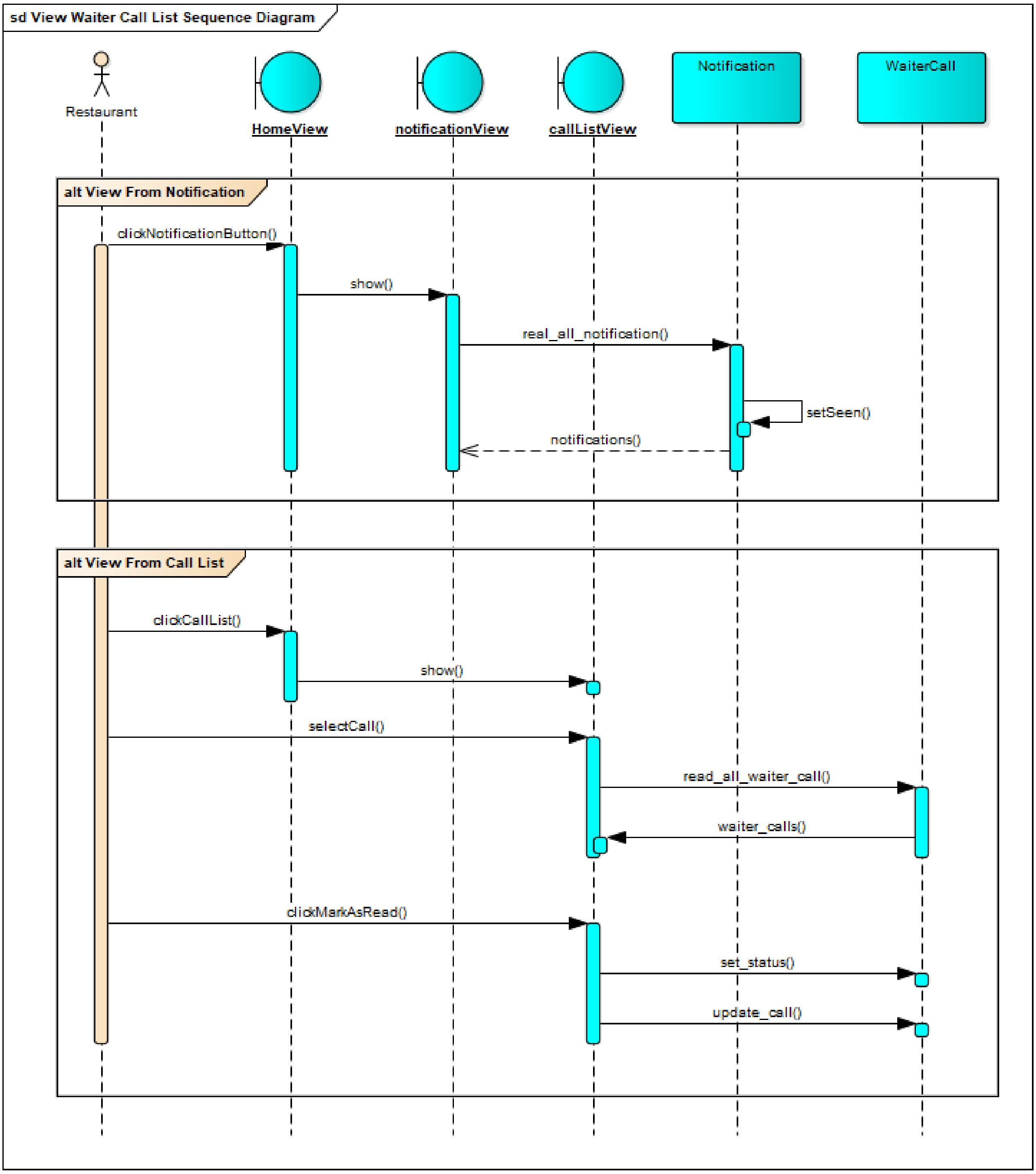
AD013: Activity Diagram for UC013 Call Waiter

3.2.3.5 UC014 View Waiter Call List

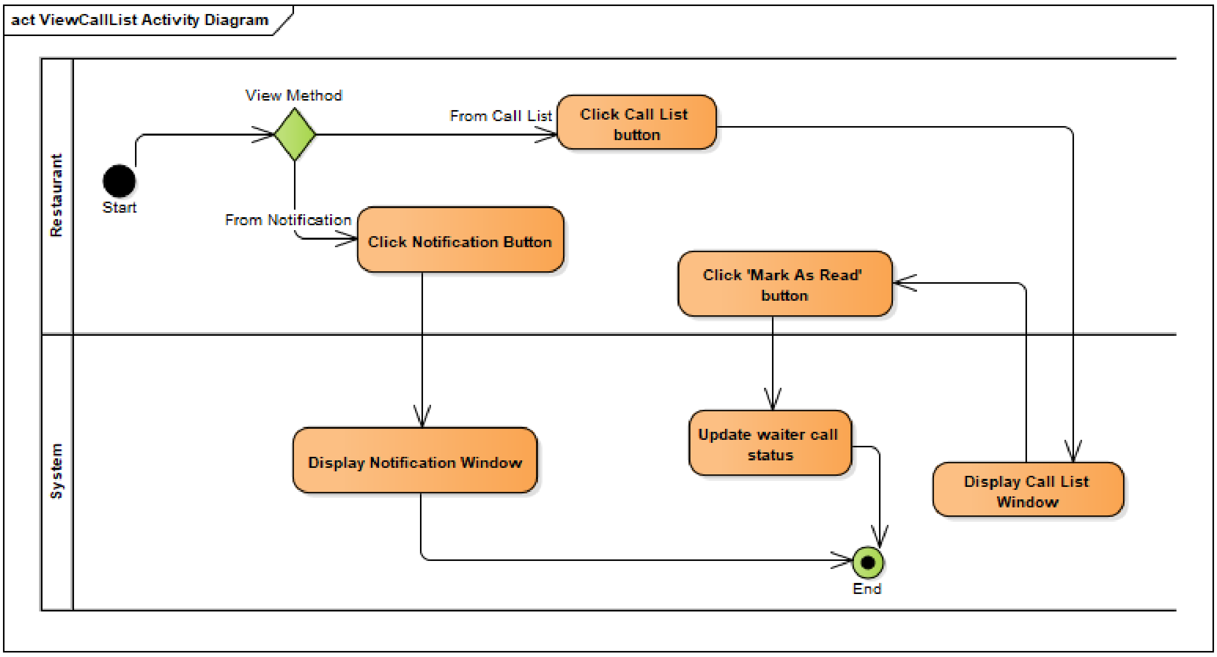
Use Case ID	UC0014
Use Case Name	View Waiter Call List
Actors	Restaurant
Trigger Event	The restaurant wants to view the call from the customer.
Brief Description	This use case is performed by restaurant to view or check the waiter call from customer.
Pre-Condition	The user is checked in to the restaurant
Post-Condition	Notification is added to the restaurant's waiter call list.
Flow of events	1. Actor click notification button. If actor clicks from call list, perform alternative flow 1A. 2. System display notification window with a list of notification including waiter call notification. 3. Use case end
Alternative Flow	1A: View from Call List 1. Actor click call list button 2. System display call list window with a list of waiter call list

	3. Back to main flow step 3
Exception Flow	-

UCD014: View Waiter Call List



SD014: Sequence Diagram for UC014 View Waiter Call List


act ViewCallList Activity Diagram
Restaurant
System
Start
View Method
From Call List
Click Call List button
From Notification
Click Notification Button
Click 'Mark As Read' button
Display Notification Window
Update waiter call status
Display Call List Window
End

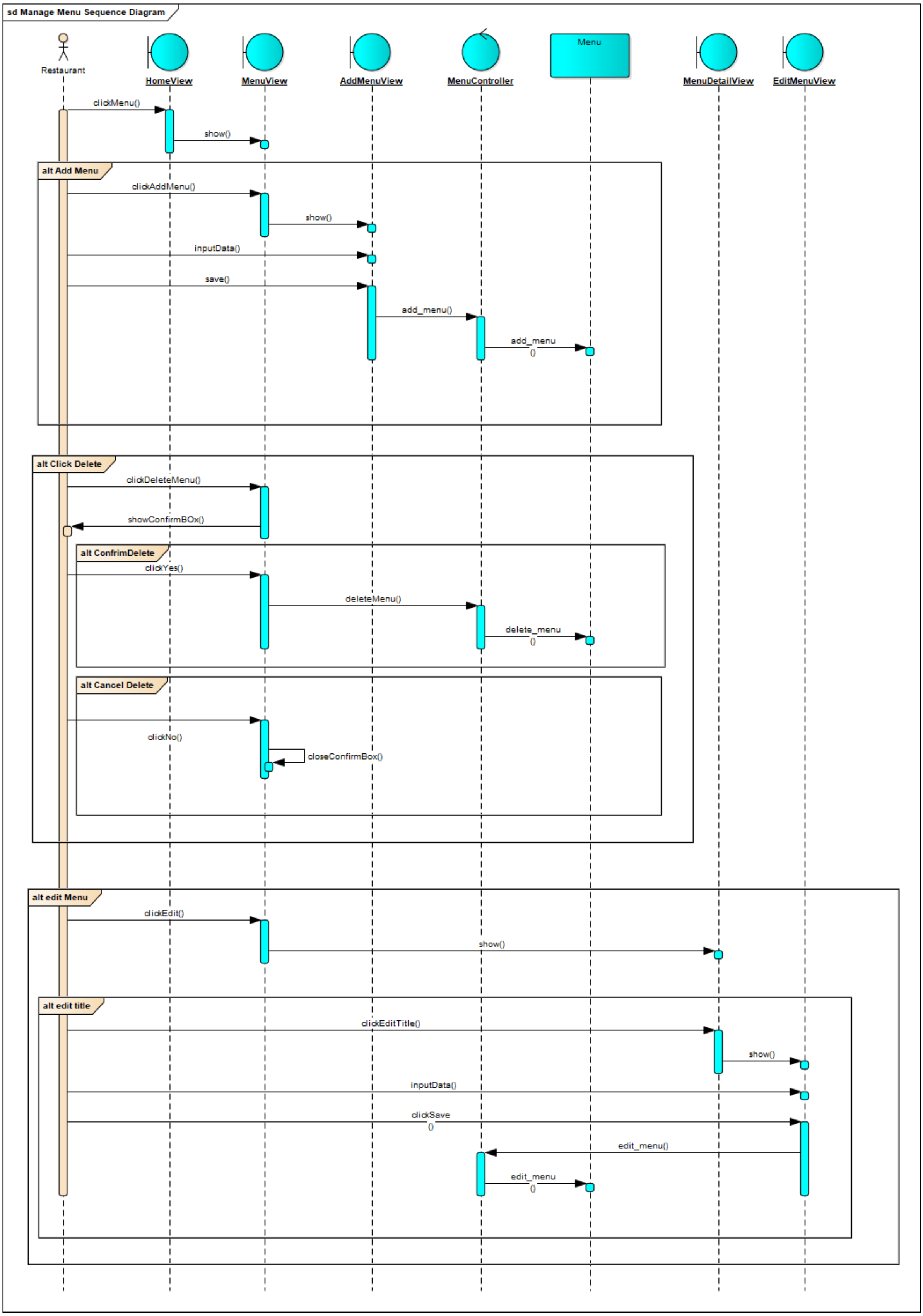
AD014: Activity Diagram for UC014 View Waiter Call List.

3.2.3.6 UC015 Manage Menu

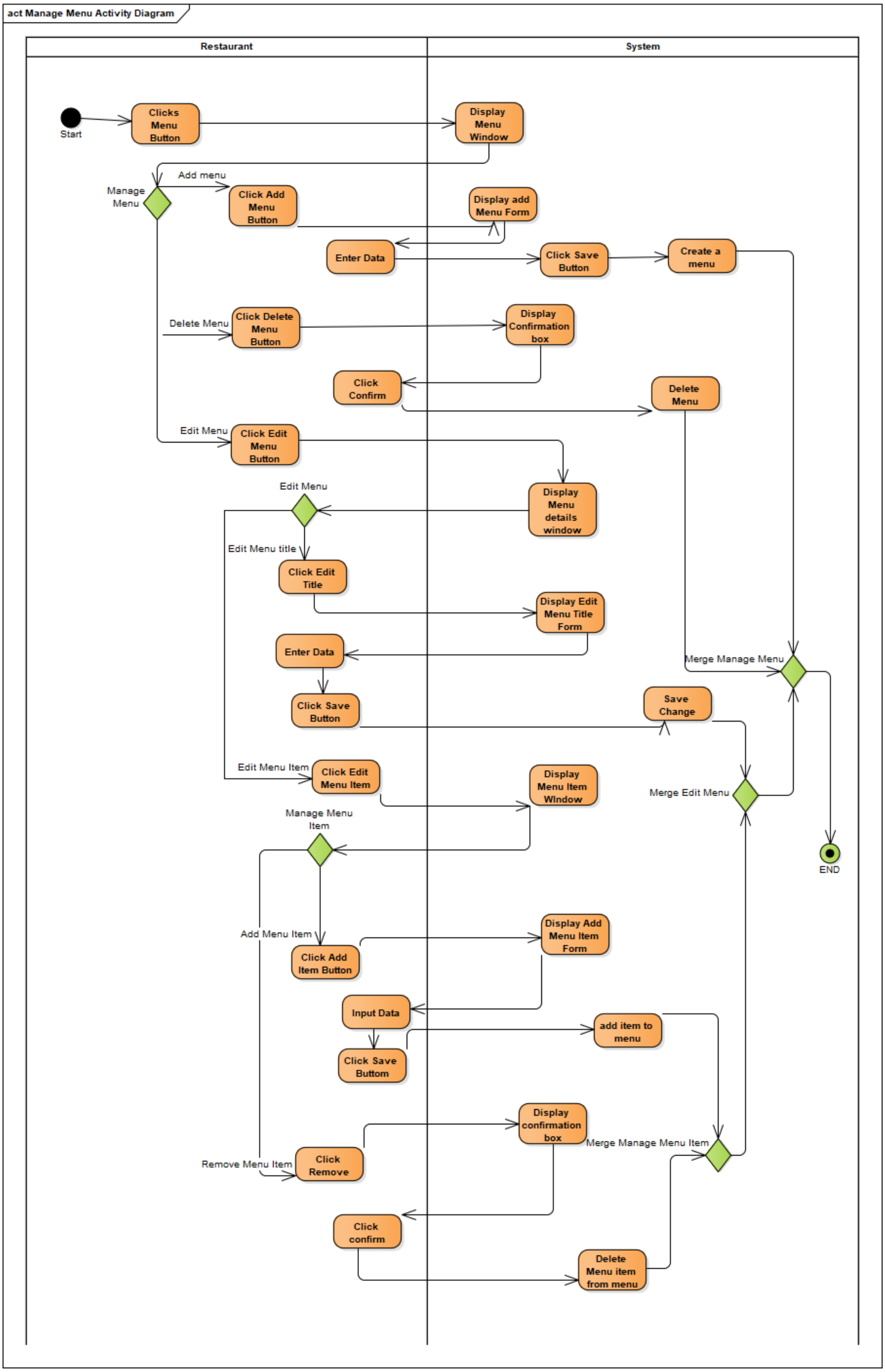
Use Case ID	UC015
Use Case Name	Manage Menu
Actors	Restaurant
Trigger Event	The restaurant wants manage the menu of the restaurant.
Brief Description	This use case is performed by restaurant to manage the menu such as create menu, add menu item, delete menu item and update menu item.
Pre-Condition	-
Post-Condition	Menu information updated.
Flow of events	1. Actor clicks menu button 2. System display menu window 3. If actor click “Add Menu” button, perform alternative flow 3A. If actor click “Edit” button, perform alternative flow 4A.If actor click “Delete button, perform alternative flow 5A.
Alternative Flow	3A: Add Menu 1. System display add menu form

	2. Actor fill in menu title. 3. Actor clicks save. 4. System create a menu. 5. Back to main flow step 2. 4A: Edit Menu 1. System display the menu details window. 2. If actor click edit menu title, perform alternative flow 4A1. If actor click ‘Add Item’ button or ‘Edit Item’ button or ‘Remove Item’ button, perform extended use case Manage Menu Item. 3. Back to main flow step 2. 4A1:Edit Menu Title 1. System display edit menu title form 2. Actor key in data 3. Actor click “Save” button. 4. System save the changed. 5. Back to main flow step 2. 5A: Delete Menu 1. System display delete confirmation box. 2. Actor clicks confirm. If actor clicks cancel, perform exception 5E. 3. System delete Menu. 4. Back to main flow step 2.
Exception Flow	5E: Cancel delete menu 1. Back to main flow step 2.

UCD015: Manage Menu



SD015: Sequence Diagram for UC015 Manage Menu



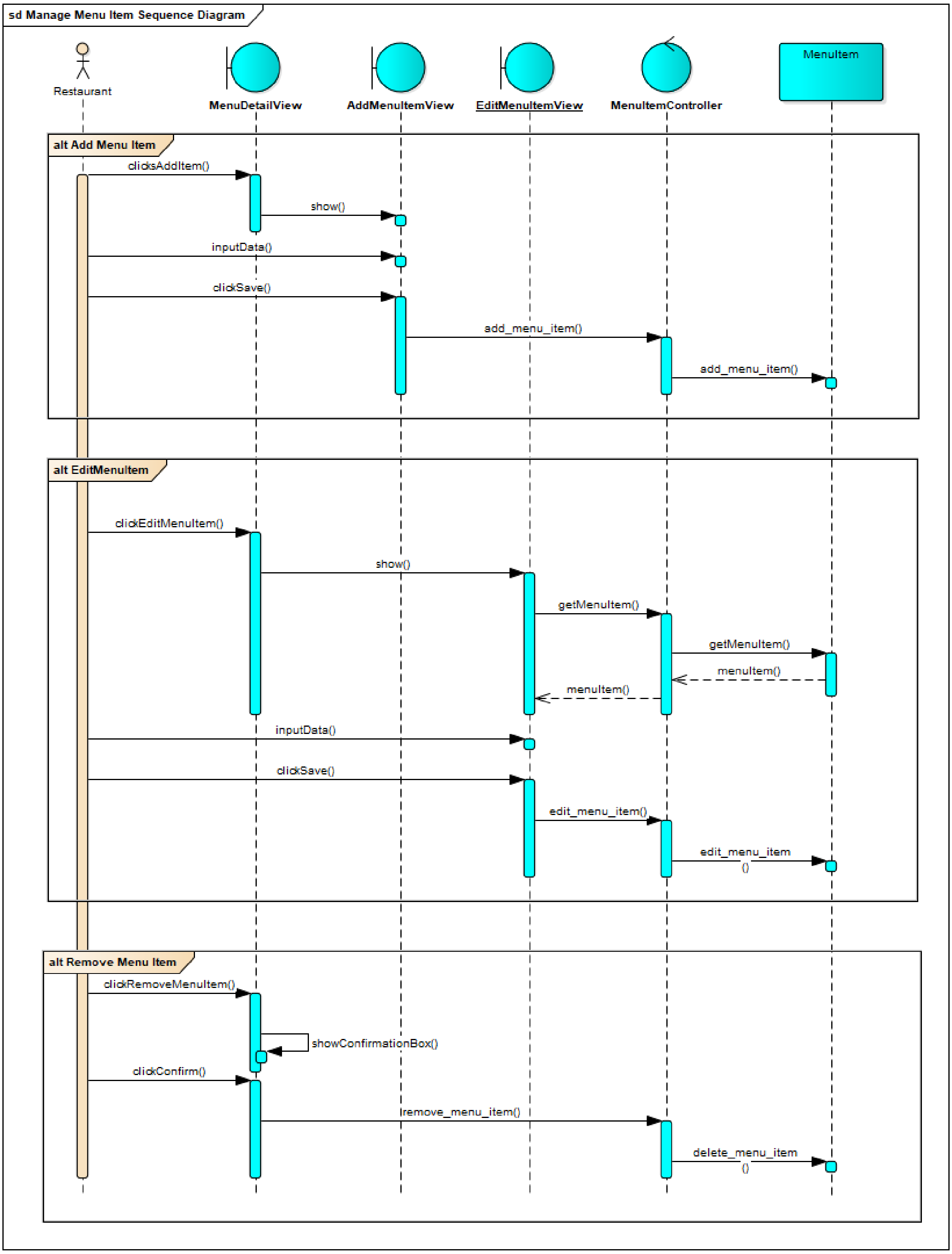
AD015: Activity Diagram for UC015 Manage Menu

3.2.3.7 UC016 Manage Menu Item

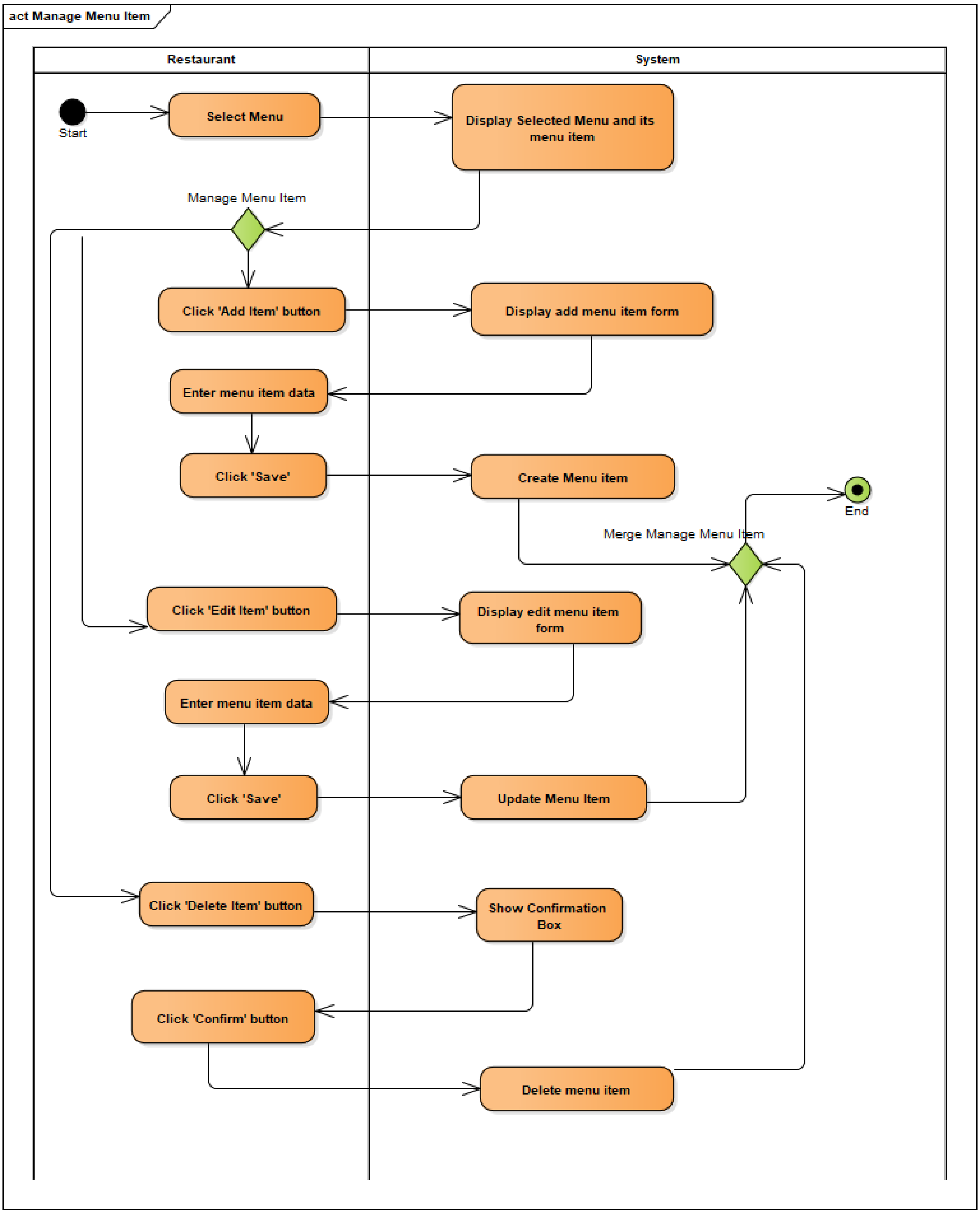
Use Case ID	UC016
Use Case Name	Manage Menu Item
Actors	Restaurant
Trigger Event	The restaurant want to add menu item to a menu.
Brief Description	This use case is performed when the restaurant clicks edit menu item button in menu window. It allow the restaurant to add, edit and remove menu item from a menu.
Pre-Condition	A menu has been selected and the edit menu item button is clicks.
Post-Condition	Menu item of the menu selected is updated.
Flow of events	1. System display menu detail window 2. If actor clicks "Add Item" button, perform alternative flow 2A. If actor clicks "Edit item" button, perform alternative flow 3A. If actor clicks "Remove Item" button, perform alternative flow 4A.
Alternative Flow	2A: Add Menu Item 1. System display add menu item form 2. Actor input the data including menu item name, menu item price, menu item ingredient, menu item allergy, menu item description. 3. Actor clicks save. 4. System add created menu item to menu. 5. Back to main flow step 1. 3A: Edit Menu Item 1. System display edit menu item form 2. Actor input the data. 3. Actor clicks save. 4. System updated menu item data 5. Back to main flow step 1 4A: Remove Menu Item 1. System display delete confirmation box.

	2. Actor clicks confirm. If actor clicks cancel, perform exception flow 4E. 3. System remove menu item from menu 4. Back to main flow step 1.
Exception Flow	4E: Remove Cancel 1. Back to main flow step 1.

UCD016: Use Case Description for Manage Menu Item



SD016: Sequence Diagram for UC017 Manage Menu Item



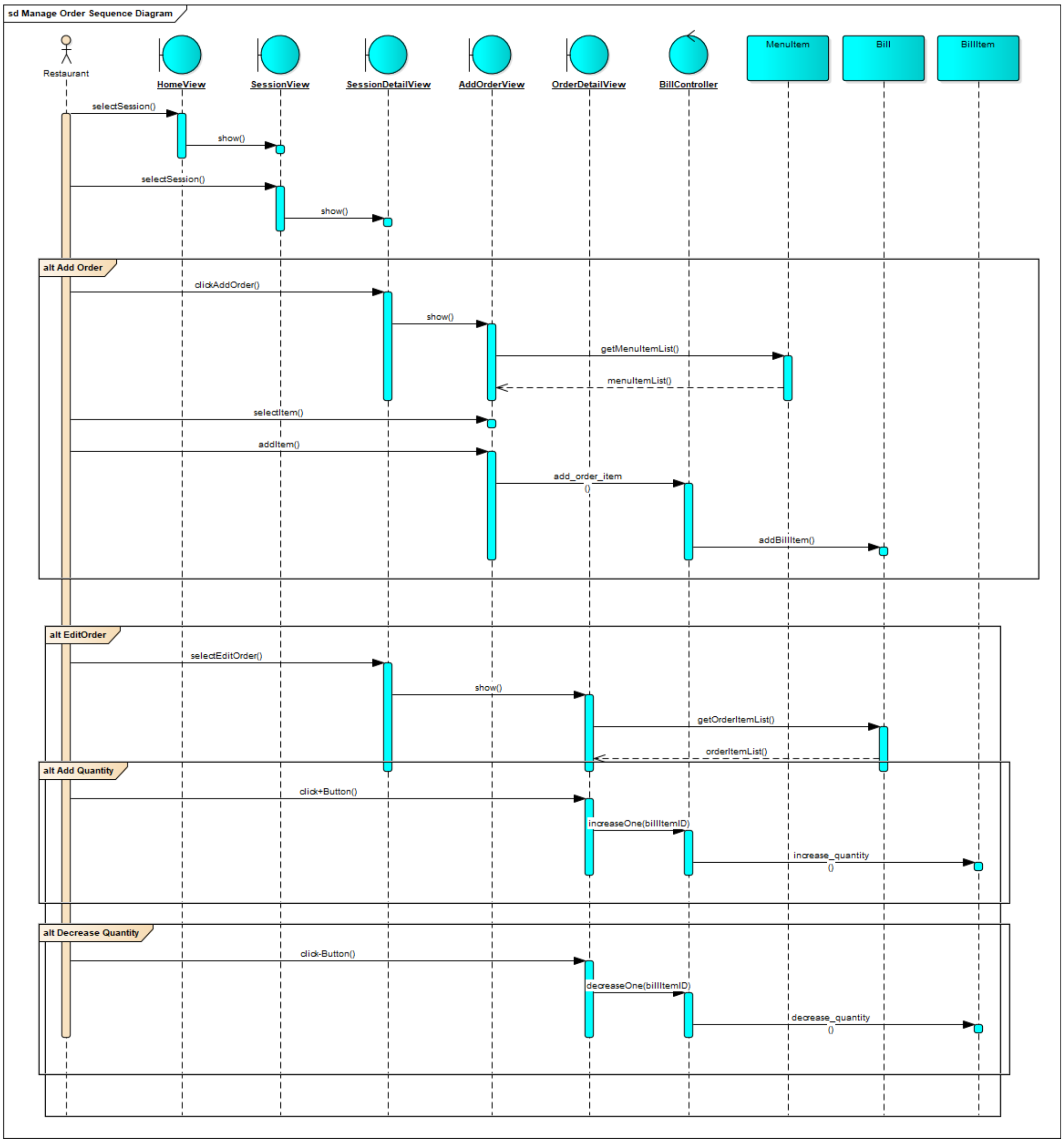
AD016: Activity Diagram for UC016 Manage Menu Item.

3.2.3.8 UC017 Manage Order

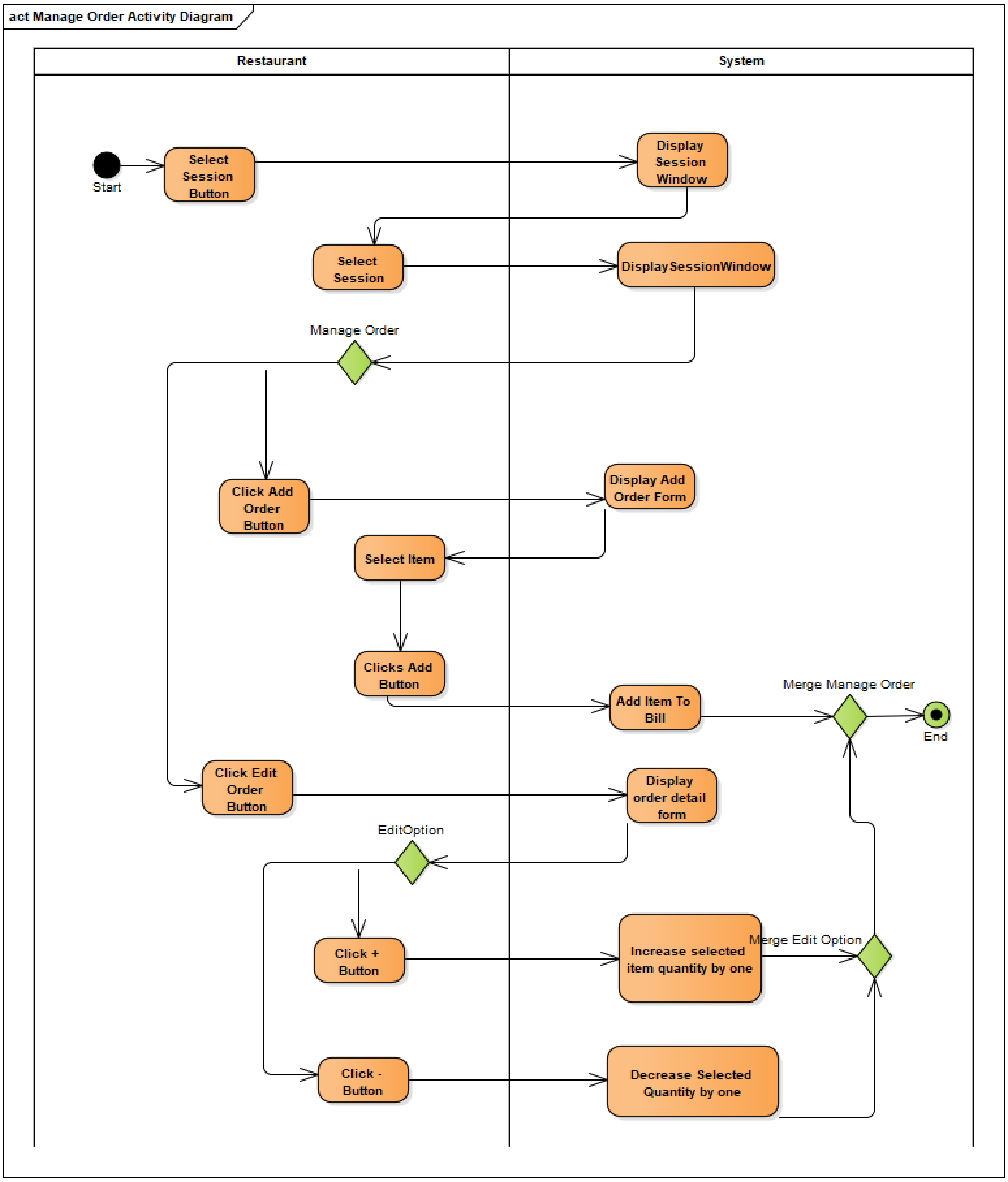
Use Case ID	UC017
Use Case Name	Manage Order
Actors	Restaurant
Trigger Event	The restaurant wants to add order to the session from its side.
Brief Description	This use case is performed by the restaurant to add order to a session.
Pre-Condition	The session is active session
Post-Condition	The session's bill is updated.
Flow of events	1. Actor click 'Session' button 2. System display session window with a list of session 3. Actor select a session. 4. System display the session details window with the sessions bill item. 5. If actor click "Add order" button , perform alternative flow 5A.If actor click "Edit order" button, perform alternative flow 6A
Alternative Flow	5A: Add order 1. System display restaurant add order form. 2. Actor select the item to add. 3. Actor input the quantity of the item to add. 4. Actor clicks add item button 5. System add the item to bill. 6. Back to main flow step 4. 6A: Edit order 1. System display order detail form. 2. If actor click "+" button of the order item, perform alternative flow 6A1.If actor click "-" button of the order item, perform alternative flow 6A2. If actor click "Remove" button of the order item, perform alternative

	flow 6A3. 3. Actor clicks ‘Exit Edit Order’ button 4. System display session detail window 6A1: Add Quantity 1. System increase the quantity of the selected item by one. 2. Back to alternative flow 6A step 1 6A2: Reduce Quantity 1. System decrease the quantity of the selected item by one. 2. Back to alternative flow 6A step 1. 6A3:Remove order item 1. System display confirmation box. 2. Actor clicks confirm. If actor clicks cancel, perform exception flow 6E. 3. System remove the item from bill 4. Back to alternative flow 6A step 1.
Exception Flow	6E: Remove Cancel 1. Actor clicks cancel button 2. Back to alternative flow 6A step 1.

UCD017: Manage Order



SD017: Sequence Diagram for UC017 Mange Order.



AD017: Activity Diagram for UC017 Manage Order.

3.2.4 Reporting Subsystem

This subsystem includes the functional requirement regarding the reporting in user and restaurant side. The use cases included in this subsystem are View Consumption, View Visited Restaurant, Add Feedback, View Feedback, Reply Feedback and Generate Report.

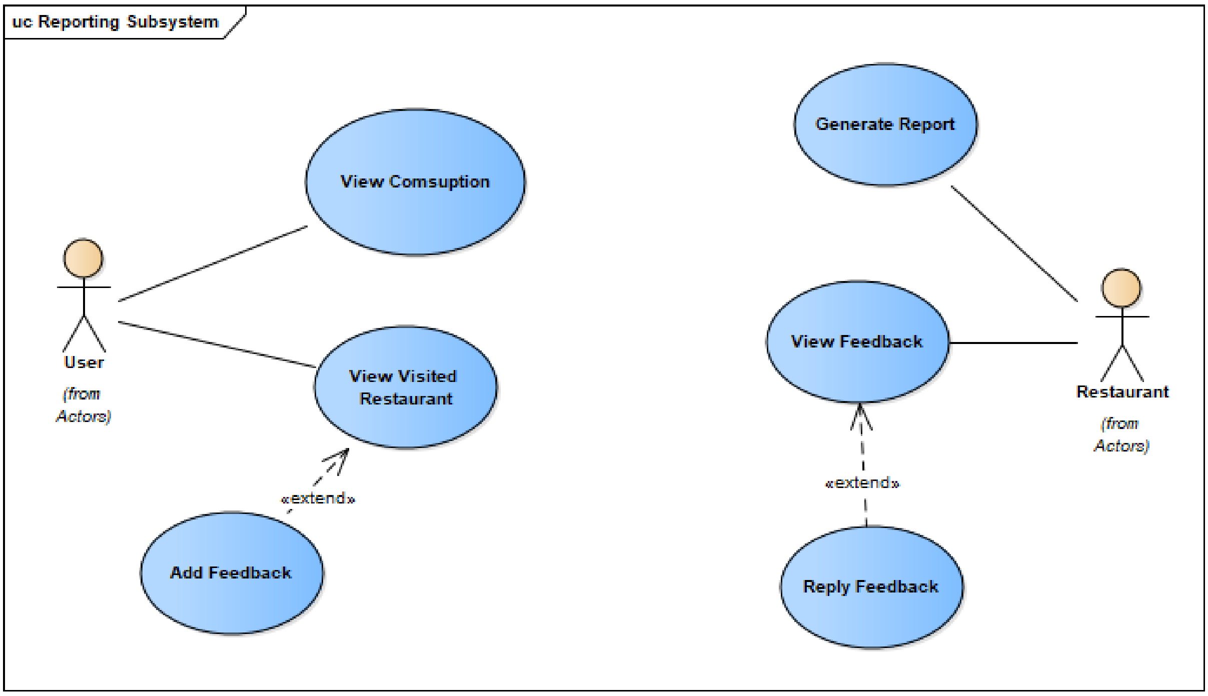


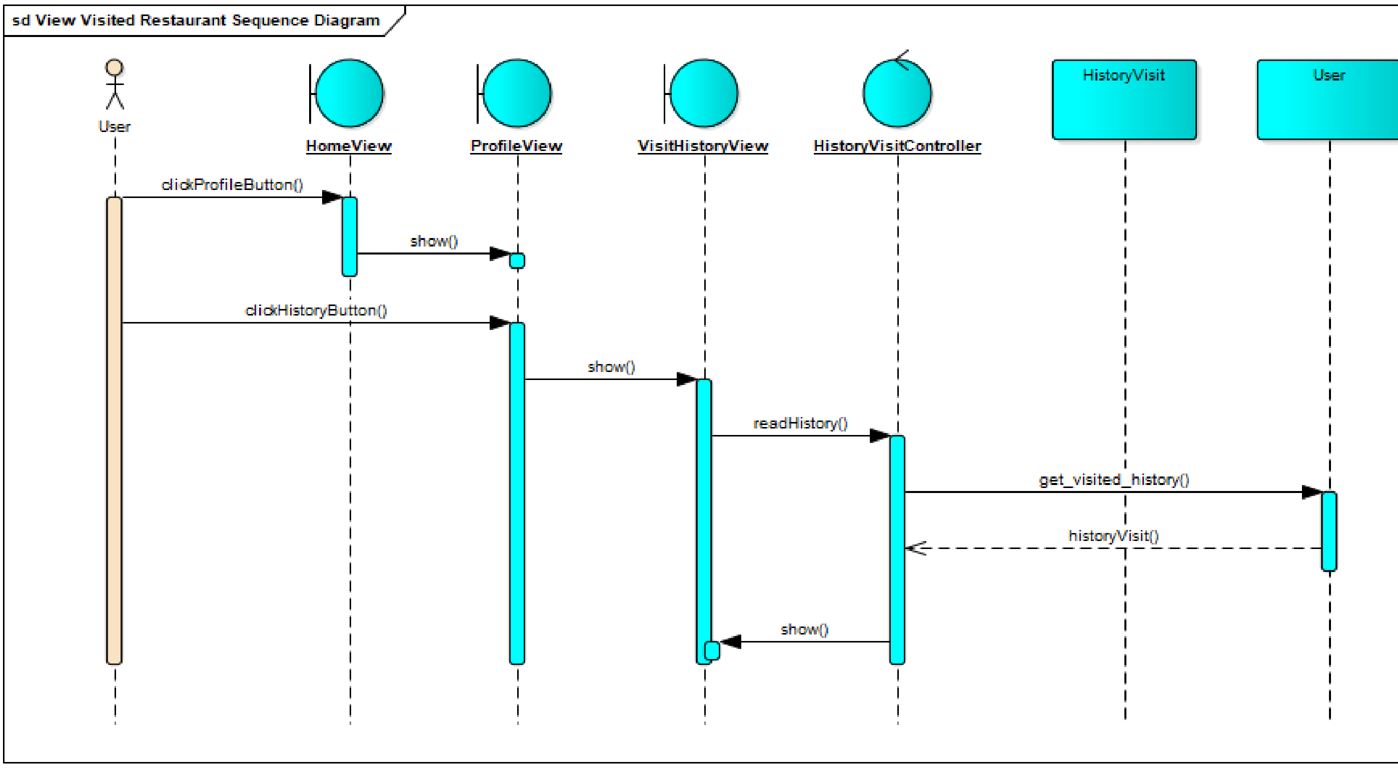
Figure 3.23: Use Case Diagram for Reporting Subsystem

3.2.4.1 UC018 View Visited Restaurant

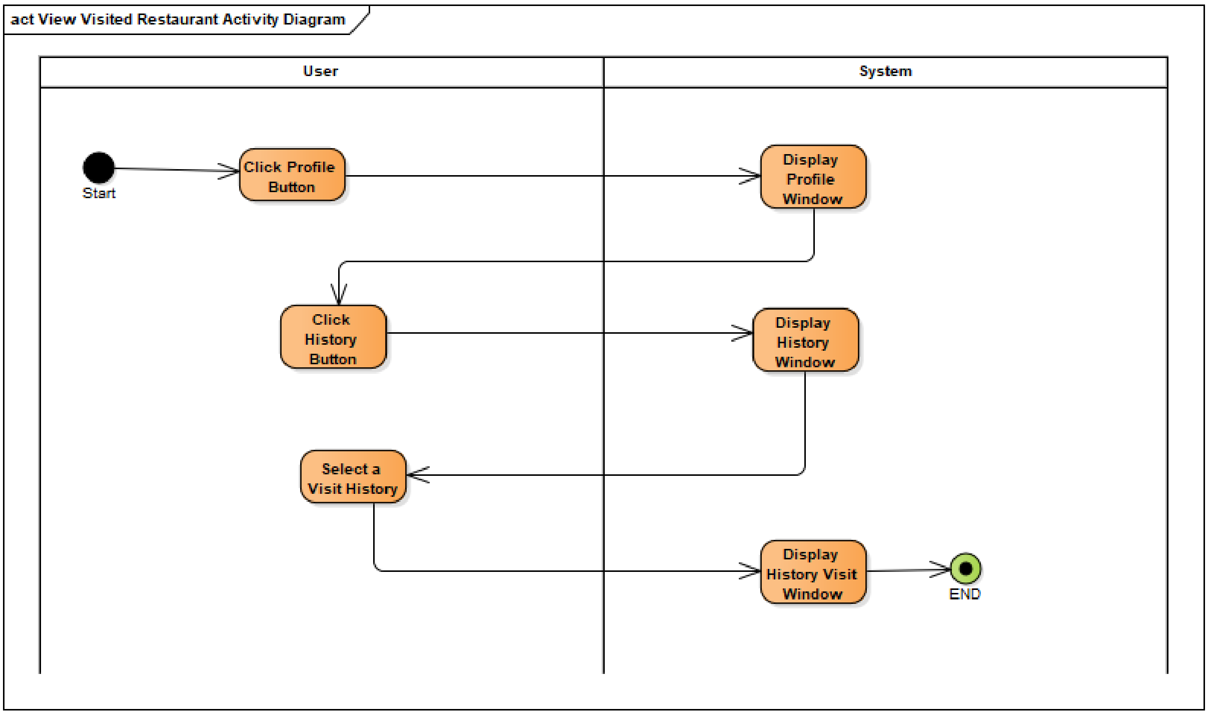
Use Case ID	UC018
Use Case Name	View Visited Restaurant
Actors	User
Trigger Event	The customer wants to know the restaurant he visited before.
Brief Description	This use case is performed by customer who want to ask for help from waiter. This use case has an extended use case which is “Add Feedback”
Pre-Condition	The user is checked in to the restaurant
Post-Condition	Notification is added to the restaurant’s waiter call list.
Flow of events	1. Actor click “Profile” button. 2. System display profile window 3. Actor click “History” Button 4. System display history window with the list of

	restaurants the user visited arranged by time ascending. If there is no history in system, perform exception flow 4E. 5. Actor select a visit history. 6. If actor click add feedback button, perform extended use case “Add Feedback” If the visited date exceeds 3 months, perform exception flow 7E.
Alternative Flow	-
Exception Flow	4E: No history restaurant. 1. System display “No history” in the history window and use case end. 7E: Visited Date exceeds 3 months 1. System display message “This visit exceeds 3 months, you are not allowed to add feedback.” and use case end.

UCD018: View Visited Restaurant



SD018: Sequence Diagram for UC018 View Visited Restaurant



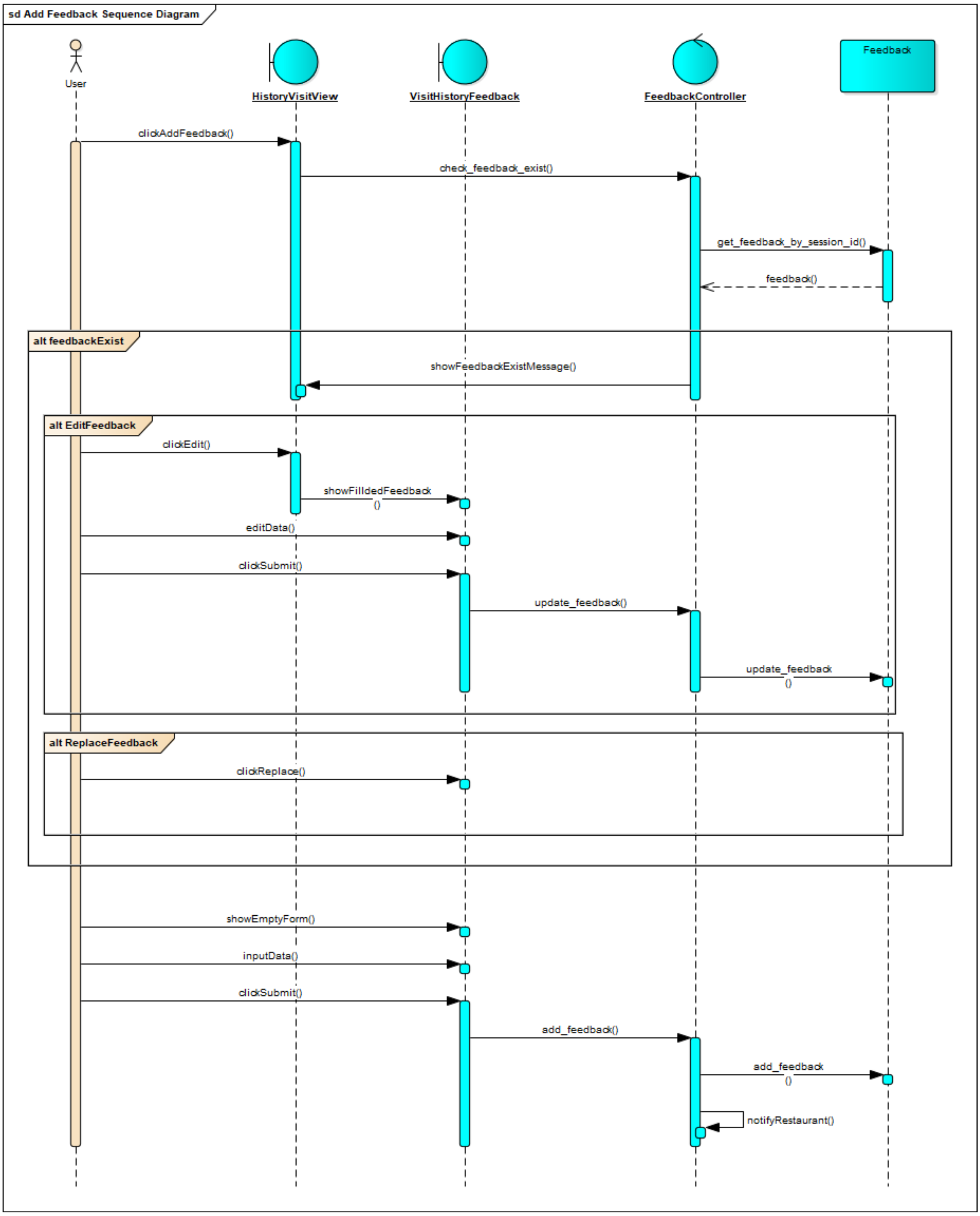
AD018: Sequence Diagram for UC018 View Visited Restaurant

3.2.4.2 UC019 Add Feedback

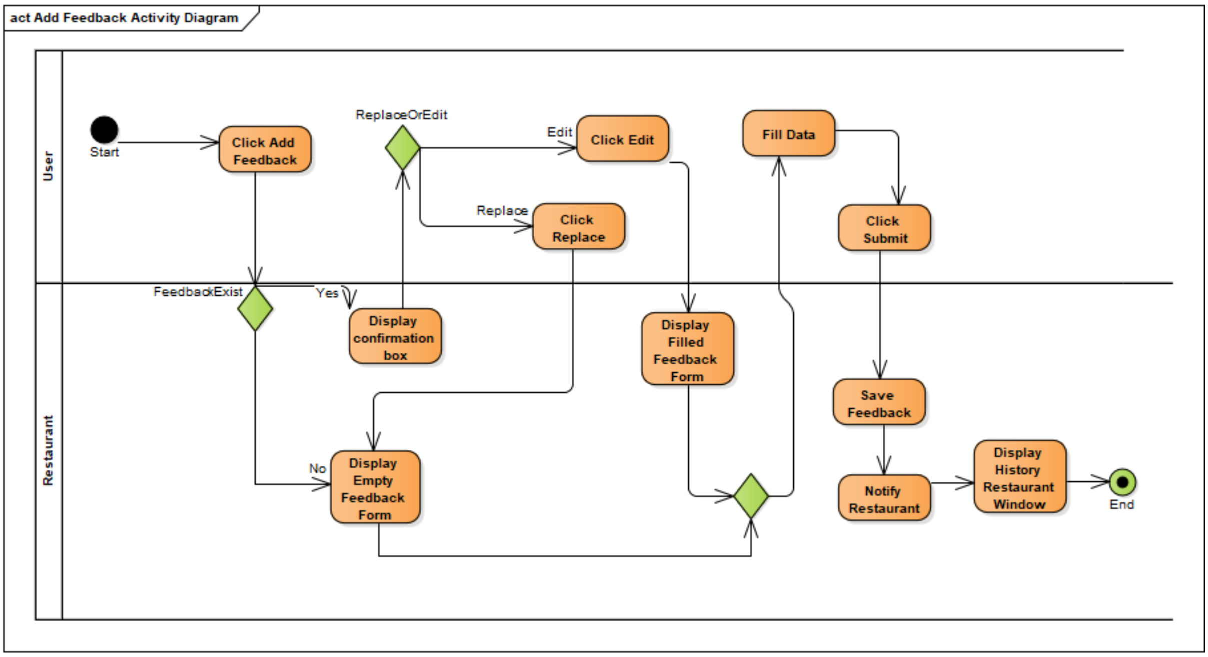
Use Case ID	UC019
Use Case Name	Add Feedback
Actors	User
Trigger Event	The user wants to add feedback to a visited restaurant
Brief Description	This use case is performed by user who want to add feedback to reflect their opinion and experience to the restaurant.
Pre-Condition	Use Case view visited restaurant is performed.
Post-Condition	Notification is added to the restaurant’s waiter call list.
Flow of events	1. System display feedback form to actor. If there is feedback exist, perform exception flow 1E. 2. Actor fill in information 3. Actor click “Submit” button. 4. System notify the restaurant on this feedback form

	submitted. 5. System display history restaurant window.
Alternative Flow	-
Exception Flow	1E: feedback exist. 1. System prompt out selection dialog box with message "You have added feedback. Do you want to Replace or Edit it?" 2. If actor click Replace, back to main flow step 2. 3. If actor click Edit, system display feedback form with data to actor. 4. Back to main flow step 2.

UCD019: Add Feedback



SD019: Sequence Diagram for UC019 Add Feedback



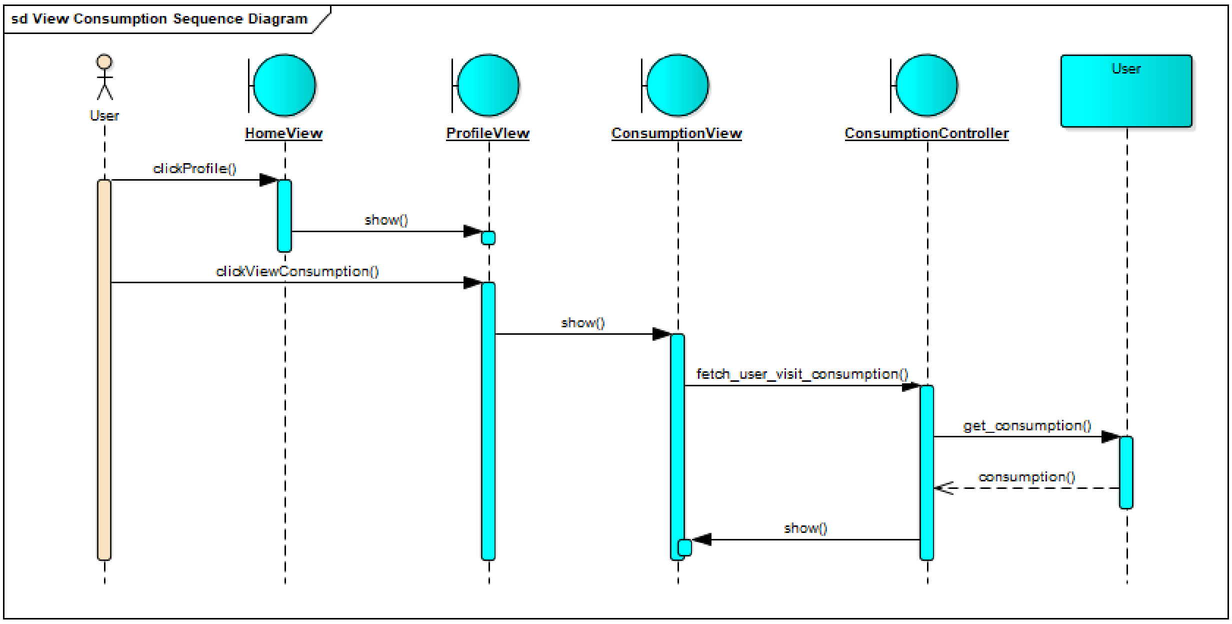
AD019: Activity Diagram for UC019 Add Feedback

3.2.4.3 UC020 View Consumption

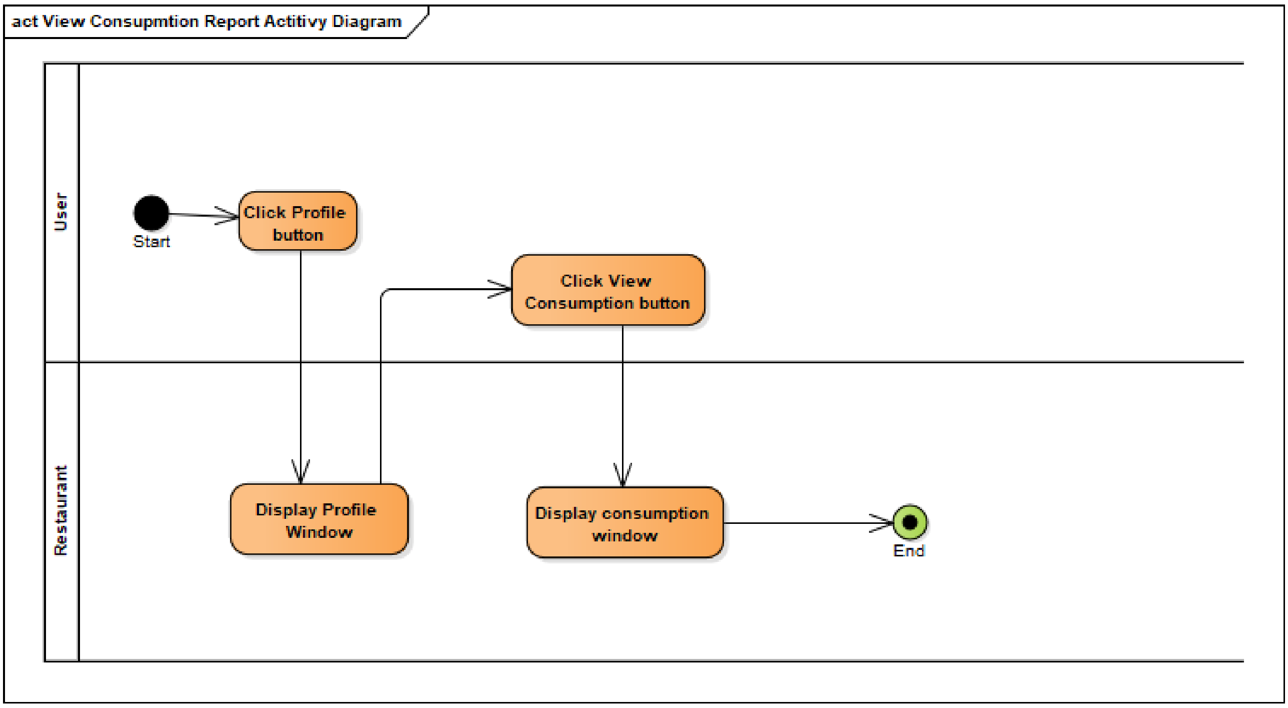
Use Case ID	UC020
Use Case Name	View Consumption
Actors	User
Trigger Event	The user wants to know their consumption of dine in.
Brief Description	This use case is performed by user who want to add feedback to reflect their opinion and experience to the restaurant.
Pre-Condition	Use Case view visited restaurant is performed.
Post-Condition	Notification is added to the restaurant's waiter call list.
Flow of events	1. Actor click profile button 2. System display profile window 3. Actor click "View Consumption" button 4. System display consumption window with a list of visit consumption
Alternative Flow	-
Exception Flow	4E: No consumption for user 1. System display "No Consumption" message in

	consumption window. 2. Use case end.

UCD020: View Consumption



SD020: Sequence Diagram for UC020 View Consumption


act View Consupmtion Report Actitivy Diagram
User
Restaurant
Start
Click Profile button
Click View Consumption button
Display Profile Window
Display consumption window
End

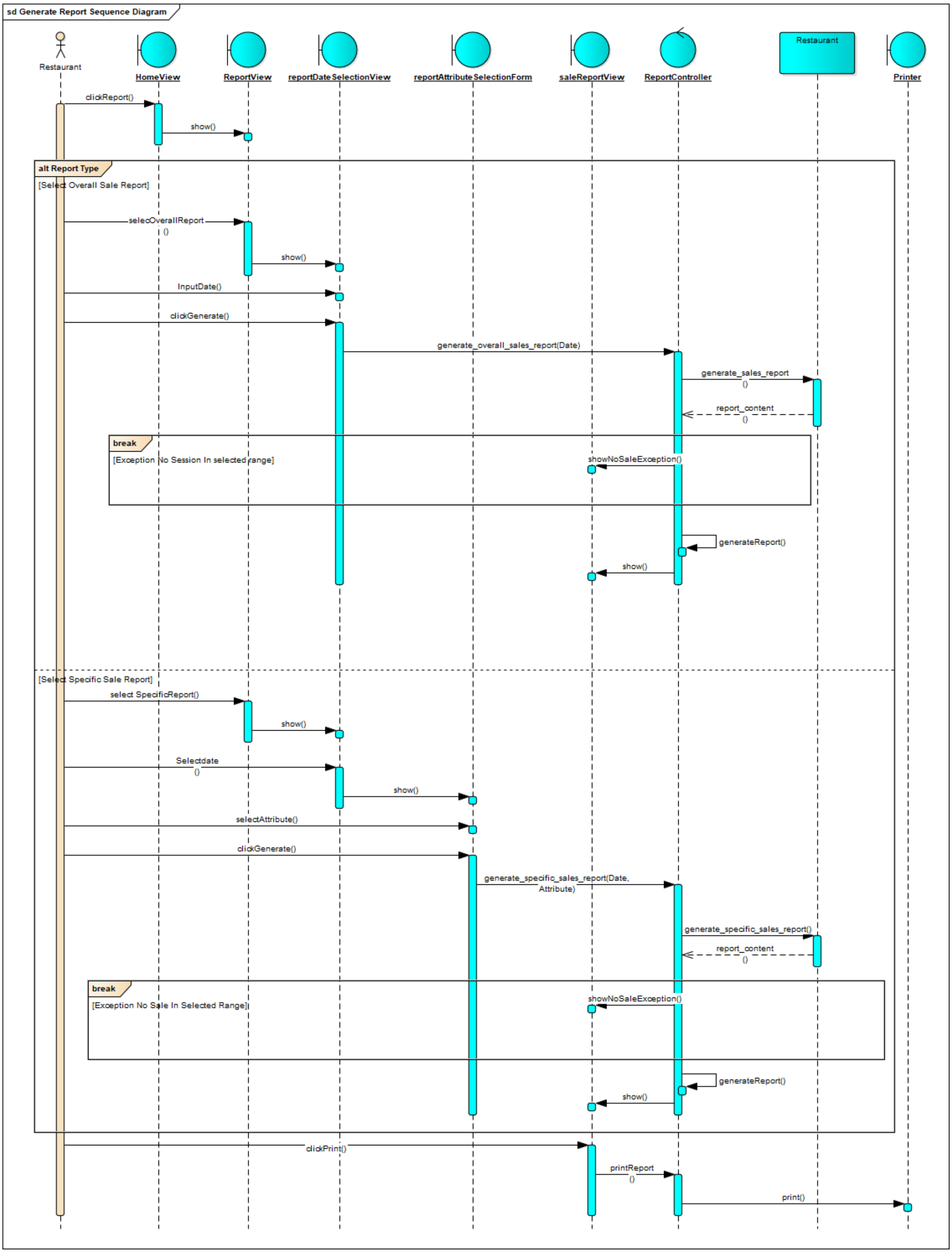
AD020: Activity Diagram for UC020 View Consumption

3.2.4.4 UC021 Generate Report

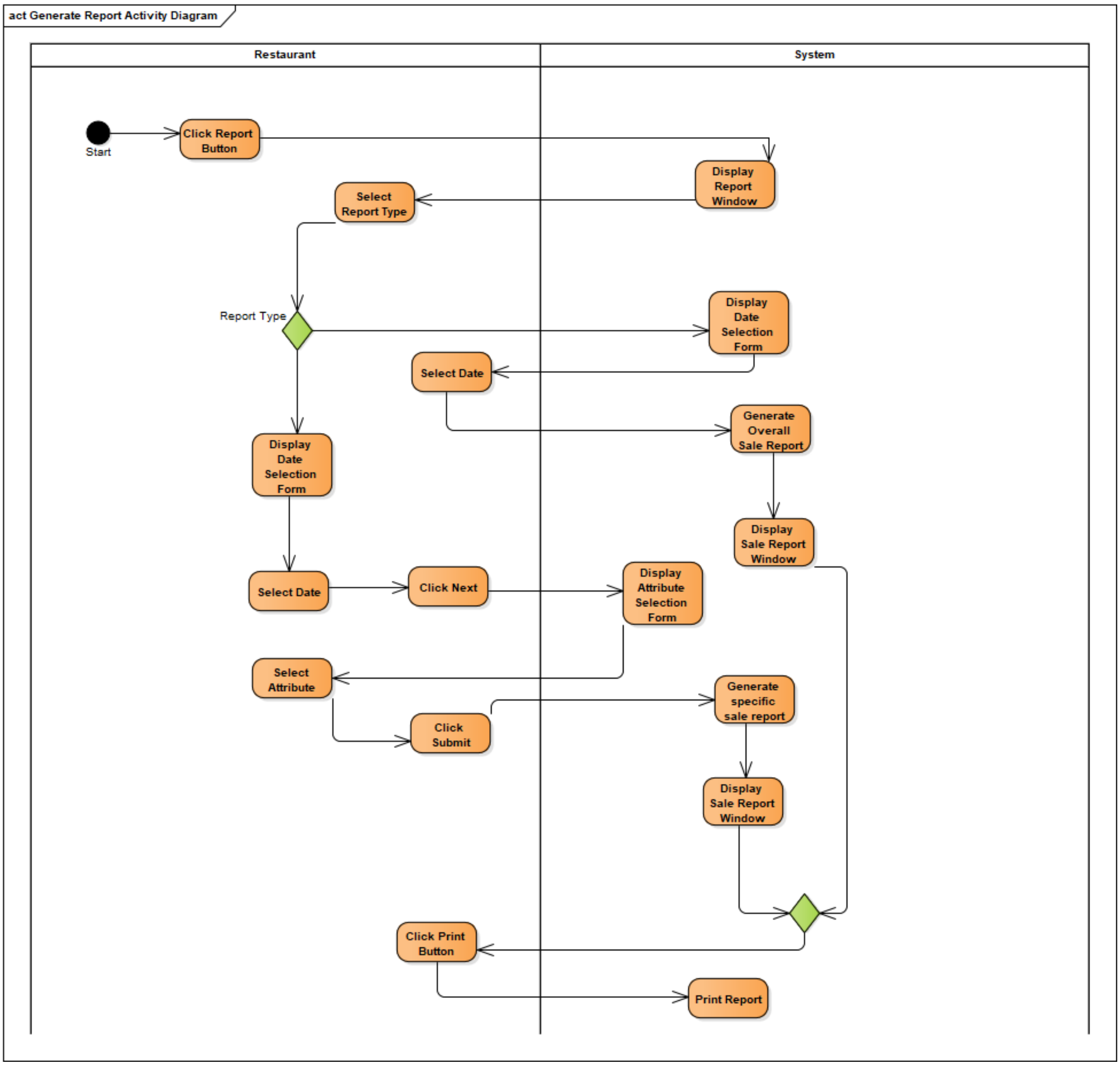
Use Case ID	UC021
Use Case Name	Generate report
Actors	Restaurant
Trigger Event	The restaurant wants to view the performance of the restaurant
Brief Description	This use case is performed by the restaurant to generate restaurant report based on the selected attribute.
Pre-Condition	-
Post-Condition	A report is generated.
Flow of events	1. Actor clicks “Report” button 2. System display report window. 3. Actor select the report type to generated. 4. If actor click “Overall Sale Report” button, perform alternative flow 4A. If actor click “Specific Sale Report” button, perform alternative flow 4B. 5. Actor clicks “Print” button. 6. System display print dialog
Alternative Flow	4A: Overall Sale Report. 1. System display date range selection form 2. Actor select from date and to date. 3. System generate overall sale report. 4. System display sale report window with the information. If there is no session in the selected range, perform exception flow 4E. 5. Back to main flow step 5. 4B:Specific Sale Report 1. System display date selection form. 2. Actor select from date and to date 3. Actor click next button 4. System display attribute selection form. The attribute includes user age.

	5. Actor select an attribute. 6. System generate specific sale report. If there is no session in the selected range, perform exception flow 4E. 7. System display sale report window with the information 8. Back to main flow step 5.
Exception Flow	4E: No session in the selected date range. 1. System display “No sales in this date range” message to actor and use case end.

UCD021: Generate Report



SD021: Sequence Diagram for UC021 Generate Report


act Generate Report Activity Diagram
Restaurant
System
Start
Click Report Button
Display Report Window
Select Report Type
Report Type
Display Date Selection Form
Select Date
Generate Overall Sale Report
Display Date Selection Form
Display Sale Report Window
Select Date
Click Next
Display Attribute Selection Form
Select Attribute
Generate specific sale report
Click Submit
Display Sale Report Window
Click Print Button
Print Report

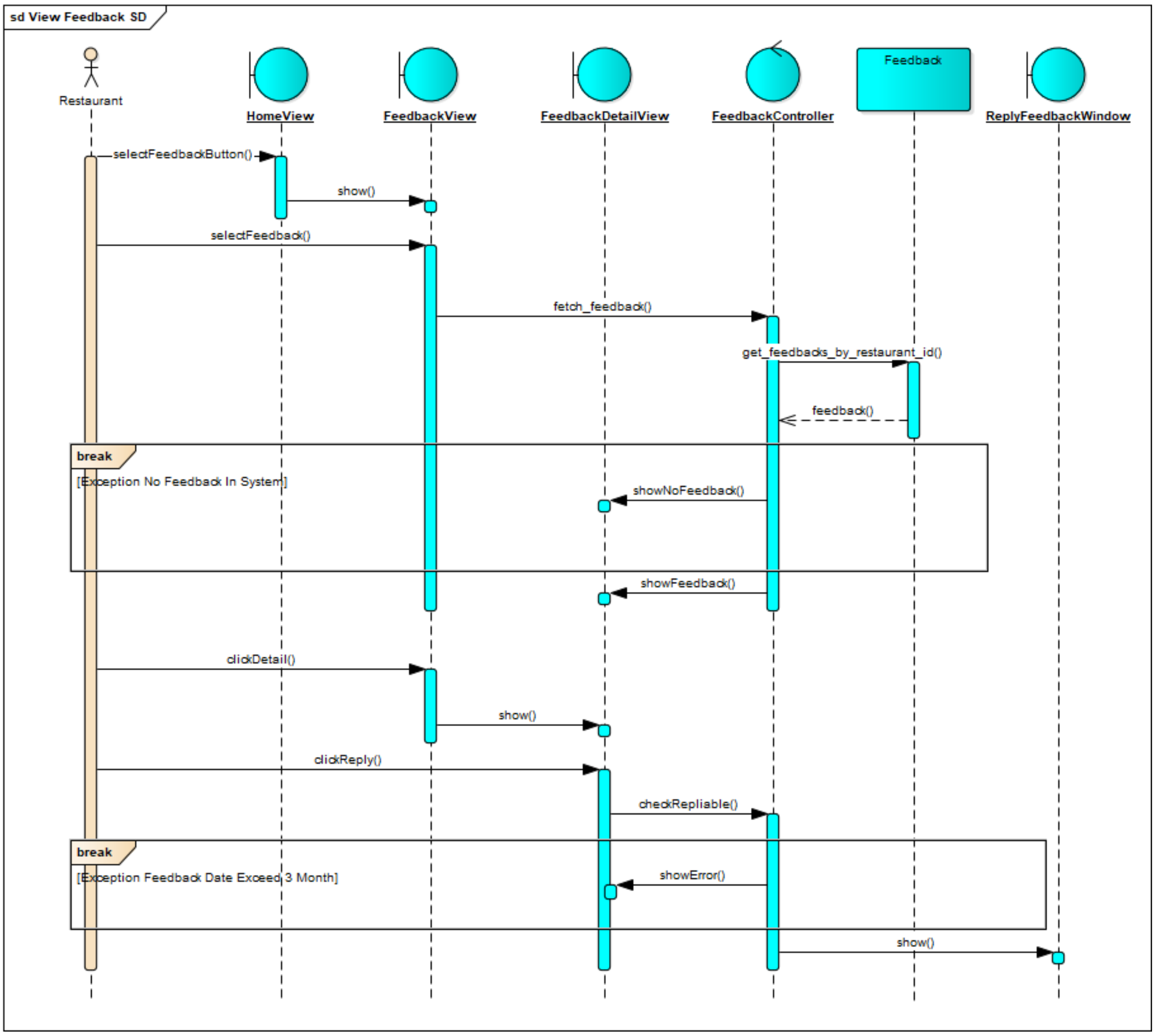
AD021: Activity Diagram for UC021 Generate Report

3.2.4.5 UC022 View Feedback

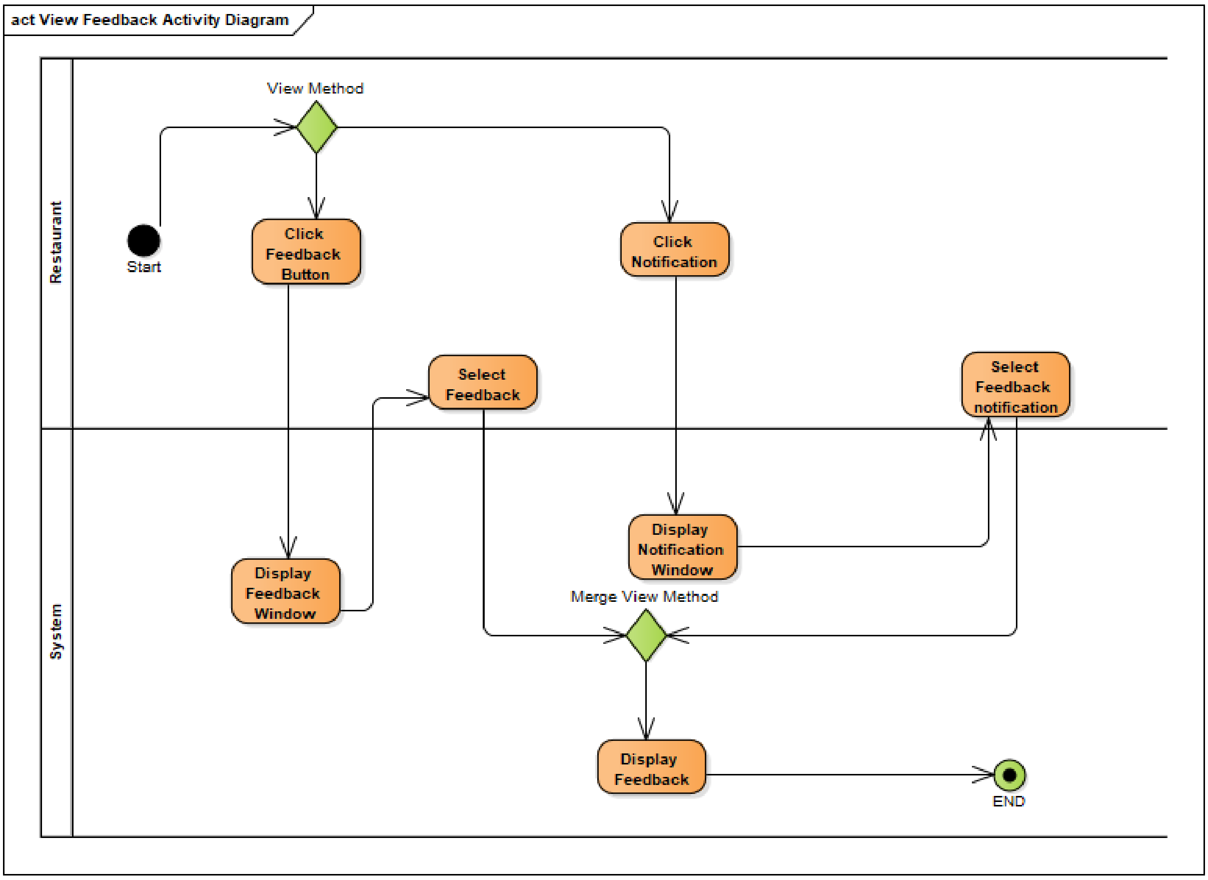
Use Case ID	UC022
Use Case Name	View Feedback
Actors	Restaurant
Trigger Event	The restaurant wants to view the feedback added to the user.
Brief Description	This use case is trigger when the user want to view the feedback from customer from notification or feedback window.
Extended Use Case	Reply Feedback.

Pre-Condition	-
Post-Condition	-
Flow of events	1. Actor click “Feedback” button. If actor clicks notification button, perform alternative flow 1A. 2. System display feedback window with a list of feedback. If there is no feedback, perform exception flow 2E. 3. Actor select a feedback to view. 4. System display customer feedback window. 5. If actor click reply button, perform extended use case. If the feedback exceed 3 months, perform exception flow 5E.
Alternative Flow	1A: View Feedback from notification 1. Actor click notification button 2. System display notification window. 3. Actor select a feedback notification. 4. Back to main flow step 4.
Exception Flow	2E: No feedback in system. 1. System display “No feedback found” message to user and use case end. 5E: Feedback date exceed 3 months: 1. System display “It is too late to reply this feedback” message and use case end.

UCD022: View Feedback



SD022: Sequence Diagram for UC022 View Feedback



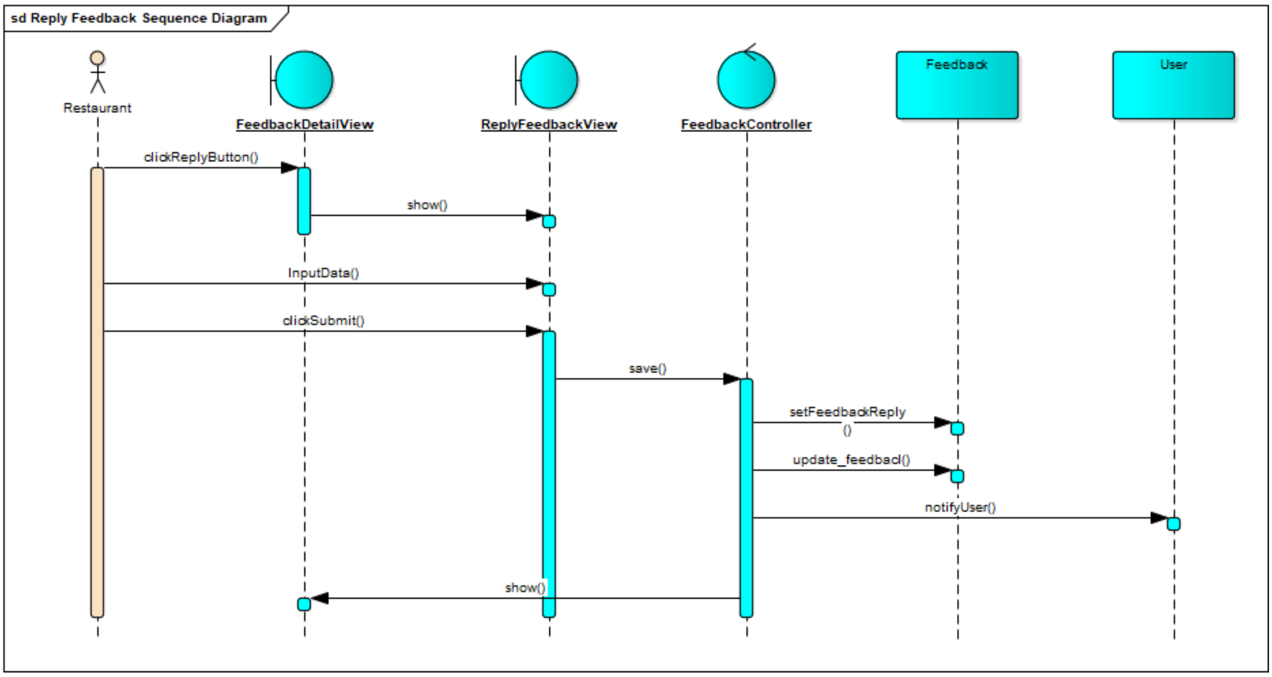
AD022: Activity Diagram for UC022 View Feedback

3.2.4.6 UC023 Reply Feedback

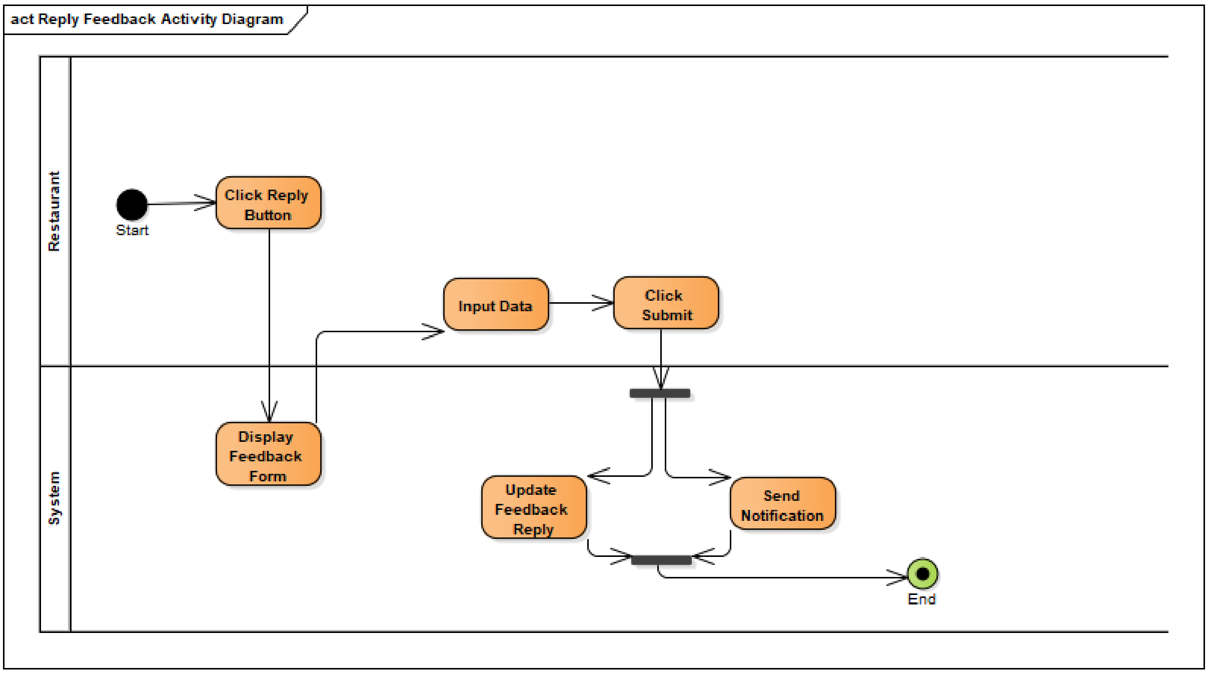
Use Case ID	UC023
Use Case Name	Reply Feedback
Actors	Restaurant
Trigger Event	The restaurant wants to reply feedback of a user.
Brief Description	This use case is an extended use case for view feedback. It is performed when the restaurant clicks reply button on a feedback.
Pre-Condition	Reply button is clicked in the customer feedback window.
Post-Condition	A email notification is sent to user.
Flow of events	1. Actor clicks reply button 2. System display reply feedback form. 3. Actor input data. 4. Actor click submit.

	5. System save reply. 6. System send email to user. 7. System display customer feedback window.
Alternative Flow	-
Exception Flow	-

UCD023: Reply Feedback



SD023: Sequence Diagram for UC023 Generate Report



AD023: Activity Diagram for UC023 Reply Feedback

3.3 Performance Requirements

2.2.3 NF001: The system down time shall not exceed 20 minutes in a day.

2.2.4 NF002: The system shall not happen synchronization problem in writing data.

3.4 Design Constraints

2.2.5 NF003: The system shall indicate if the food is halal.

3.5 Software System Attributes

2.2.6 NF004: The system shall support multiple language

2.2.7 NF005: The system shall contain an easy and simple user interface.

3.6 Other Requirements

No other requirement.