Software Requirements Specification

for

Cafeteria Ordering System

Prepared by Tianna Cano, Jay Patel, Jennifer Wu, Krishna Bhatia, Rohan Patwardhan

December 19, 2021

**Table of Contents**

[1. Introduction 3](#_Toc90837919)

[1.1 Purpose 3](#_Toc90837920)

[1.2 Scope 3](#_Toc90837921)

[1.3 Definitions, acronyms, and abbreviations 3](#_Toc90837922)

[1.4 References 3](#_Toc90837923)

[1.5 Overview 3](#_Toc90837924)

[2. Overall Description 4](#_Toc90837925)

[2.1 Product Perspective 4](#_Toc90837926)

[2.2 Product Features (FEs) 4](#_Toc90837927)

[2.3 User Classes and Characteristics 5](#_Toc90837928)

[2.4 Operating Environment 6](#_Toc90837929)

[2.5 Design and Implementation Constraints 6](#_Toc90837930)

[2.6 User Documentation 6](#_Toc90837931)

[2.7 Assumptions and Dependencies 6](#_Toc90837932)

[3. Use-Case Diagram with Use-Case Descriptions (UCs) 7](#_Toc90837933)

[3.1 Use-case Diagram 7](#_Toc90837934)

[3.2 Use-Case Description(s) 8](#_Toc90837935)

[4. External Interface Requirements 44](#_Toc90837936)

[4.1 User Interfaces 44](#_Toc90837937)

[4.2 Hardware Interfaces 44](#_Toc90837938)

[4.3 Software Interfaces 44](#_Toc90837939)

[4.4 Communications Interfaces 45](#_Toc90837940)

[5. Other Nonfunctional Requirements 45](#_Toc90837941)

[5.1 Performance Requirements 45](#_Toc90837942)

[5.2 Safety Requirements 45](#_Toc90837943)

[5.3 Security Requirements 45](#_Toc90837944)

[5.4 Software Quality Attributes 45](#_Toc90837945)

[6. Other Requirements 46](#_Toc90837946)

[7. Functional Requirements (FRs) 46](#_Toc90837947)

[7.1 FR-01: <Order meals> 46](#_Toc90837948)

[7.2 FR-02: <View menu> 46](#_Toc90837949)

[7.3 FR-03: <Submit payment> 46](#_Toc90837950)

[7.4 FR-04: <Create account> 47](#_Toc90837951)

[7.5 FR-05: <Manage payment methods> 47](#_Toc90837952)

[7.6 FR-06: <Earn rewards points> 47](#_Toc90837953)

[7.7 FR-07: <Favorite meals> 47](#_Toc90837954)

[7.8 FR-08: <Order custom meals> 48](#_Toc90837955)

[7.9 FR-09: <Give feedback> 48](#_Toc90837956)

[7.10 FR-10: <Order from local restaurants> 48](#_Toc90837957)

[7.11 FR-11: <Order multiple meals and menu items> 48](#_Toc90837958)

[7.12 FR-12: <Create a meal plan subscription> 49](#_Toc90837959)

[7.13 FR-13: <View meal plan subscriptions> 49](#_Toc90837960)

[7.14 FR-14: <Modify meal plan subscriptions> 49](#_Toc90837961)

[7.15 FR-15: <Cancel meal plan subscriptions> 50](#_Toc90837962)

[7.16 FR-16: <Select delivery or pickup> 50](#_Toc90837963)

[7.17 FR-17: <Prepare meal> 50](#_Toc90837964)

[7.18 FR-18: <Request delivery from cafeteria> 50](#_Toc90837965)

[7.19 FR-19: <Update order status> 51](#_Toc90837966)

[7.20 FR-20: <Print delivery instructions> 51](#_Toc90837967)

[7.21 FR-21: <Deliver meals> 51](#_Toc90837968)

[7.22 FR-22: <Update delivery status> 51](#_Toc90837969)

[7.23 FR-23: <Confirm delivery> 52](#_Toc90837970)

[7.24 FR-24: <Create cafeteria menus> 52](#_Toc90837971)

[7.25 FR-25: <View cafeteria menus> 52](#_Toc90837972)

[7.26 FR-26: <Modify cafeteria menus> 52](#_Toc90837973)

[7.27 FR-27: <Delete cafeteria menus> 53](#_Toc90837974)

[7.28 FR-28: <Track order status> 53](#_Toc90837975)

[7.29 FR-29: <Staff login> 53](#_Toc90837976)

[7.30 FR-30: <Manage inventory> 53](#_Toc90837977)

[7.31 FR-31: <Display restaurant specials> 54](#_Toc90837978)

[7.32 FR-32: <Modify restaurant specials> 54](#_Toc90837979)

[7.33 FR-33: <Delete restaurant specials> 54](#_Toc90837980)

[7.34 FR-34: <Request delivery from restaurant> 54](#_Toc90837981)

[7.35 FR-35: <Manage order status> 55](#_Toc90837982)

[7.36 FR-36: <Print order record> 55](#_Toc90837983)

[7.37 FR-37: <Update account> 55](#_Toc90837984)

[7.38 FR-38: <Delete account> 55](#_Toc90837985)

[7.39 FR-39: <Redeem rewards points> 55](#_Toc90837986)

[7.40 FR-40: <Rewards points expire> 56](#_Toc90837987)

[7.41 FR-41: <Create staff account> 56](#_Toc90837988)

[7.42 FR-42: <View order history> 56](#_Toc90837989)

[TABLE 2: Traceability Matrix (Use-cases & FRs). 57](#_Toc90837990)

[8. Functional Modeling (DFDs) 58](#_Toc90837991)

[8.1 Context Diagram (level 0) 58](#_Toc90837992)

[8.2 Level 1 58](#_Toc90837993)

[8.3 Level 2 59](#_Toc90837994)

[TABLE 3: Traceability Matrix (FRs and DFD Processes) 63](#_Toc90837995)

[9. Class Analysis Modeling 65](#_Toc90837996)

[TABLE 4: Traceability Matrix (FRs and Initial Classes) 67](#_Toc90837997)

[TABLE 5: Traceability Matrix (Initial Classes (IC) and modified Classes (MC)) 70](#_Toc90837998)

[10. Sequence Diagram (SDs) 72](#_Toc90837999)

[10.1 SD-01: *<Order Meal & Submit Payment>* 72](#_Toc90838000)

[10.2 SD-02: *<View Menu>* 72](#_Toc90838001)

[10.3 SD-03: *<Create Account>* 73](#_Toc90838002)

[10.4 SD-04: *<Manage Payment Methods>* 73](#_Toc90838003)

[10.5 SD-05: *<Rewards Points>* 74](#_Toc90838004)

[10.6 SD-06: *<Favorite Meals>* 74](#_Toc90838005)

[10.7 SD-07: *<Order Custom Meals>* 75](#_Toc90838006)

[10.8 SD-08: *<Feedback>* 75](#_Toc90838007)

[10.9 SD-09: *<Order Meal FR-om Restaurant>* 76](#_Toc90838008)

[10.10 SD-10: *<Meal Subscriptions>* 76](#_Toc90838009)

[10.11 SD-11: *<Meal Preparation>* 77](#_Toc90838010)

[10.12 SD-12: *<Delivery>* 77](#_Toc90838011)

[10.13 SD-13: *<Menu Management>* 78](#_Toc90838012)

[10.14 SD-14: *<View Order Tracking Status>* 79](#_Toc90838013)

[10.15 SD-15: *<Manage Inventory>* 79](#_Toc90838014)

[10.16 SD-16: *<Display Restaurant Specials>* 80](#_Toc90838015)

[10.17 SD-17: *<Request Delivery from Restaurants>* 80](#_Toc90838016)

[10.18 SD-18: *<Manage Orders>* 81](#_Toc90838017)

[TABLE 6: Traceability Matrix (UCs and Sequence Diagram (SD) ) 82](#_Toc90838018)

[11. Detailed Class Diagram 83](#_Toc90838019)

# Introduction

## Purpose

The following document has been created to describe the software requirements for the implementation of a Cafeteria Ordering System. The group Empty Coffee Cups will be using this document to ensure all requirements have been implemented and verified for release.

## Scope

Like any busy company, Process Impact would like to improve the efficiency of its company cafeteria when providing meals to its employees. This application will allow the company’s employees to order meals for pickup or delivery via the company cafeteria to lessen the time these employees spend on their lunch breaks.

Using the reward points system to encourage guests to pick up meals at designated

locations can reduce staff consumption and increase meal retrieval efficiency.

Develop a favorite system to allow customers to save historical orders or customize their

favorite order so that they can choose the same meal in the future. This system can save customers time for ordering, and the automatic ordering mechanism can reduce the queue for ordering in restaurants.

The notification system allows customers to track their food delivery status. This system requires GPS and Internet to show the location of the orders. System information needs to access the location of the delivery.

Forecasting ordering system can predict the orders in the next 2 hours. It helps the kitchen to monitor the ingredient inventory and get ready for large crowds of customers.

## Definitions, acronyms, and abbreviations

COS Cafeteria Ordering System

GPS Global Positioning System

## References

[1] Vision and Scope F2021.doc

[2] IEEE Software Engineering Standards Committee, IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications, October 20, 1998.

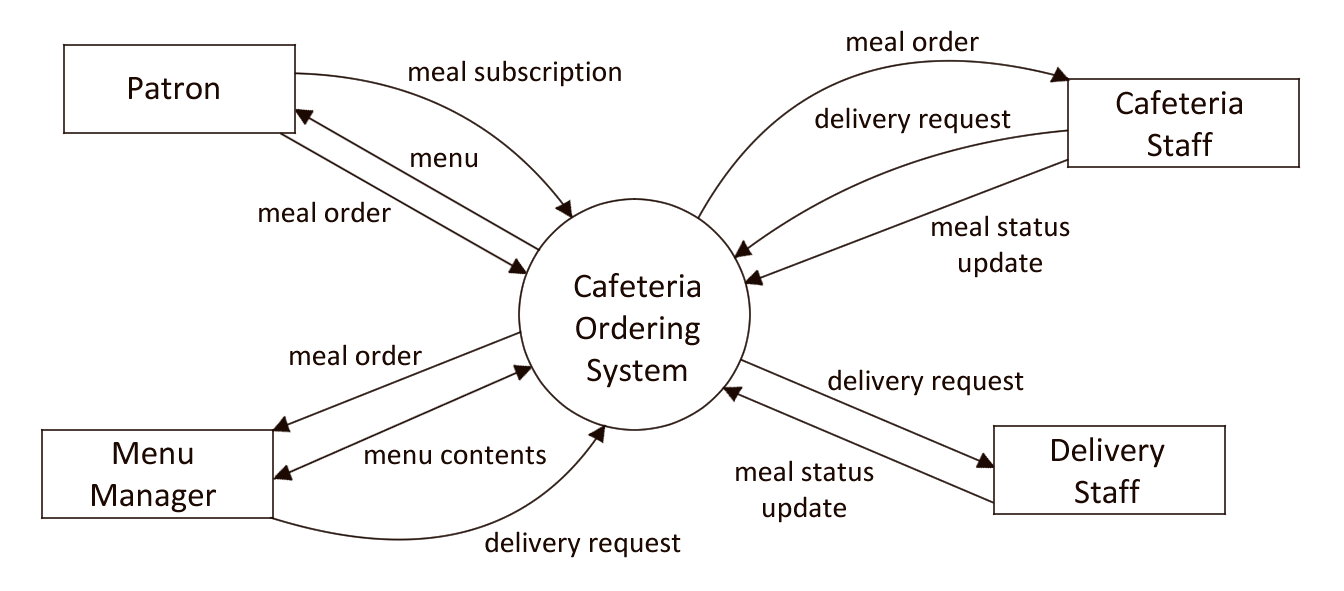
## Overview

This document contains an overview for all the requirements and documentation needed for the development of this project. It includes 6 sections.

Section 2, Overall Description, discusses the product feature, user classes, constraint of design and implementation. Section 3, Use-Case Descriptions, describes the user case diagrams and description. Section 4, External Interface Requirements, overviews external interface requirements, including the variety of software interfaces, and hardware interfaces. Section 5, Other Nonfunctional Requirements, talks about the non-functional requirement. Section 6, Other Requirements, mentions the other requirements.

# Overall Description

## Product Perspective



**Figure 1** Context Diagram

The Cafeteria Ordering System will be a new system created for the ease of ordering and obtaining food from the company cafeteria. The system is expected to evolve over the course of subsequent iterations to allow for ordering from restaurants outside of the company and other features as the needs of the employees evolve.

## Product Features (FEs)

FE-01: Order meals from the cafeteria menu to be picked up or delivered

* The user places an order from the cafeteria menu
* The user is able to choose multiple items from the menu each time
* The menu needs to keep the record of the order in shopping cart
* The user is able to choose the way for getting the order

FE-02: Order meals from local restaurants to be delivered

* The user shall place an order of other local restaurants from the menu
* The user chooses to pick up the order from the local restaurant or deliver to the desired address.

FE-03: Create, view, modify, and delete meal service subscriptions

* The use can subscribe a meal service from the menu
* The user can create their own meal service.

FE-04: Register for meal payment options

* The user can choose the payment methods.

FE-05: Request meal delivery

* The user can request a meal delivery to the desired location at the specific time

FE-06: Create, view, modify, and delete cafeteria menus

* The user can modify the cafeteria menu

FE-07: Order custom meals that aren’t on the cafeteria menu

* The user can customize their meal based on their demand.

FE-08: Set up reward points for picking up orders at designated places on the campus, including the cafeteria.

* The user can pick up the order at the designed location and get the rewards
* The restaurant can reduce the delivery times and increase the efficiency.

FE-09: Favorite system where customers can save an order and can automatically order the same meal next time, including customizations.

* The user can build their own favorite meals record

FE-10: Notification feature for tracking the readiness of meals and where the meal will be delivered to if the customer orders delivery.

* The user can track the meal and get ready for picking up

FE-11: A feedback system where customers can provide feedback and ratings for the system.

* The user can provide feedback for the quality of the meal.
* The user can provide feedback for the quality of the delivery.

## User Classes and Characteristics

USR-Patron Patrons of the cafeteria consist of Process Impact’s employees who have voiced their desire to order meals for delivery from the company cafeteria. They are our favored user class. We expect them to mainly use the application while at work in their offices but should also be expected to be able to order meals from their cellphones or from home. They may want to create an account so that they can manage their payment methods and purchase meal subscriptions. Through meal subscriptions, they must be able to choose whether to have the same meal delivered every day or if they want to order specific meals from the menu that are offered on certain days.

USR-Cafeteria Staff The cafeteria staff consist of all current employees hired to work at the cafeteria. They will receive the orders from the system, prepare the meals, and package them for delivery if necessary. Should the meal require delivery, they must also provide delivery instructions and send it for delivery with the delivery staff. They should be familiar with operating the system once the system has been released since they will update the meal status until it is sent out for delivery.

USR-Menu Manager The menu manager can create and update the menu throughout the week. They will manage orders for meals from restaurants and arrange their delivery. They must also manage the inventory for the cafeteria.

USR-Delivery Staff The delivery staff are responsible for delivering the meals to their designated locations according to the delivery instructions that should be provided by the cafeteria staff. They will receive the food and delivery instructions then deliver each meal to the proper customer. They should use the system to provide additional updates on the delivery status of the order, then confirm or deny the delivery’s completion once the customer does or does not receive their order.

## Operating Environment

OE-1: The software is expected to operate on desktop computers and mobile devices alike.

OE-2: it should be capable of running on machines using Windows and Linux operating systems. These versions are assumed to be up to date.

OE-3: The system will need to be accessible via an internet connection or through the company’s Intranet

## Design and Implementation Constraints

DIC-1: A database will need to be used to store records of employee's orders and their meal subscriptions.

DIC-2: Security will also need to be considered since employees will be allowed to store payment methods, such as credit and debit card information.

## User Documentation

UD-1: User tutorials can be provided upon creating an account through the system as infographics.

UD-2: Tutorials for the delivery and cafeteria staff can be provided via online tutorials and/or printed manuals.

## Assumptions and Dependencies

AD-1: The system would need an account creation system for the employees to be able to manage their payment methods and meal subscriptions.

# Use-Case Diagram with Use-Case Descriptions (UCs)

## Use-Case Diagram

**Figure 2** Use-Case Diagram

## Use-Case Description(s)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | **UC-01** | | | |
| **Use Case Name:** | **Order Meal** | | | |
| **Related Features** | **FE-01** | | | |
| **Created By:** | **Tianna Cano** | | **Last Updated By:** | **Tianna Cano** |
| **Date Created:** | **9/29/2021** | | **Last Revision Date:** | **12/18/2021** |
| **Actors:** | | **Patrons** | | |
| **Description:** | | When a patron accesses the COS, they have the option to view the menu for any day, select food items, then place an order for the meal. They can also select a meal for delivery to one of the designated locations or for pickup at the cafeteria within an appropriate time window. | | |
| **Trigger:** | | System event - tap or click on the order button | | |
| **Preconditions:** | | 1. Patron has a COS account. 2. Patron is logged into COS. | | |
| **Postconditions:** | | 1. Meal order is sent to the cafeteria for preparation. 2. Inventory of food items and/or ingredients is updated because of the order. 3. The delivery staff updates the availability of delivery personnel once the delivery request is made and accepted. | | |
| **Normal Flow:** | | 1. Patron views the menu for meals offered on any selected day. 2. COS displays the menu of available food items for that day. 3. Patron selects one or more food items from the menu. 4. Patron confirms their order is complete. 5. Patron selects delivery or pickup. 6. COS displays their order- listing the food items, individual prices, and total price, including any taxes and delivery costs. 7. Patron confirms order or chooses to modify it. 8. Patron selects payment method. 9. COS confirms the order. 10. COS sends a notification via email or the application itself to display the order details, prices, and total price, as well as the delivery instructions if selected. 11. COS stores the order into a database. 12. COS notifies cafeteria staff of the order, prints out order tickets. | | |
| **Alternative Flows:** | | **Order multiple items**  3a. In step 3 of the normal flow, the patron orders at least one food item.   1. COS updates the order item count and prompts the patron to checkout if they are finished ordering. 2. Patron returns to step 2   **Order multiple of the same item**  3b. In step 3 of the normal flow, the patron orders at least one food item.   1. COS updates the order item count and prompts the patron to checkout if they are finished ordering. 2. Patrons can select the same item to update the number of meals they wish to purchase. | | |
| **Exceptions:** | | **Time of order is after serviceable hours**  1a. In step 2 of the normal flow if the patrons try to select a date that does not provide adequate time to complete the order.   1. COS notifies the patron that it is too late to order for that day.   2a. Patron cancels order  2b. Patron selects another date  **Delivery staff unavailable**  5a. In step 5 of the normal flow if the patron selects delivery when delivery personnel are already out delivering meals.   1. COS notifies the patron that delivery is unavailable at the time or ordering.   2a. Patron cancels order  2b. Patron selects pickup  **Requested items are unavailable**  1b. In step 2 of the normal flow, if the patrons select an item or reaches a number of items that exceed what is available at the cafeteria   1. COS notifies the patron that the item or requested number of items is unavailable.   2a. Patron cancels order  2b. Patron updates number of ordered items or selects a different item | | |
| **Includes:** | | UC-03, UC-05, UC-08 | | |
| **Frequency of Use:** | | Approximately 400 users, at least one use per patron per day | | |
| **Special Requirements:** | | 1. Patrons can cancel their order at any time. 2. Patrons can view past orders. | | |
| **Assumptions:** | | 1. Patron has created an account for the system. | | |
| **Notes and Issues:** | | 1. The menu date is updated to the current date of when the patron accesses the system, or the following date and menu is displayed. | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | **UC-02** | | | |
| **Use Case Name:** | **View Menu** | | | |
| **Related Features** | **FE-01, FE-02, FE-05, FE-07, FE-09** | | | |
| **Created By:** | **Jennifer Wu** | | **Last Updated By:** | **Tianna Cano** |
| **Date Created:** | **9/29/2021** | | **Last Revision Date:** | **12/18/2021** |
| **Actors:** | | Patrons | | |
| **Description:** | | Reason: To provide a list of meals for patrons to select from.  Outcome: Patrons understands what meals are provided by the restaurant. | | |
| **Trigger:** | | Patrons selects the menu page on the order system. | | |
| **Preconditions:** | | 1. Patron has already logged into the menu system. 2. Patron is ready to order. | | |
| **Postconditions:** | | 1. A list of meals shows on the menu to patrons. | | |
| **Normal Flow:** | | 1. Patron views the main menu 2. Patron selects a particular meal page. 3. Patron zoom-in/zoom-out the photos of the meals. 4. Patron traverse to the next/previous page 5. Patron clicks the meals button and views the details. 6. Patron sees the ordering bottom and customize zone. | | |
| **Alternative Flows:** | | AF-1. In step 1 of normal flow, if the patron checks other restaurants’ menu   1. System will jump back to the main page. 2. Patron selects the restaurant. 3. Use Case resumes on step 2 | | |
| **Exceptions:** | | E-1. In any step, if the patron logs out of the ordering system accidentally.   1. re-log in the ordering system 2. Use Case resumes on step 1 | | |
| **Includes:** | | UC-01 | | |
| **Frequency of Use:** | | Approximately 400 users, at least one use per patron per day | | |
| **Special Requirements:** | | The menu contains an audio display menu tool for visually impaired patrons. | | |
| **Assumptions:** | | The patron understands either English or Spanish language. | | |
| **Notes and Issues:** | | 1. How to notify the patrons while updating the menu? 2. What is the maximum number of meals that can be displayed on the menu? | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | **UC-03** | | | |
| **Use Case Name:** | **Submit Payment** | | | |
| **Related Features** | **FE-01, FE-02, FE-04, FE-07** | | | |
| **Created By:** | **Tianna Cano** | | **Last Updated By:** | **Tianna Cano** |
| **Date Created:** | **9/30/2021** | | **Last Revision Date:** | **12/18/2021** |
| **Actors:** | | Patrons | | |
| **Description:** | | When paying for their order, patrons can select any stored payment methods on their COS account. | | |
| **Trigger:** | | System Event - tap or click on payment method choice. | | |
| **Preconditions:** | | 1. Patron has an account. 2. Patron has saved their preferred payment methods. 3. Patron has placed an order. | | |
| **Postconditions:** | | 1. Meal order is sent to the cafeteria for preparation. 2. Inventory of food items and/or ingredients is updated because of the order. 3. The delivery staff updates the availability of delivery personnel once the delivery request is made and accepted. Order is confirmed by the COS. 4. COS sends a notification via email or the application itself to display the order details, prices and total price, as well as the delivery instructions if selected. 5. COS stores the order into a database. 6. COS notifies cafeteria staff of the order, prints out order tickets. | | |
| **Normal Flow:** | | 1. Patron is paying for their order. 2. Patron selects the preferred payment method from a selection of pre-saved methods. 3. Steps after step 8 from UC-01 are carried out. | | |
| **Alternative Flows:** | |  | | |
| **Exceptions:** | | **There is no payment method available**  2a. The patron has not entered or saved a payment method beforehand.   1. No payment method is displayed 2. COS prompts patron to enter a payment method. 3. Patron can select to save the method for future orders. 4. Use Case resumes on step 3 of normal flow   **Payment method has expired**  2b. The selected payment method’s credentials are outdated.   1. COS notifies that there is an error with the payment method 2. COS prompts the patron to delete or update the payment method’s information. 3. Use Case resumes on step 3 of normal flow | | |
| **Includes:** | | UC-06 | | |
| **Frequency of Use:** | | Approximately 400 users, at least one use per patron per day | | |
| **Special Requirements:** | | 1. Patrons can add, view, edit, or remove payment methods at any time. | | |
| **Assumptions:** | | 1. Patron has already created an account | | |
| **Notes and Issues:** | | 1. Security for storing payment methods? | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | **UC-04** | | | |
| **Use Case Name:** | **Create account** | | | |
| **Related Features** | **FE-01, FE-02, FE-03, FE-04, FE-05, FE-06, FE-07, FE-08, FE-09, FE-10, FE-11** | | | |
| **Created By:** | **Rohan Patwardhan** | | **Last Updated By:** | **Tianna Cano** |
| **Date Created:** | **9/30/2021** | | **Last Revision Date:** | **12/18/2021** |
| **Actors:** | | Patrons | | |
| **Description:** | | Reason: To subscribe to meals and to order food.  Outcome: Patron can access all features of COS like order food, subscribe to meals, etc. | | |
| **Trigger:** | | Patron selects the ‘create account’ button | | |
| **Preconditions:** | | 1. Patron has a device to access the cafeteria ordering system 2. Patron has internet access | | |
| **Postconditions:** | | 1. Patron will be able to view menus, subscribe to meal plans, order food, etc. | | |
| **Normal Flow:** | | 1. Patron views the login page 2. Patron selects the 'create new account’ option. 3. Patron fills in all fields like name, number, email, address, password on ‘create new account’ page 4. Patron clicks the ‘create account’ button | | |
| **Alternative Flows:** | | N/A | | |
| **Exceptions:** | | **E-1.** If a patron already has an account, then the patron needs to go to login page and login using appropriate credentials  **E-2** In step3 of the normal form, if the patron enters invalid information   1. Information is not saved. 2. COS notifies patron that the entered information is incorrect 3. COS prompts patron to re-enter information 4. Patron enters valid information 5. Use Case resumes on step 4 of normal flow | | |
| **Includes:** | | N/A | | |
| **Frequency of Use:** | | Once | | |
| **Special Requirements:** | | N/A | | |
| **Assumptions:** | | The patron understands either English or Spanish language. | | |
| **Notes and Issues:** | | N/A | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | **UC-05** | | | |
| **Use Case Name:** | **Manage Payment Methods** | | | |
| **Related Features** | **FE-04** | | | |
| **Created By:** | **Tianna Cano** | | **Last Updated By:** | **Tianna Cano** |
| **Date Created:** | **9/30/2021** | | **Last Revision Date:** | **12/18/2021** |
| **Actors:** | | Patrons | | |
| **Description:** | | Patrons with COS accounts can manage their saved payment methods or add more if they want to. | | |
| **Trigger:** | | System event - upon creating an account, the COS gives the patron the option of entering and saving payment methods for ease of ordering. | | |
| **Preconditions:** | | 1. Patron has created an account. | | |
| **Postconditions:** | | 1. Patron’s payment method is saved to apply to future orders. | | |
| **Normal Flow:** | | 1. Patron has created an account 2. COS prompts patron to enter payment method information to save for next use 3. Patron can enter information for credit or debit cards 4. COS saves information and will make it available for use | | |
| **Alternative Flows:** | | N/A | | |
| **Exceptions:** | | **Patron enters invalid payment method information** 2a. In step 2 of the normal flow, if the patron enters invalid information   1. Information is not saved. 2. COS notifies patron that the entered information is incorrect 3. COS prompts patron to re-enter information 4. Patron enters valid information 5. Use Case resumes on step 3 of normal flow | | |
| **Includes:** | | N/A | | |
| **Frequency of Use:** | | Approximately 400 users | | |
| **Special Requirements:** | | 1. Patrons must have valid payment methods | | |
| **Assumptions:** | | 1. Patrons will have already created an account | | |
| **Notes and Issues:** | | 1. Security for storing payment method information? | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | **UC-06** | | | |
| **Use Case Name:** | **Earn Reward Points** | | | |
| **Related Features** | **FE-08** | | | |
| **Created By:** | **Tianna Cano** | | **Last Updated By:** | **Tianna Cano** |
| **Date Created:** | **9/30/20201** | | **Last Revision Date:** | **12/18/2021** |
| **Actors:** | | Patrons | | |
| **Description:** | | Patrons will earn reward points for ordering meals using the COS. These points will yield certain rewards when certain amounts have been reached. These rewards are often discounts on certain items or even a free item. | | |
| **Trigger:** | | Each time an order has been confirmed. The checkout for the order can display a button to view available rewards that will automatically apply the discount to the current order. | | |
| **Preconditions:** | | 1. Patron has created an account 2. Patron has placed an order 3. The order has been paid for and confirmed | | |
| **Postconditions:** | | 1. Points are calculated based on the cost of the order 2. The total number of points is added to the patron’s account 3. If a reward goal is reached, the patron will be notified that the reward is available to use next time | | |
| **Normal Flow:** | | 1. Patron orders from the COS 2. Patron successfully pays for the meal 3. The COS calculates the number of points based on the cost for the food items 4. The total amount of rewards is added to the patron’s account | | |
| **Alternative Flows:** | | **Patron chooses to use reward points**  1a. In step 1 of the normal flow, the patron has existing rewards available, so they decide to apply the reward to their order.   1. The patron will select to view available rewards. 2. If a reward is available, the patron can select to apply it to their current order 3. The reward will apply a discount to the total cost 4. Use Case from UC-01 can resume from step 7 | | |
| **Exceptions:** | | **Patron does not have an account**   1. COS will prompt the patron to create an account 2. Patron creates an account 3. The ability to earn points is now available 4. Use Case resumes normal flow from step 2 | | |
| **Includes:** | | UC-01, UC-04 | | |
| **Frequency of Use:** | | Approximately 400 users, at least one use per user per day | | |
| **Special Requirements:** | | 1. Patron must have valid payment methods | | |
| **Assumptions:** | | 1. Patron will have already created an account | | |
| **Notes and Issues:** | | 1. How to calculate points? 2. What rewards to offer? How many? 3. Discount percentage for each reward? | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | **UC-07** | | | |
| **Use Case Name:** | **Favorite Meals** | | | |
| **Related Features** | **FE-09** | | | |
| **Created By:** | **Tianna Cano** | | **Last Updated By:** | **Tianna Cano** |
| **Date Created:** | **9/30/2021** | | **Last Revision Date:** | **12/18/2021** |
| **Actors:** | | Patrons | | |
| **Description:** | | When viewing the menu, the patron can select certain food items as their ‘Favorites.’ These favorite items can be ordered automatically if the patron desires. | | |
| **Trigger:** | | System event - tap or click a button to add item to ‘Favorites’ category | | |
| **Preconditions:** | | 1. Patron has created an account 2. Patron is viewing the menu | | |
| **Postconditions:** | | 1. COS stores these selected items to a ‘Favorites’ category for each account 2. The item is offered for automatic ordering at the next use | | |
| **Normal Flow:** | | 1. Patron is viewing the menu 2. Patron selects a food item 3. Patron adds item to their ‘Favorites’ 4. The ‘Favorites’ list on their account is updated 5. COS stores the list in a database | | |
| **Alternative Flows:** | | **Patron removes an item from Favorites**  1a. When a patron is viewing the menu, their favorites will have an icon indicating it is an item on that list.   1. Patron selects the icon to remove the item from the list. 2. Use Case resumes normal flow from step 4   **Patron selects multiple items**  3a. After adding at least one item to the ‘Favorites’ list, the patron can return to the menu to select more items   1. the steps in normal flow will loop until patron is satisfied | | |
| **Exceptions:** | | **Patron does not have an account**   1. COS will prompt the patron to create an account 2. Patron creates an account 3. The ability to manage Favorites is now available | | |
| **Includes:** | | UC-04 | | |
| **Frequency of Use:** | | Approximately 400 users, at least one use per user | | |
| **Special Requirements:** | | 1. Patron must have a COS account 2. Patron can view, add, remove items from Favorites at any time | | |
| **Assumptions:** | | 1. Patron will be able to include customizations when saving favorite meals  2. Patron can automatically include these items or meals when placing an order | | |
| **Notes and Issues:** | | N/A | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | **UC-08** | | | |
| **Use Case Name:** | **Order a Customized Meal** | | | |
| **Related Features** | **FE-01, FE-07** | | | |
| **Created By:** | **Krishna Bhatia** | | **Last Updated By:** | **Tianna Cano** |
| **Date Created:** | **10/1/2021** | | **Last Revision Date:** | **12/18/2021** |
| **Actors:** | | End User | | |
| **Description:** | | This use case helps user to place a customized meal order | | |
| **Trigger:** | | User might order his own recipe. Cafeteria provides the space for the user to create his own recipe | | |
| **Preconditions:** | | End user is registered with COS | | |
| **Postconditions:** | | User successfully places a customized meal order and e-receipt, and SMS receipt will be generated as a token of meal confirmation | | |
| **Normal Flow:** | | 1.0 User opens the COS application.  2.0 COS displays *home screen* including *customize a mea*l, *most rated recipes, account activity, meal plan* and a *search bar* on the top of the screen.  3.0 User selects *customize a meal* option  4.0 COS displays customized meal option screen including *– Ingredient’s list and recipe list*  5.0 User selects the ingredients or recipe and those quantities which he feels to have on his recipe.  6.0 User shall make a mandatory note for his *cooking instructions* or an *additional note* for his customized meal.  7.0 User selects ‘*Add to my Dine’* option to have the meal order in his basket. 8.0 User enters the delivery details and clicks ‘*proceed to checkout’.*  9.0 At the checkout screen, user reconfirms his order and clicks ‘*finalize and pay’*  10.0 COS generates e-receipt and a SMS receipt to the user’s registered mobile number with an ‘*Customized meal Order placed successfully*’ acknowledgement. | | |
| **Alternative Flows:** | | N/A | | |
| **Exceptions:** | | 4.0.E.1 Invalid address or zip code.  COS displays an error message if the user enters an incorrect address detail.  7.0.E.1 Transaction not processed  COS displays an error message if the user’s card details are incorrect or expired and cancels the transaction | | |
| **Includes:** | | 1. COS displays integrated menu for the keyword entered by the user. 1.2. Integrated menu includes the recipe list from Cafeteria menu and from another local restaurant’s menu. | | |
| **Frequency of Use:** | | Approximately 50 users a day and 500 users in a month. | | |
| **Special Requirements:** | | 1. Patron must have a COS account 2. Patron can view, add, remove items from Favorites at any time | | |
| **Assumptions:** | | The End user understands English | | |
| **Notes and Issues:** | | N/A | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | **UC-09** | | | |
| **Use Case Name:** | **Feedback** | | | |
| **Related Features** | **FE-01, FE-02, FE-05, FE-07, FE-011** | | | |
| **Created By:** | **Tianna Cano** | | **Last Updated By:** | **Tianna Cano** |
| **Date Created:** | **9/30/20201** | | **Last Revision Date:** | **12/18/2021** |
| **Actors:** | | Patrons | | |
| **Description:** | | Patrons can rate and provide feedback on the COS which focus on quality of meals and delivery service. | | |
| **Trigger:** | | After receiving their order, the patron will have the option to give feedback on their experience. | | |
| **Preconditions:** | | 1. Patron has an account 2. Patron has placed the order 3. The order has been prepared and delivered or picked up 4. The order or delivery is marked as complete | | |
| **Postconditions:** | | 1. Ratings will be visible as star ratings per food item. 2. Feedback will be available to cafeteria staff, delivery staff, and menu manager to view. | | |
| **Normal Flow:** | | 1. Patron has an account and places an order 2. Patron has either picked up their order or received the delivery 3. The order is marked as complete on the COS by patron or delivery personnel 4. Patron is prompted to leave feedback if they desire 5. Patron provides feedback 6. Feedback is submitted for viewing 7. Star ratings are updated per food item 8. Feedback is emailed to cafeteria staff, delivery staff, and menu manager per week | | |
| **Alternative Flows:** | | **Patron only leaves feedback and rating for meal**  5a. When filling out the survey, the patron can decide what feedback they want to leave   1. Patron can leave the section regarding delivery empty or mark ‘N/A’ 2. normal flow for Use Case resumes from step 6   **Patron only leaves feedback and rating for delivery**  5a. When filling out the survey, the patron can decide what feedback they want to leave   1. Patron can leave the section regarding meal quality empty 2. normal flow for Use Case resumes from step 6   **Patron decides not to give feedback**  4a. When prompted to provide feedback, the user can opt out of the survey   1. Patron selects to opt out of the survey | | |
| **Exceptions:** | | N/A | | |
| **Includes:** | | UC-01, UC-04, UC-017 | | |
| **Frequency of Use:** | | at least once per order | | |
| **Special Requirements:** | | N/A | | |
| **Assumptions:** | | 1. Patron must have an account | | |
| **Notes and Issues:** | | 1. (Optional) Feedback can be displayed for other patrons to view publicly on the COS. 2. (Optional) Staff members can respond to feedback. | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | **UC-10** | | | |
| **Use Case Name:** | **Order From Restaurants** | | | |
| **Related Features** | **FE-01, FE-02** | | | |
| **Created By:** | **Jay Patel** | | **Last Updated By:** | **Tianna Cano** |
| **Date Created:** | **10/2/2021** | | **Last Revision Date:** | **12/2/2021** |
| **Actors:** | | Patrons | | |
| **Description:** | | Patrons will be able to order from different local restaurants | | |
| **Trigger:** | | When Patron goes for selecting their meals, they get an option of placing an order from a selection of local restaurants | | |
| **Preconditions:** | | 1. Patrons need to login to their account to orders.  2. Patrons will select meals from different restaurants. | | |
| **Postconditions:** | | Patron will place the orders from different restaurant | | |
| **Normal Flow:** | | 1. Patron will login to his account  2. Patron will view the menu and any specials offered by local restaurants.  3. Patrons will select to order a meal from a restaurant.  4. Patrons will place the order. | | |
| **Alternative Flows:** | | They might want to remove an item from one restaurant and add same item from another restaurant. | | |
| **Exceptions:** | | Restaurant which are very far away from each other are not feasible for custom orders | | |
| **Includes:** | | UC-01, UC-02, UC-03, UC-04, UC-022 | | |
| **Frequency of Use:** | | Every Time whenever patron selects meal option | | |
| **Special Requirements:** | | N/A | | |
| **Assumptions:** | | That restaurants are nearby and custom orders can be made in a minimum span of time to satisfy patrons as much as possible. | | |
| **Notes and Issues:** | | N/A | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | **UC-11** | | | |
| **Use Case Name:** | **Subscribe to Meal Plan** | | | |
| **Related Features** | **FE-03** | | | |
| **Created By:** | **Krishna Bhatia** | | **Last Updated By:** | **Tianna Cano** |
| **Date Created:** | **10/1/2021** | | **Last Revision Date:** | **12/19/2021** |
| **Actors:** | | Patron | | |
| **Description:** | | This use case helps user to subscribe a meal plan | | |
| **Trigger:** | | Rather than placing a meal order daily, user wishes to subscribe a meal plan, so that meal will be delivered to the user regularly. | | |
| **Preconditions:** | | End user is registered with COS  End user is not subscribed with meal plan | | |
| **Postconditions:** | | User successfully subscribes a meal plan | | |
| **Normal Flow:** | | 1.0 User opens the COS application.  2.0 COS displays *home screen* including *customize a mea*l, *most rated recipes, account activity, meal plan* and a *search bar* on the top of the screen.  3.0 COS displays *‘Meal plan’* screen including ‘subscribe a meal plan’ option. 4.0 User selects meal plan option  5.0 COS displays list of meal plans  6.0 User selects an option among those options and selects the quantity  7.0 User selects ‘*Add to my subscription’* option to have the meal order in his basket.  8.0 User enters the delivery details and clicks ‘*proceed to checkout’.*  9.0 At the checkout screen, user reconfirms his order and clicks ‘*finalize and pay’*  10.0 COS generates e-receipt and a SMS receipt to the user’s registered mobile number with an ‘Meal plan successfully subscribed’ acknowledgement. | | |
| **Alternative Flows:** | |  | | |
| **Exceptions:** | | 8.0.E.1 Invalid address or zip code.  COS displays an error message if the user enters an incorrect address detail.  9.0.E.1 Transaction not processed  COS displays an error message if the user’s card details are incorrect or expired and cancels the transaction | | |
| **Includes:** | | The user has a valid email address | | |
| **Frequency of Use:** | | Approximately 20 users a day, mainly depends on user’s interest | | |
| **Special Requirements:** | | 1. Patron must have a COS account 2. Patron can view, add, remove items from Favorites at any time | | |
| **Assumptions:** | | The End user understands English | | |
| **Notes and Issues:** | | N/A | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | **UC-12** | | | |
| **Use Case Name:** | **Prepare Meal** | | | |
| **Related Features** | **FE-01, FE-02, FE-07, FE-10** | | | |
| **Created By:** | **Rohan Patwardhan** | | **Last Updated By:** | **Tianna Cano** |
| **Date Created:** | **9/30/2021** | | **Last Revision Date:** | **12/19/2021** |
| **Actors:** | | Cafeteria Staff | | |
| **Description:** | | When COS receives an order, COS staff start preparing order. | | |
| **Trigger:** | | Patron places an order | | |
| **Preconditions:** | | 1. COS Staff has all ingredients to prepare meal | | |
| **Postconditions:** | | 1. Notify patron that meal is prepared 2. Request for delivery staff to deliver the order | | |
| **Normal Flow:** | | 1. Patron places an order 2. Cafeteria receives order | | |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | | 1. Patron subscribes to a meal plan 2. Cafeteria receives order according to meal plan selected by patron | | |
| **Exceptions:** | | E-1. If a patron cancels an order, then cafeteria staff need to stop preparing order | | |
| **Includes:** | | UC-01, UC-08, UC-010 | | |
| **Frequency of Use:** | | Approximately 400 | | |
| **Special Requirements:** | | N/A | | |
| **Assumptions:** | | N/A | | |
| **Notes and Issues:** | | N/A | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | **UC-13** | | | |
| **Use Case Name:** | **Request Delivery** | | | |
| **Related Features** | **FE-05, FE-10** | | | |
| **Created By:** | **Rohan Patwardhan** | | **Last Updated By:** | **Tianna Cano** |
| **Date Created:** | **9/30/2021** | | **Last Revision Date:** | **12/19/2021** |
| **Actors:** | | Cafeteria Staff, Delivery Staff | | |
| **Description:** | | Once an order is prepared, COS will look for delivery guys to pickup the order | | |
| **Trigger:** | | Meal is prepared | | |
| **Preconditions:** | | 1. COS receives an order | | |
| **Postconditions:** | | 1. COS staff will hand over the prepared food to delivery guy for delivery. 2. COS staff will update status of order | | |
| **Normal Flow:** | | 1. COS receives order 2. COS staff prepares meal according to the order 3. COS staff press ‘Request for delivery’ option | | |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | | 1. COS staff prepares meals according to meal subscriptions | | |
| **Exceptions:** | | E-1. If a patron cancels an order then delivery guy should not be assigned for delivery | | |
| **Includes:** | | UC-011 | | |
| **Frequency of Use:** | | Approximately | | |
| **Special Requirements:** | |  | | |
| **Assumptions:** | | Delivery Guy is available to pickup the order | | |
| **Notes and Issues:** | |  | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | **UC-14** | | | |
| **Use Case Name:** | **Update Order Status** | | | |
| **Related Features** | **FE-10** | | | |
| **Created By:** | **Rohan Patwardhan** | | **Last Updated By:** | **Tianna Cano** |
| **Date Created:** | **9/30/2021** | | **Last Revision Date:** | **12/19/2021** |
| **Actors:** | | Cafeteria Staff, Delivery Staff | | |
| **Description:** | | Patron can track status of order | | |
| **Trigger:** | | Meal is prepared | | |
| **Preconditions:** | | 1. COS has received an order 2. Order is prepared | | |
| **Postconditions:** | | 1. Delivery guy will pick up the order for delivery | | |
| **Normal Flow:** | | 1. Meal is prepared 2. COS staff press ‘Request for delivery’ option 3. Delivery Guy is assigned for the order 4. COS staff hand over meal to delivery guy | | |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | | 1. Meal is prepared 2. Hand over food to patron | | |
| **Exceptions:** | |  | | |
| **Includes:** | | UC-011, UC-012 | | |
| **Frequency of Use:** | | Approximately 400 | | |
| **Special Requirements:** | |  | | |
| **Assumptions:** | |  | | |
| **Notes and Issues:** | |  | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | **UC-15** | | | |
| **Use Case Name:** | **Print Delivery Instructions** | | | |
| **Related Features** | **FE-01, FE-02, FE-05, FE-07, FE-10** | | | |
| **Created By:** | **Jennifer Wu** | | **Last Updated By:** | **Tianna Cano** |
| **Date Created:** | **9/30/2021** | | **Last Revision Date:** | **12/19/2021** |
| **Actors:** | | Cafeteria Staff | | |
| **Description:** | | After payment is completed, COS prints out the delivery instruction on a printer. And a delivery staff will pick up the delivery instruction printout. The instruction includes patron information, order meals, delivery address, and note. | | |
| **Trigger:** | | Upon an order is finalized | | |
| **Preconditions:** | | Patron completed the payment. | | |
| **Postconditions:** | | delivery staff can deliver the order. | | |
| **Normal Flow:** | | 1. Patrons complete the order payment. 2. COS connects printer to print the order delivery instruction 3. Cafeteria staff receive the instruction. 4. Cafeteria staff pass the instruction to delivery staff. | | |
| **Alternative Flows:** | | 3a. In normal flow step 3, If patron change the order address:  1. cafeteria staff manually change the address on the instruction.  2. Use Case resumes on step 4 of normal flow | | |
| **Exceptions:** | | Printer is not available to use   1. COS will buffer will send email or other electronic massage before a new printer is available. | | |
| **Includes:** | | N/A | | |
| **Frequency of Use:** | | On every order which payment is completed. | | |
| **Special Requirements:** | | Printer must support printing at a speed of 60 color pages per minute. | | |
| **Assumptions:** | | The order payment is completed | | |
| **Notes and Issues:** | | 1. Printer getting older and experience paper jam  2. Need to check printer regularly to avoid out of ink on a busy day. | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | **UC-16** | | | |
| **Use Case Name:** | **Deliver Meals** | | | |
| **Related Features** | **FE-05, FE-10, FE-11** | | | |
| **Created By:** | **Rohan Patwardhan** | | **Last Updated By:** | **Tianna Cano** |
| **Date Created:** | **9/30/2021** | | **Last Revision Date:** | **12/19/2021** |
| **Actors:** | | Delivery Staff | | |
| **Description:** | | Delivery guy will pick up meals from COS staff and deliver the meals to their designated destinations. | | |
| **Trigger:** | | Delivery Guy picks up meal | | |
| **Preconditions:** | | 1. Order is placed 2. Meal is prepared 3. Delivery is requested | | |
| **Postconditions:** | | 1. Meal is delivered to the patron | | |
| **Normal Flow:** | | 1. Delivery Guy picks up the order and delivery instructions 2. Delivery Guy log in COS system and select ‘order on its way’ function to update status of order | | |
| **Alternative Flows:** | | N/A | | |
| **Exceptions:** | | N/A | | |
| **Includes:** | | N/A | | |
| **Frequency of Use:** | | Approximately 300 | | |
| **Special Requirements:** | | N/A | | |
| **Assumptions:** | | Delivery Guys are available  Delivery Guys have resources like vehicle to deliver the meals | | |
| **Notes and Issues:** | | N/A | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | **UC-17** | | | |
| **Use Case Name:** | **Update Delivery Status** | | | |
| **Related Features** | **FE-01, FE-05, FE-10** | | | |
| **Created By:** | **Jay Patel** | | **Last Updated By:** | **Tianna Cano** |
| **Date Created:** | **10/02/2021** | | **Last Revision Date:** | **12/2/2021** |
| **Actors:** | | Delivery staff | | |
| **Description:** | | After picking up the order by delivery guy, location of the order will be tracked real time so that patrons know about delivery status | | |
| **Trigger:** | | At that time when the delivery guy picks up an order from SRS staff and he turns on location tracking on his device. | | |
| **Preconditions:** | | Delivery staff should collect the order first, in order to turn on tracking and link the order number to his device. | | |
| **Postconditions:** | | 1. Patrons can track the status of their orders every minute. 2. Whether the order is delayed or going to reached early at their destination, status update of the order will be given to the patron. | | |
| **Normal Flow:** | | 1. The delivery staff open COS system 2. The delivery staff login into his account 3. The delivery staff select an order that he want to delivery 4. The delivery staff go to restaurant and pick up the meal 5. The delivery staff turns on his location so that patron can track the order 6. The delivery staff will go out to deliver the meal. | | |
| **Alternative Flows:** | | 1. Delivery time may vary due to unconditional circumstances 2. Sometimes orders may get canceled during the delivery process so delivery staff may need to update to order status and also inform the COS staff. | | |
| **Exceptions:** | | 6a. In normal flow step5, if the delivery staff does not have a stable internet, then one needs to ‘Find a location with a stable internet’ so that he can update the estimated delivery time when he picks up the order. | | |
| **Includes:** | | N/A | | |
| **Frequency of Use:** | | Confirm Delivery is needed on every delivery order. | | |
| **Special Requirements:** | | N/A | | |
| **Assumptions:** | | We assume that delivery guy has proper device and good internet service to be able to keep on updating delivery status | | |
| **Notes and Issues:** | | N/A | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | **UC-18** | | | |
| **Use Case Name:** | **Confirm Delivery** | | | |
| **Related Features** | **FE-05, FE-10** | | | |
| **Created By:** | **Jennifer Wu** | | **Last Updated By:** | **Tianna Cano** |
| **Date Created:** | **9/30/2021** | | **Last Revision Date:** | **12/19/2021** |
| **Actors:** | | Delivery staff | | |
| **Description:** | | When a delivery staff delivers the order to the patron, he/she will mark the order on COS as completed, and COS will close the order case. | | |
| **Trigger:** | | At the time when a delivery staff successfully delivered a patron’s order to the patron. | | |
| **Preconditions:** | | A delivery staff gets the meal that a patron ordered and delivery instruction to where to deliver the order to the patron. | | |
| **Postconditions:** | | The order is delivered successfully, and order is marked on the COS as closed | | |
| **Normal Flow:** | | 1. 1. The delivery staff open COS system 2. 2. The delivery staff login into his account 3. 3. The delivery staff select an order that he want to delivery 4. 4. The delivery staff go to restaurant and pick up the meal 5. 5. The delivery staff turns on his location so that patron can track the order 6. 6. The delivery staff will go out to deliver the meal | | |
| **Alternative Flows:** | | **9. COS cannot get confirmation from the patron**  9a. A patron may postpone the feedback to COS   1. COS will temporarily propone to get the confirmation with the patron 2. COS will give a temporary Confirm Delivery to the delivery staff 3. The delivery staff resumes to Normal Flow step10. | | |
| **Exceptions:** | | 6a. In normal flow step6, if delivery staff does not have stable internet to confirm the order delivery   1. Find a location with a stable internet. 2. Use Case resumes on step 6of normal flow   6b. In normal flow step6, if delivery staff does not have stable internet to confirm the order delivery   1. Call the restaurant and complete the order delivery by oral.   Use Case resumes on step 6of normal flow. | | |
| **Includes:** | | N/A | | |
| **Frequency of Use:** | | Confirm Delivery is needed on every delivery order. | | |
| **Special Requirements:** | | 1. Need stable internet connection for a delivery staff to login COS from the field. | | |
| **Assumptions:** | | 1. Almost all patrons will give feedback or confirmation after receiving from delivery staff. | | |
| **Notes and Issues:** | | 1. Some reward systems would be helpful to encourage patrons to confirm the delivery as soon as possible. | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | **UC-19** | | | |
| **Use Case Name:** | **Create Menu** | | | |
| **Related Features** | **FE-06** | | | |
| **Created By:** | **Krishna Bhatia** | | **Last Updated By:** | **Tianna Cano** |
| **Date Created:** | **10/1/2021** | | **Last Revision Date:** | **12/19/2021** |
| **Actors:** | | Menu Manager | | |
| **Description:** | | This use case helps menu manager to create a menu | | |
| **Trigger:** | | Menu manager collects various menus from Cafeteria and other restaurants and creates an integrated menu list for the users. | | |
| **Preconditions:** | | Menu manager is registered with COS using his employee ID. | | |
| **Postconditions:** | | Menu Manager successfully creates a new menu | | |
| **Normal Flow:** | | 1.0 Menu manager opens COS from his device.  2.0 COS displays *home screen* for menu manager including – *create menu, current menu, update ingredients list*  3.0 Menu manager selects his option from the home screen.  4.0 Menu manager provides menu name, menu category, various recipes in the menu and ingredients for the recipes.  5.0 Menu manager rechecks and clicks save.  6.0 COS generates an acknowledgement intimating *Menu created successfully* | | |
| **Alternative Flows:** | | N/A | | |
| **Exceptions:** | | 3.0.E.1 Menu name not provided  3.0.E.2 Menu category not provided | | |
| **Includes:** | | Menu manager collects various menus from Cafeteria and other restaurants. | | |
| **Frequency of Use:** | | Once Per week only done by the Menu Manager. | | |
| **Special Requirements:** | | Menu manager must have an ID | | |
| **Assumptions:** | | Menu Manager must know about all the meals | | |
| **Notes and Issues:** | | N/A | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | **UC-20** | | | |
| **Use Case Name:** | **Update Menu** | | | |
| **Related Features** | **FE-06** | | | |
| **Created By:** | **Krishna Bhatia** | | **Last Updated By:** | **Tianna Cano** |
| **Date Created:** | **10/1/2021** | | **Last Revision Date:** | **12/19/2021** |
| **Actors:** | | Menu Manager | | |
| **Description:** | | This use case helps menu manager to update a menu | | |
| **Trigger:** | | Menu manager collects various menus from Cafeteria and other restaurants and creates an integrated menu list for the users. | | |
| **Preconditions:** | | Menu manager is registered with COS using his employee ID. | | |
| **Postconditions:** | | Menu Manager successfully performs his activity with the menu list | | |
| **Normal Flow:** | | 1.0 Menu manager opens COS from his device.  2.0 COS displays *home screen* for menu manager including – *create menu, current menu list, update ingredients list*  3.0 Menu manager selects *current menu* option from the home screen.  4.0 COS displays list of menus on the screen  ​​5.0 Menu manager selects his choice  6.0 Menu manager updates any of the following options - menu name, menu category, various recipes in the menu and ingredients for the recipes, delete this menu.  7.0 Menu manager rechecks and clicks save.  8.0 COS generates an acknowledgment intimating *Changes saved successfully.* | | |
| **Alternative Flows:** | | N/A | | |
| **Exceptions:** | | 8.0.E.1 Invalid address or zip code.  COS displays an error message if the user enters an incorrect address detail.  9.0.E.1 Transaction not processed  COS displays an error message if the user’s card details are incorrect or expired and cancels the transaction | | |
| **Includes:** | | Menu manager collects various menus from Cafeteria and other restaurants. | | |
| **Frequency of Use:** | | 5 – 10 times a day, depending on the need | | |
| **Special Requirements:** | | Menu Manager has a valid ID | | |
| **Assumptions:** | | The Menu Manager understands English | | |
| **Notes and Issues:** | | N/A | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | **UC-21** | | | |
| **Use Case Name:** | **Manage Inventory** | | | |
| **Related Features** | **FE-01, FE-02, FE-03, FE-04, FE-05, FE-06, FE-07, FE-09, FE-11** | | | |
| **Created By:** | **Jennifer Wu** | | **Last Updated By:** | **Tianna Cano** |
| **Date Created:** | **9/30/2021** | | **Last Revision Date:** | **12/2/2021** |
| **Actors:** | | Inventory manager | | |
| **Description:** | | Inventory managers update, create, retrieve, delete the inventory to reflect the real stock on inventory. | | |
| **Trigger:** | | 1. When ingredients are purchased 2. When ingredients are consumed for cooking meals. | | |
| **Preconditions:** | | 1. Ingredients purchasing numbers are reported by purchasing staff either online or recorded on paper. 2. Kitchen staff report ingredients on a daily basis. | | |
| **Postconditions:** | | 1. Inventory of all ingredients are correctly recorded on COS on a daily basis. 2. Inventory of all ingredients can be searched on COS in real-time. | | |
| **Normal Flow:** | | 1. New data of Ingredients are reported by meal preparing staff.    1. Purchasing staff report the increase of some ingredients    2. kitchen staff report the consumption of some ingredients in inventory 2. An inventory login COS, and updates the new ingredient numbers either from purchasing staff or kitchen staff on COS. 3. The inventory report to his/her inventory manager for the updating of new inventory. 4. The inventory manager checks the inventory number of all ingredients on COS and compares it with stock in inventory. 5. The Inventory manager confirms the new inventory is correct. | | |
| **Alternative Flows:** | | 2a. In normal flow step 2 to 5, if kitchen staff need the new purchasing ingredients urgently, then it might not have enough time to wait for all steps to finish.   1. Special notes need to be recorded on COS with the permission of inventory for kitchen staff to use the new ingredients first. 2. Step 2 to 5 still need to be filled in by an inventory staff. | | |
| **Exceptions:** | | The information of purchasing a particular ingredient comes after it is consumed by kitchen staff, hence a minus number is shown on inventory.   1. Inventory managers need to make a special note on COS when such issues happen and correct the number on COS.. | | |
| **Includes:** | | N/A | | |
| **Frequency of Use:** | | Inventory are updated about three to five times every day | | |
| **Special Requirements:** | | 1. COS gives higher security clearance to inventory managers than ordinary inventory staff. 2. COS will show clear information for the inventory number of all ingredients. 3. When the stock of an ingredient is insufficient, COS will issue an alarm. | | |
| **Assumptions:** | | 1. Kitchen staff will calculate clearly about what and how many ingredients they have used on a daily basis. | | |
| **Notes and Issues:** | | 1. COS must be able to record the different expiration dates of the same ingredient | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | **UC-22** | | | |
| **Use Case Name:** | **Display Restaurant Specials** | | | |
| **Related Features** | **FE-01, FE-02** | | | |
| **Created By:** | **Tianna Cano** | | **Last Updated By:** | **Tianna Cano** |
| **Date Created:** | **10/2/2021** | | **Last Revision Date:** | **12/2/2021** |
| **Actors:** | | Menu Manager | | |
| **Description:** | | The menu manager will display specials offered by local restaurants on the COS menu page. | | |
| **Trigger:** | | Local restaurants offer specials for the company | | |
| **Preconditions:** | | 1. Menu manager is in contact with local restaurants  2. Local restaurants agree to offer specials that can be displayed on the COS | | |
| **Postconditions:** | | Patron will be able to select meals from these restaurants through COS | | |
| **Normal Flow:** | | 1. Menu manager logs in as a staff member on COS.  2. COS displays the menu manager interface.  3. Menu manager selects to create a menu.  4. Menu manager enters order information for the restaurant menu items.  5. Menu manager saves this as a menu item.  6. COS confirms this item’s creation and displays it on the menu page. | | |
| **Alternative Flows:** | | Menu manger needs to edit information  3a. Menu manager can select an existing menu item that displays specials offered by restaurants and edit the information  System resumes normal flow at step 4  Menu manager needs to delete restaurant special  3b. Menu manager can delete an existing menu item that displays specials offered by restaurants  4. COS confirms item was deleted and it is removed from the menu page | | |
| **Exceptions:** | | N/A | | |
| **Includes:** | | N/A | | |
| **Frequency of Use:** | | Every time a local restaurant wants to partner with the company’s cafeteria staff on COS | | |
| **Special Requirements:** | | Restaurants must be in contact and in agreement with what is offered on COS. | | |
| **Assumptions:** | | N/A | | |
| **Notes and Issues:** | | N/A | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | **UC-23** | | | |
| **Use Case Name:** | **Request Delivery from Restaurant** | | | |
| **Related Features** | **FE-02, FE-05** | | | |
| **Created By:** | **Krishna Bhatia** | | **Last Updated By:** | **Tianna Cano** |
| **Date Created:** | **10/1/2021** | | **Last Revision Date:** | **12/2/2021** |
| **Actors:** | | Menu Manager | | |
| **Description:** | | This use case helps menu manager request meal delivery for the completed restaurant order | | |
| **Trigger:** | | A local restaurant completes the meal order that was placed by a Patron through COS | | |
| **Preconditions:** | | 1. Menu manager is registered with COS using his employee ID.  2. Local restaurant has accepted the order. | | |
| **Postconditions:** | | Menu manager successfully request delivery for the completed meal. | | |
| **Normal Flow:** | | 1.0 Menu manager opens COS from his device.  2.0 COS displays a home *screen* for menu manager including – *Accept Order, Current Orders, Completed orders.*  3.0 Menu manager selects *Current orders* from the home screen.  4.0 COS displays a list of current orders on the screen.  5.0 Menu manager selects his desired current order.  6.0 COS displays the details of the current order including *request delivery* option.  7.0 Menu manager selects request delivery on the screen.  8.0 COS generates an acknowledgement on the screen intimating *“Request delivery sent”.* | | |
| **Alternative Flows:** | | 8.1 COS automatically updates the Order status as *“Requested for Delivery”* 6.0 COS also displays the order to the meal deliverer as *“Incoming delivery request”.* | | |
| **Exceptions:** | | N/A | | |
| **Includes:** | | N/A | | |
| **Frequency of Use:** | | 50 – 60 times a day for a menu manager | | |
| **Special Requirements:** | | N/A | | |
| **Assumptions:** | | The menu manager know the ingredients of the meal by its name. | | |
| **Notes and Issues:** | | N/A | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | **UC-24** | | | |
| **Use Case Name:** | **Manage Orders** | | | |
| **Related Features** | **FE-01, FE-02, FE-05, FE-10** | | | |
| **Created By:** | **Jennifer Wu** | | **Last Updated By:** | **Tianna Cano** |
| **Date Created:** | **10/1/2021** | | **Last Revision Date:** | **12/2/2021** |
| **Actors:** | | Patron, menu manager, cafeteria staff, delivery staff | | |
| **Description:** | | Actors update the order status from ordering, pending, payment completed, preparing, delivering, and order completed | | |
| **Trigger:** | | Whenever the status of an order changed, from payment pending to payment completed, from meal preparing to delivering, and to delivery confirmed. | | |
| **Preconditions:** | | The status and the update of status of an order status are clearly recorded. | | |
| **Postconditions:** | | The order case is completed and COS close the case. | | |
| **Normal Flow:** | | 1. Patron opens COS on his device. 2. Patron adds meals to the shopping cart, COS create a order and updates order status to “ordering”. 3. Patron confirms order, COS updates order status to “order checkout”. 4. Patron completes payment, COS updates order status to “payment complete”. 5. Menu manager receives the order instruction, menu manager click receive button on his COS device, COS updates order status to “preparing” 6. Order is ready and passed to cafeteria staff. COS updates order status to “order is ready for delivering” 7. Delivery staff receive order and start delivering. COS update order status to “order is delivering” 8. Order is delivered to the patron delivery staff click the complete bottom on his COS device. COS update order status to “order delivered”. 9. COS closes the order case 10. Menu manager can access order cases 11. Menu manager views orders 12. Menu manager can print order records | | |
| **Alternative Flows:** | | 4a. In normal flow step4, if the payment is decline   1. Update status to “payment declined” - the payment is declined. 2. Patron retry the payment method 3. Use Case resumes on step 4 of normal flow   6a. In normal flow step 6, if the patron picks up in a restaurant or the assigned location.   1. Update status to “order is ready for pickup” 2. Patron come to pick up, COS updates order status to “order is picked up” 3. Use Case resumes on step 0 of normal flow | | |
| **Exceptions:** | | N/A | | |
| **Includes:** | | N/A | | |
| **Frequency of Use:** | | every order update | | |
| **Special Requirements:** | | Actors have stable internet | | |
| **Assumptions:** | | If patron doesn’t order completely in 30 minutes, the order case will close by COS | | |
| **Notes and Issues:** | | N/A | | |

**TABLE 1: Traceability Matrix (Features & Use-cases).**

|  |  |
| --- | --- |
|  | **Related Use-cases (UCs)** |
| **FE-01** | **UC-01, UC-02, UC-03, UC-04** |
| **FE-02** | **UC-01, UC-03, UC-04, UC-08** |
| **FE-03** | **UC-04, UC-10** |
| **FE-04** | **UC-04, UC-05** |
| **FE-05** | **UC-01, UC-04** |
| **FE-06** | **UC-18, UC-19** |
| **FE-07** | **UC-01, UC-04, UC-08** |
| **FE-08** | **UC-04, UC-06** |
| **FE-09** | **UC-04, UC-07** |
| **FE-10** | **UC-01, UC-04** |
| **FE-11** | **UC-01, UC-04, UC-09** |

# External Interface Requirements

## User Interfaces

UC-1: The COS application shall conform to user interface standards for web and mobile applications.

UC-2: The system shall provide a link to the tutorial HTML page from either the web or mobile applications.

UC-3: The web and mobile applications shall contain forward and backwards navigation between application pages.

UC-4: When accessing the application on a mobile device, the application pages will automatically adjust to the device screen size and resolution.

## Hardware Interfaces

No hardware interfaces had been identified.

## Software Interfaces

SI-1: Cafeteria Inventory System

SI-1.1: The COS staff shall input the quantities of food items ordered to the Cafeteria Inventory System through an interface.

SI-1.2: The COS shall use the Cafeteria Inventory System to determine whether a requested food item is available.

SI-1.3: When the Cafeteria Inventory System notifies the COS staff that a specific food item is no longer available, the COS shall remove that food item from the menu for the current date.

SI-2: Patron Membership System

SI-2.1: The COS shall validate the membership details of the user to validate their membership through a database:

SI-2.2: To allow patron to register for their membership if they are not part of it.

SI-2.3: To give perks and rewards of different types of patron membership.

SI-2.4: To check whether a patron’s membership has expired of not.

SI-3: Patron Favorites System

SI-3.1: The COS shall hold a record of food items that patrons have saved as ‘Favorites’ which will be made available to automatically order in a database.

SI-3.2: The patron shall be able to add or remove food items at which the system will update the records.

SI-4: Order Tracking System

SI-4.1: The COS shall display updated order statuses when an order is placed.

SI-4.2: The system will allow the patron to view the status of their order after it is placed.

SI-4.3 The cafeteria staff shall be able to update the order status when the meal is being prepared and when it is ready for pickup.

SI-4.4: If the order is marked for delivery, the delivery staff will update the order status when the delivery is in progress and when it has been delivered to the designated location.

SI-4.5: When an order is picked up, the delivery staff shall mark the order as complete.

## Communications Interfaces

CI-1: The COS shall send an email receipt to the patron with order details and delivery instructions as entered by the patron.

CI-2: The COS shall send an email to the patron to acknowledge a successful meal plan subscription with details of the meal plan.

CI-3: The COS shall send a notification to COS staff once patron places an order.

CI-4: COS staff shall send a notification to delivery staff once meal is prepared.

# Other Nonfunctional Requirements

## Performance Requirements

User Experience is utmost in this market and is critical for success. Thus, system performance should adhere to strict requirements.

PE-1: The application will show order confirmation to the users within 5 seconds after the transaction is completed.

PE-2: Response time of all the webpages / applications takes at most 1 seconds.

PE-3: All queries which are submitted online shall be addressed within 2 seconds.

PE-4: The system should support 1000 patrons at peak hours.

## Safety Requirements

SR-1: All meals containing items that might be allergic, shall be highlighted.

SR-2: Calories and Nutrition value of each meal shall be highlighted.

## Security Requirements

SE-1: All personal data of users shall be encrypted such as mobile number and credit card information.

SE-2: All personal information of the employees / delivery personnel shall be encrypted.

SE-3: Users shall only be able to look at their own information and will be restricted from looking at information of any other user.

## Software Quality Attributes

QA-1: The system would be available for both IOS and Android devices.

QA-2: The system would be flexible and easy to use for every group age (13+).

# Other Requirements

OR-1: The system shall incorporate the usage of multiple databases to maintain organization of data, such as data pertaining to individual patrons, like their information and account features, and the organization of orders and cafeteria inventory.

OR-2: The system shall calculate rewards points for every dollar spent in an order.

OR-3: The system will create an expiration date for the rewards points based on the date the points were earned.

OR-4: Reward items and their values will be determined by the cafeteria staff.

# Functional Requirements (FRs)

## FR-01: <Order meals>

a. Introduction/Functionality: *This functional requirement shall allow the patron to place a meal order from the company cafeteria.*

b. Traced: *UC-01*

c. Inputs: *The patron shall have an account on the COS. The patron shall select their meal from the menu.*

d. Processing: *The patron purchases the meal using a valid payment method of their choosing* *and the patron’s bank organization shall process the payment details.*

e. Outputs: *An electronic receipt shall be provided via email and/or text message to confirm the order along with order status.*

f. Error Handling: *COS displays an error message if the patron’s payment information is incorrect or expired and will prompt the user to re-enter the information. COS displays an error message if the date or time upon ordering the meal is too late.*

## FR-02: <View menu>

a. Introduction/Functionality: *This functional requirement shall allow the patron to view the cafeteria menu*

b. Traced: *UC-02*

c. Inputs: *The patron shall have internet or intranet access to view the COS application.*

d. Processing: *Viewing the menu shall allow patrons to select items to order.*

e. Outputs: *COS displays a list of the menu items available throughout each day.*

f. Error Handling: *No error handling available.*

## FR-03: <Submit payment>

a. Introduction/Functionality: *This functional requirement shall allow the patron to submit payments when placing an order*

*b. Traced: UC-03*

c. Inputs: *The patron shall have an account on the COS and have placed an order.*

d. Processing: *When paying for the order, the patron can select a saved payment method or add a new one.*

e. Outputs: *COS displays a list of payment methods to select.*

f. Error Handling: *No error handling available.*

## FR-04: <Create account>

a. Introduction/Functionality: *This functional requirement shall allow the user to create an account on the COS using their employee information*

*b. Traced: UC-04*

c. Inputs: *The user shall be a current employee.*

d. Processing: *If the employee wishes to make ordering from the cafeteria easier, they can create an account using their employee information such as an employee ID number and name. They can use this information and create a password for the account.*

e. Outputs: *COS displays a welcome message*

f. Error Handling: *The COS shall display an error message if the employee information is incorrect and prompt reentry.*

## FR-05: <Manage payment methods>

a. Introduction/Functionality: *This functional requirement shall allow the patron to add valid payment methods to their account for paying for their meals. It is recommended to save the card details in advance for faster ordering.*

*b. Traced: UC-05*

c. Inputs: *The patron shall have an account on the COS.*

d. Processing: *The patron can add, edit, or delete payment information such as card details.*

e. Outputs: *COS displays an updated list of the saved payment methods.*

f. Error Handling: *COS displays an error message if the patron’s payment information is incorrect or expired and will prompt the patron to re-enter the information.*

## FR-06: <Earn rewards points>

a. Introduction/Functionality: *This functional requirement shall allow the patron to earn rewards points for using COS.*

*b. Traced: UC-06*

c. Inputs: *The patron shall have an account on the COS.*

d. Processing: *The patron shall earn points for every order they place using COS*

e. Outputs: *COS maintains and displays the number of points earned and spent by the patron.*

f. Error Handling: *No error handling available.*

## FR-07: <Favorite meals>

a. Introduction/Functionality: *This functional requirement shall allow the patron to add or delete meal items to a ‘Favorites’ category on their COS account. This allows the user to automatically order the item if they wish to.*

*b. Traced: UC-07*

c. Inputs: *The patron shall have an account on the COS.*

d. Processing: *The patron can add or delete a meal item from their ‘Favorites’ category.*

e. Outputs: *COS displays an updated list of the patron’s preferred menu items.*

f. Error Handling: *No error handling available*

## FR-08: <Order custom meals>

a. Introduction/Functionality: *This functional requirement shall allow the patron to customize meals when placing an order from the cafeteria menu.*

*b. Traced: UC-08*

c. Inputs: *The patron shall have an account on the COS.*

d. Processing: *The patron can request a modification to a menu item they wish to order.*

e. Outputs: *An electronic receipt shall be provided via email and/or text message to confirm the order along with order status.*

f. Error Handling: *If a customization is not possible, COS will notify the patron.*

## FR-09: <Give feedback>

a. Introduction/Functionality: *This functional requirement shall allow the user to leave feedback on meals and/or delivery via rating and reviews.*

*b. Traced: UC-09*

c. Inputs: *The patron shall have an account on the COS and have received a meal they ordered.*

d. Processing: *Sometimes after an order was placed, the patron will receive the option to leave feedback regarding the meal and delivery if selected.*

e. Outputs: *COS displays a ‘Thank you’ message.*

f. Error Handling: *No error handling available.*

## FR-10: <Order from local restaurants>

a. Introduction/Functionality: *This functional requirement shall allow the patron to place a meal order from restaurants outside of the company.*

*b. Traced: UC-10*

c. Inputs: *The patron shall have an account on the COS. The patron shall select their meal and the quantity from the restaurant menu.*

d. Processing: *The patron purchases the meal using a valid payment method of their choosing* *and the patron’s bank organization shall process the payment details.*

e. Outputs: *An electronic receipt shall be provided via email and/or text message to confirm the order along with order status.*

f. Error Handling: *COS displays an error message if the patron’s payment information is incorrect or expired and will prompt the user to re-enter the information.*

## FR-11: <Order multiple meals and menu items>

a. Introduction/Functionality: *This functional requirement shall allow the patron to order multiple meals and multiple menu items.*

*b. Traced: UC-01*

c. Inputs: *The patron shall have an account on the COS.*

d. Processing: *The patron can order up to the maximum number of meals or items available in the cafeteria’s inventory.*

e. Outputs: *COS displays a number beside each menu item that the patron can add or subtract from when ordering.*

f. Error Handling: *If the maximum number is selected, COS will not allow the patron to order any more.*

## FR-12: <Create a meal plan subscription>

a. Introduction/Functionality: *This functional requirement shall allow the patron to create a subscription to a meal plan.*

*b. Traced: UC-11*

c. Inputs: *The patron shall have an account on the COS.*

d. Processing: *For various prices, the patron can select a number of meals or menu items pertaining to the selected cost.* *The patron can create a meal plan by selecting meals or menu items for each day they wish to order from the cafeteria.*

e. Outputs: *COS displays a message to confirm the patron has created a meal plan.*

f. Error Handling: *If the patron selects the maximum number of items allowed by the selected plan, COS will not allow the patron to select more or suggest the patron upgrade to a larger plan. If the patron selects too few items, COS will suggest the patron downgrade to a smaller plan.*

## FR-13: <View meal plan subscriptions>

a. Introduction/Functionality: *This functional requirement shall allow the patron to view existing meal plan subscriptions.*

*b. Traced: UC-11*

c. Inputs: *The patron shall have an account on the COS and have created a meal subscription.*

d. Processing: *The patron can view the meal subscription.*

e. Outputs: *COS displays the patron’s meal plan and the patron can view plan details if selecting a specific date.*

f. Error Handling: *No error handling available*

## FR-14: <Modify meal plan subscriptions>

a. Introduction/Functionality: *This functional requirement shall allow the patron to modify an existing meal plan subscription.*

*b. Traced: UC-11*

c. Inputs: *The patron shall have an account on the COS and have created a meal subscription.*

d. Processing: *The patron can select to upgrade or downgrade their meal subscription and recreate their meal plan according to that change.*

e. Outputs: *COS displays the patron’s update meal plan subscription.*

f. Error Handling: *If the patron selects the maximum number of items allowed by the selected plan, COS will not allow the patron to select more or suggest the user upgrade to a larger plan. If the patron selects too few items, COS will suggest the patron downgrade to a smaller plan.*

## FR-15: <Cancel meal plan subscriptions>

a. Introduction/Functionality: *This functional requirement shall allow the patron to cancel an existing meal plan subscription.*

*b. Traced: UC-11*

c. Inputs: *The patron shall have an account on the COS and have created a meal subscription.*

d. Processing: *The patron can select to cancel an existing meal plan subscription.*

e. Outputs: *COS displays a confirmation that the patron has cancelled their subscription.*

f. Error Handling: *No error handling available*

## FR-16: <Select delivery or pickup>

a. Introduction/Functionality: *This functional requirement shall allow the patron to request meal delivery or pickup.*

*b. Traced: UC-1*

c. Inputs: *The patron shall have an account on the COS.*

d. Processing: *When placing an order, the patron can select to have their meal delivered to a designated location or picked up at the cafeteria.*

e. Outputs: *COS displays a confirmation message.*

f. Error Handling: *COS displays an error message if the date or time upon ordering the meal is too late.*

## FR-17: <Prepare meal>

a. Introduction/Functionality: *This functional requirement shall allow the cafeteria staff to update order status for meals in preparation.*

*b. Traced: UC-12*

c. Inputs: *The cafeteria staff shall be logged in as staff members.*

d. Processing: *Upon receiving an order, the cafeteria staff will update the order status to indicate the meal is being prepared. The cafeteria staff will then work on preparing the order.*

e. Outputs: *COS displays an updated status for each meal.*

f. Error Handling: *No error handling available.*

## FR-18: <Request delivery from cafeteria>

a. Introduction/Functionality: *This functional requirement shall allow the cafeteria staff to request delivery for orders.*

*b. Traced: UC-13*

c. Inputs: *The cafeteria staff as a staff member.*

d. Processing: *When meals have been marked for delivery, cafeteria staff will be responsible for sending a delivery request to the delivery staff.*

e. Outputs: *COS displays a request for the delivery along with delivery instructions.*

f. Error Handling: *No error handling available.*

## FR-19: <Update order status>

a. Introduction/Functionality: *This functional requirement shall allow the cafeteria staff to update order status for meals ready for pickup.*

*b. Traced: UC-14*

c. Inputs: *The cafeteria staff shall be logged in as staff members.*

d. Processing: *When meals have been marked for pickup, cafeteria staff will be responsible for updating the order status as ready for pickup.*

e. Outputs: *COS displays an updated status to inform the patron that their meal is ready for pickup.*

f. Error Handling: *No error handling available.*

## FR-20: <Print delivery instructions>

a. Introduction/Functionality: *This functional requirement shall allow the cafeteria staff to print delivery instructions for the delivery staff.*

*b. Traced: UC-15*

c. Inputs: *The cafeteria staff shall be logged in as a staff member.*

d. Processing: *COS will provide delivery instructions for the cafeteria staff to print for the delivery staff.*

e. Outputs: *COS displays the delivery instructions such as the location, the items in the order, time constraints, and any other details if needed.*

f. Error Handling: *No error handling available.*

## FR-21: <Deliver meals>

a. Introduction/Functionality: *This functional requirement shall allow the delivery staff to pick up delivery orders and update the status.*

*b. Traced: UC-16*

c. Inputs: *The delivery staff shall be logged in as staff members.*

d. Processing:  *When delivery staff pick up a delivery order, they will be responsible for delivering it to the designated location in a timely manner by following printed delivery instructions.*

e. Outputs: *COS displays an updated status for a patron’s order.*

f. Error Handling: *No error handling available.*

## FR-22: <Update delivery status>

a. Introduction/Functionality: *This functional requirement shall allow the delivery staff to update the status of a delivery order.*

*b. Traced: UC-17*

c. Inputs: *The delivery staff shall be logged in as staff members.*

d. Processing: *When delivery staff picks up a delivery order and delivers it to the designated location, they can update the delivery to indicate it is ready for pickup.*

e. Outputs: *COS displays an updated status for a patron’s order.*

f. Error Handling: *No error handling available.*

## FR-23: <Confirm delivery>

a. Introduction/Functionality: *This functional requirement shall allow the delivery staff to mark an order as complete once it is picked up*

*b. Traced: UC-18*

c. Inputs: *The delivery staff shall be logged in as staff members.*

d. Processing: *When delivery staff successfully deliver an order to a customer, they can mark the order as complete.*

e. Outputs: *COS displays an updated status for a patron’s order.*

f. Error Handling: *No error handling available.*

## FR-24: <Create cafeteria menus>

a. Introduction/Functionality: *This functional requirement shall allow the menu manager to create cafeteria menus.*

*b. Traced: UC-19*

c. Inputs: *The menu manager shall have an account on the COS with the approved administrative abilities.*

d. Processing: *The menu manager has the ability to create the cafeteria menu by adding meals and items offered by the cafeteria.*

e. Outputs: *COS displays the created menu.*

f. Error Handling: *No error handling available*

## FR-25: <View cafeteria menus>

a. Introduction/Functionality: *This functional requirement shall allow the menu manager to view cafeteria menus.*

*b. Traced: UC-19*

c. Inputs: *The menu manager shall have an account on the COS with the approved administrative abilities and have created a menu.*

d. Processing: *The menu manager can view the menu and select to view more detailed information for each meal or item.*

e. Outputs: *COS displays the menu for the menu manager to review.*

f. Error Handling: *No error handling available*

## FR-26: <Modify cafeteria menus>

a. Introduction/Functionality: *This functional requirement shall allow the menu manager to modify cafeteria menus.*

*b. Traced: UC-20*

c. Inputs: *The menu manager shall have an account on the COS with the approved administrative abilities and have created a menu.*

d. Processing: *The menu manager has the ability to modify the menu by editing meals or menu items accordingly.*

e. Outputs: *COS displays an updated menu with any changes made by the menu manager.*

f. Error Handling: *No error handling available*

## FR-27: <Delete cafeteria menus>

a. Introduction/Functionality: *This functional requirement shall allow the menu manager to delete cafeteria menus.*

*b. Traced: UC-20*

c. Inputs: *The menu manager shall have an account on the COS with the approved administrative abilities and have made a menu.*

d. Processing: *The menu manager can select to delete the menu.*

e. Outputs: *COS displays confirmation that the menu was deleted.*

f. Error Handling: *No error handling available*

## FR-28: <Track order status>

a. Introduction/Functionality: *This functional requirement shall allow the patron to view the status of their order.*

*b. Traced: UC-24*

c. Inputs: *The patron shall have an account on the COS and have placed an order.*

d. Processing: *Viewing the status of the order becomes available after placing an order.*

e. Outputs: *COS displays a status that will update as the order is prepared and once it is ready for pickup or delivery.*

f. Error Handling: *No error handling available.*

## FR-29: <Staff login>

a. Introduction/Functionality: *This functional requirement shall allow the user to log in as a staff member for the COS.*

*b. Traced: UC-12, UC-13, UC-14, UC-15, UC-16, UC-17, UC-18, UC-19, UC-20, UC-21, UC-22, UC-23, UC-24*

c. Inputs: *The user shall have an account on the COS as cafeteria staff, menu manager, or delivery staff.*

d. Processing: *Login shall be accepted using staff employee information and account creation like that for patrons.*

e. Outputs: *COS displays administration access to certain aspects of COS depending on the type of staff.*

f. Error Handling: *COS will display an error message if employee information is incorrect.*

## FR-30: <Manage inventory>

a. Introduction/Functionality: *This functional requirement shall allow the menu manager to maintain the inventory of ingredients and menu items for the cafeteria.*

*b. Traced: UC-21*

c. Inputs: *The menu manager shall be logged in as a staff member.*

d. Processing: *The menu manager shall keep track of the cafeteria’s inventory and update them accordingly.*

e. Outputs: *COS displays an updated list of the cafeteria’s inventory.*

f. Error Handling: *No error handling available.*

## FR-31: <Display restaurant specials>

a. Introduction/Functionality: *This functional requirement shall allow the menu manager to display specials and meals offered by local restaurants.*

*b. Traced: UC-22*

c. Inputs: *The menu manager shall be logged in as a staff member.*

d. Processing: *The menu manager can create specials to be displayed on the COS menu to promote meals offered by local restaurants.*

e. Outputs: *COS displays an updated display of the menu with these deals.*

f. Error Handling: *No error handling available.*

## FR-32: <Modify restaurant specials>

a. Introduction/Functionality: *This functional requirement shall allow the menu manager to modify specials and meals offered by local restaurants.*

*b. Traced: UC-22*

c. Inputs: *The menu manager shall be logged in as a staff member and have created deals offered by local restaurants.*

d. Processing: *The menu manager can edit the meals or specials offered by local restaurants.*

e. Outputs: *COS displays an updated display of the menu with these deals.*

f. Error Handling: *No error handling available.*

## FR-33: <Delete restaurant specials>

a. Introduction/Functionality: *This functional requirement shall allow the menu manager to delete specials and meals offered by local restaurants.*

*b. Traced: UC-22*

c. Inputs: *The menu manager shall be logged in as a staff member and have created deals offered by local restaurants.*

d. Processing: *The menu manager can delete the meals or specials offered by local restaurants.*

e. Outputs: *COS displays an updated display of the menu with these deals.*

f. Error Handling: *No error handling available.*

## FR-34: <Request delivery from restaurant>

a. Introduction/Functionality: *This functional requirement shall allow the menu manager to request delivery for orders.*

*b. Traced: UC-23*

c. Inputs: *The menu manager shall be logged in as a staff member.*

d. Processing: *When meals have been marked for delivery, the menu manager will be responsible for sending a delivery request to the delivery staff.*

e. Outputs: *COS displays a request for the delivery along with delivery instructions.*

f. Error Handling: *No error handling available.*

## FR-35: <Manage order status>

a. Introduction/Functionality: *This functional requirement shall allow the menu manager to update the order status for restaurant orders.*

*b. Traced: UC-24*

c. Inputs: *The menu manager shall be logged in as a staff member.*

d. Processing: *When a restaurant order is placed, the menu manager will update the order status when the meal is prepared to notify the patron.*

e. Outputs: *COS displays an updated status for a patron’s order.*

f. Error Handling: *No error handling available.*

## FR-36: <Print order record>

a. Introduction/Functionality: *This functional requirement shall allow the menu manager to print records of closed orders.*

*b. Traced: UC-24*

c. Inputs: *The menu manager shall be logged in as a staff member.*

d. Processing: *When a restaurant order is placed, the menu manager can view closed orders and print them to maintain records.*

e. Outputs: *COS displays a list of closed orders.*

f. Error Handling: *No error handling available.*

## FR-37: <Update account>

a. Introduction/Functionality: *This functional requirement shall allow the user to update information on their account.*

*b. Traced: UC-04*

c. Inputs: *The user shall be logged in as either a patron or staff member.*

d. Processing: *The user can edit information such as name, email, phone number, etc.*

e. Outputs: *COS displays an updated user interface with these changes.*

f. Error Handling: *If the user changes their mind, they can select to cancel any changes.*

## FR-38: <Delete account>

a. Introduction/Functionality: *This functional requirement shall allow the user to delete their account.*

*b. Traced: UC-04*

c. Inputs: *The user shall be logged in as a patron or staff member.*

d. Processing: *The user will select the option to delete their account.*

e. Outputs: *COS displays a message to indicate the account’s deletion.*

f. Error Handling: *Before initiating the user’s request, COS will display a warning message.*

## FR-39: <Redeem rewards points>

a. Introduction/Functionality: *This functional requirement shall allow the patron to redeem their rewards points for discounts or free items.*

*b. Traced: UC-06*

c. Inputs: *The patron shall be logged in as a patron.*

d. Processing: *When viewing their points or placing an order, patrons can select to redeem points if they equate to an applicable reward.*

e. Outputs: *COS displays an updated count of the points and any discounts applied to the total.*

f. Error Handling: *If there is an insufficient number of points, the patron will not have the option to use points.*

## FR-40: <Rewards points expire>

a. Introduction/Functionality: *This functional requirement shall notify patrons when and how many points are set to expire*

*b. Traced: UC-06*

c. Inputs: *The patron shall be logged in as a patron.*

d. Processing: *When points are set to expire soon, COS will notify the patron. Points will expire*

e. Outputs: *COS displays an updated status for a patron’s order.*

f. Error Handling: *No error handling available.*

e. Outputs: *COS displays an updated status for a patron’s order.*

f. Error Handling: *No error handling available.*

## FR-41: <Create staff account>

a. Introduction/Functionality: *This functional requirement shall allow users to create accounts with staff credentials*

*b. Traced: UC-04, UC-12, UC-13, UC-14, UC-15, UC-16, UC-17, UC-18, UC-19, UC-20, UC-21, UC-22, UC-23, UC-24*

c. Inputs: *The user shall be a current employee.*

d. Processing: *If the employee is a cafeteria or delivery staff member, they can create an account using their employee information such as an employee ID number and name. They can use this information and create a password for the account.*

e. Outputs: *COS displays a welcome message*

f. Error Handling: *The COS shall display an error message if the employee information is incorrect and prompt reentry.*

## FR-42: <View order history>

a. Introduction/Functionality: *This functional requirement shall allow patrons to view previous orders*

*b. Traced: UC-01, UC-04*

c. Inputs: *The patron shall have a COS account and placed orders in the past.*

d. Processing: *Patrons can select to view past orders and can select individual orders to view more detailed information.*

e. Outputs: *COS displays a list of previous orders.*

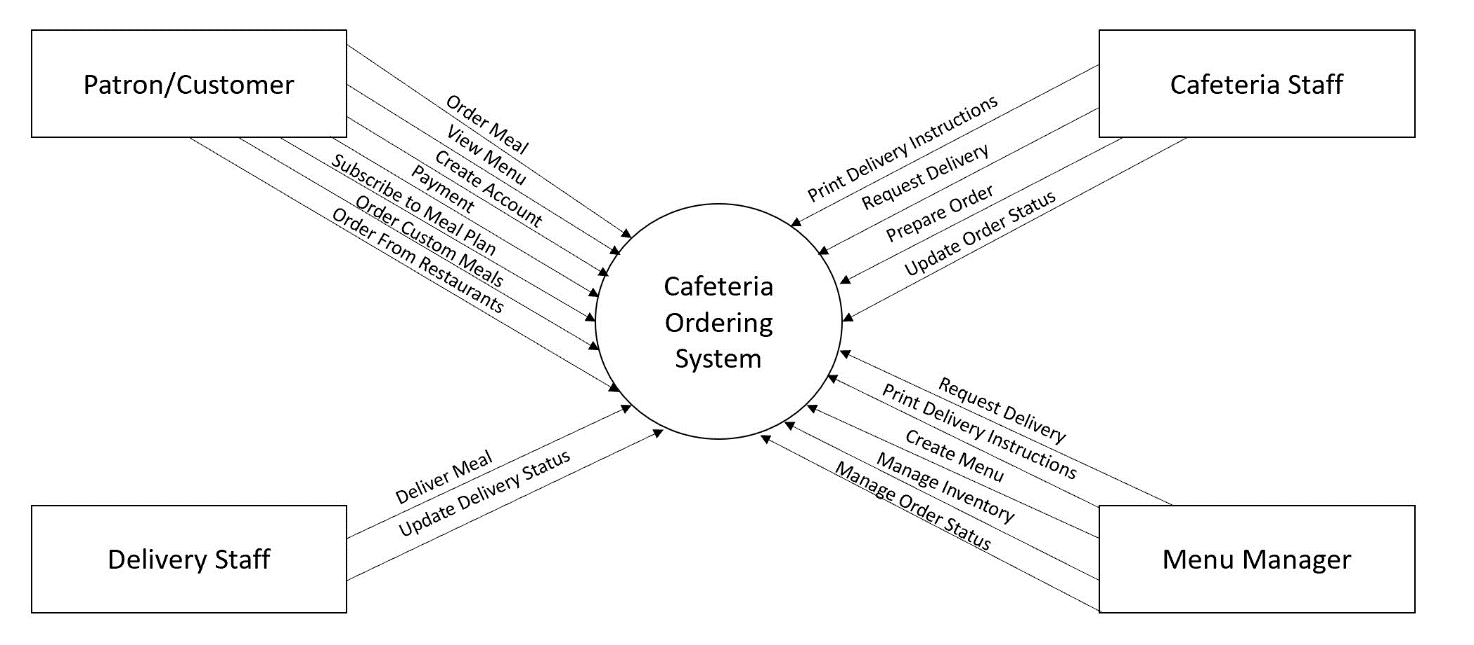
f. Error Handling: *No error handling available*

# TABLE 2: Traceability Matrix (Use-cases & FRs).

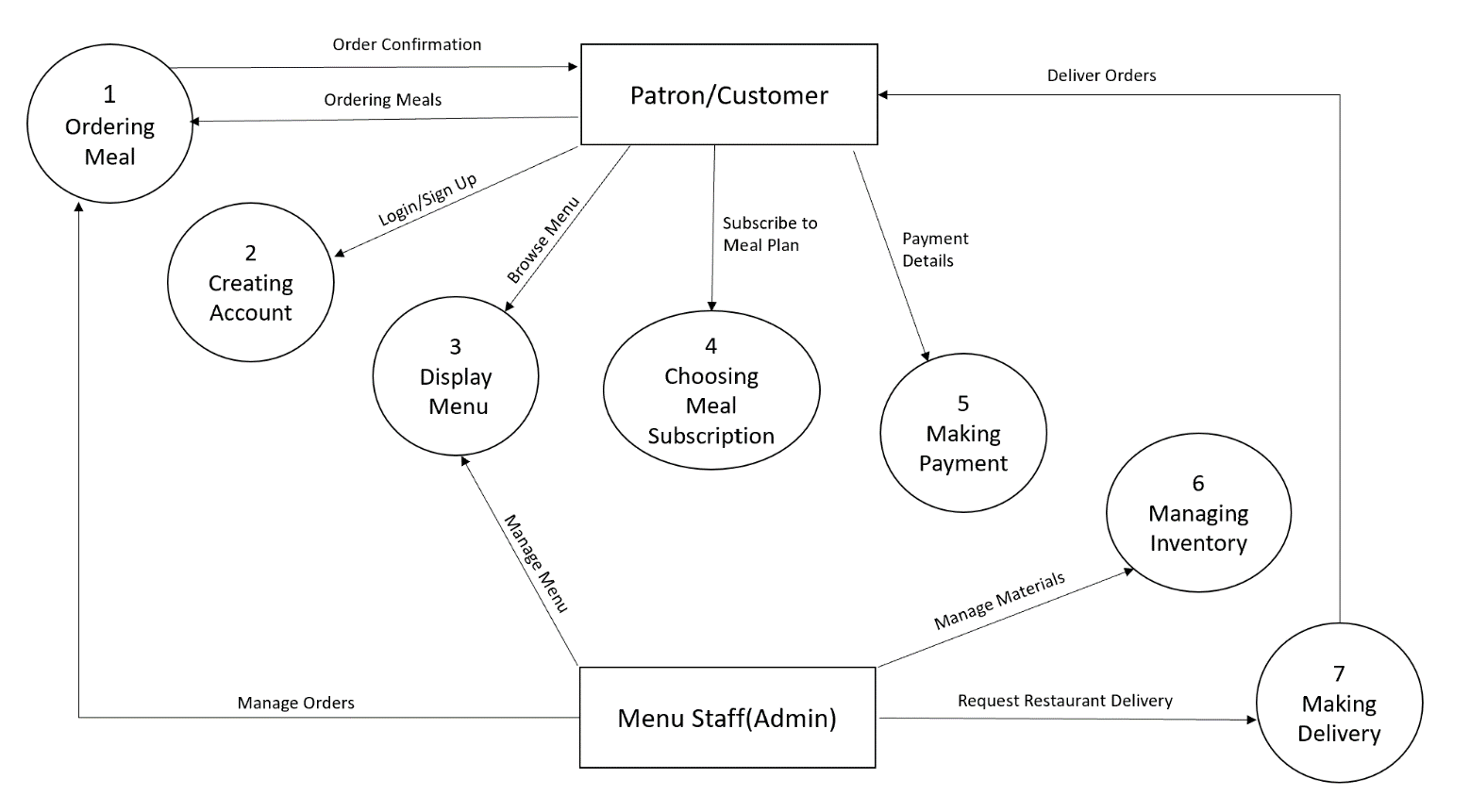
|  |  |
| --- | --- |
|  | **Related FRs** |
| **UC-01** | FR-01, FR-11, FR-16, FR-42 |
| **UC-02** | FR-02 |
| **UC-03** | FR-03 |
| **UC-04** | FR-04, FR-37, FR-38, FR-40, FR-41, FR-42 |
| **UC-05** | FR-05 |
| **UC-06** | FR-06, FR-39, FR-40 |
| **UC-07** | FR-07 |
| **UC-08** | FR-08 |
| **UC-09** | FR-09 |
| **UC-10** | FR-10 |
| **UC-11** | FR-12, FR-13, FR-14, FR-15 |
| **UC-12** | FR-17, FR-29, FR-41 |
| **UC-13** | FR-18, FR-29, FR-41 |
| **UC-14** | FR-19, FR-29, FR-41 |
| **UC-15** | FR-20, FR-29, FR-41 |
| **UC-16** | FR-21, FR-29, FR-41 |
| **UC-17** | FR-22, FR-29, FR-41 |
| **UC-18** | FR-23, FR-29, FR-41 |
| **UC-19** | FR-24, FR-25, FR-29, FR-41 |
| **UC-20** | FR-26, FR-27, FR-29, FR-41 |
| **UC-21** | FR-29, FR-30, FR-41 |
| **UC-22** | FR-29, FR-31, FR-32, FR-33 FR-41 |
| **UC-23** | FR-29, FR-34, FR-41 |
| **UC-24** | FR-28, FR-29, FR-35, FR-36, FR-41 |

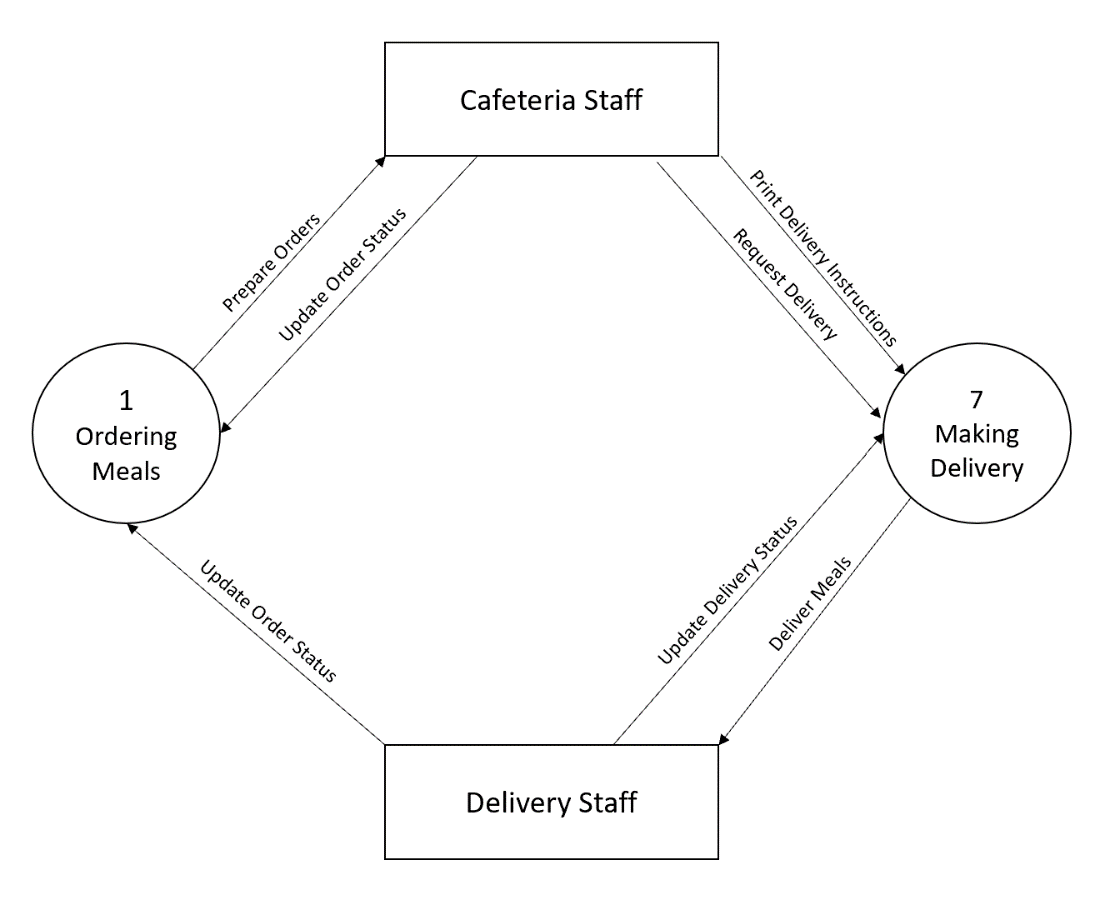
# Functional Modeling (DFDs)

## Context Diagram (level 0)

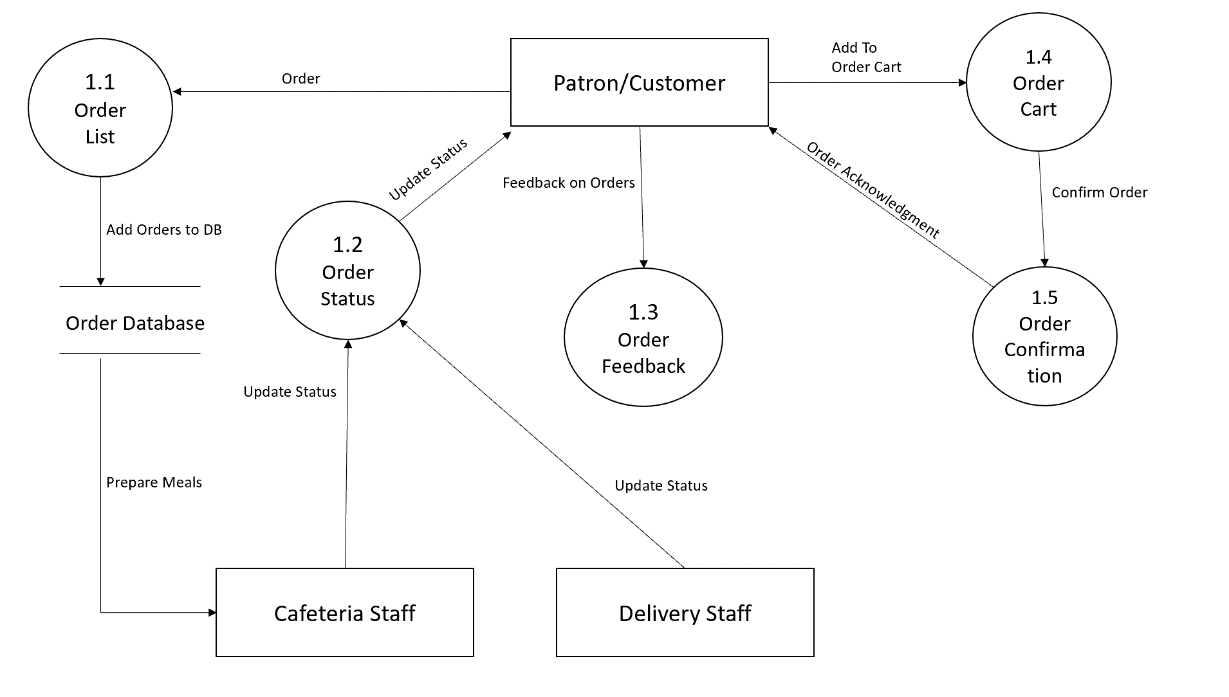


## Level 1

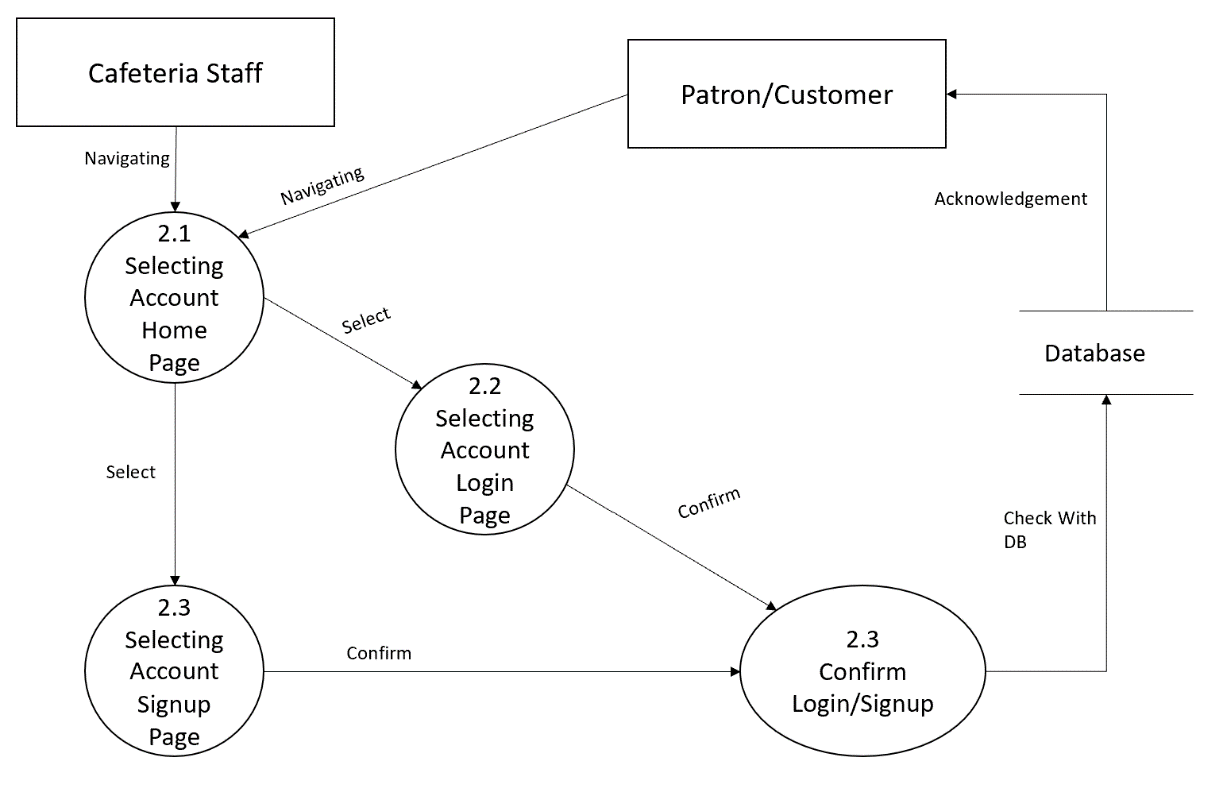




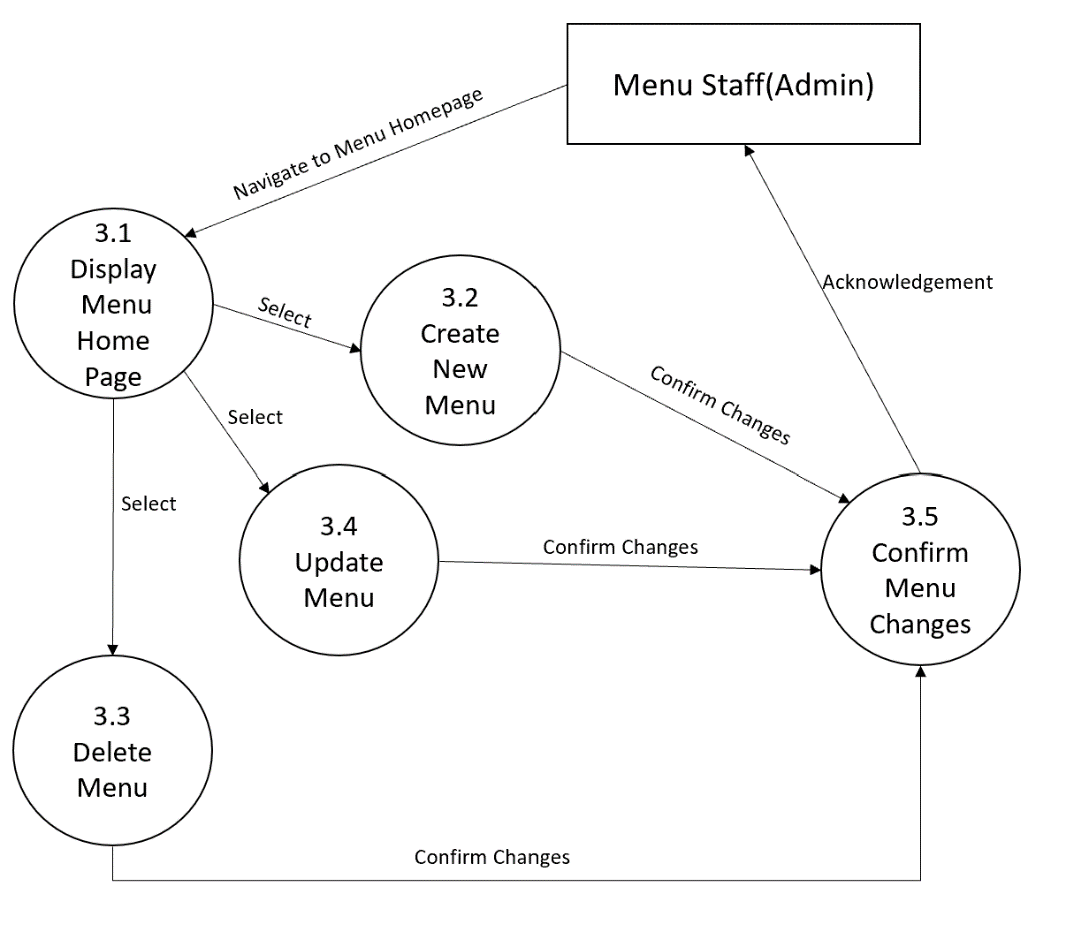
## Level 2

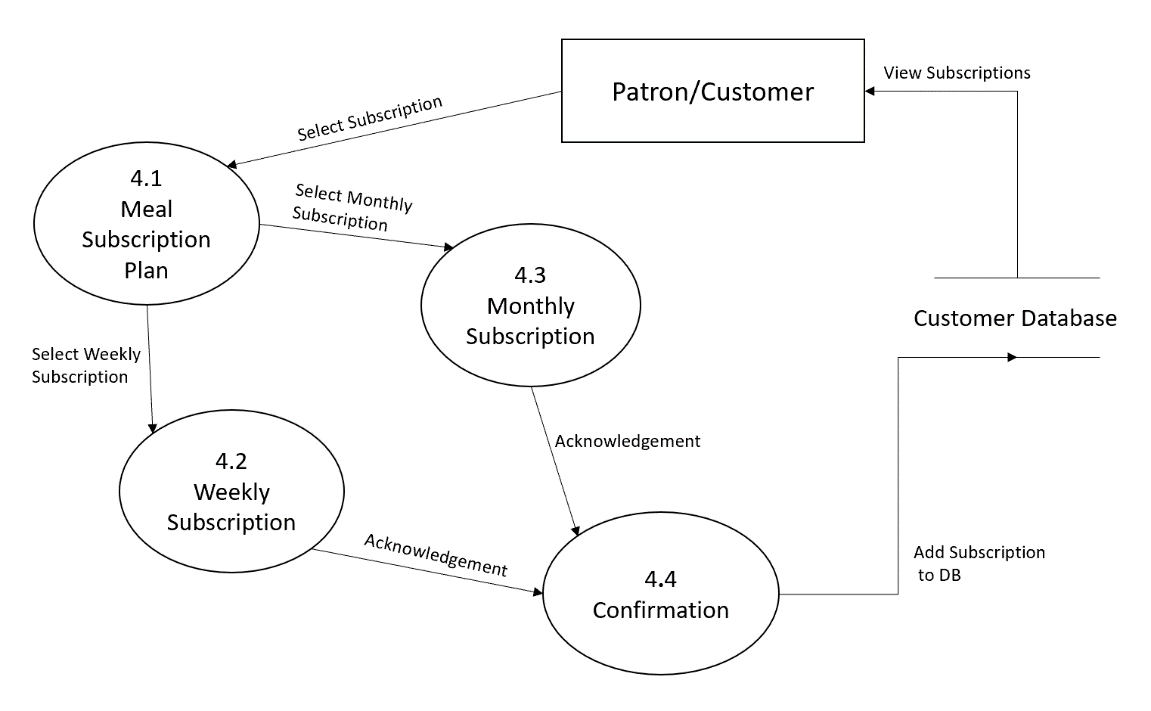


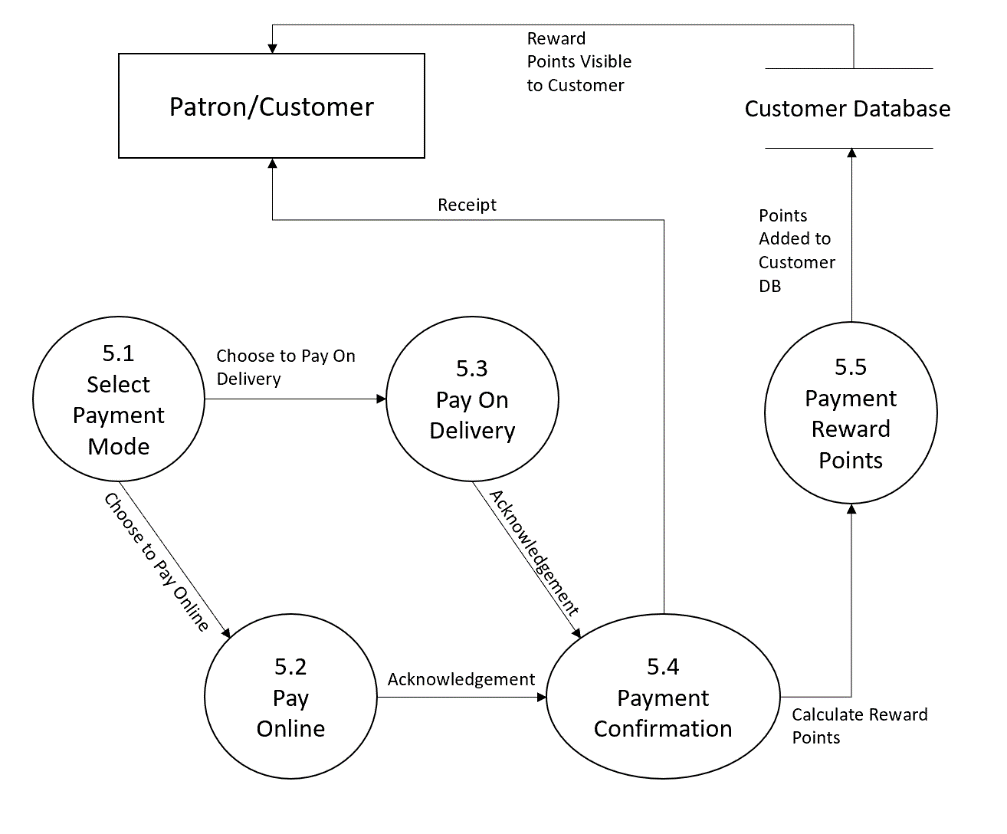


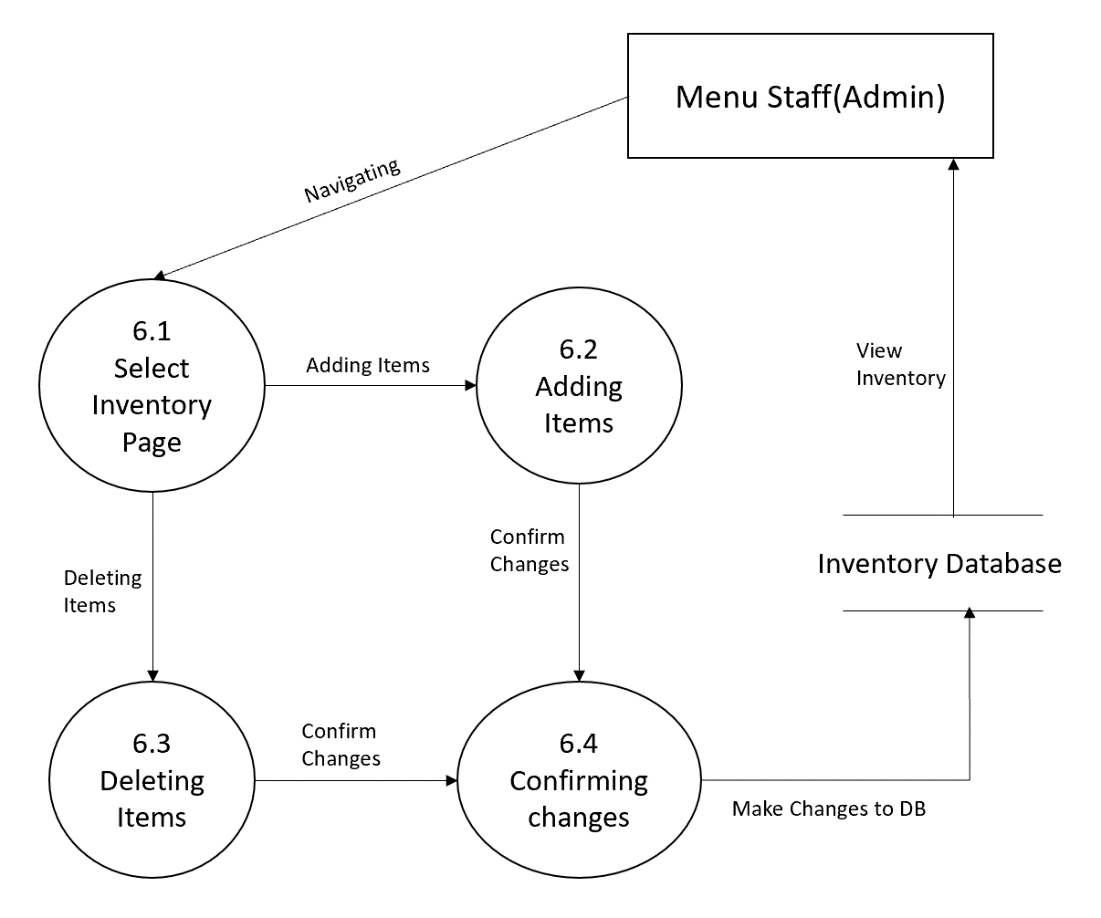


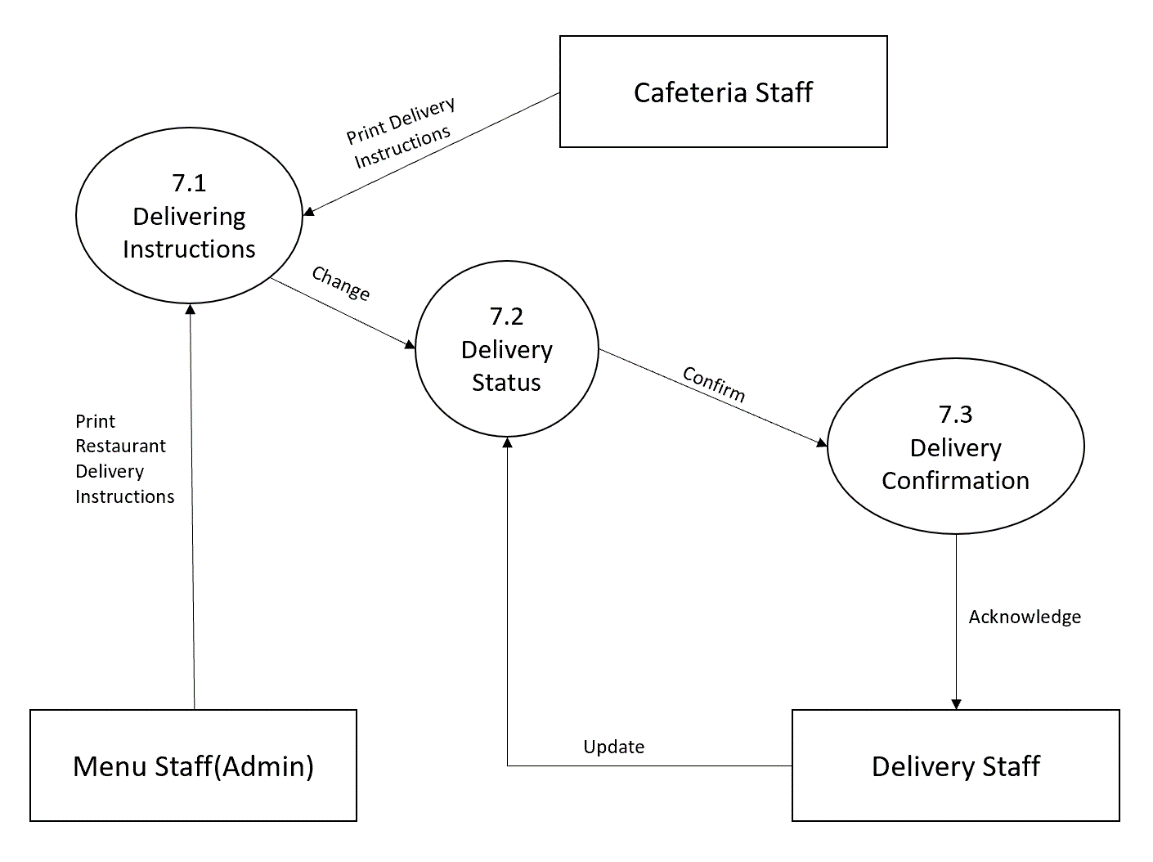










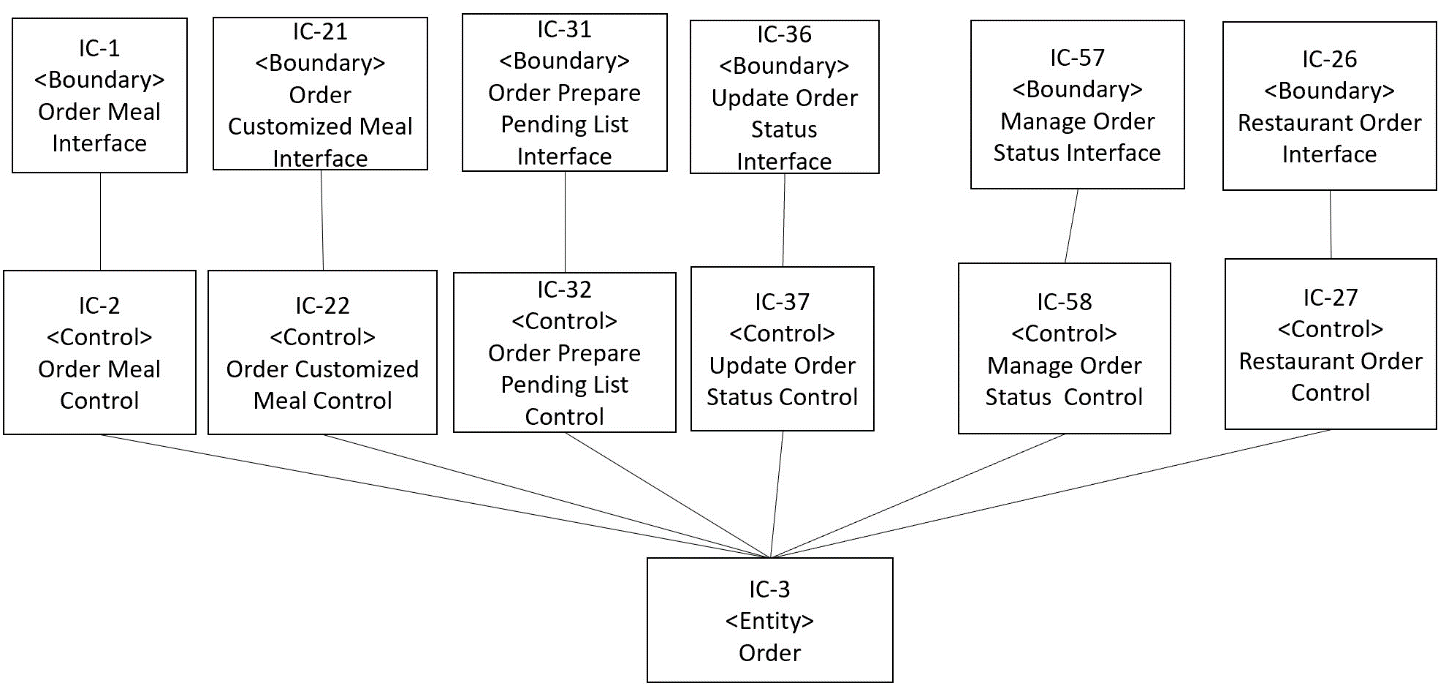


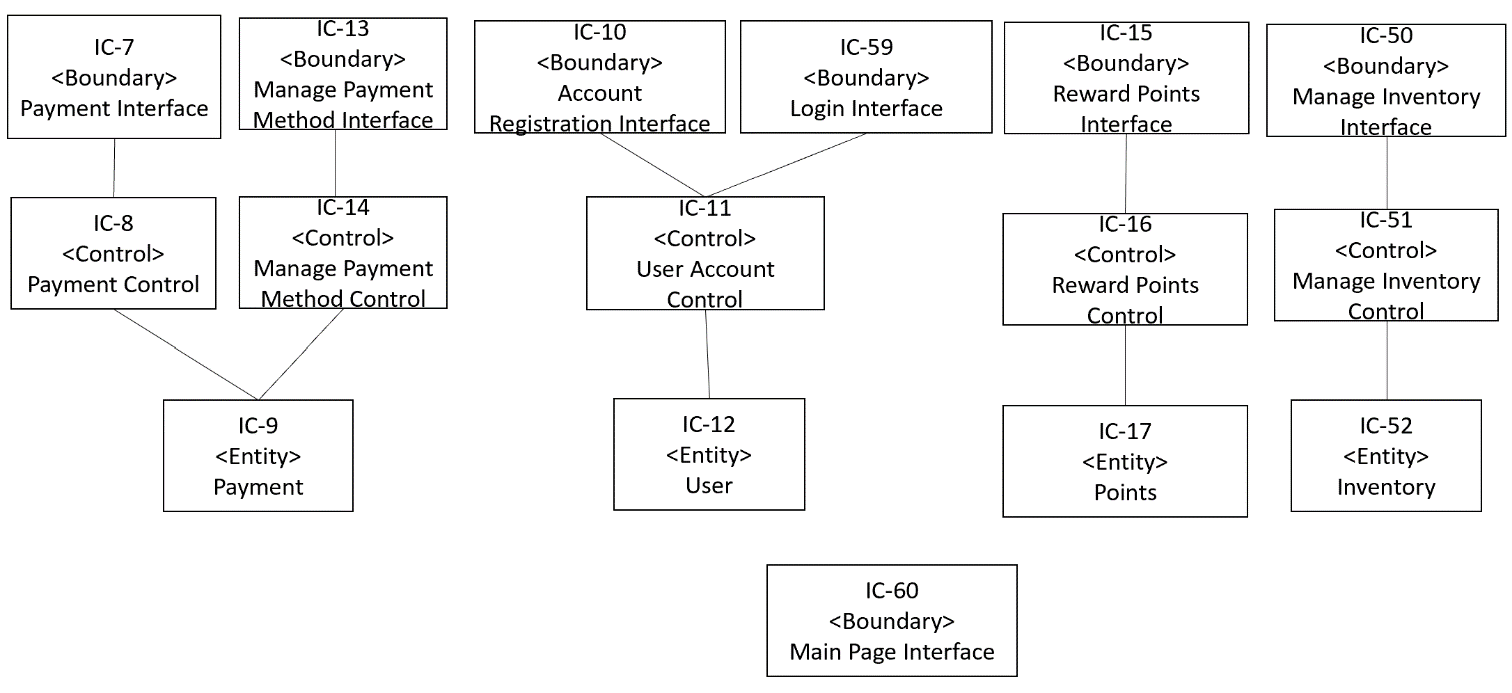
# TABLE 3: Traceability Matrix (FRs and DFD Processes)

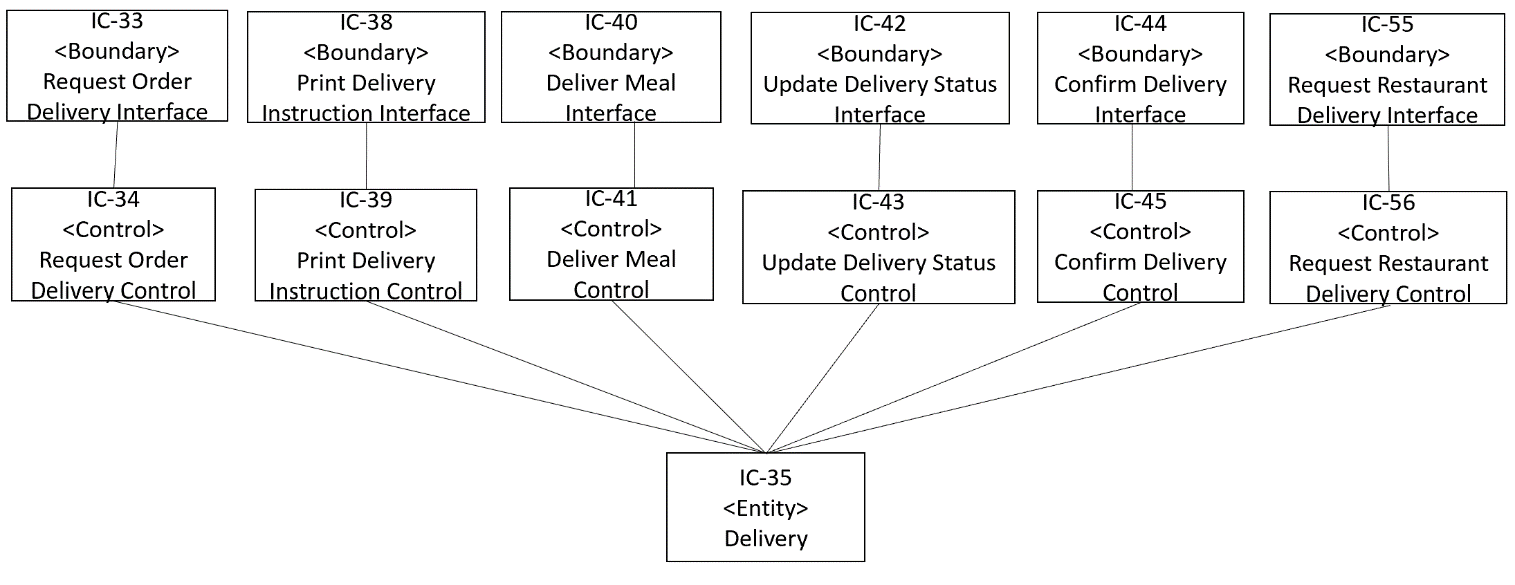
|  |  |
| --- | --- |
| **Functional Requirements** | **Related Processes in DFD** |
| FR-01 | 1,1.4,5.1,7 |
| FR-02 | 1.1 |
| FR-03 | 4,4.1,4.2,4.3 |
| FR-04 | 5.1,5.2,5.3 |
| FR-05 | 7.1,7.2 |
| FR-06 | 3.1,3.2,3.3,3.4 |
| FR-07 | 1.1 |
| FR-08 | 1.1 |
| FR-09 | 1.3 |
| FR-10 | 1.2 |
| FR-11 | 2.1,2.2,2.3 |
| FR-12 | 3 |
| FR-13 | 5.4 |
| FR-14 | 5.5 |
| FR-15 | 2.1 |
| FR-16 | 1.1 |
| FR-17 | 1.2 |
| FR-18 | 7.2,7.3 |
| FR-19 | 7.1 |
| FR-20 | 7.3 |
| FR-21 | 7.2 |
| FR-22 | 7.3 |
| FR-23 | 6.1,6.2,6.3 |
| FR-24 | 1.1,7,1 |
| FR-25 | 1.2 |

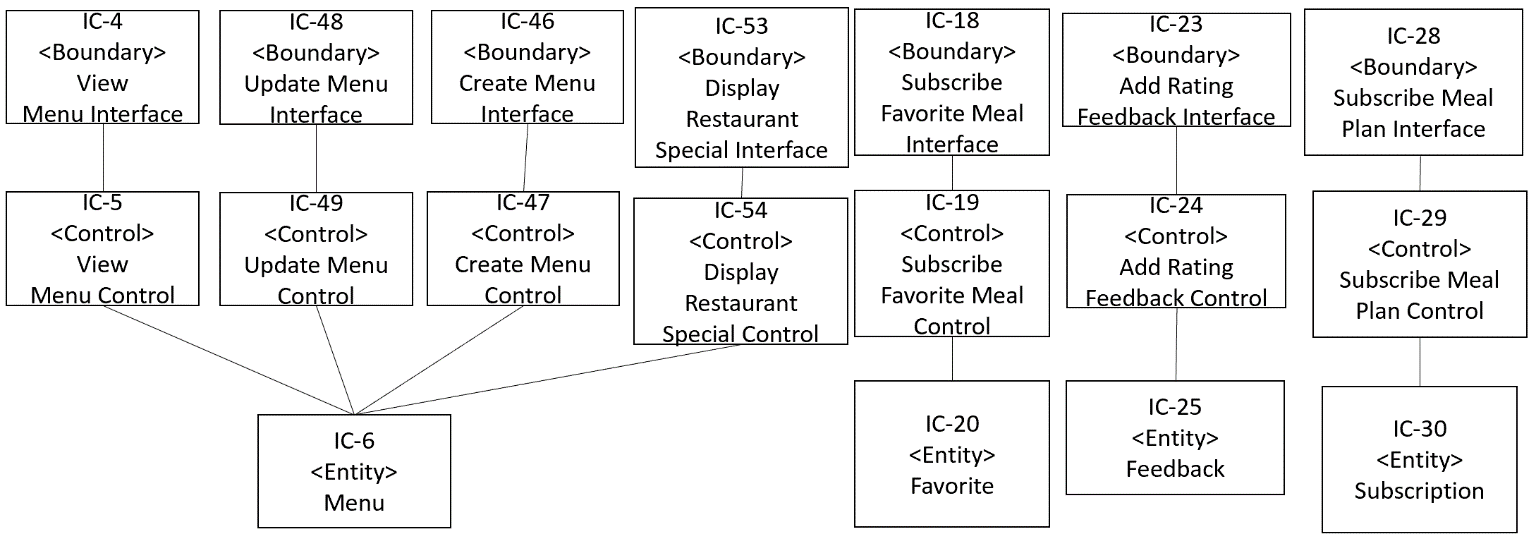
# Class Analysis Modeling

**9.1 Initial Class Diagram (ICs)**

****

****

****

****

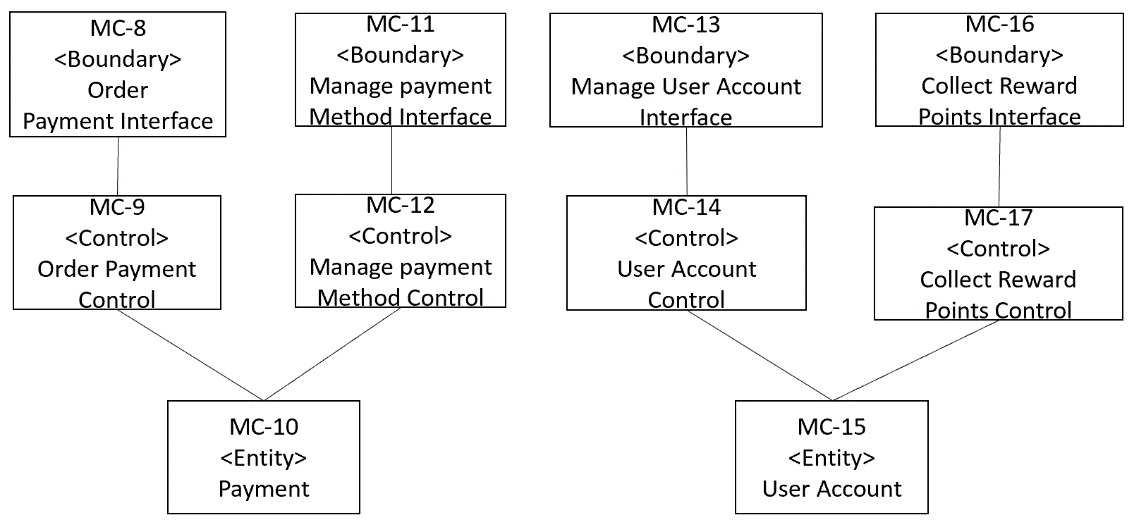
## TABLE 4: Traceability Matrix (FRs and Initial Classes)

|  |  |
| --- | --- |
| **FR No** | **Related IC’S** |
| FR-01 | IC-01, IC-02, IC-03, IC-59, IC-60 |
| FR-02 | IC-04, IC-05, IC-06, IC-59, IC-60 |
| FR-03 | IC-07, IC-08, IC-09, IC-59, IC-60 |
| FR-04 | IC-10, IC-11, IC-12, IC-59, IC-60 |
| FR-05 | IC-13, IC-14, IC-09, IC-59, IC-60 |
| FR-06 | IC-15, IC-16, IC-17, IC-59, IC-60 |
| FR-07 | IC-18, IC-19, IC-20, IC-59, IC-60 |
| FR-08 | IC-21, IC-22, IC-03, IC-59, IC-60 |
| FR-09 | IC-23, IC-24, IC-25, IC-59, IC-60 |
| FR-10 | IC-26, IC-27, IC-03, IC-59, IC-60 |
| FR-11 | IC-01, IC-02, IC-03, IC-59, IC-60 |
| FR-12 | IC-28, IC-29, IC-30, IC-59, IC-60 |
| FR-13 | IC-28, IC-29, IC-30, IC-59, IC-60 |
| FR-14 | IC-28, IC-29, IC-30, IC-59, IC-60 |
| FR-15 | IC-28, IC-29, IC-30, IC-59, IC-60 |
| FR-16 | IC-01, IC-02, IC-03, IC-59, IC-60 |
| FR-17 | IC-31, IC-32, IC-03, IC-59, IC-60 |
| FR-18 | IC-33, IC-34, IC-35, IC-59, IC-60 |
| FR-19 | IC-28, IC-29, IC-30, IC-59, IC-60 |
| FR-20 | IC-38, IC-39, IC-35, IC-59, IC-60 |
| FR-21 | IC-40, IC-41, IC-35, IC-59, IC-60 |
| FR-22 | IC-42, IC-43, IC-35, IC-59, IC-60 |
| FR-23 | IC-44, IC-45, IC-35, IC-59, IC-60 |
| FR-24 | IC-46, IC-47, IC-06, IC-59, IC-60 |
| FR-25 | IC-46, IC-47, IC-06, IC-59, IC-60 |
| FR-26 | IC-48, IC-49, IC-06, IC-59, IC-60 |
| FR-27 | IC-48, IC-49, IC-06, IC-59, IC-60 |
| FR-28 | IC-57, IC-58, IC-03, IC-59, IC-60 |
| FR-29 | IC-10, IC-11, IC-12. IC-59, IC-60 |
| FR-30 | IC-50, IC-51, IC-52, IC-59, IC-60 |
| FR-31 | IC-53, IC-54, IC-6, IC-59, IC-60 |
| FR-32 | IC-53, IC-54, IC-06, IC-59, IC-60 |
| FR-33 | IC-53, IC-54, IC-06, IC-59, IC-60 |
| FR-34 | IC-55, IC-56, IC-35, IC-59, IC-60 |
| FR-35 | IC-36, IC-37, IC-03, IC-59, IC-60 |
| FR-36 | IC-57, IC-58, IC-03, IC-59, IC-60 |
| FR-37 | IC-10, IC-11, IC-12, IC-59, IC-60 |
| FR-38 | IC-10, IC-11, IC-12. IC-59, IC-60 |
| FR-39 | IC-15, IC-16, IC-17, IC-59, IC-60 |
| FR-40 | IC-15, IC-16, IC-17, IC-59, IC-60 |
| FR-41 | IC-10, IC-11, IC-12, IC-59, IC-60 |
| FR-42 | IC-01, IC-02, IC-03, IC-10, IC-11, IC-12, IC-59, IC-60 |

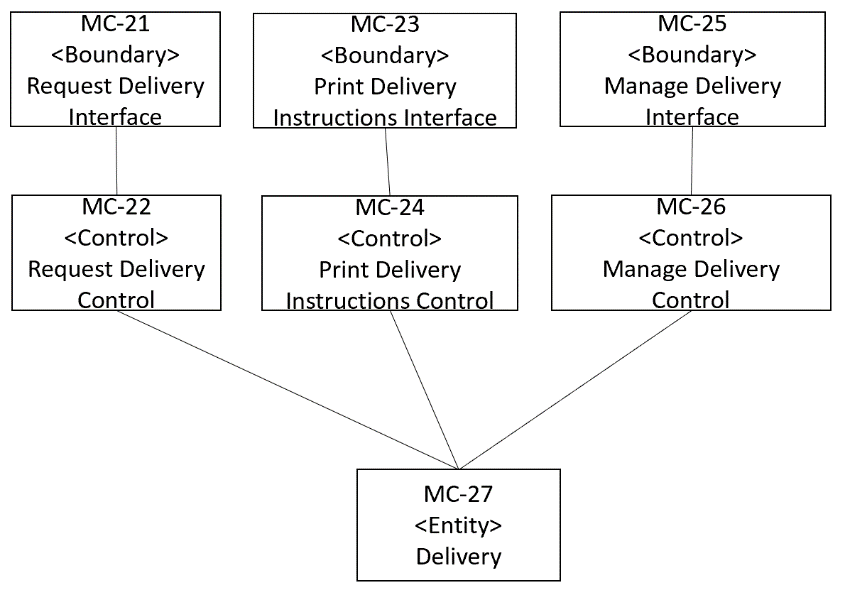
**9.2 Modified Class Diagram (MCs)**

Diagram

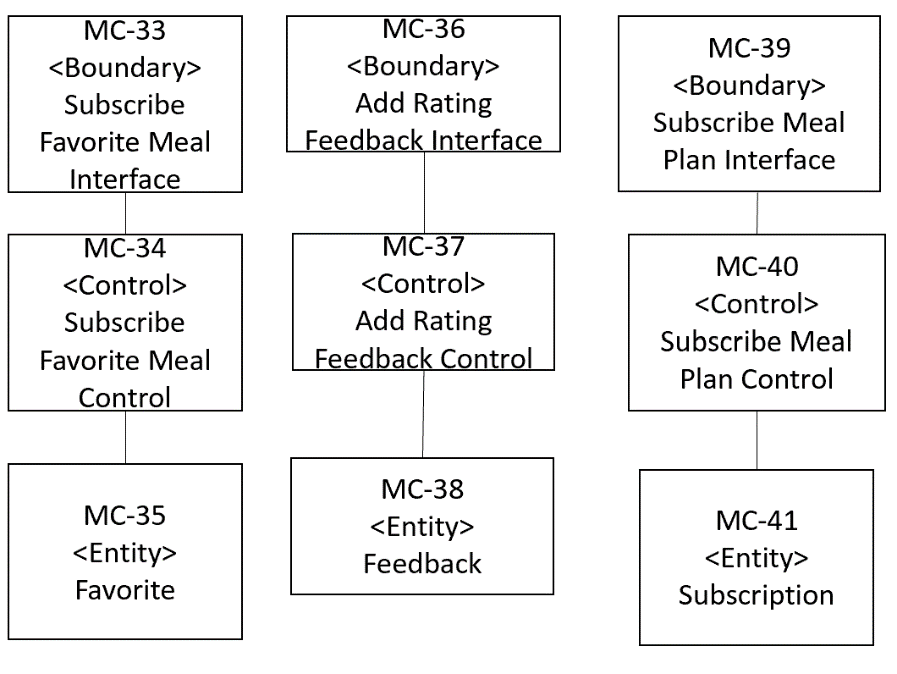
Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated

Text

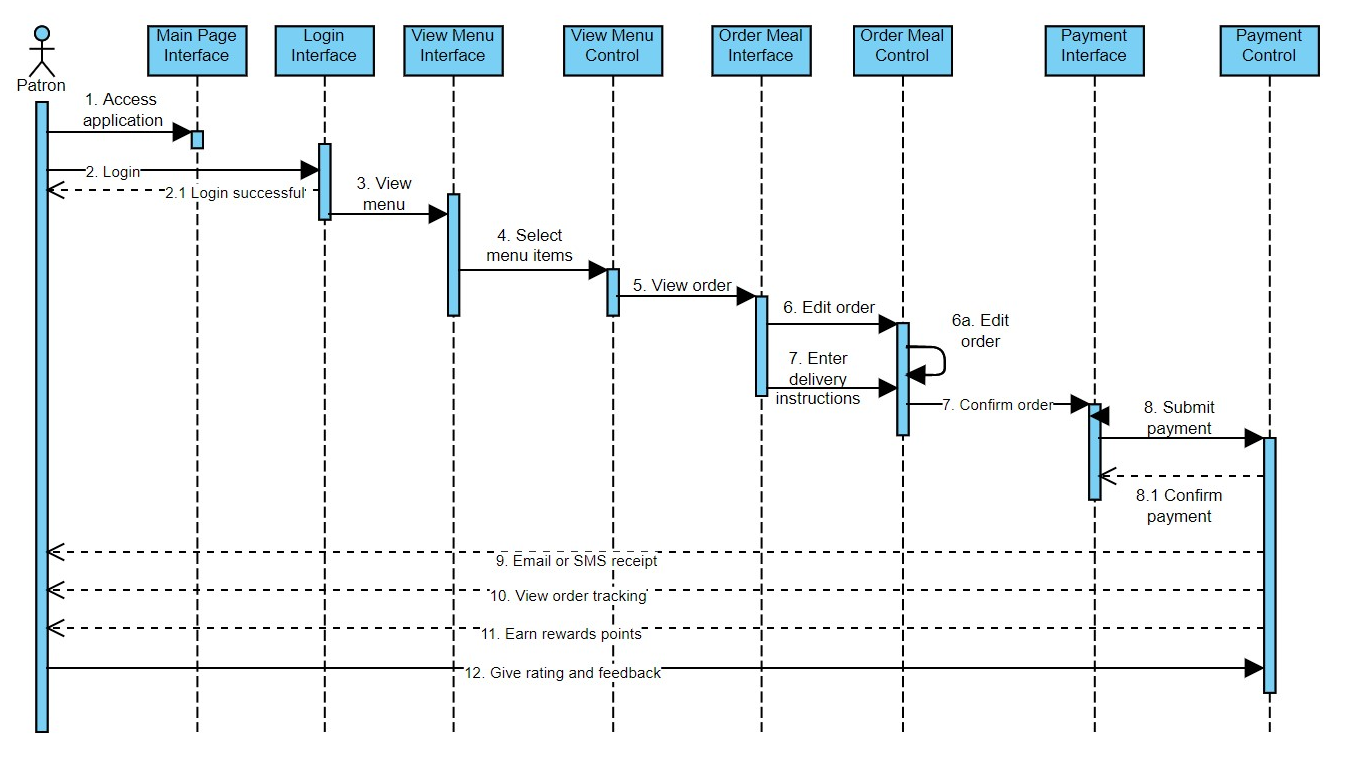
Description automatically generated with medium confidence

## TABLE 5: Traceability Matrix (Initial Classes (IC) and modified Classes (MC))

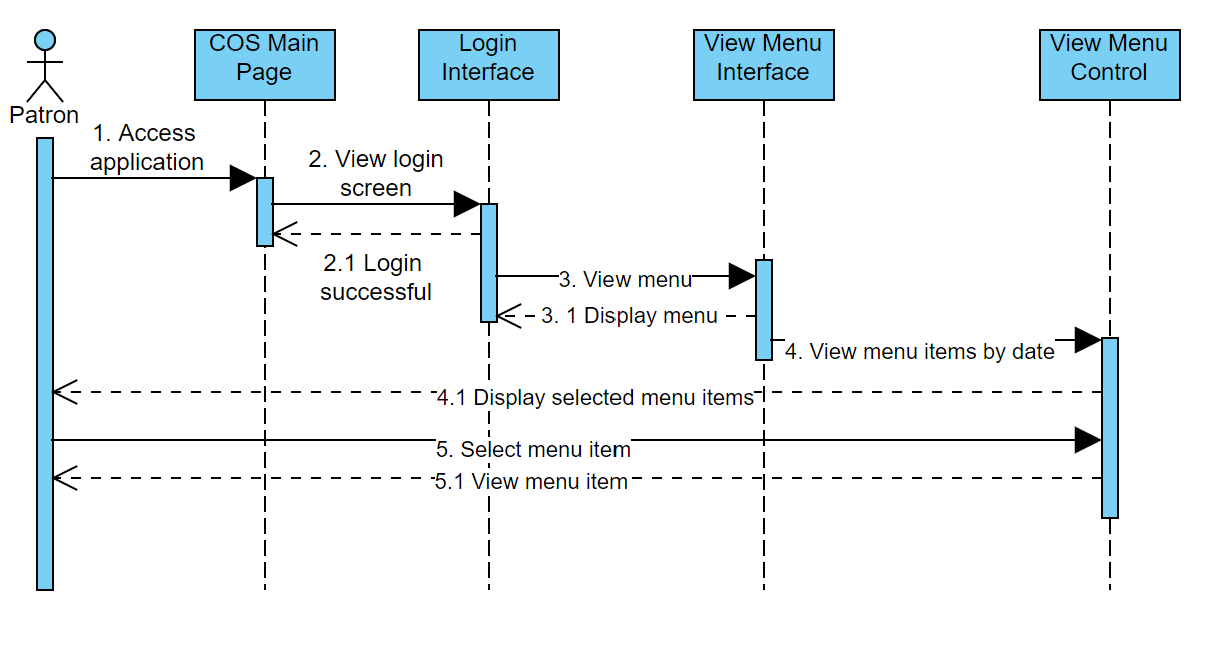
|  |  |
| --- | --- |
| **IC’s** | **Related MC’s** |
| IC-01 | MC-01, MC-02, MC-03, MC-42 |
| IC-02 | MC-01, MC-02, MC-03, MC-42 |
| IC-03 | MC-01, MC-02, MC-03, MC-42 |
| IC-04 | MC-28, MC-29, MC-30, MC-42 |
| IC-05 | MC-28, MC-29, MC-30, MC-42 |
| IC-06 | MC-28, MC-29, MC-30, MC-42 |
| IC-07 | MC-08, MC-09, MC-10, MC-42 |
| IC-08 | MC-08, MC-09, MC-10, MC-42 |
| IC-09 | MC-08, MC-09, MC-10, MC-42 |
| IC-10 | MC-13, MC-14, MC-15, MC-42 |
| IC-11 | MC-13, MC-14, MC-15, MC-42 |
| IC-12 | MC-13, MC-14, MC-15, MC-42 |
| IC-13 | MC-11, MC-12, MC-13, MC-42 |
| IC-14 | MC-11, MC-12, MC-13, MC-42 |
| IC-15 | MC-16, MC-17, MC-15, MC-42 |
| IC-16 | MC-16, MC-17, MC-15, MC-42 |
| IC-17 | MC-16, MC-17, MC-15, MC-42 |
| IC-18 | MC-33, MC-34, MC-35, MC-42 |
| IC-19 | MC-33, MC-34, MC-35, MC-42 |
| IC-20 | MC-33, MC-34, MC-35, MC-42 |
| IC-21 | MC-01, MC-02, MC-03, MC-42 |
| IC-22 | MC-01, MC-02, MC-03, MC-42 |
| IC-23 | MC-36, MC-37, MC-38, MC-42 |
| IC-24 | MC-36, MC-37, MC-38, MC-42 |
| IC-25 | MC-36, MC-37, MC-38, MC-42 |
| IC-26 | MC-01, MC-02, MC-03, MC-42 |
| IC-27 | MC-01, MC-02, MC-03, MC-42 |
| IC-28 | MC-39, MC-40, MC-41, MC-42 |
| IC-29 | MC-39, MC-40, MC-41, MC-42 |
| IC-30 | MC-39, MC-40, MC-41, MC-42 |
| IC-31 | MC-06, MC-07, MC-03, MC-42 |
| IC-32 | MC-06, MC-07, MC-03, MC-42 |
| IC-33 | MC-21, MC-22, MC-27, MC-42 |
| IC-34 | MC-21, MC-22, MC-27, MC-42 |
| IC-35 | MC-21, MC-22, MC-27, MC-42 |
| IC-36 | MC-04, MC-05, MC-03, MC-42 |
| IC-37 | MC-04, MC-05, MC-03, MC-42 |
| IC-38 | MC-23, MC-24, MC-27, MC-42 |
| IC-39 | MC-23, MC-24, MC-27, MC-42 |
| IC-40 | MC-21, MC-22, MC-27, MC-42 |
| IC-41 | MC-21, MC-22, MC-27, MC-42 |
| IC-42 | MC-25, MC-26, MC-27, MC-42 |
| IC-43 | MC-25, MC-26, MC-27, MC-42 |
| IC-44 | MC-25, MC-26, MC-27, MC-42 |
| IC-45 | MC-25, MC-26, MC-27, MC-42 |
| IC-46 | MC-28, MC-29, MC-30, MC-42 |
| IC-47 | MC-28, MC-29, MC-30, MC-42 |
| IC-48 | MC-31, MC-32, MC-30, MC-42 |
| IC-49 | MC-31, MC-32, MC-30, MC-42 |
| IC-50 | MC-18, MC-19, MC-20, MC-42 |
| IC-51 | MC-18, MC-19, MC-20, MC-42 |
| IC-52 | MC-18, MC-19, MC-20, MC-42 |
| IC-53 | MC-31, MC-32, MC-30, MC-42 |
| IC-54 | MC-31, MC-32, MC-30, MC-42 |
| IC-55 | MC-21, MC-22, MC-27, MC-42 |
| IC-56 | MC-21, MC-22, MC-27, MC-42 |
| IC-57 | MC-04, MC-05, MC-03, MC-42 |
| IC-58 | MC-04, MC-05, MC-03, MC-42 |
| IC-59 | MC-13, MC-14, MC-15, MC-42 |
| IC-60 | MC-42 |

# Sequence Diagram (SDs)

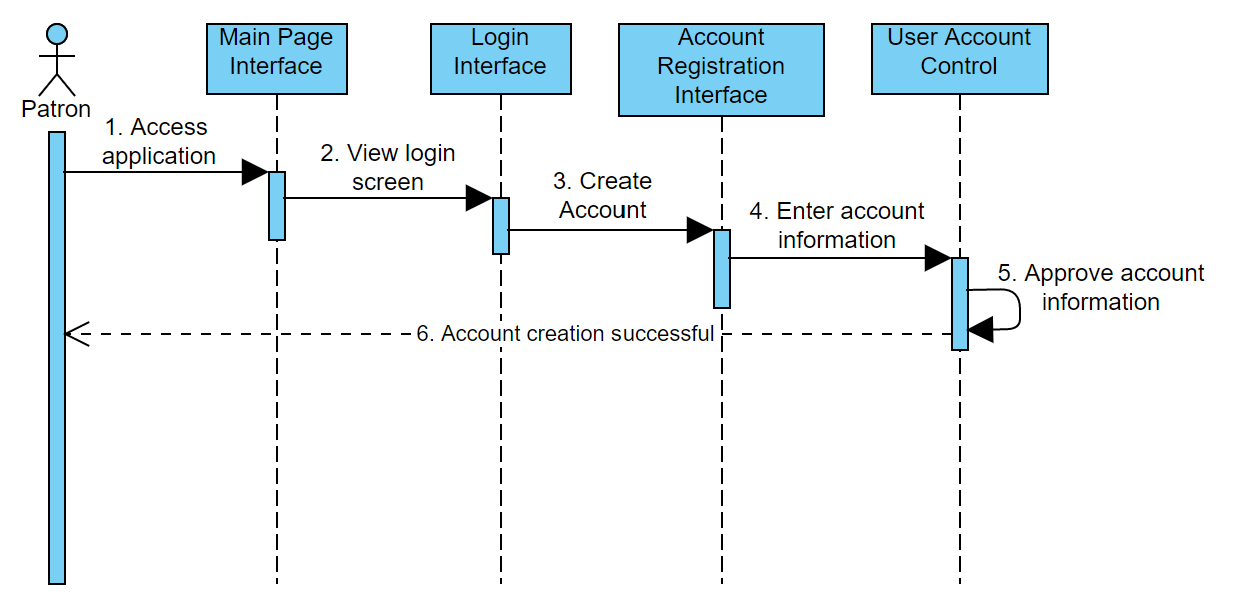
## SD-01: *<Order Meal & Submit Payment>*



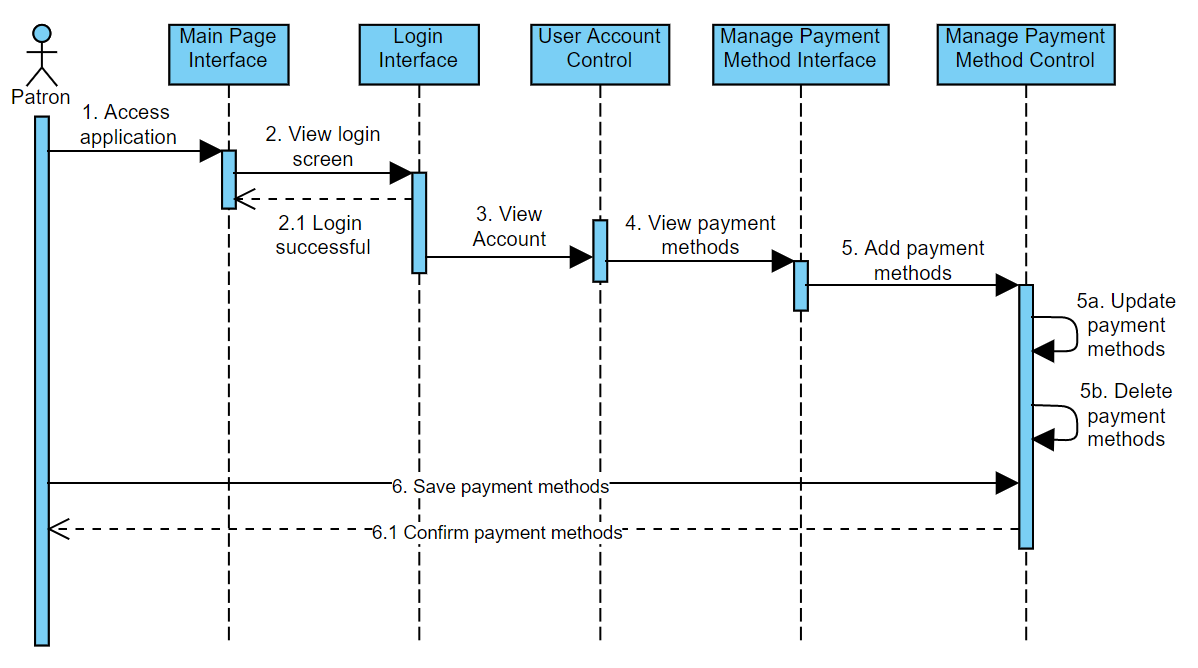
## SD-02: *<View Menu>*



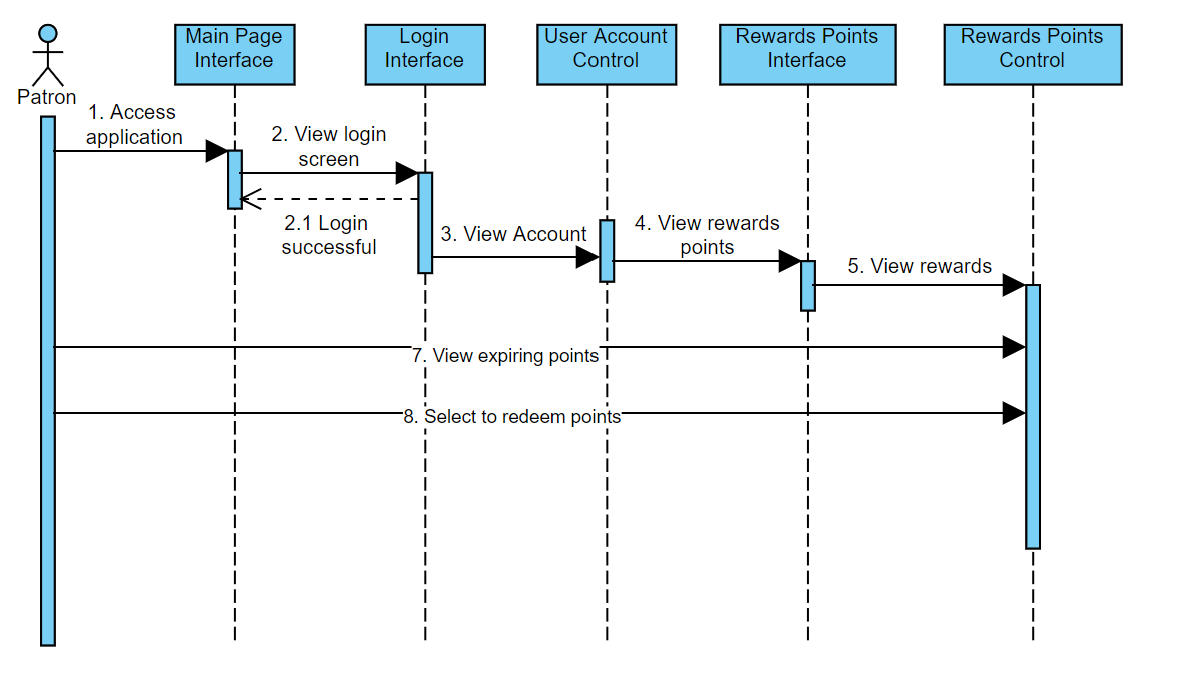
## SD-03: *<Create Account>*

****

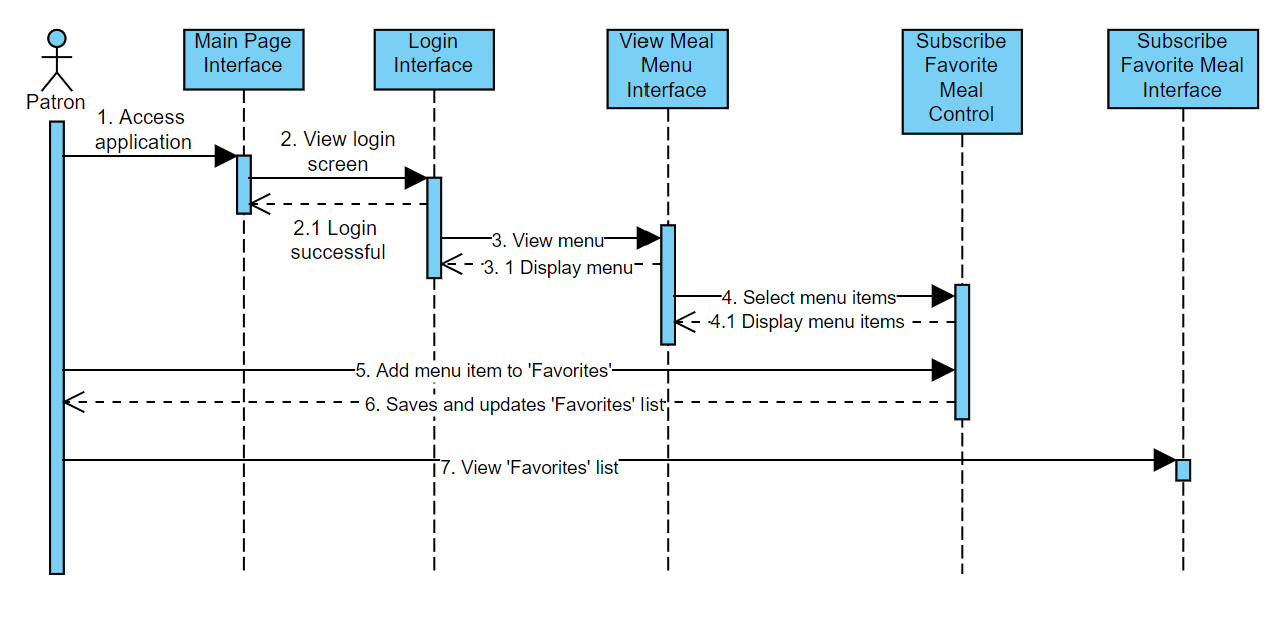
## SD-04: *<Manage Payment Methods>*

****

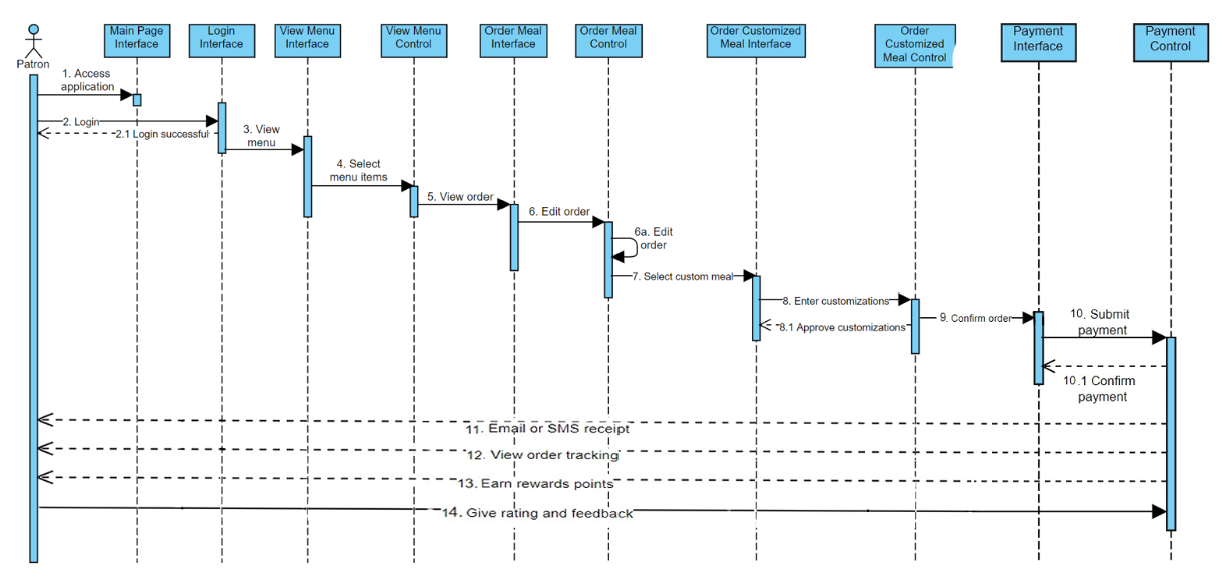
## SD-05: *<Rewards Points>*

****

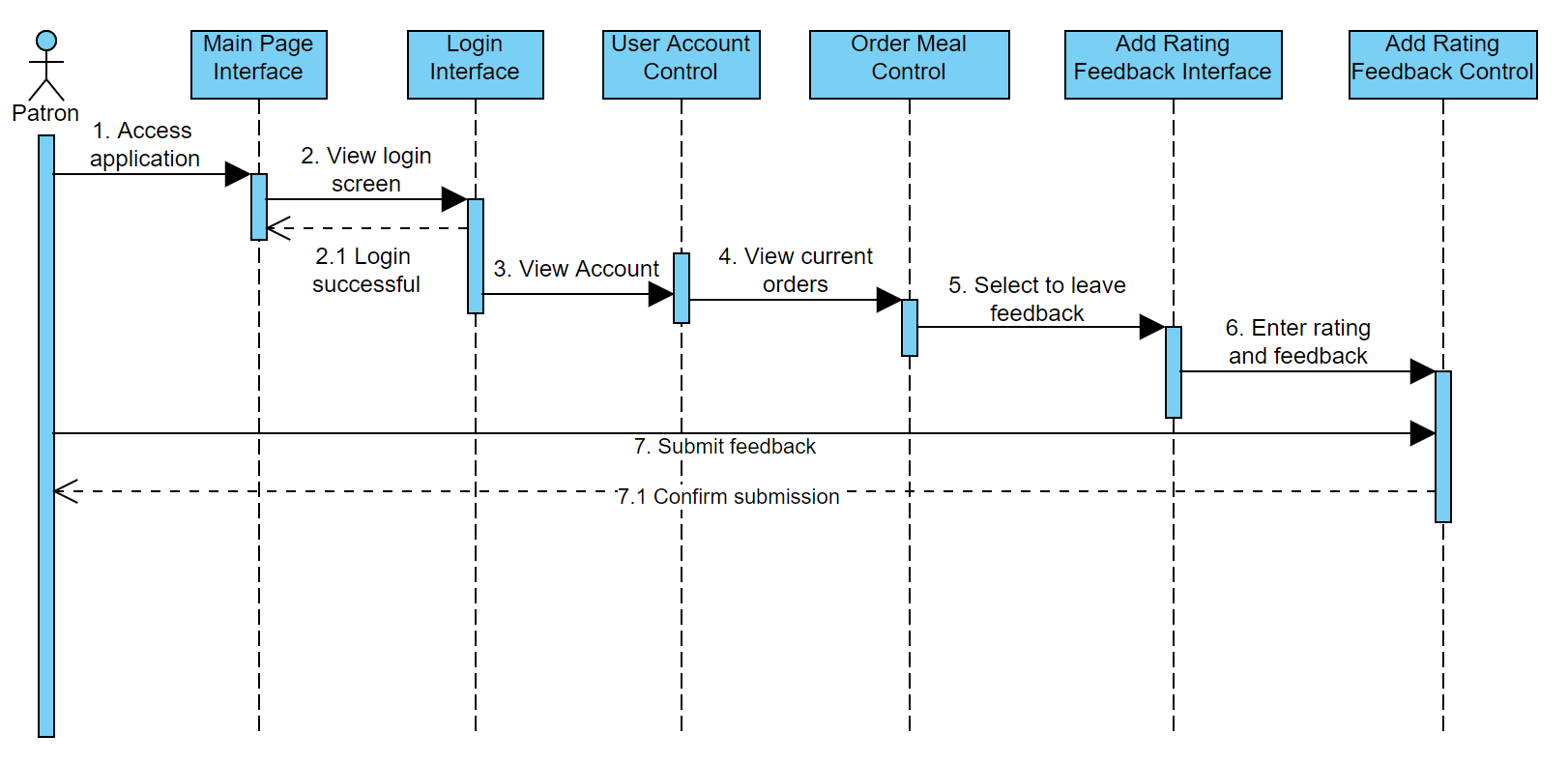
## SD-06: *<Favorite Meals>*

****

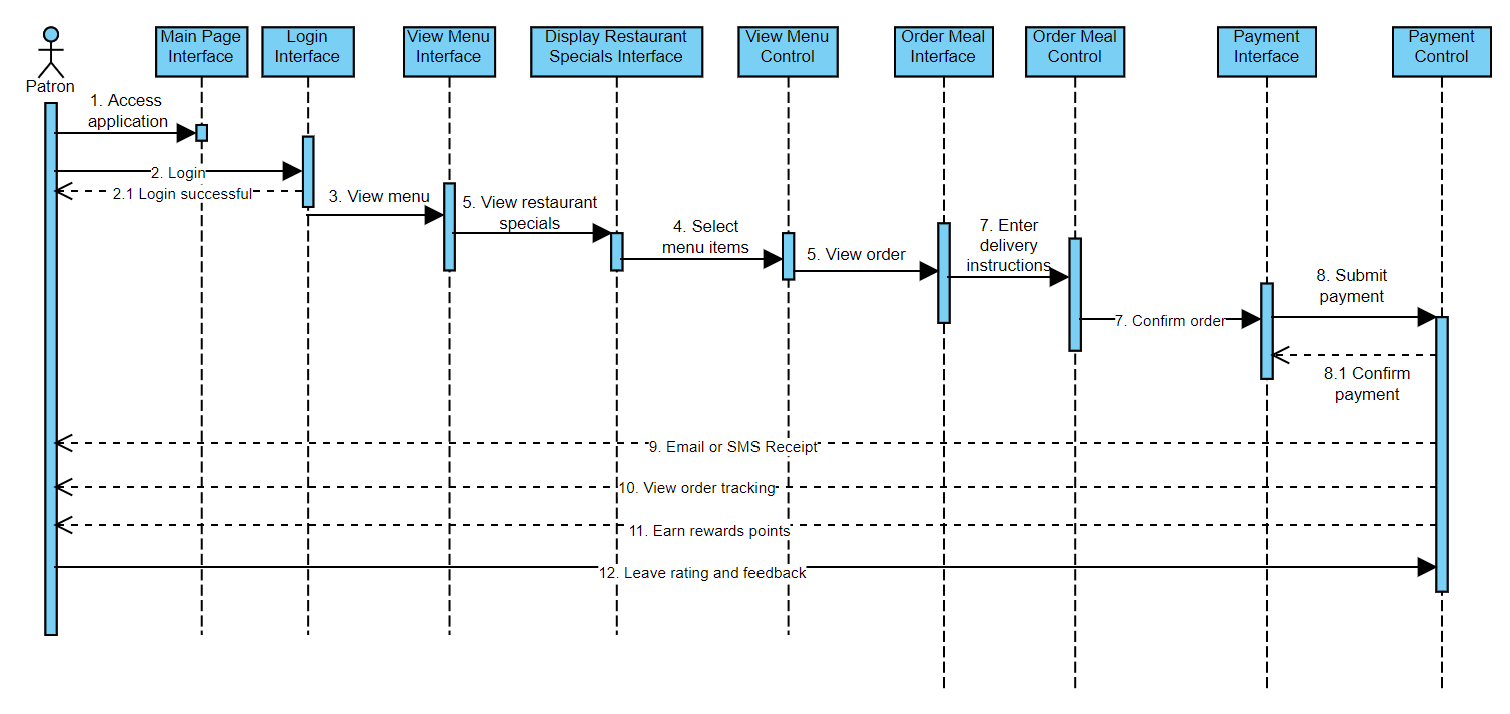
## SD-07: *<Order Custom Meals>*

****

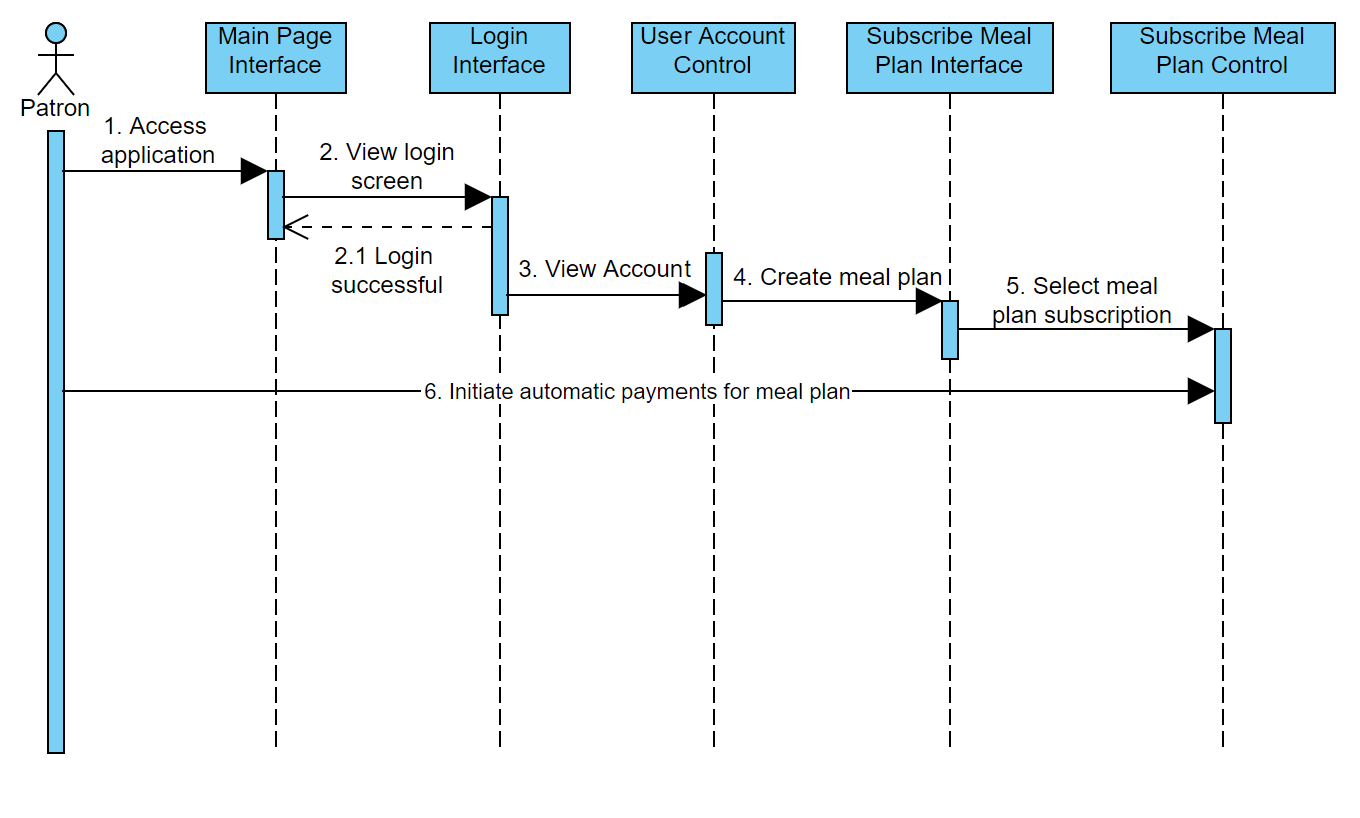
## SD-08: *<Feedback>*

****

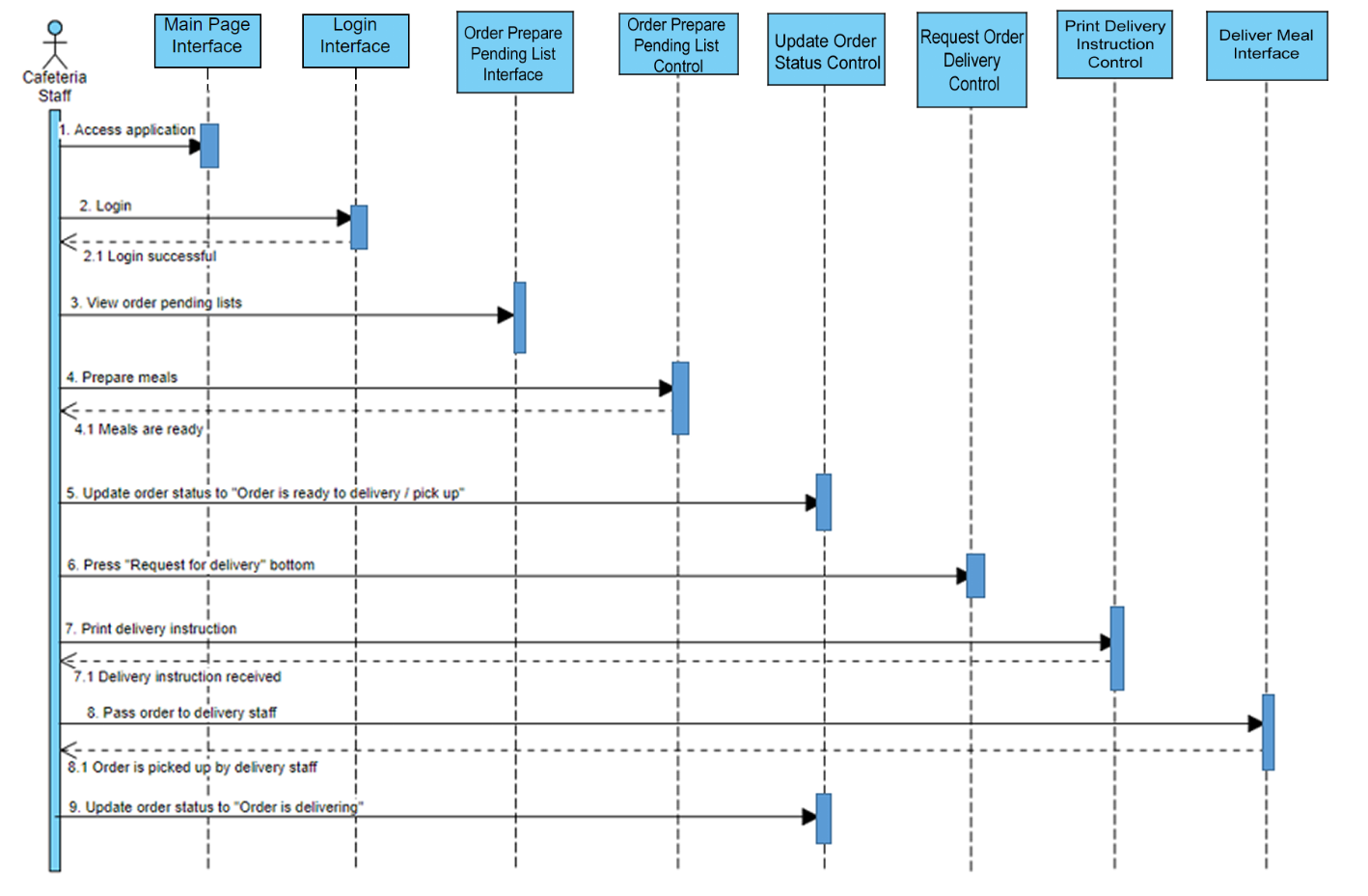
## SD-09: *<Order Meal FR-om Restaurant>*

****

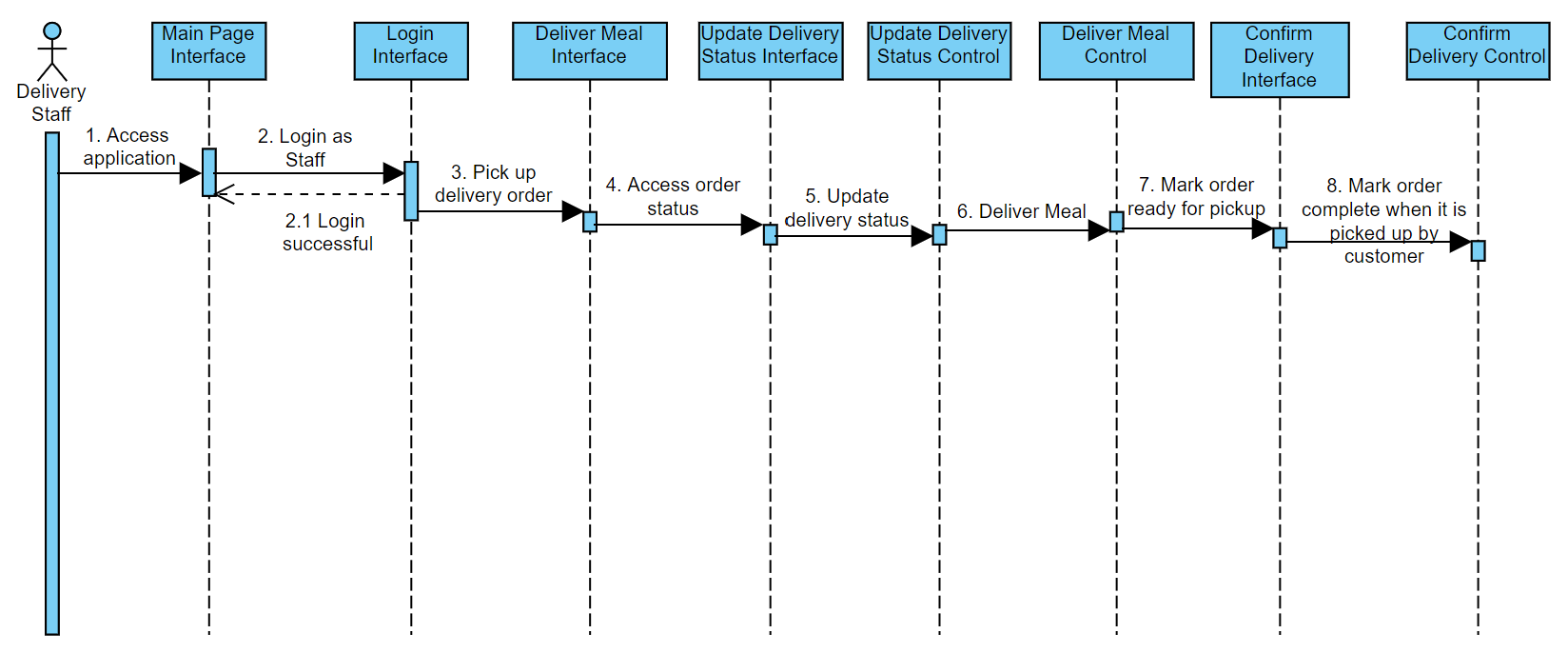
## SD-10: *<Meal Subscriptions>*



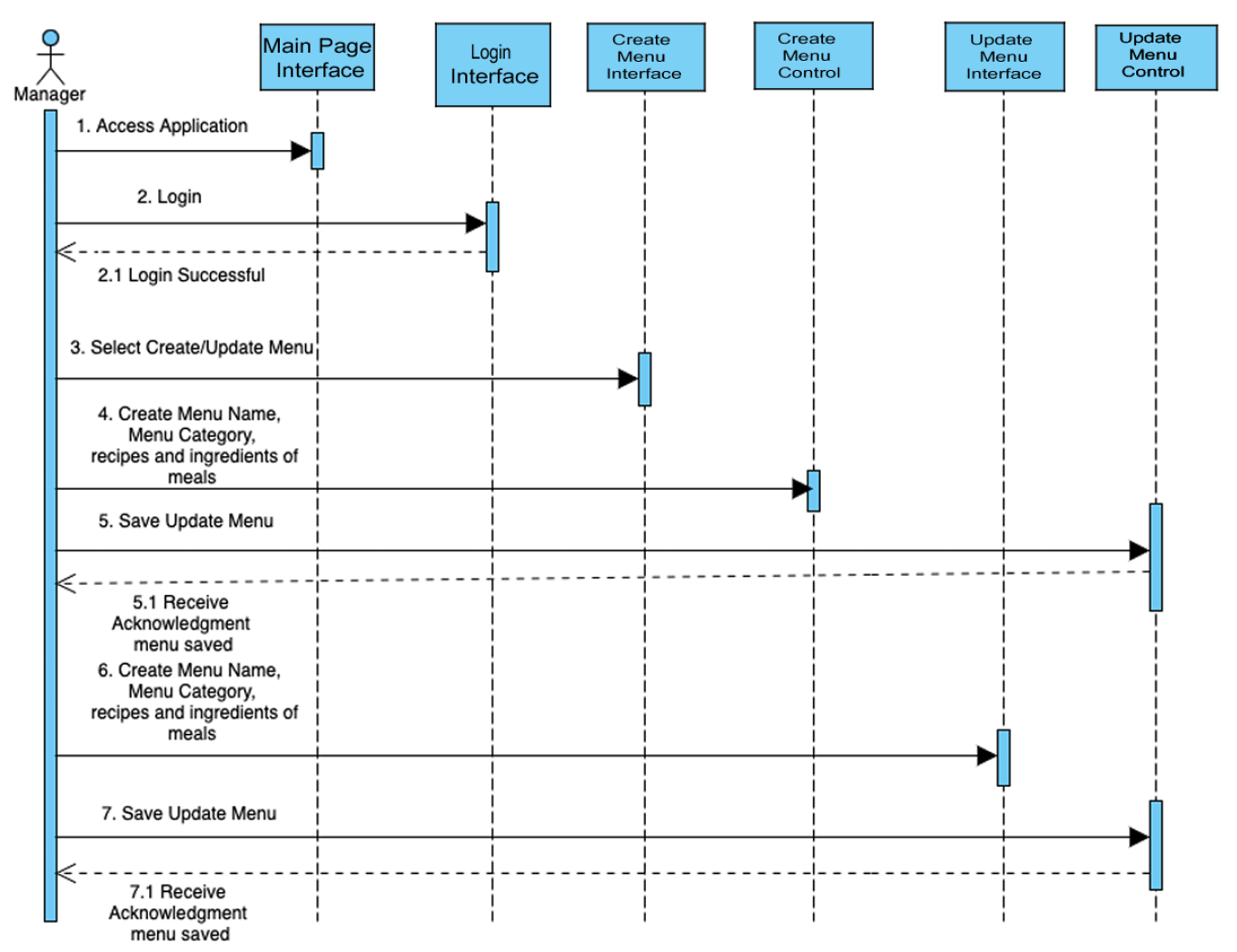
## SD-11: *<Meal Preparation>*

****

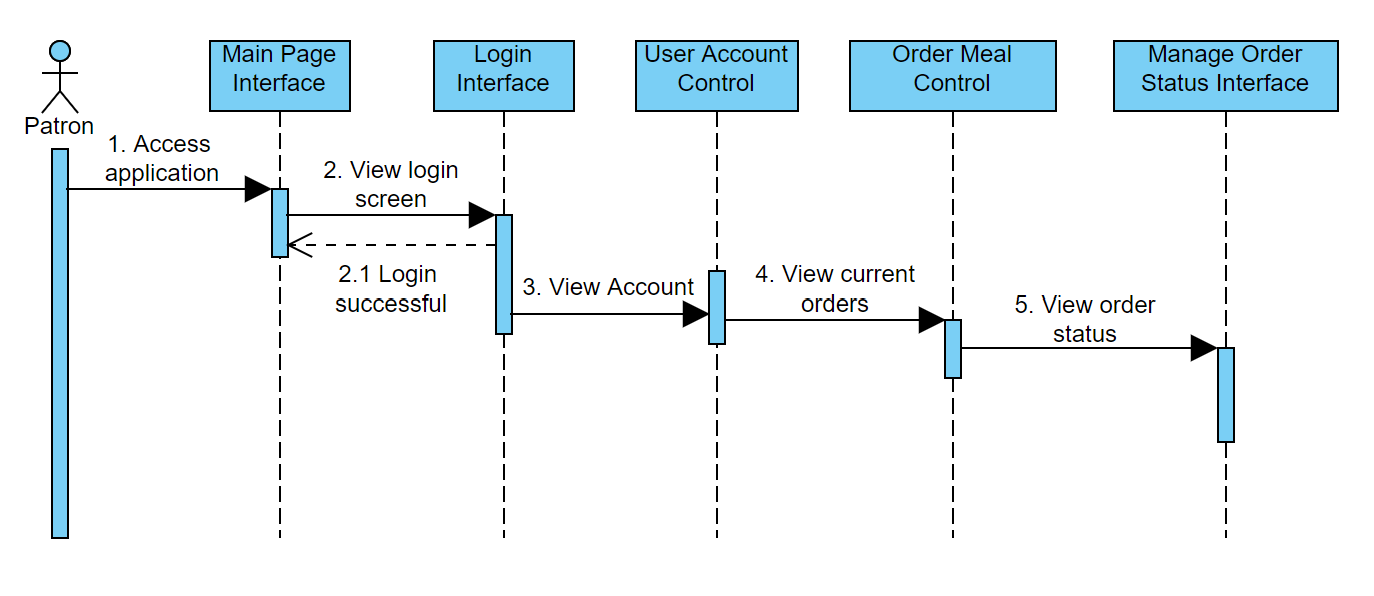
## SD-12: *<Delivery>*

****

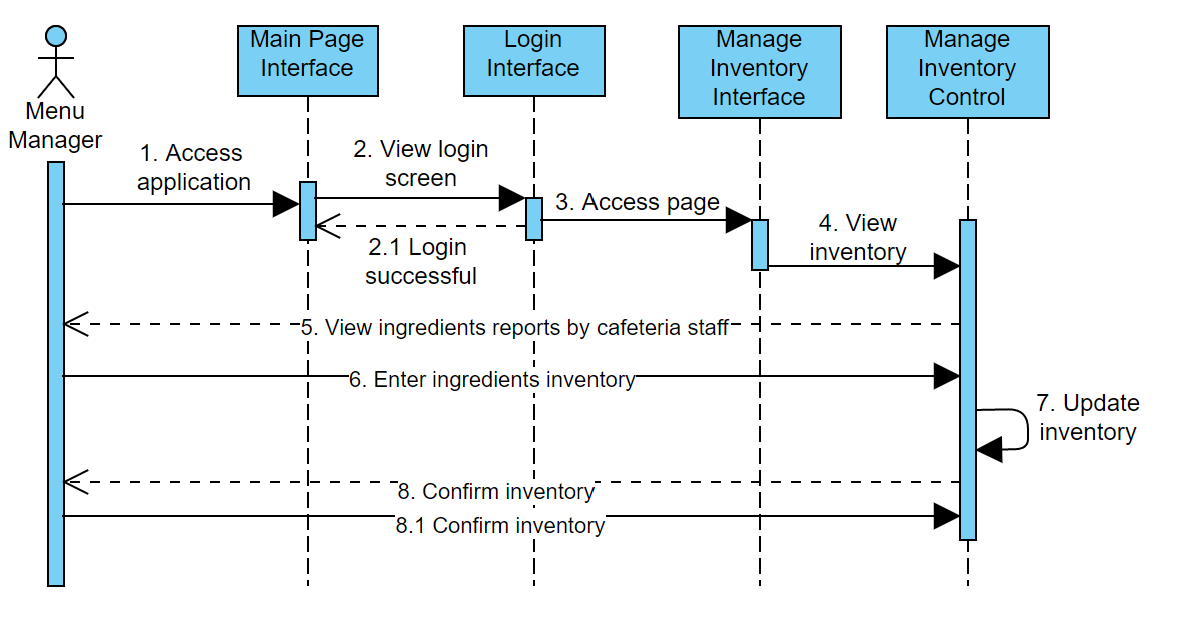
## SD-13: *<Menu Management>*

****

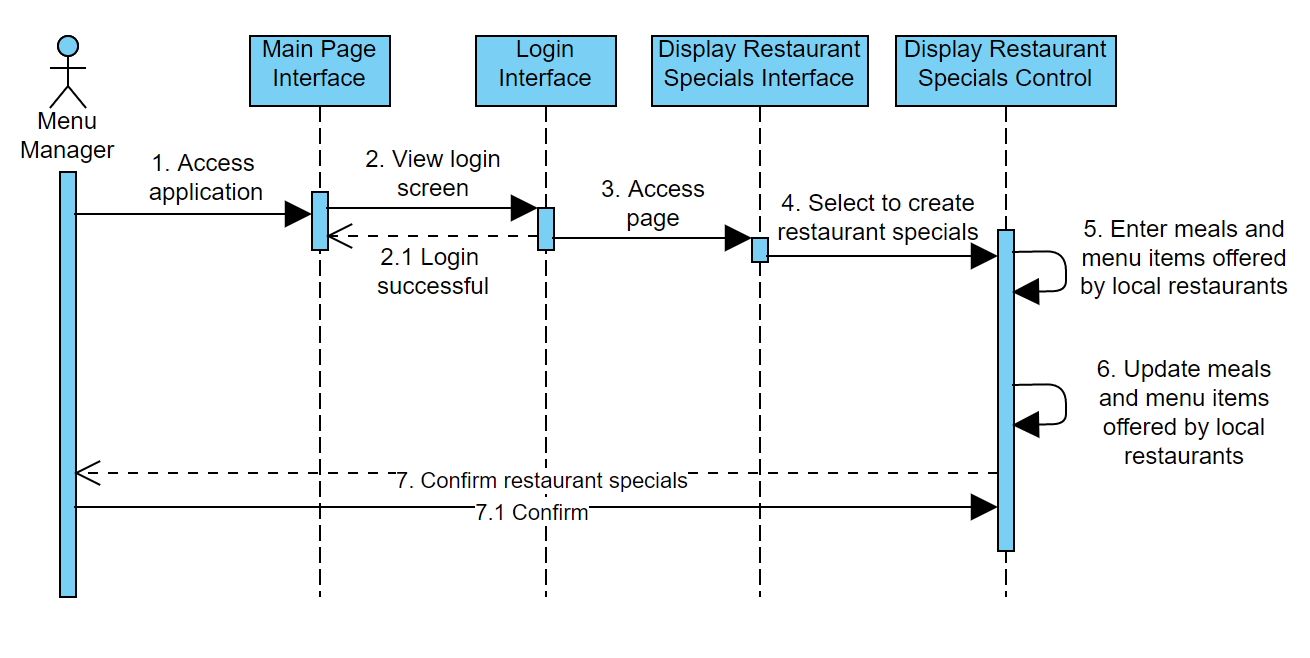
## SD-14: *<View Order Tracking Status>*

****

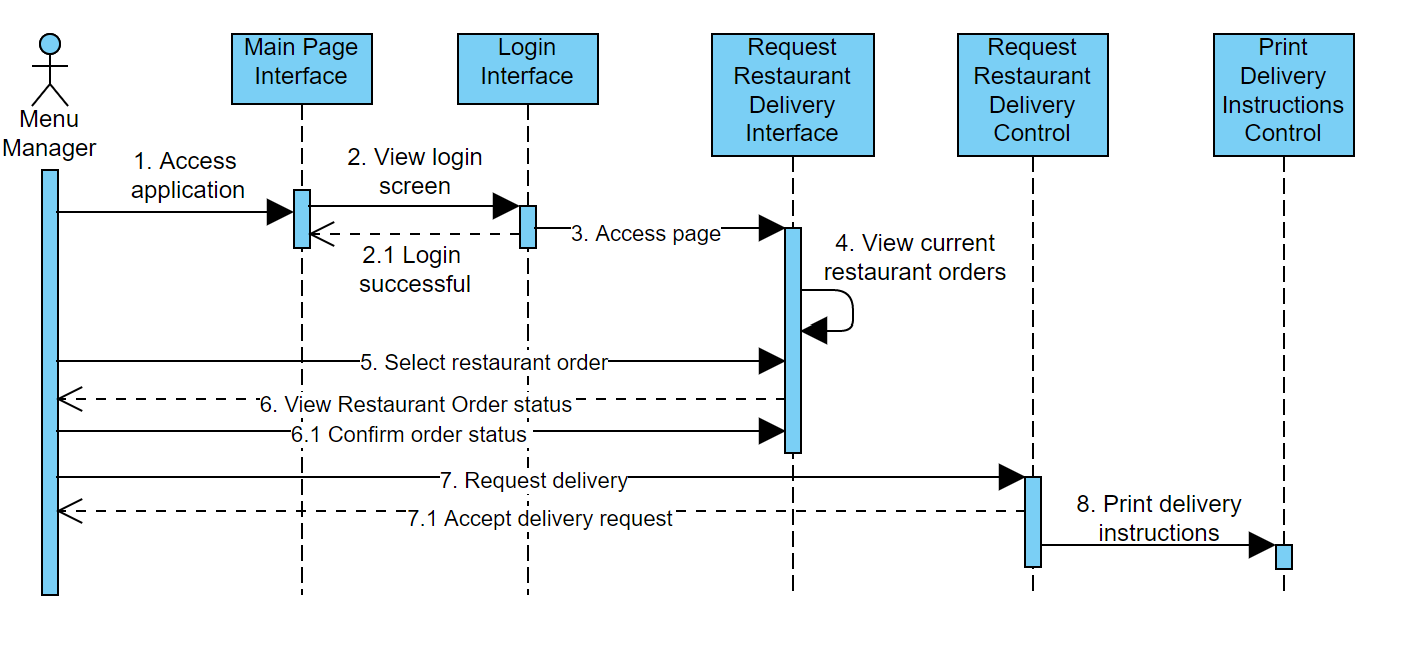
## SD-15: *<Manage Inventory>*

****

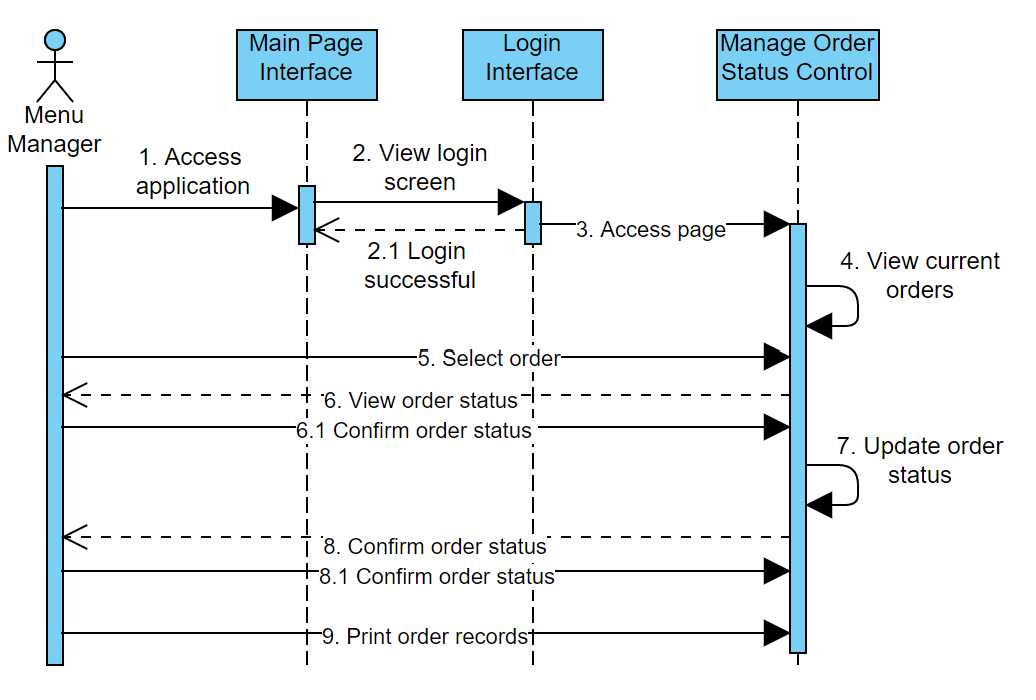
## SD-16: *<Display Restaurant Specials>*

****

## SD-17: *<Request Delivery from Restaurants>*

****

## SD-18: *<Manage Orders>*

****

|  |  |
| --- | --- |
|  | **Related SDs** |
| **UC-01** | **SD-01** |
| **UC-02** | **SD-02** |
| **UC-03** | **SD-01** |
| **UC-04** | **SD-03** |
| **UC-05** | **SD-04** |
| **UC-06** | **SD-05** |
| **UC-07** | **SD-06** |
| **UC-08** | **SD-07** |
| **UC-09** | **SD-08** |
| **UC-10** | **SD-09** |
| **UC-11** | **SD-10** |
| **UC-12** | **SD-11** |
| **UC-13** | **SD-11** |
| **UC-14** | **SD-11** |
| **UC-15** | **SD-11** |
| **UC-16** | **SD-12** |
| **UC-17** | **SD-12** |
| **UC-18** | **SD-12** |
| **UC-19** | **SD-13** |
| **UC-20** | **SD-13** |
| **UC-21** | **SD-15** |
| **UC-22** | **SD-16** |
| **UC-23** | **SD-17** |
| **UC-24** | **SD-14, SD-18** |

# TABLE 6: Traceability Matrix (UCs and Sequence Diagram (SD) )

# Detailed Class Diagram

