Software Requirements Specification

for

Cafeteria Ordering System

Prepared by

Jose Morales, Marcello Sautto, Jon Smith, Kyle Urban

December 6, 2023

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this software requirement specification is to outline in clear and consistent detail the requirements involving brief descriptions and conditions for each use case. This will give us the ability to trace each use case for situations such as finding the originator of the use case, and program flow for the case itself. The intended audience for this document is for anyone involved in the management and development of this product since this document is concise enough to provide ample details no matter who is reading with any technical knowledge.

1.2 Scope

The Cafeteria Ordering System is an internet-based application that will allow Process Impact employees in the Clackamas campus in Oregon to order food that can be either picked up or delivered to their location. Meal choices can be customized to suit the customer's preferences and thus, their dietary needs. This application will take individual or group orders, schedule a pickup/delivery time, and process payments. The current initial release will only allow customers to order from the cafeteria on campus; however, the Cafeteria Ordering System will allow customers to order food from local restaurants in later releases. Additionally, the application provides cafeteria employees with menu management capabilities, allowing for menu updates and the ability to mark items that are not available for delivery. They can also view customer feedback to improve the customer experience even further.

The goal is to have the majority of Process Impact employees utilize the Cafeteria Ordering System to some degree. This application can reduce the amount of time that employees take to travel to, order, and obtain their meals from the cafeteria. Giving the option to have meals delivered shortens the time even more, presenting valuable convenience for the user. Additionally, more time can be dedicated to their work, as well as promote better productivity.
Cafeteria food wastage could decrease due to cafeteria employees being able to view future orders and plan accordingly. Similarly, operating costs should be decreased as a result.

1.3 Definitions, acronyms, and abbreviations

SRS = Software Requirement specification
Use Case = An operation or feature in the system to achieve some action by the user
Constraints = A restriction or limitation in system functionality
COS = Cafeteria Ordering System
Program flow = The subsequent steps a user takes in order to achieve a certain action
Alternate flow = Same as program flow, however the user takes different steps to achieve the same action
User Classes = A category of different user types based on role and needs for the system
Patrons = Process Impact employees that will use the application primarily during lunch hours
Operating Environment = This is where the software will operate including the hardware itself and any other components within the application or related to it
Use case diagram = A diagram that displays the flow of the system and different paths within the application for all users
Intranet = Private network used by the system organization
GUI = Graphical User Interface or the displays you see when using the system
UI = User Interface or the display an application user sees in a particular instance during use
CRUD = Create, Read, Update, Delete, these are the operations in which the system takes part of to complete a certain functionality

1.4 References

Use case Diagram, 10/01/2023, Marcello Sautto, Sources that can be reached are the use case features and different flows for the application.

Cafeteria Ordering System, 10/01/2023, Development Team, Sources that can be reached is the application itself that allows Clackamas campus employees to order food that can be picked up or delivered and customized as well.

Vision and Scope, 10/01/2023, Development team, Sources that can be reached is business requirements, scope and limitations as well as business context for this application.

1.5 Overview

This Software Requirements Specification contains overall information about the Cafeteria Ordering System's features, user classes, constraints, and environment. Section 2 will show the overall description of the Cafeteria Ordering System. Section 3 will display the use case diagram, as well as use case descriptions. Section 4 describes the external interface requirements, such as user interfaces, hardware interfaces, software interfaces, and communications interfaces. All nonfunctional requirements will be described in section 5 and any other requirements will be shown in section 6. Section 7 will display all functional requirements and section 8 will display all functional models (DFDs). Finally, section 9 will display all class analysis models (ICs and MCs).

2. Overall Description

2.1 Product Perspective

Before the inception and implementation of the Cafeteria Ordering System (COS), the organization's process-impact staff experienced significant inefficiencies. On average, they spent an arduous 90 minutes off-site, with nearly half of that time, approximately 40 minutes, devoted to travel, meal ordering, and payment. This investment of time often resulted in the process-impact staff receiving meals that didn't align with their preferences, as their desired choices were frequently sold out, compelling them to consider fewer ideal alternatives. This costs employees valuable time and money and incurs equivalent costs for the organization. Furthermore, these inefficiencies took a toll on cafeteria staff who found themselves dedicating extra hours to prepare meals that might ultimately go uneaten due to the lack of insight into daily popularity trends. As a forward-looking solution, COS will enable the organization to transition both the responsibility and the associated expenses for breakfast and supper provision to employees and local restaurants. This strategic shift not only promises a reduction in waste and inefficiency but also holds the potential to enhance overall employee happiness and satisfaction.

2.2 Product Features (FEs)

FE-1: Order meals from the cafeteria menu to be picked up or delivered

- The user can select and order a meal of their choice from the menu available on the application.
- The user can choose to pick up the meal or have it delivered.
- If delivery is selected, a cafeteria deliverer will have a specified location to drop off the food.

FE-2: Order meals from local restaurants to be delivered

- The Cafeteria Ordering System acts as a middleman between the user and the restaurant's online ordering system, if they have one.
- The user can choose to have the meal delivered to their location by the restaurant's delivery system.

FE-3: Create, view, modify, and delete meal service subscriptions

- The user can subscribe to a meal plan option.
- The user can view their subscription status (payment dates/other meal plans) as well as have the option to cancel their subscription.

FE-4: Register for meal payment options

- The user can input credit card details when it is time to pay for a meal and have the option to save it on their account.
- The user can choose to pay the deliverer or at the cafeteria when picking up the order.

FE-5: Create, view, modify, and delete cafeteria menus

- The cafeteria staff can view their menu on the Cafeteria Ordering System and edit/delete any items listed.

FE-6: Order custom meals that aren't on the cafeteria menu

- The cafeteria staff can list available ingredients for users to view.
- Users can order a meal not shown on the menu using available ingredients shown.
- Users can add, delete, or modify the ingredients used.

FE-7: Request feedback from customers

- Users will be asked to fill out a survey after their order has been fulfilled.
- Corporate management can view feedback on the application.

FE-8: Allow cafeteria staff to display menu items as "not available for delivery."

- Cafeteria staff can display the entire menu on the application with the option to mark some items as not available for delivery.

- Cafeteria staff can choose to display or withhold menu items that cannot be delivered.

FE-9: Allow custom instructions to be added. e.g. how a meat-based item should be cooked (medium, rare, well, etc.)

- Users can customize how they want a menu item to be cooked.
- The cafeteria staff can limit how much customization can be done for menu items.

FE-10: Allow a user to insert ingredients into a filter setting to filter out potential allergens. (reduces potential liability)

- Users can input their allergies before selecting a meal.
- Users can view a filtered menu without menu items containing their specified allergens.

2.3 User Classes and Characteristics

The following user classes feature their characteristics and actions that may be performed using the Cafeteria Ordering System.

UCC-1: Patrons (favored)

The patrons are made up of Process Impact employees that will use the application primarily during lunch hours. The patron will select a meal from the menu that can be further customized. After selection, the patron will enter their payment information. Patrons may also order meals for a group of people, which would follow the same process of meal customization, then payment.

UCC-2: Cafeteria Staff (favored)

The cafeteria staff will need to be able to learn how to use the Cafeteria Ordering System. The staff will be able to view upcoming orders (time and date) and prepare the meals accordingly. The cafeteria staff will also be able to modify the menu and mark the items that cannot be delivered.

UCC-3: Delivery Staff (favored)

The delivery staff will be able to view the delivery details, including a deadline and location of the delivery.

UCC-4: Corporate Management

The corporate management team will be able to view trends, sales, and overall productivity with the Cafeteria Ordering System. Management will also be able to view customer feedback and reviews.

UCC-5: Local Restaurant Managers

Restaurant managers will be able to view orders on their own ordering application that were communicated from the Cafeteria Ordering System. Any adjustments made on their own application will reflect on the Cafeteria Ordering System. When the application is expanded to local restaurants, the majority of the orders will be placed during breakfast, dinner, and weekends.

UCC-6: Payroll Department

The payroll department will be able to process payments made through the Cafeteria Ordering System. They will be able to deduct the charge of orders from payroll.

2.4 Operating Environment

OE-1: The Cafeteria Ordering System (COS) shall possess the capability to run seamlessly on all major operating systems that are currently supported by their respective owners, encompassing Windows, Linux, and macOS.
OE-2: The Cafeteria Ordering System (COS) shall interface with local restaurants and their ordering systems, necessitating robust communication and coexistence capabilities.
OE-3: The Cafeteria Ordering System (COS) shall interact with various hardware devices, including (potentially) Point of Service stations and printers, to enable the printing of order forms for each meal request.

2.5 Design and Implementation Constraints

LI-1: Some food items that are available from the cafeteria will not be suitable for delivery, so the menus available to patrons of the Cafeteria Ordering System will be a subset of the full cafeteria menus.

LI-2: The Cafeteria Ordering System shall be used only for the cafeteria at the main Process Impact campus in Clackamas, Oregon.

CO-1: The cafeteria staff will require training for internet usage; additional staff and vehicles for deliveries are also needed.

CO-2: Patrons need corporate Intranet to access the Cafeteria Ordering System.

CO-3: The payroll department has no identifiable resources to make the required software changes for the payroll deduction registration scheme.

CO-4: Restaurant managers may lack staff and capacity to handle order levels; may need Internet access.

CO-5: All features scheduled for release 1.0 must be fully operational.

CO-6: 95% of user acceptance tests must pass; all security tests must pass; compliance with corporate security standards must be demonstrated for all secure transactions

2.6 User Documentation

User manuals on how to fulfill major system features such as placing an order for pickup or delivery. This will include common scenarios and how to accomplish them. Employee training for internet and application usage will also be provided.

2.7 Assumptions and Dependencies

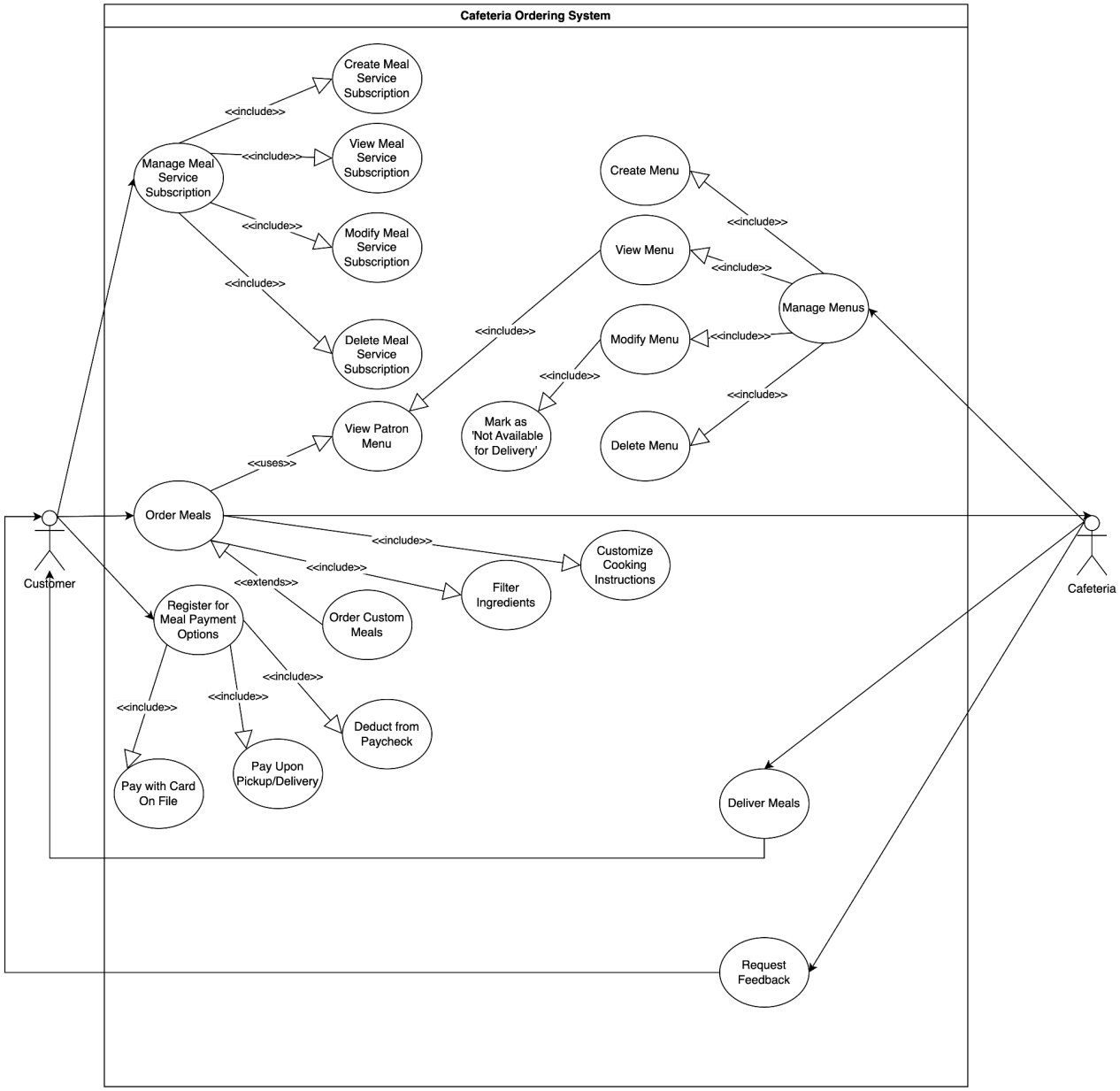
AS-1: Intranet-enabled computers and printers will be available in the cafeteria to permit cafeteria employees to process the expected volume of orders without missing any delivery time windows.

AS-2: Cafeteria staff and vehicles will be available to deliver all orders within 15 minutes of the requested delivery time.

DE-1: If a restaurant has its own on-line ordering system, the Cafeteria Ordering System must be able to communicate with it bi-directionally.

3. Use-Case Diagram with Use-case descriptions (UCs)

3.1 Use-case Diagram



3.2 Use-Case Description(s)

Use Case ID:	UC-1		
Use Case Name:	Order Meals		
Related Features	FE-1, FE-2		
Created By:	Marcello Sautto	**Last Updated By:**	Marcello Sautto
Date Created:	09/26/2023	**Last Revision Date:**	09/26/2023
Actors:	Customer, Cafeteria		
Description:	The customer accesses the cafeteria ordering system and views the menu. The customer then picks a meal from the menu and places an order.		
Trigger:	The customer clicks the “place an order” button on the user interface.		
Preconditions:	1. Customer has access to the Cafeteria Ordering System.		
Postconditions:	1. The order specified by the customer is sent to the cafeteria. 2. The customer is charged for the order to their registered method of payment.		
Normal Flow:	1. Customer clicks the “place an order” button on the UI. 2. System displays the available menu items specified by the cafeteria. 3. Customer adds meal(s) to the shopping cart. 4. System updates the shopping cart with the meal(s) added by the customer. 5. Customer clicks the “checkout” button at the bottom of the “order” page. 6. System prompts user with a list of the meals they added to their shopping cart along with an option to choose a payment method and a method of delivery. 7. Customer picks preferred payment method. 8. Customer selects “pickup” or “delivery” 9. System prompts user with button to “confirm order”		

	10. User clicks “confirm order” 11. Order is sent to the cafeteria. 12. Confirmation of the order placement is displayed on the screen for the customer.
Alternative Flows: **[Alternative Flow 1 – Not in Network]**	6a. For step 6, if the customer does not have any items in their shopping cart 1. A message will appear at the top of the page stating “Please add at least 1 item to your shopping cart to checkout” 2. The system remains on the current page. 8a. For step 8, if the customer selects “delivery” 1. Provide a prompt to enter the address for delivery. 2. Customer provides address for delivery. 3. System returns to interface on step 9
Exceptions:	7a. On step 7, if the payment method is invalid 1. An error message will appear on the screen saying “Invalid Payment Method” 2. The user will be redirected to the display on step 6. 8a. On step 8, if the customer selects “delivery” and the order contains a menu item not available for delivery. 1. The system will specify an error on the screen stating “your order contains a meal which is not available for delivery” 2. The system will default the delivery method to “pickup” 11a. On step 11, if the order fails to reach the cafeteria 1. An error message will appear on the screen saying “System error”. 2. The user will be redirected to the display on step 6.
Includes:	UC-3, UC-4, UC-5
Frequency of Use:	Every time a customer intends to order a meal.
Special Requirements:	User interface for ordering should be intuitive to navigate.
Assumptions:	1. The customer has internet access. 2. The customer is registered for the Cafeteria Ordering System.
Notes and Issues:	None

Use Case ID:	UC-2		
Use Case Name:	View Patron Menu		
Related Features	FE-3		
Created By:	Marcello Sautto	**Last Updated By:**	Marcello Sautto
Date Created:	09/26/2023	**Last Revision Date:**	09/26/2023
Actors:	Customer		
Description:	A customer views the menu items available for order.		
Trigger:	A customer clicks the “menu” button in the navigation bar at the top of the page.		
Preconditions:	1. Customer has access to the Cafeteria Ordering System.		
Postconditions:	1. System displays menu items on the screen		
Normal Flow:	1. Customer clicks the “menu” button in the navigation bar. 2. The system retrieves the menu and displays it on the screen.		
Alternative Flows: **[Alternative Flow 1 – Not in Network]**	2a. In step 2, if a menu item is not available for delivery 1. The menu item will have a ‘*’ next to it, indicating that it isn’t available for delivery (a legend will specify this at the top of the menu) Note: Insert a new row for each distinctive alternative flow.]		
Exceptions:	No identifiable exceptions.		
Includes:	None		
Frequency of Use:	When users want to browse the menu.		
Special Requirements:	Menu should be partitioned into categories based on the time of day and should be easy to read.		
Assumptions:	The customer understands English.		
Notes and Issues:	None		

Use Case ID:	UC-3		
Use Case Name:	Order Custom Meals		
Related Features	FE-1, FE-2, FE-6, FE-9, FE-10		
Created By:	Marcello Sautto	**Last Updated By:**	Marcello Sautto
Date Created:	09/30	**Last Revision Date:**	09/30
Actors:	Customer, Cafeteria		
Description:	The customer wants to order a meal which is not on the cafeteria menu.		
Trigger:	Customer clicks the “order” button.		
Preconditions:	1. Customer has access to the system.		
Postconditions:	1. The customer’s order is sent to the cafeteria. 2. The customer receives a confirmation that their order was received by the cafeteria. 3. The order is charged to the customer’s registered form of payment.		
Normal Flow:	1. Customer clicks the “place an order” button on the UI. 2. System displays available menu items available for the customer to order 3. System displays option to “customize” their order 4. Customer clicks “make your own meal” button on the UI. 5. Customer is presented with select menu options with available ingredients listed. 6. Customer creates meal and adds it to their shopping cart. 7. System updates the shopping cart with the meal(s) added by the customer. 8. Customer clicks the “checkout” button at the bottom of the page. 9. System displays list of items in shopping card with the option to choose a payment method and a delivery method. 10. Customer picks preferred payment method. 11. Customer selects “pickup” or “delivery”		

	12. System prompts user with button to “confirm order” 13. User clicks “confirm order” 14. Order is sent to the cafeteria 15. Confirmation of the order placement is displayed on the screen for the customer.
Alternative Flows: [Alternative Flow 1 – Not in Network]	8a. For step 8, if the customer does not have any items in their shopping cart 1. A message will appear at the top of the page stating “Please add at least 1 item to your shopping cart to checkout” 2. The system remains on the current page. 11a. For step 11, if the customer selects “delivery” 4. Provide a prompt to enter the address for delivery. 5. Customer provides address for delivery. 6. System returns to interface on step 9
Exceptions:	10a. On step 10, if the payment method is invalid 1. An error message will appear on the screen saying “Invalid Payment Method” 2. The user will be redirected to the display on step 9. 14a. On step 14, if the order fails to reach the cafeteria 1. An error message will appear on the screen saying “System error”. 2. The user will be redirected to the display on step 9.
Includes:	
Frequency of Use:	When users aren’t able to find a meal on the menu they want.
Special Requirements:	The custom order form should start with a general food item and become more specific with available ingredients.
Assumptions:	The customer understands that the meals and ingredients are limited to what the cafeteria has.
Notes and Issues:	None

Use Case ID:	UC-4		
Use Case Name:	Filter Ingredients		
Related Features	FE-9		
Created By:	Marcello Sautto	**Last Updated By:**	Marcello Sautto
Date Created:	09/30	**Last Revision Date:**	09/30
Actors:	Customer		
Description:	The customer can filter out ingredients when looking at the menu		
Trigger:	The customer is ordering a meal.		
Preconditions:	1. Customer is on the “menu” or “place an order” page.		
Postconditions:	1. The menu only includes items which do not include the filtered out ingredients.		
Normal Flow:	1. Customer views the menu 2. System displays the menu to the user 3. Customer clicks “filter” button 4. System presents a list of ingredients which are filterable 5. Customer selects/deselects ingredients to filter out and clicks “done” 6. System updates the menu to exclude meals containing ingredients in the filter		
Alternative Flows: [Alternative Flow 1 – Not in Network]	None		
Exceptions:	None		
Includes:	None		
Frequency of Use:	When customers have dietary restrictions and/or preferences.		
Special Requirements:	Ingredients available to filter should start with the most common allergens for ease of use.		

Assumptions:	The customer is aware of their own dietary restrictions.
Notes and Issues:	1. The menu must be parsable by the filter in order to perform accurately.

Use Case ID:	UC-5		
Use Case Name:	Customize Cooking Instructions		
Related Features	FE-10		
Created By:	Marcello Sautto	**Last Updated By:**	Marcello Sautto
Date Created:	09/30	**Last Revision Date:**	09/30
Actors:	Customer		
Description:	Customers can specify how they want their food prepared.		
Trigger:	Customer starts an order.		
Preconditions:	1. Customer has access to the system. 2. Customer has begun an order.		
Postconditions:	1. Customer has added a meal to their order with specific instructions		
Normal Flow:	1. Customer starts an order. 2. System presents the user with a menu to choose from. 3. Customer adds a meal. 4. System presents a text box which the customer can enter instructions. 5. Customer enters notes and adds item to the shopping cart. 6. System updates the shopping cart with the meal and includes the notes specified by the customer.		
Alternative Flows: [Alternative Flow 1 – Not in Network]	4a. In step 4 of the normal flow, if the customer does not have specific instructions		

	1. The customer can add the meal to their shopping cart without filling out the text box.
Exceptions:	None
Includes:	None
Frequency of Use:	When users want to take an ingredient out of the meal or want to specify the way it is cooked.
Special Requirements:	None
Assumptions:	The customer understands that the request can only be used to hold ingredients/add-ons or specify the rarity of their meat within guidelines for safe consumption.
Notes and Issues:	1. None

Use Case ID:	UC-6		
Use Case Name:	Register for Meal Payment Options		
Related Features	FE-3, FE-4		
Created By:	Kyle Urban	**Last Updated By:**	Kyle Urban
Date Created:	09/29/2023	**Last Revision Date**	09/29/2023
Actors:	Process Impact Employee		
Description:	The user can select their payment option depending on their preference.		
Trigger:	After selecting a meal, the final step displayed in the ordering process is to identify the payment method.		
Preconditions:	1. The user has at least one item in their cart. 2. The user does not have a subscription.		
Postconditions:	1. The user has credit card information saved on their account. 2. The user has paid for their meal. 3. The user is now awaiting delivery or will go to the cafeteria to pay.		
Normal Flow:			

	1. User opens the application on their device. 2. User logs into account. 3. User starts a new order. 4. User adds menu items to their cart. 5. User selects the “checkout” option. 6. User chooses “input credit card information.” 7. User is prompted to save credit card information if they wish. 8. User selects “place order.” 9. User can check their saved card information on their account.
Alternative Flows: **[Alternative Flow 1 – Pay at cafeteria]**	5a. User chooses to pay at the cafeteria. 1. User is given their total amount due and pickup time.
Alternative Flow 2 - [Decline to save card information]	6a. User declines to save credit card information. 1. User will be prompted to enter card information on all future orders.
Exceptions:	4a. No menu items placed in cart. 1. “Checkout” option cannot be selected. 2. Message popup indicates that the cart is empty. 3. User adds an item to the cart. 4. User can now select “checkout.” 5. Use Case resumes on step 5. 5a. Incorrect credit card information 1. Message popup indicates invalid card information. 2. User enters correct information. 3. Use case resumes on step 6.
Includes:	UC-7, UC-8, UC-9
Frequency of Use:	60 per hour.
Special Requirements:	Security requirements need to be addressed for this use case.

Assumptions:	The user has created an account.
Notes and Issues:	What other forms of payment can the user choose from?

Use Case ID:	UC-7		
Use Case Name:	Pay Upon Pickup/Delivery		
Related Features	FE-3, FE-4		
Created By:	Kyle Urban	**Last Updated By:**	Kyle Urban
Date Created:	09/29/2023	**Last Revision Date**	09/29/2023
Actors:	Process Impact Employee		
Description:	The user has chosen to pay at the cafeteria or directly to the deliverer.		
Trigger:	The user declined to enter their payment information and selected the option to pay upon pickup/delivery.		
Preconditions:	1. User has declined to input payment information. 2. User has at least one menu item in their cart.		
Postconditions:	1. Confirmation page is displayed. 2. The user is now awaiting delivery or will go to the cafeteria to pay.		
Normal Flow:	1. User opens the application on their device. 2. User logs into account. 3. User starts a new order. 4. User adds menu items to their cart. 5. User selects the “checkout” option. 6. User chooses “pay at time of pickup/delivery.” 7. User selects “place order.”		
Alternative Flows: **[Alternative Flow 1 –**	5a. Menu item cannot be delivered.		

Delivery check]	1. User is prompted to pick up their order at an estimated time. 5b. User chooses delivery. 1. User is prompted to enter an address.
Exceptions:	5a. User is not in range of allowable delivery. 1. User receives an error message.
Includes:	UC-6
Frequency of Use:	Approximately 60 per hour.
Special Requirements:	None
Assumptions:	User is an employee at the Clackamas location.
Notes and Issues:	None

Use Case ID:	UC-8		
Use Case Name:	Pay with Card on File		
Related Features	FE-2, FE-4		
Created By:	Kyle Urban	**Last Updated By:**	Kyle Urban
Date Created:	09/29/2023	**Last Revision Date**	09/29/2023
Actors:	Process Impact Employee		
Description:	User can select their card on file at time of payment.		
Trigger:	User is prompted to add a card on file through their account or at the time of first purchase.		
Preconditions:	1. User has an active account.		
Postconditions:	1. User is charged the amount due before receiving the order.		
Normal Flow:	1. User opens the application on their device.		

	2. User logs into account. 3. User starts a new order. 4. User adds menu items to their cart. 5. User selects the “checkout” option. 6. User uses saved card information to pay. 7. User selects “place order.”
Alternative Flows: **[Alternative Flow 1 – Change card information]**	6a. User can change card information 1. User selects “modify payment information.” 2. User edits the information. 6b. User can add new card 1. User selects “modify payment information.” 2. User selects “add new card.” 3. User inputs new card information.
Exceptions:	6a, 6b Exception 1: User enters invalid card information. 1. User is prompted to correct invalid information.
Includes:	UC-6
Frequency of Use:	Approximately 60 per hour.
Special Requirements:	Security requirements
Assumptions:	User’s card on file is active.
Notes and Issues:	How many active cards on file can be stored?

Use Case ID:	UC-9		
Use Case Name:	Deduct from Paycheck		
Related Features	FE-3, FE-4		
Created By:	Kyle Urban	**Last Updated By:**	Kyle Urban
Date Created:	09/29/2023	**Last Revision Date**	09/29/2023
Actors:	Process Impact Employee		
Description:	The user can opt to have their meal payments deducted from their paycheck		

Trigger:	The user selected the option to have meal order payments deducted from their paycheck when they created their account.
Preconditions:	1. User must be a Process Impact Employee and receive paychecks from the Clackamas location. 2. User must have opted to use paycheck as default payment method when creating their account.
Postconditions:	1. User successfully pays for meal with paycheck. 2. User can pay for part of the meal with paycheck.
Normal Flow:	1. User opens the application on their device. 2. User logs into account. 3. User starts a new order. 4. User adds menu items to their cart. 5. User selects the “checkout” option. 6. User confirms the paycheck option to pay for the order in its entirety. 7. User selects “place order.”
Alternative Flows: **[Alternative Flow 1 – Partial payment]**	6a. User selects partial payment 1. User specifies the amount to deduct from paycheck. 2. User confirms the amount. 3. User proceeds to input credit card information/use card on file. 4. Resumes on step 7.
Exceptions:	None
Includes:	UC-6
Frequency of Use:	Approximately 60 per hour.
Special Requirements:	Application can access their employee portal to view paycheck information.

Assumptions:	User will receive a paycheck large enough to cover payment.
Notes and Issues:	None

Use Case ID:	UC-10		
Use Case Name:	Manage Meal Subscription Service		
Related Features	FE-3		
Created By:	Kyle Urban	**Last Updated By:**	Kyle Urban
Date Created:	09/29/2023	**Last Revision Date**	09/29/2023
Actors:	Process Impact Employee		
Description:	The user can access the meal subscription service interface to manage their subscription.		
Trigger:	The user wants to view or change their subscription status.		
Preconditions:	1. User must have an active account.		
Postconditions:	1. User will access their active meal subscription service screen. 2. User will be prompted to select a subscription.		
Normal Flow:	1. User opens the application. 2. User logs into account. 3. User clicks on their account tab. 4. User selects “manage meal subscription.” 5. User is redirected to the subscription page.		
Alternative Flows: **[Alternative Flow 1 – No subscription]**	4a. User does not have an active subscription. 1. User is redirected to an alternate page to select a subscription option.		
Exceptions:	None		

Includes:	UC-11, UC-12, UC-13, UC-14
Frequency of Use:	Once a week per person
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

Use Case ID:	UC-11		
Use Case Name:	Create Meal Service Subscription		
Related Features	FE-3		
Created By:	Kyle Urban	**Last Updated By:**	Kyle Urban
Date Created:	09/29/2023	**Last Revision Date**	09/29/2023
Actors:	Process Impact Employee		
Description:	User wants to subscribe for the meal service plan		
Trigger:	When creating their account, they are automatically offered to subscribe to the meal service. The user can opt to subscribe at any time through their account page.		
Preconditions:	1. User has an active account.		
Postconditions:	1. User has successfully subscribed for the meal service.		
Normal Flow:	1. User opens application. 2. User logs into account. 3. User visits their account page. 4. User selects “meal service subscription.” 5. User decides which subscription to use. 6. User selects “subscribe.” 7. User selects payment method. 8. User selects “confirm.”		
Alternative Flows: **[Alternative Flow 1 –**	4a. User has had an active subscription in the past.		

User had a past subscription]	1. User is prompted with the details of their previous subscription and if they want to renew it. 2. User selects “renew.” 3. Resume on step 7.
Exceptions:	None
Includes:	UC-10, UC-12, UC-13, UC-14
Frequency of Use:	Approximately 60 times per week.
Special Requirements:	None
Assumptions:	User has never subscribed to the meal service.
Notes and Issues:	None

Use Case ID:	UC-12		
Use Case Name:	View Meal Service Subscription		
Related Features	FE-3		
Created By:	Kyle Urban	**Last Updated By:**	Kyle Urban
Date Created:	09/29/2023	**Last Revision Date**	09/29/2023
Actors:	Process Impact Employee		
Description:	User wants to review the details of their subscription, such as their next payment date.		
Trigger:	User visits their account page and selects “meal service subscription.”		
Preconditions:	1. User has an active meal service subscription.		
Postconditions:	1. User successfully views all the details of their subscription.		
Normal Flow:	7. User opens application. 8. User logs into account. 9. User visits their account page. 10. User selects “meal service subscription.” 11. User is redirected to a new page with subscription details.		
Alternative Flows: **[Alternative Flow 1 –**	User has had an active subscription in the past.		

Subscription has ended]	4. User is prompted with the details of their previous subscription and if they want to renew it.
Exceptions:	None
Includes:	UC-10, UC-11, UC-13, UC-14
Frequency of Use:	Approximately 60 times per week.
Special Requirements:	None
Assumptions:	User
Notes and Issues:	None

Use Case ID:	UC-13		
Use Case Name:	Modify Meal Service Subscription		
Related Features	FE-3, FE-4		
Created By:	Jose Morales	**Last Updated By:**	Jose Morales
Date Created:	10/01/2023	**Last Revision Date**	10/01/2023
Actors:	Customer		
Description:	The customer will have the ability to manage their meal service subscription and then modify the meal service subscription directly.		
Trigger:	Customer clicks the button for modifying their service.		
Preconditions:	1. A customer exists with a meal service subscription in the system.		
Postconditions:	1. Customer user will be able to modify their service subscription		
Normal Flow:	1. Customer user logs into application 2. Customer views the UI 3. Customer clicks the button that reads “manage meal service subscription” 4. Customer clicks on the button that says “modify meal service subscription”		
Alternative Flows: [Alternative Flow 1 – Not in Network]	None		
Exceptions:	3a. Customer has no meal service subscription		

	1. Customer will be prompted back to main menu 2. Customer is prompted to create new meal service subscription
Includes:	UC-10
Frequency of Use:	Approximate 500 per week
Special Requirements:	None
Assumptions:	There exists a meal service subscription from a customer.
Notes and Issues:	Handle potential creation of multiple meal service subscriptions.

Use Case ID:	UC-14		
Use Case Name:	Delete Meal Service Subscription		
Related Features	FE-3, FE-4		
Created By:	Jose Morales	**Last Updated By:**	Jose Morales
Date Created:	10/01/2023	**Last Revision Date**	10/01/2023
Actors:	Customer		
Description:	The customer will have the ability to manage their meal service subscription and then choose to delete the meal service subscription directly.		
Trigger:	A customer engages in meal service subscription management functionality and click the button to delete their service.		
Preconditions:	1. A customer exists with a meal service subscription in the system.		
Postconditions:	1. Customer user will be able to delete their service subscription		
Normal Flow:	1. Customer user logs into application 2. Customer views the UI 3. Customer clicks the button that reads “manage meal service subscription” 4. Customer clicks on the button that says “delete meal service subscription”		
Alternative Flows: [Alternative Flow 1 – Not in Network]	None		

Exceptions:	3a. Customer has no meal service subscription 1. Customer will be prompted back to main menu 2. Customer is prompted to create new meal service subscription
Includes:	UC-10
Frequency of Use:	Approximate 500 per week
Special Requirements:	None
Assumptions:	A meal service subscription exists from customer.
Notes and Issues:	Handle any active orders after deletion that are still in progress.

Use CaseID:	UC-15		
Use Case Name:	Manage Menus		
Related Features	FE-5		
Created By:	Jon Smith	Last Updated By:	Jon Smith
Date Created:	10/1/2023	Last Revision Date:	10/1/2023

Actors:	Cafeteria Staff
Description:	Cafeteria Staff will be able to manage the menus that have been created.
Trigger:	Cafeteria Staff must log in and choose to manage the menu.
Preconditions:	1.The user must have an active account with the required permissions.
Postconditions:	1. Cafeteria staff will have access to the menu modification page. 2. Cafeteria staff will be prompted to select a modification to make

Normal Flow:	1. Cafeteria staff open applications. 2. Cafeteria staff authenticates themselves to application. 3. User clicks “Manage Menus.” 4. System validates if the client has permission to access menu modifications. 5. Cafeteria staff redirected to the menu modification page.
Alternative Flows:	None
Exceptions:	2a. In step 2 of the normal flow, if the staff member fails to authenticate: 1. The system will prompt staff to re enter credentials. 2. Upon successful entry Use Case resumes on step 3 4a. In step 4 of the normal flow, if the user doesn’t have the necessary authorization to manage the menus: 1. The system will display to the user they don’t have the necessary privileges and to contact an administrator if this is in error. 2. The user is redirected to the home page. Use Case resumes on step 3 of normal flow.
Includes:	None
Frequency of Use:	Daily
Special Requirements:	Authentication and general security requirements.
Assumptions:	The user has an active account with the service.
Notes and Issues:	Should all cafeteria staff be able to modify the menu?

Use Case ID:	UC-16		
Use Case Name:	Create Menu		
Related Features	FE-5		
Created By:	Jon Smith	Last Updated By:	Jon Smith
Date Created:	10/1/2023	Last Revision Date:	10/1/2023
Actors:	Cafeteria Staff		
Description:	Cafeteria Staff will be able to use the manage menu option to create a new menu.		

Trigger:	Cafeteria staff must be logged in to the application
Preconditions:	1. The user must be authenticated with the service
Postconditions:	1. Cafeteria Staff will have created a new menu
Normal Flow:	1. Cafeteria staff open applications. 2. Cafeteria staff authenticates themselves to application. 3. User clicks “Manage Menus.” 4. System validates user has permissions required to manage menus 5. User clicks “Create Menu” 6. User enters menu items into new menu template 7. User clicks “Publish Menu " to add the new menu to the service.
Alternative Flows: [Alternative Flow 1 – Not in Network]	5a. In step 5 of the normal flow, if the client decides to restart the menu creation: 1. User clicks the undo button to remove all information previously typed. 2. System prompts user for confirmation of data removal 3. Upon receipt of approval all prior information is deleted and the user starts with a new blank menu. 4. Resume use case from step 5
Exceptions:	6a. In step 6 of the normal flow, if the client submits an empty menu: 1. System alerts user that the have tried to submit an empty menu 2. Resume use case from step 5
Includes:	None
Frequency of Use:	Daily
Special Requirements:	None
Assumptions:	User has an active account with the service with the required permissions to add a new menu.
Notes and Issues:	Limit the amount of menus that can be created?

Use Case ID:	UC-17		
Use Case Name:	View Menu		
Related Features	FE-5		
Created By:	Jon Smith	Last Updated By:	Jon Smith
Date Created:	10/1/2023	Last Revision Date:	10/1/2023
Actors:	Cafeteria Staff		
Description:	Cafeteria Staff will be able to view menus created by colleagues		
Trigger:	Cafeteria staff must login and choose to manage the menu		
Preconditions:	1. . The user must have an active account with the required permissions.		
Postconditions:	1. Cafeteria Staff will have all of the created menus open to view 2. Cafeteria Staff will be able to choose which menus patrons can view		
Normal Flow:	1. Cafeteria staff open applications. 2. Cafeteria staff authenticates themselves to application. 3. User clicks “Manage Menus.” 4. User clicks “View Menus” 5. User redirected to available menus that can be viewed		
Alternative Flows: [Alternative Flow 1 – Not in Network]	None		
Exceptions:	[Describe any anticipated **error conditions** that could occur during execution of the use case, and define how the system is to respond to those conditions. e.g. Exceptions to the Withdraw Case transaction 5a. In step 5 of the normal flow, if there are no menus to be viewed: 1. System alerts user there are no menus to view 2. Use case resumes on step 4		
Includes:	UC-2		
Frequency of Use:	Daily		
Special Requirements:	None		

Assumptions:	1. User has the necessary permissions needed to manage the menus
Notes and Issues:	1. Should all menus be viewable by all cafeteria staff?

Use Case ID:	UC-18		
Use Case Name:	Modify Menu		
Related Features	FE-5		
Created By:	Jon Smith	Last Updated By:	Jon Smith
Date Created:	10/1/2023	Last Revision Date:	11/1/2023
Actors:	Cafeteria Staff		
Description:	Cafeteria Staff will be able to modify and existing menu to add or remove items		
Trigger:	Cafeteria staff is logged in and chooses to manage the menu		
Preconditions:	1. Cafeteria staff must have an active account		
Postconditions:	1. Cafeteria staff will be able to modify existing menus		
Normal Flow:	1. Cafeteria staff will open the application 2. Cafeteria staff will login to the application 3. User clicks “Manage Menus” 4. User redirected to the manage menus screen 5. User clicks “Modify Menus” 6. user is redirected to the modify menu screen where they can make adjustments to available menus		
Alternative Flows: [Alternative Flow 1 – Not in Network]	None		
Exceptions:	[Describe any anticipated **error conditions** that could occur during execution of the use case, and define how the system is to respond to those conditions. e.g. Exceptions to the Withdraw Case transaction 5a. In step 5 of the normal flow, if the user enters teh modify menus page and no menus have been published:		

	1. System alerts user that there are no menus to modify 2. User is redirected to home page 3. Use Case resumes on step 4 of normal flow
Includes:	None
Frequency of Use:	Daily
Special Requirements:	None
Assumptions:	1. User has the necessary permissions needed
Notes and Issues:	1. Can all menus be modified by all cafeteria staff? 2. Can menus that have been published to patrons be modified as well?

Use Case ID:	UC-19		
Use Case Name:	Delete Menu		
Related Features	FE-5		
Created By:	Jon Smith	Last Updated By:	Jon Smith
Date Created:	10/1/2023	Last Revision Date:	11/1/2023
Actors:	Cafeteria Staff		
Description:	Cafeteria Staff will be able to delete menus		
Trigger:	Cafeteria staff must open the application and login		
Preconditions:	1. Users must have an active account with the service.		
Postconditions:	1. User will be able to delete menus from the application		
Normal Flow:	1. Cafeteria Staff opens application 2. Cafeteria Staff logins and authenticates themselves 3. Cafeteria staff select manage menus 4. Cafeteria staff are redirected to manage menu screen 5. Cafeteria staff select delete menus 6. Cafeteria staff are redirected to page that allows them to delete available menus		
Alternative Flows: **[Alternative Flow 1 – Not in Network]**	None		

Exceptions:	[Describe any anticipated **error conditions** that could occur during execution of the use case, and define how the system is to respond to those conditions. e.g. Exceptions to the Withdraw Case transaction 6a. In step 6 of the normal flow, if there are non menus that can be deleted from the service: 1. System alerts user that n menus are current available 2. System redirects user to modify menu page 3. Use Case resumes from step 4 of normal flow
Includes:	None
Frequency of Use:	Daily
Special Requirements:	None
Assumptions:	1. User has the required permissions needed to manage the menus
Notes and Issues:	1. Can a menu be deleted while there are pending orders from it?

Use Case ID:	UC-20		
Use Case Name:	Deliver Meals		
Related Features	FE-1, FE-2		
Created By:	Jose Morales	**Last Updated By:**	Jose Morales
Date Created:	09/29/2023	**Last Revision Date:**	09/29/2023
Actors:	Cafeteria staff and customer		
Description:	The cafeteria staff will have the ability to deliver meals to/for the customer with the specified location to drop off the food		
Trigger:	1. User places order as delivery 2. Restaurant receive delivery order		
Preconditions:	1. User completes food delivery order option 2. Food item is available for delivery		
Postconditions:	1. Cafeteria receives food delivery order 2. Customer receives food order		

	3. If use case fails then customer needs to be refunded of any payment
Normal Flow:	1. Customer logs into application 2. Customer orders meal 3. Cafeteria system receive order 4. Cafeteria communicates with restaurant for food order 5. Food is delivered to the customer
Alternative Flows: [Alternative Flow 1 – Not in Network]	3a. Customer filters ingredients 3b. Customer customizes cooking instructions 3c. Customer views patron menu 3d. Customer sends custom meals
Exceptions:	2a. The desired food item is not available for delivery 1. Alert user 2. User can try different food item 4a. Restaurant does not have food item 1. Alert user for appropriate step such as trying a different food item or canceling
Includes:	None
Frequency of Use:	On demand
Special Requirements:	Accurate real time updates on food availability and delivery
Assumptions:	1. User has placed an order 2. There is existing food quantities available 3. Restaurant or food provider will have delivery system 4. Drivers available to fulfill delivery order
Notes and Issues:	1. Ensure correct delivery address validation 2. Handle cases where resistance ran out of desired food item

Use Case ID:	UC-21		
Use Case Name:	Request Feedback		
Related Features	FE-7		
Created By:	Jose Morales	**Last Updated By:**	Jose Morales
Date Created:	09/29/2023	**Last Revision Date:**	09/29/2023
Actors:	Cafeteria staff and Customer		
Description:	Cafeteria will have the feature to request feedback from the customer regarding ordering meals, registering for meal payment options and managing meal service subscriptions		
Trigger:	1. The process of the customer sends order and it gets fulfilled will trigger this use case 2. Cafeteria user will be able to initiate feedback request from customer		
Preconditions:	1. Customer has created an order 2. Customer order has been received in the system 3. Cafeteria staff logs into application and initiates request feedback flow		
Postconditions:	1. Feedback will contain information about the system or procedure 2. Feedback is sent back to the system 3. At the end the customer will be able to choose an appropriate path for their features or start from the beginning		
Normal Flow:	1. Cafeteria initiates request feedback action for user		
Alternative Flows: **[Alternative Flow 1 – Not in Network]**	1a. User orders meal and is requested for feedback 1b. User register for meal payment option and is requested for feedback 1c. User manages meal service subscription and is requested for feedback		
Exceptions:	1a. User's feedback unsuccessfully 1. System needs to validate successful status code else resubmit 2. If feedback is not sent then prompt user to try again or cancel		
Includes:	None		
Frequency of Use:	Approximately 20 per hour		

Special Requirements:	None
Assumptions:	The user has sent out an order to be fulfilled
Notes and Issues:	Ensure user is not spammed for feedback prompts

Use Case ID:	UC-22		
Use Case Name:	Mark as 'Not Available for Delivery'		
Related Features	FE-1, FE-8		
Created By:	Jose Morales	**Last Updated By:**	Jose Morales
Date Created:	09/29/2023	**Last Revision Date**	09/29/2023
Actors:	Cafeteria/Staff,		
Description:	Cafeteria staff will be able to modify the cafeteria menu options for delivery		
Trigger:	The cafeteria staff will login and choose to manage the menu and then modify it.		
Preconditions:	1. When a food item is not available to deliver such as 0 quantity 2. Not in the working hours in which delivery would be available. 3. Cafeteria staff entering menu management		
Postconditions:	1. The particular food item will be displayed as not available for delivery after modify menu change 2. Food item will not be offered for delivery 3. Food item will not be shown 4. If actor's goal is not achieved then food item should still not be delivered		
Normal Flow:	1. Cafeteria staff or user opens the application on their end device 2. User logs into appropriate account 3. User enters menu management mode 4. Modifies menu 5. Mark item(s) as not available for delivery 6. Corresponding items will be not shown or simply marked as not available for delivery		

Alternative Flows: [Alternative Flow 1 – Not in Network]	3a. User views menu to see which items to mark 1. From here, user can backout of view mode 2. User can enter menu management mode 5a. User chooses not to display food item on menu 1. No food item will be shown to the end user
Exceptions:	2a. User enters wrong login information 1. Prompt user to try again 5a. User unsuccessfully marks item as not available for delivery 1. Prompt user to clarify if changes are correct 2. If correct then move on else try again
Includes:	UC-15, UC-18
Frequency of Use:	On demand weekly
Special Requirements:	Reliable validation for real time updates
Assumptions:	1. Some food items are marked as delivery 2. Existing populated menu with food items 3. Food items can be available for delivery
Notes and Issues:	1. Automatically marking foot item as not available for delivery if quantity is 0

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

	Related Use-cases (UCs)
FE-1	UC-1, UC-3,
FE-2	UC-1
FE-3	UC-6, UC-7, UC-8, UC-9, UC-10, UC-11, UC-12, UC-13, UC-14
FE-4	UC-6, UC-7, UC-8, UC-9, UC-10, UC-11, UC-12, UC-13, UC-14
FE-5	UC-15, UC-16, UC-17, UC-18, UC-19
FE-6	UC-2, UC-3
FE-7	UC-21
FE-8	UC-22
FE-9	UC-4,
FE-10	UC-5

4. External Interface Requirements

4.1 User Interfaces

Standards that should follow for all UI's would ensure that the user will see the correct display based on the user's machine or device used. Additionally, any and all error alerts will display why the error occurred and possible actions to be taken to either proceed in the application or an attempt to correct the error. All buttons and text fields will share a common theme that way it all looks uniform to the user.

Customer UI

The customer user interface will include GUI's for programs flows including ordering meals, registering for meal payment options and managing meal service subscriptions. To dive into the ordering meals user interface, the user will see options in the forms of buttons to customize their potential cooking preferences, filtering ingredients for their food order and even ordering custom meals if the restaurant allows for it. Should any error happen here which could include a missing ingredient or unable to make alterations to the food order then the user will be prompted via alert

with the appropriate message and actions like canceling or simply preceding forward without the customization.

Furthermore, another UI for the customer involves registering for meal payment options. Once this path is taken, the user will see options such as pay with card on file, pay upon pickup/delivery option and deduct from paycheck for payment in the system. This will be in the form of buttons or boxes to click. Should any error occur here which would involve wrong card information on file then the user will be asked to enter the correct credentials. Other anticipated errors would be placing a delivery order for a food item that is not actually available for delivery then the user should be alerted for such a case.

One of the last UI for the user is managing meal service subscriptions. At this point, the user has chosen this path to accomplish functionality such as creating a meal service subscription, viewing meal survive subscriptions, modifying them, deleting, and viewing patron menus. Each button is then accompanied by more program flows to take such as user inputs in the form of text fields, boxes and CRUD operations. In order to prevent CRUD errors, there should be validation methods in place to ensure correct data format for user input and data manipulation that the system will undergo.

Cafeteria Ordering system UI

On the other hand, we have a major user interface designed for cafeteria use which can involve some admin functionality. Within this scope, possible paths in this user interface include managing menus which then branch out to creating menus, viewing them, modifying and deleting menus. Besides that, this section of UI will also display information and instructions on how to deliver the meals to the customer as well as a button option to request feedback from customers after they place food orders. Anticipated application errors that might occur would be CRUD operations that are not successful. In which case the system should recognize the attempt by the user, return an error code and alert the user that the CRUD operation failed and should try again.

4.2 Hardware Interfaces

The application will need to access the user’s location to gain access to certain functionalities, such as a minimum distance supported for delivery services. Supported device types include mobile devices and computers.

4.3 Software Interfaces

Software Components

Component	Name	Version	Description
Database Management System	MySQL	8.0	The cafeteria ordering system relies on a MySQL database to store and manage various data, including menu items, customer orders, user accounts, and transaction records.
Operating System	Linux (Ubuntu)	20.04 LTS	The system runs on a Linux-based server to ensure stability, security, and scalability. The operating system provides the environment for hosting the software components.
Web Server	Apache HTTP Server	2.4	Apache serves as the web server for the cafeteria ordering system, handling incoming HTTP requests from clients and forwarding them to the appropriate components for processing.
Programming Language	Python	3.9	Python is used for developing the core application logic and handling business logic. It also connects to various APIs and libraries for specific functionalities.
Frontend Framework	React	17	React is used to create the user interface (UI) for customers and cafeteria staff. It communicates with the backend via API calls to display menus, process orders, and update order statuses in real-time.
Payment Gateway API	Stripe	2023-08-16	Stripe will be used to securely process payments for customer orders.
User Authentication API	JWT	RFC 7519	JWT will be used for user registration and login.

Message Broker	RabbitMQ	3.8	RabbitMQ is used for asynchronous communication within the system. For example, it helps in managing order queue processing and sending notifications to customers when their orders are ready.

Data Items and Messages

Data Item	**Description**
Incoming Data	Customer orders, user registration/login requests, menu item selections, and payment details are incoming data items.
Outgoing Data	Confirmation messages, order updates, payment confirmations, and email notifications are outgoing data items.

Services and Communications

Service	**Description**
Order Processing Service	Communicates with the database to store and retrieve order details, RabbitMQ for order queuing, and the payment gateway for processing payments.
User Management Service	Handles user registration, authentication, and authorization, including communication with the authentication API.
Menu Service	Retrieves menu information from the database to display it to customers.
Notification Service	Sends email or SMS notifications to customers and staff regarding order updates and status changes.

Shared Data

Data Item	**Description**

Customer and Order Information	Shared across multiple components, including the frontend, backend, and database, ensuring consistency and accuracy in order processing.

4.4 Communications Interfaces

CI-1: The Cafeteria Ordering Systems (COS) interfacing shall exclusively utilize HTTPS, ensuring the encryption of user data during transit to either our system or the local restaurant's system.
CI-2: The Cafeteria Ordering System (COS) shall retain high data transfer rates and throughput consistently with a high level of performance to prevent any degradation in the service's usability, thereby mitigating user frustration.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Stable real time updates

This would include when any users attempt to modify a field attribute such as CRUD operations for menus including creating a menu, accessing a menu, updating a menu and deleting a menu. After each CRUD operation, the customer should see the updated change in real time right after the operation is executed. By doing this, the customer can receive accurate menu choices instead of taking the time to carry out a process only to find out at the end it was all for nothing especially if a food item is not available or not accurately marked as available for delivery.

Ensure user is not spammed with feedback requests

Getting feedback from your clients is a crucial feature to ensure continuous and lasting success on any application. However, if the user is spammed by feedback requests then they are likely to get frustrated and not submit anything at all.

5.2 Safety Requirements

Featured menu items will show the potential allergens, calories, and spice level.

5.3 Security Requirements

SR-1: Under the current operating assumptions, the Cafeteria Ordering System shall ensure users undergo authentication by providing their name, following which they will receive an incrementing ID. This ID will be sent to them upon confirmation of their order, and they can then present it at the location to validate both payment and ownership of the requested meal.
SR-2: The Cafeteria Ordering System shall ensure safeguarding PII/SPII such as names, possible addresses, payment methods, and order ID numbers to prevent any potential theft of meals, which could diminish the system's value.
SR-3: The Cafeteria Ordering System shall handle all payment information in strict compliance with the Payment Card Industry Data Security Standards (PCI DSS) or consider employing a third-party payment gateway. While this may lead to a slight increase in administrative costs, it significantly reduces liability.
SR-4: For enhanced security, the Cafeteria Ordering System (COS) shall implement two-factor authentication (2FA) and encryption protocols for any user information that necessitates storage to facilitate the service.

5.4 Software Quality Attributes

ID	Quality Attribute	Description	Priority
QA-1	Availability	In order to maximize the usage of this system by employees, It is important that downtime of the Cafeteria Ordering System is minimized, especially during business hours.	Critical
QA-2	Correctness	The patron menu should reflect the latest changes implemented by the cafeteria staff at all times. This will prevent cases where a customer attempts to order a meal which is currently unavailable.	Critical

QA-3	Flexibility	A major appeal of the Cafeteria Ordering System is convenience, so allowing the user to customize their payment options and cafeteria staff to customize their menu options will encourage a mutual benefit from both sides of the user spectrum.	Critical
QA-4	Usability	A customer should easily be able to find the meals they are interested in and place an order.	High
QA-5	Interoperability	All orders sent from the customer should indicate whether they've successfully reached the cafeteria. If not, an error message should be displayed on the customer's interface.	High
QA-6	Portability	Customers should be able to access the Cafeteria Ordering System from their computer or a mobile device. The user interface should support all available platforms.	Medium

6. Other Requirements

OR-1: Menu items can be marked as "unavailable" or "ingredients out of stock" by cafeteria staff.
OR-2: Language preference can be selected at any time in settings.

7. Functional Requirements (FRs)

7.1 FR1: Save card information

a. Introduction/Functionality: The payment screen shall allow customers to save their card information for easier payments in the future.
b. Traced: *UC-6*
c. Inputs: *Customer's card information.*
d. Processing: *The Cafeteria Ordering System shall save the information on their account.*
e. Outputs: *Account settings option to view/edit/delete active card on file.*
f. Error Handling: *If the card is declined, an error message shall prompt the user to review their card information and try again.*

7.2 FR2: Set payment status to deduct from paycheck

a. Introduction/Functionality: The account page shall give users the option to have their payments deducted from their paycheck with the help from the payroll department.
b. Traced: *UC-9*
c. Inputs: *Employee information.*
d. Processing: *The Cafeteria Ordering System shall contact the payroll department, which will await their approval to access the employee's payroll.*
e. Outputs: *Account page shall show user's payroll deduction details.*
f. Error Handling: *The system shall cancel the process if the user is no longer an employee.*

7.3 FR3: Subscribe for meal service subscription

a. Introduction/Functionality: The Cafeteria Ordering System gives users the option to subscribe to a monthly meal service plan of their choice.
b. Traced: *UC-11*
c. Inputs: *User's card information or successful payroll deduction.*
d. Processing: *The system shall give certain accessibilities to the user with the chosen subscription.*
e. Outputs: *Account page will show the details of their subscription and gives users extra perks depending on the type of subscription.*
f. Error Handling: *<any error conditions and how to handle these error conditions>*

7.4 FR4: View/modify/cancel meal service subscription

a. Introduction/Functionality: The account page shall display the user's active meal service subscription status as well as a button to cancel the subscription.
b. Traced : *UC-12, UC-13, UC-14*
c. Inputs: *User needs an active Cafeteria Ordering System account.*
d. Processing: *If the user edits their payment information, the account page shall display the new monthly payment details. If the user cancels, the system shall state that the subscription has been canceled.*
e. Outputs: *Account page shall have an updated subscription status based on modifications or cancellation.*
f. Error Handling: *If the user's subscription has ended due to card declining, the account page will notify the user.*

7.5 FR5: Choose to pay upon pickup/delivery

a. Introduction/Functionality: The payment screen will give an option to allow customers to pay at the cafeteria or upon delivery.
b. Traced: *UC-7*
c. Inputs: *An order must be placed.*
d. Processing: *The cafeteria staff or deliverer will receive a notification that the customer will pay upon pickup/delivery.*
e. Outputs: *The customer shall receive an order confirmation on their account.*
f. Error Handling: *The system shall notify the cafeteria staff if the customer cancels the order.*

7.6 FR6: Order meal from cafeteria

a. Introduction/Functionality: The system will display a button to order a meal from the cafeteria.
b. Traced: *UC-1*
c. Inputs: *User clicks the 'order' button*
d. Processing: *The system will retrieve the cafeteria menu and create a forum of items the user can add to their order.*
e. Outputs: *The user is presented with a forum which they can chose menu items and add them to their order.*
f. Error Handling: *The system will notify the user if it's unable to retrieve the latest menu or if the user is ordering outside of cafeteria business hours.*

7.7 FR7: Display Menu

a. Introduction/Functionality: The system will display a button to view the cafeteria menu.
b. Traced: *UC-2*
c. Inputs: *User clicks the 'menu' button*
d. Processing: *The system will retrieve the cafeteria menu.*
e. Outputs: *The system will display the cafeteria menu on the screen.*
f. Error Handling: *The system will notify the user if the menu is undergoing revisions or is temporarily unavailable.*

7.8 FR8: Order a custom meal from cafeteria

a. Introduction/Functionality: The system will display substitutions for various meals on the menu.
b. Traced: *UC-3*
c. Inputs: *User selects a menu item*
d. Processing: *System retrieves valid substitutions for the meal*
e. Outputs: *System displays menu item with a forum to customize its ingredients*
f. Error Handling: *The system will notify the user if a meal is not available for custom order*

7.9 FR9: Filter Ingredients

a. Introduction/Functionality: The system will display a variation of the menu in accordance to the filter set by the user.
b. Traced: *UC-4*
c. Inputs: *The system shows a forum of ingredients for the user to filter.*
d. Processing: *The system reloads the menu based on the filter.*
e. Outputs: *The system displays the altered menu.*
f. Error Handling: *The system will notify the user if the altered menu encountered an error while generating.*

7.10 FR10: Customize cooking instructions

a. Introduction/Functionality: The system presents the user with a textbox to enter additional details on how they'd like the meal prepared.
b. Traced: *UC-5*
c. Inputs: *The user enters cooking/preparation instructions in a text box provided by the system.*
d. Processing: *The system appends the instructions to the meal*
e. Outputs: *The meal contains a note with the user's instructions*
f. Error Handling: *No known error conditions*

7.11 FR11: Update Meal Subscription

a. Introduction/Functionality: The system presents the user with different program flows to accomplish creating, deleting or modifying their meal subscription.
b. Traced: *UC-13*
c. Inputs: *The user chooses what option they want to execute depending on their current need.*
d. Processing: *The system executes the appropriate operation depending on the user flow.*
e. Outputs: *The user's meal subscription is modified or updated.*
f. Error Handling: *User chooses to modify their meal subscription when it does not exist, therefore guides the user to create a new one.*

7.12 FR12: Delete Meal Subscription

a. Introduction/Functionality: The user is allowed to delete their meal subscription if they want to.
b. Traced: *UC-14*
c. Inputs: *The user clicks on the delete button when accessing their meal subscription.*
d. Processing: *The system deletes or sets to null the user's meal subscription.*
e. Outputs: *No existing user meal subscription.*
f. Error Handling: *No known error conditions.*

7.13 FR13: Deliver Meals

a. Introduction/Functionality: The system gives details to the cafeteria staff to perform meal delivery to a specified location.
b. Traced: *UC-20*
c. Inputs: *The user places an order as delivery*
d. Processing: *The system alerts the cafeteria or restaurant of delivery order*
e. Outputs: *The food deliverer delivers the user's order and user receives their food.*
f. Error Handling: *No known error conditions.*

7.14 FR14: Request feedback

a. Introduction/Functionality: The system allows the cafeteria staff to request feedback from the user regarding their service.
b. Traced: *UC-21*
c. Inputs: *The system gives the cafeteria staff the option to request feedback from the customer after their food order is achieved.*
d. Processing: *The system executes the feedback option to the cafeteria staff and sends it to the customer.*
e. Outputs: *The user is allowed to respond with feedback if they desire in which the cafeteria staff will receive it.*
f. Error Handling: *No known error conditions.*

7.15 FR15: Manage Menus

a. Introduction/Functionality: The cafeteria staff shall be able to manage any menus created. This includes viewing, deleting, creating, or modifying menus.
b. Traced: UC-15, UC-17
c. Inputs: *Information on menu edits that need to be done (Such as menus that need to be deleted, items to be added to a menu, etc.).*
d. Processing: The Cafeteria Ordering System receives the request to manage the menus and transfers the authenticated user to the page where they choose how to manage the menus for the organization.
e. Outputs: *Edited and managed menus for the organization*
f. Error Handling: *The user session has expired before management can confirm their authentication. Transfer the user back to the login page.*

7.16 FR16: Manage Delivery Availability

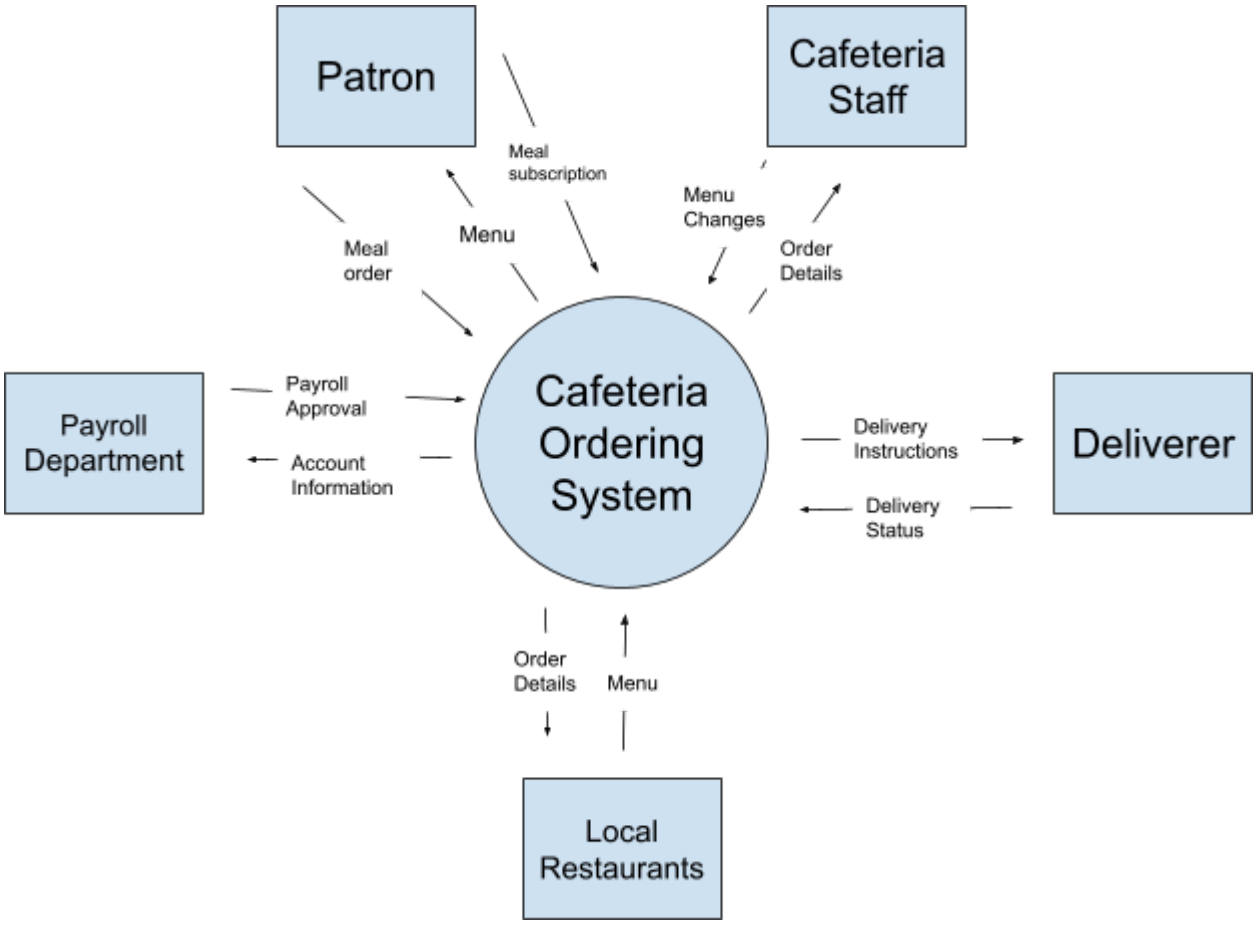
a. Introduction/Functionality: The system allows the cafeteria staff to mark a certain food option as “Not available for delivery,” making that specific meal to be not deliverable.
b. Traced: UC-22
c. Inputs: The cafeteria staff logs in to modify their menu for a certain food and marks not available for delivery option.
d. Processing: The system will change that value for that specific meal and relays that information to the customer.
e. Outputs: The system will display a food as “not deliverable” if a customer is interested in that particular food.
f. Error Handling: If a food quantity reaches 0 then the food should automatically be marked as not deliverable.

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

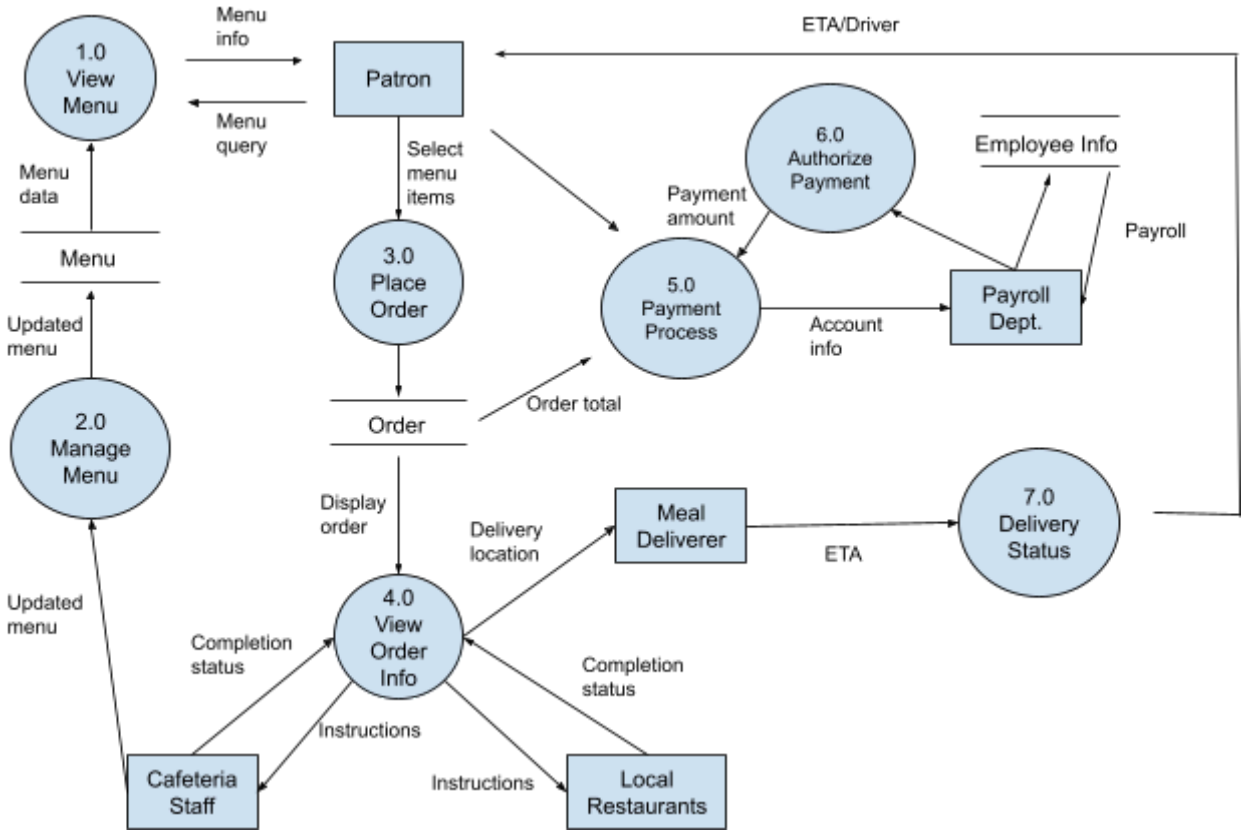
	Related FRs
UC-01	FR6
UC-02	FR7
UC-03	FR8
UC-04	FR9
UC-05	FR10
UC-06	FR1
UC-07	FR5
UC-08	FR5
UC-09	FR2
UC-10	FR11, FR12
UC-11	FR3
UC-12	FR4
UC-13	FR4
UC-14	FR4
UC-15	FR15
UC-16	FR15
UC-17	FR15
UC-18	FR15
UC-19	FR15
UC-20	FR16

8. Functional Modeling (DFDs)

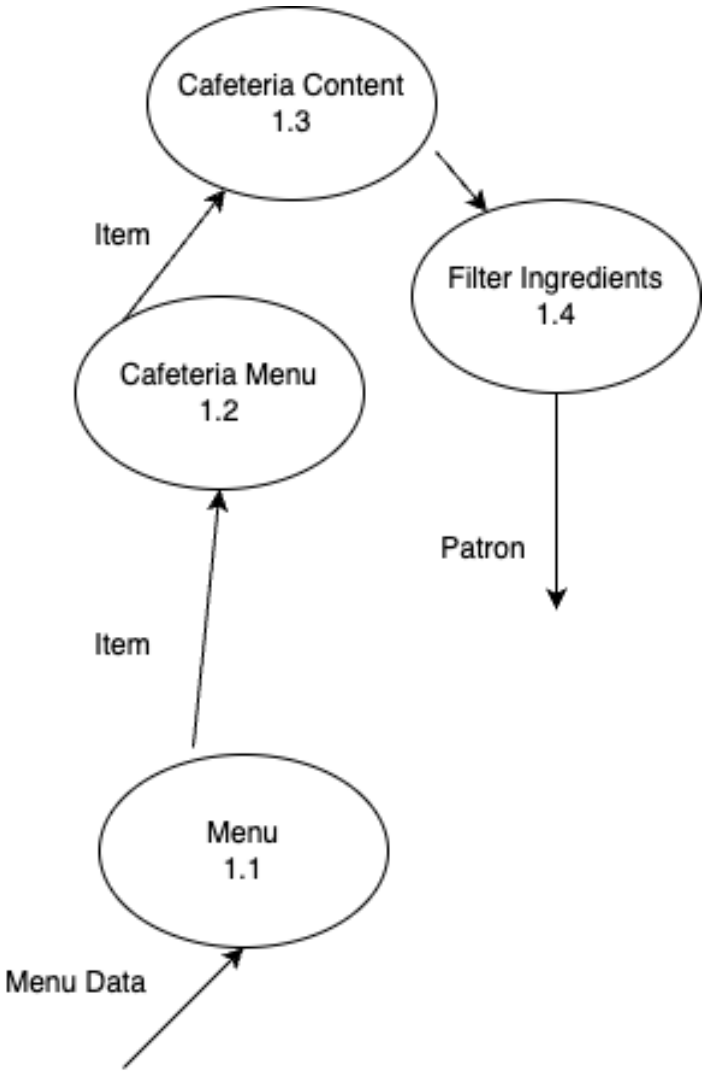
8.1 Context Diagram (level 0)

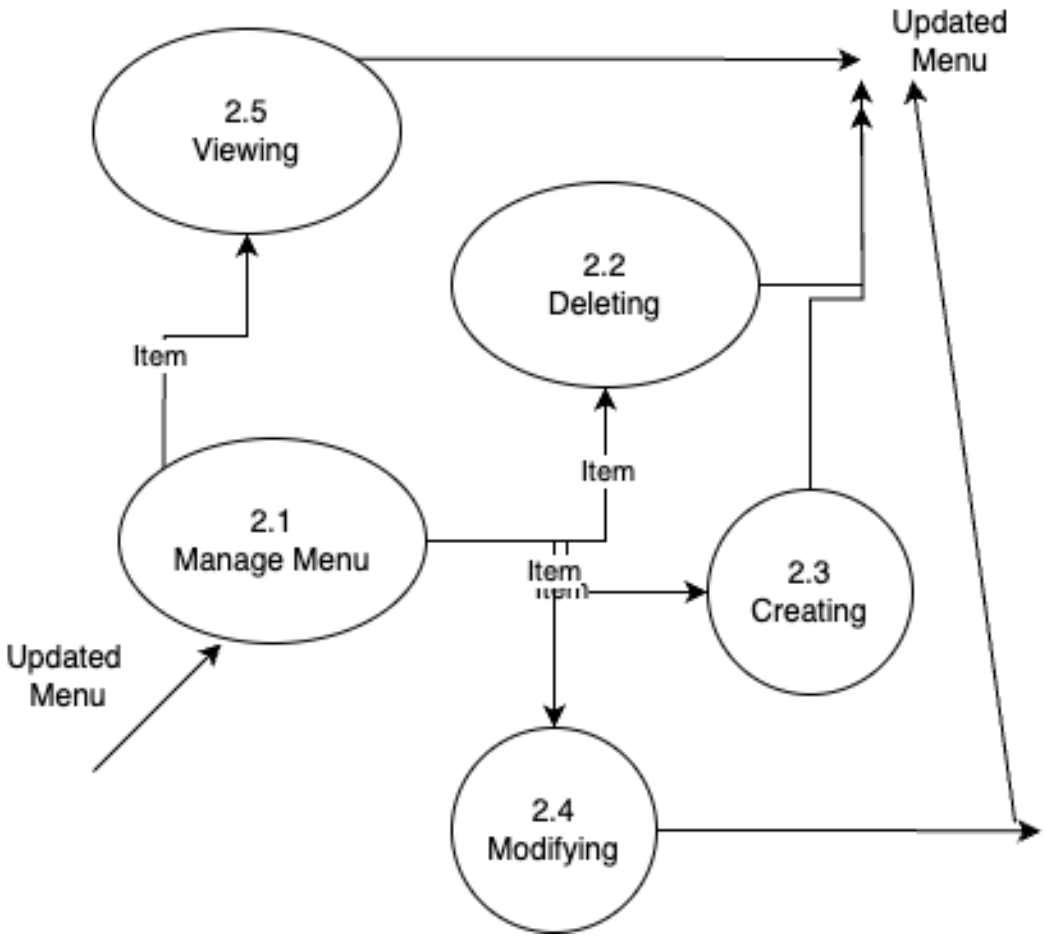


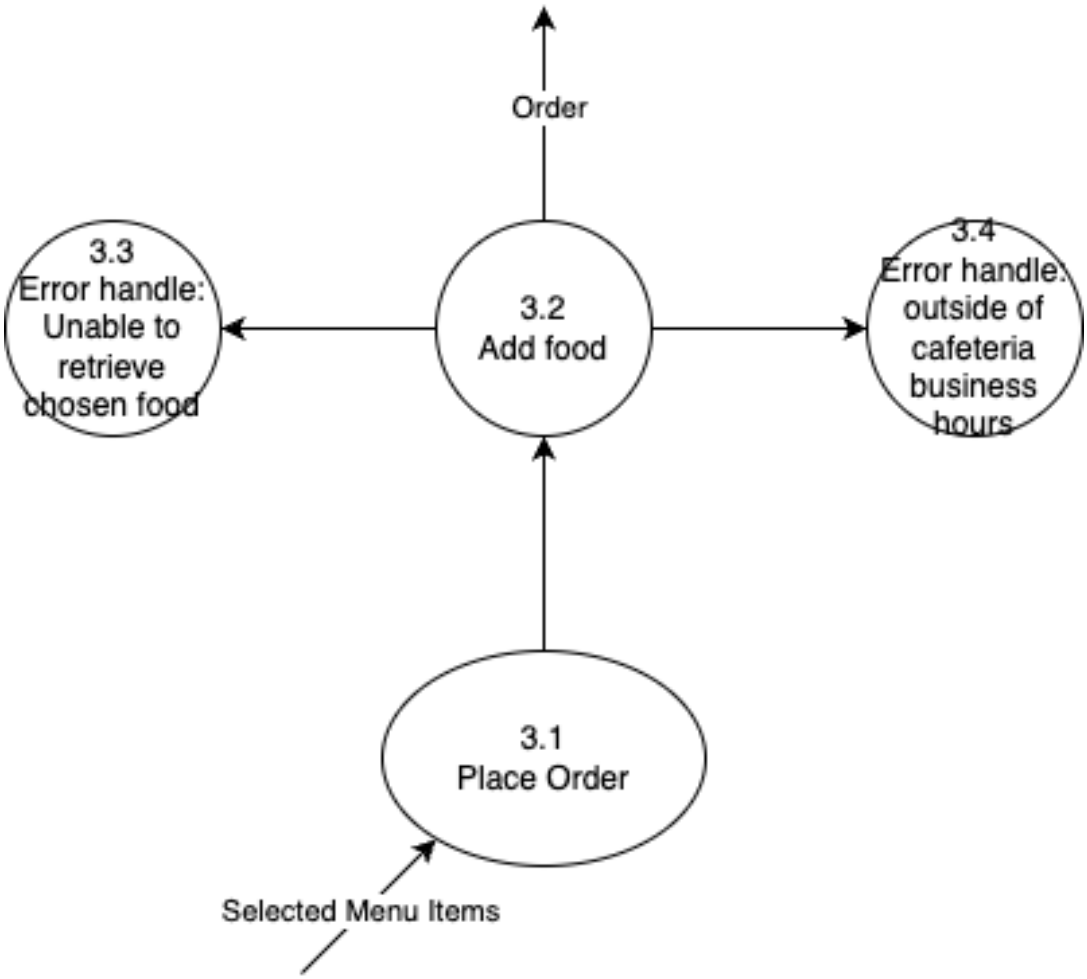
8.2 Level 1

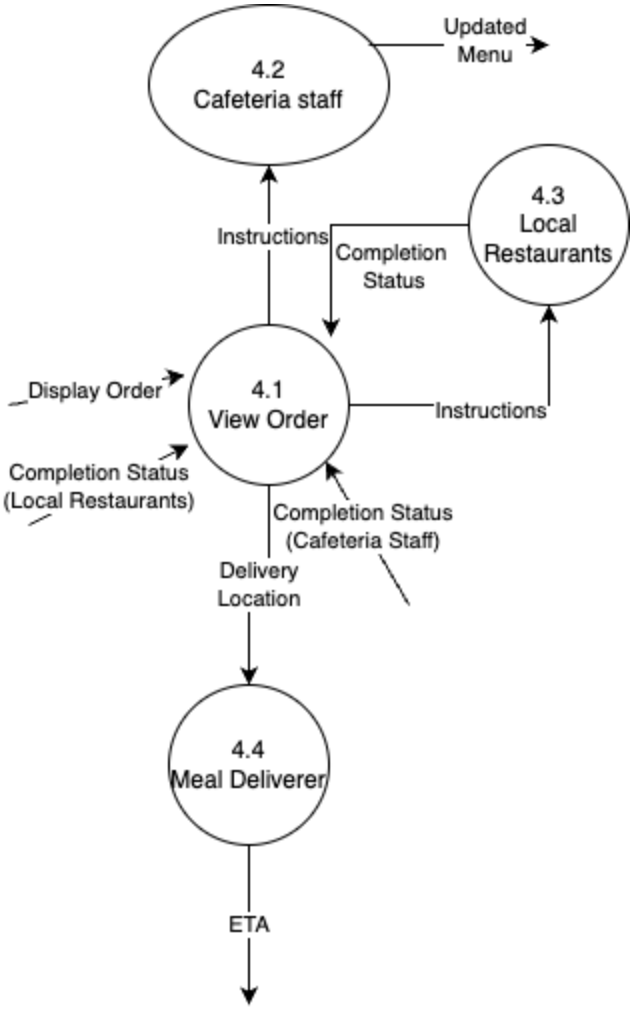

1.0
View
Menu
Menu
info
Menu
query
Patron
ETA/Driver
Menu
data
Menu
Select
menu
items
6.0
Authorize
Payment
Employee Info
Payment
amount
Payroll
3.0
Place
Order
5.0
Payment
Process
Payroll
Dept.
Account
info
Updated
menu
Order total
2.0
Manage
Menu
Order
Display
order
Delivery
location
Meal
Deliverer
7.0
Delivery
Status
ETA
Updated
menu
4.0
View
Order
Info
Completion
status
Completion
status
Instructions
Instructions
Cafeteria
Staff
Local
Restaurants

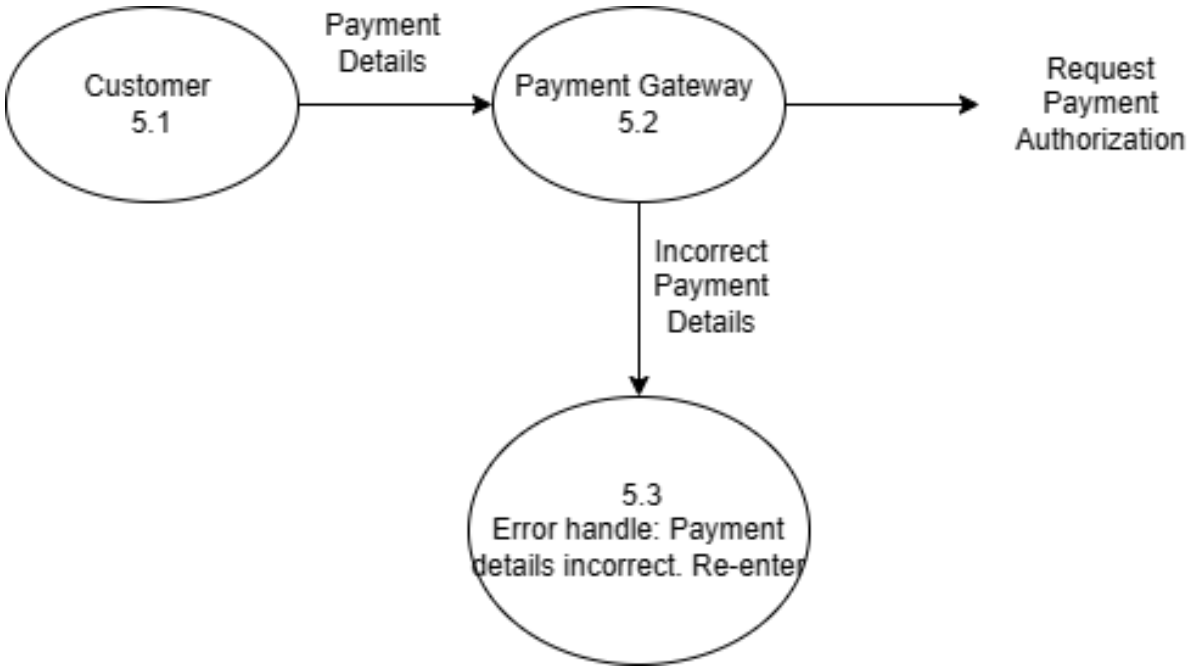
8.3 Level 2

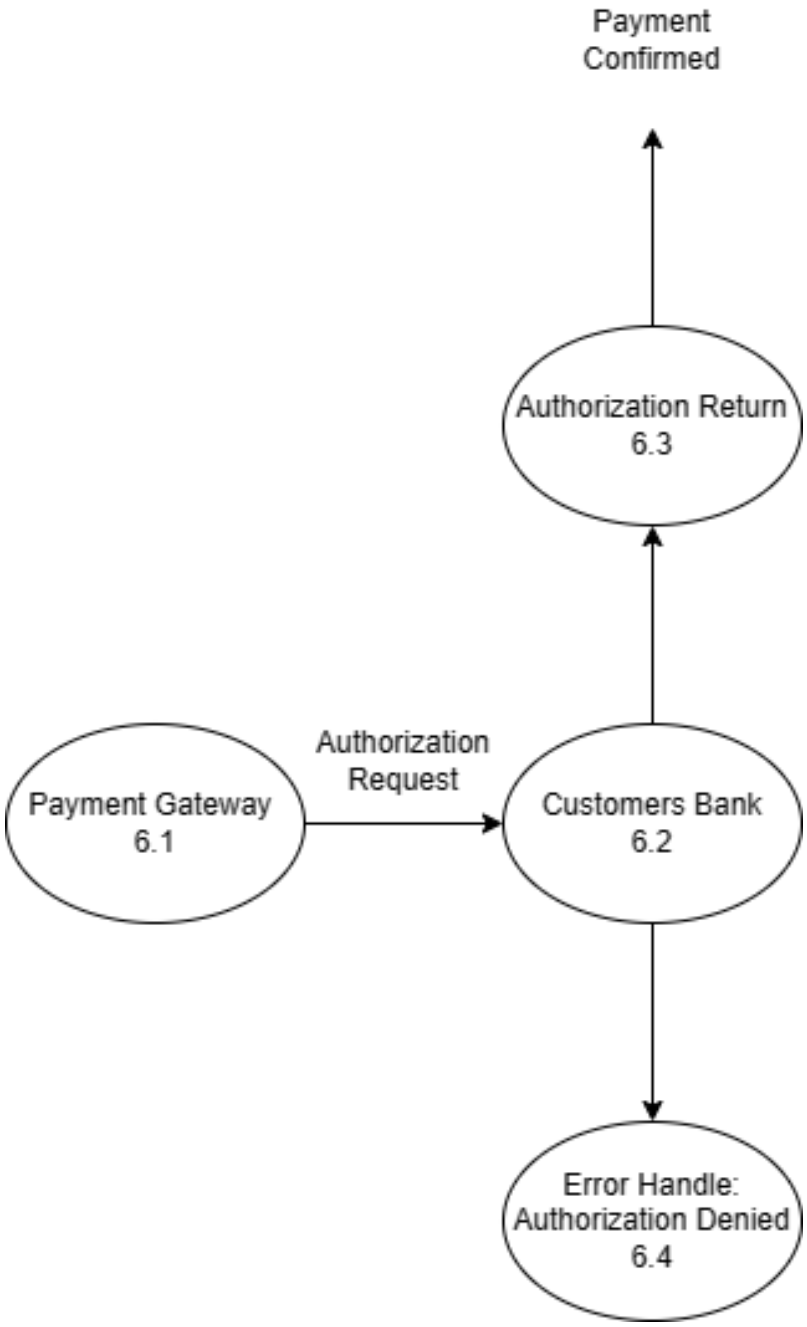

Cafeteria Content
1.3
Item
Filter Ingredients
1.4
Cafeteria Menu
1.2
Patron
Item
Menu
1.1
Menu Data

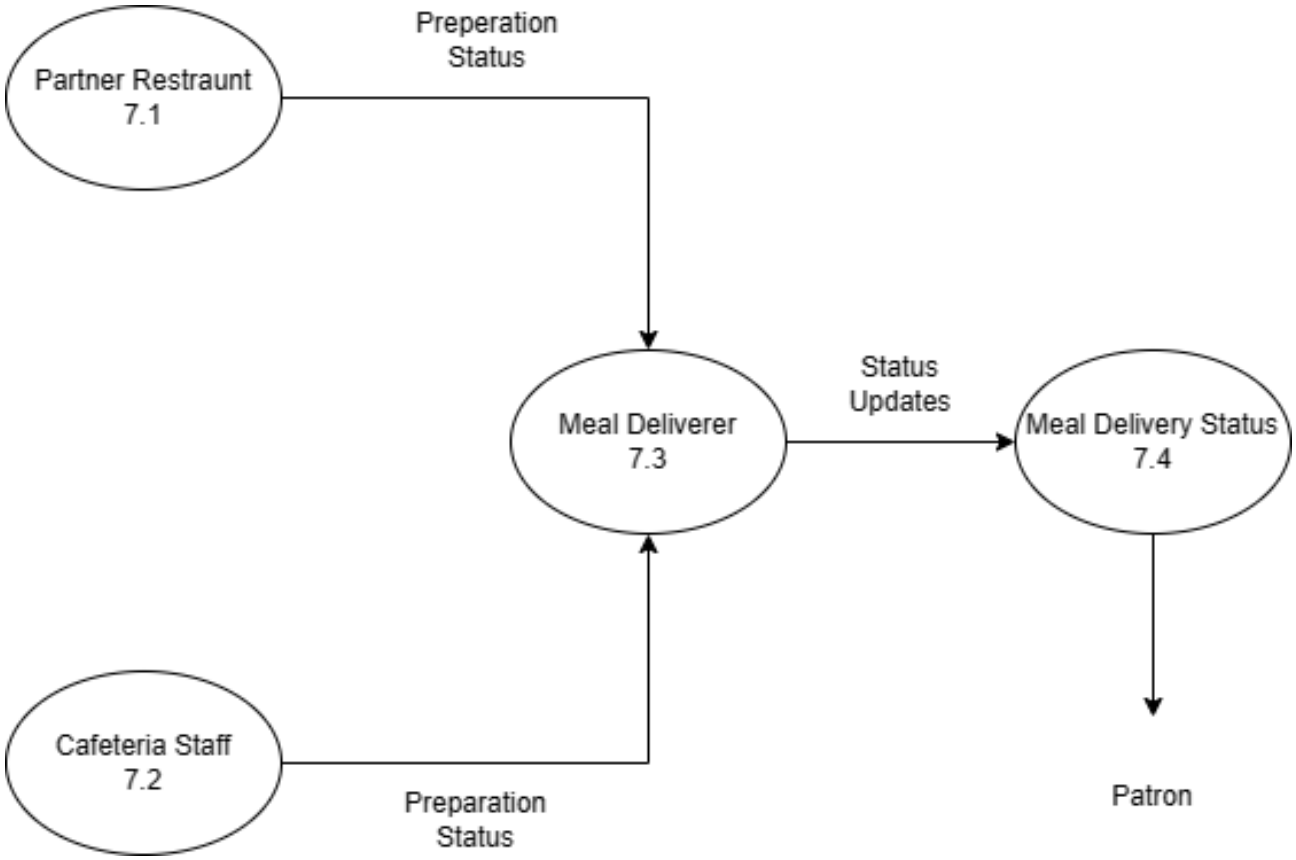

Updated
Menu
2.5
Viewing
2.2
Deleting
Item
Item
2.1
Manage Menu
Item
2.3
Creating
Updated
Menu
2.4
Modifying


Order
3.3
Error handle:
Unable to
retrieve
chosen food
3.2
Add food
3.4
Error handle:
outside of
cafeteria
business
hours
3.1
Place Order
Selected Menu Items


4.2
Cafeteria staff
Updated
Menu
4.3
Local
Restaurants
Instructions
Completion
Status
Display Order
4.1
View Order
Instructions
Completion Status
(Local Restaurants)
Completion Status
(Cafeteria Staff)
Delivery
Location
4.4
Meal Deliverer
ETA


Customer
5.1
Payment
Details
Payment Gateway
5.2
Request
Payment
Authorization
Incorrect
Payment
Details
5.3
Error handle: Payment
details incorrect. Re-enter


Payment
Confirmed
Authorization Return
6.3
Payment Gateway
6.1
Authorization
Request
Customers Bank
6.2
Error Handle:
Authorization Denied
6.4


Partner Restraunt
7.1
Preperation
Status
Meal Deliverer
7.3
Status
Updates
Meal Delivery Status
7.4
Cafeteria Staff
7.2
Preparation
Status
Patron

……..

TABLE 3: Traceability Matrix (FRs and DFD Processes)

	All Related Processes in DFD
FR1	5.0
FR2	5.0
FR3	3.0, 5.0, 6.0
FR4	3.0
FR5	3.0, 7.0
FR6	3.0, 4.0
FR7	1.0, 3.0
FR8	3.0
FR9	1.0, 3.0
FR10	3.0
FR11	1.0, 5.0
FR12	1.0
FR13	4.0, 7.0
FR14	1.0
FR15	2.0
FR16	2.0

9. Class Analysis Modeling

9.1 Initial Class Diagram (ICs)

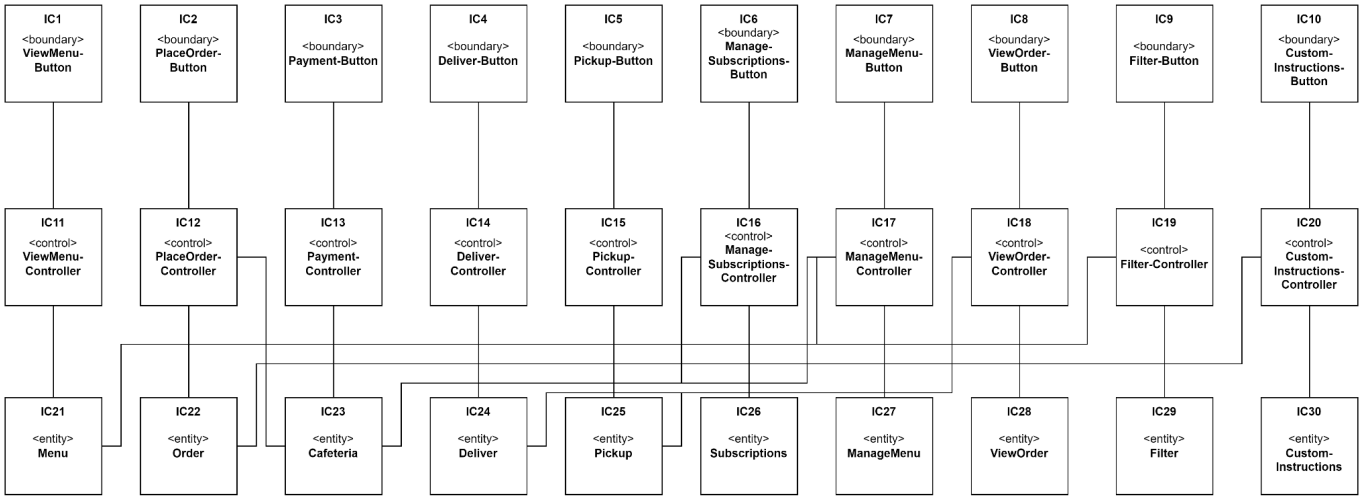


TABLE 4: Traceability Matrix (FRs and Initial Classes)

	Related ICs
FR1	IC3, IC13
FR2	IC3, IC13, IC6, IC16, IC26
FR3	IC6, IC16, IC26
FR4	IC6, IC16, IC26
FR5	IC3, IC4, IC13, IC5, IC15
FR6	IC7, IC8, IC17, IC18, IC22, IC26
FR7	IC1, IC11, IC21
FR8	IC7, IC17, IC27
FR9	IC9, IC19, IC21, IC29
FR10	IC10, IC21, IC27, IC30
FR11	IC6, IC16, IC26
FR12	IC6, IC16, IC26
FR13	IC4, IC14, IC24, IC28
FR14	IC10, IC22,
FR15	IC1, IC2, IC 11, IC12, IC27
FR16	IC2, IC4, IC12, IC14, IC24

9.2 Modified Class Diagram (MCs)

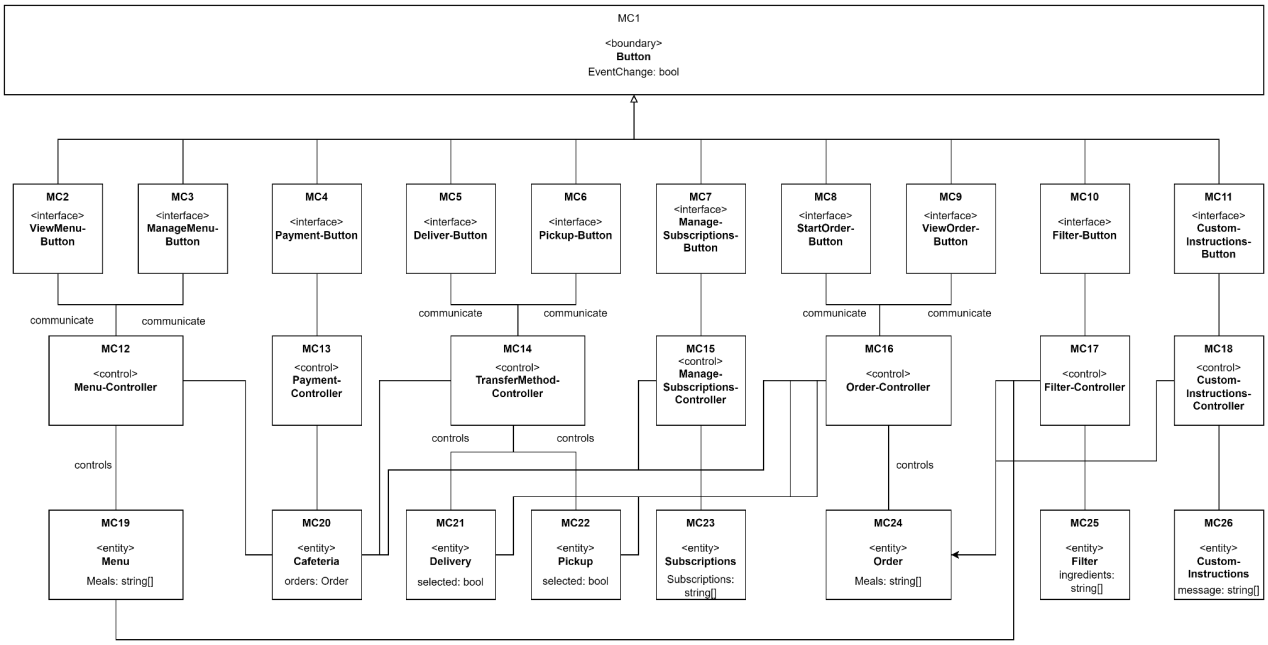


TABLE 5: Traceability Matrix (Initial Classes (IC) and modified Classes (MC))

	Related MCs
IC1	MC1, MC2, MC12
IC2	MC1, MC3, MC12
IC3	MC1, MC4, MC13
IC4	MC1, MC5, MC14
IC5	MC1, MC6, MC14
IC6	MC1, MC7, MC15
IC7	MC1, MC8, MC16
IC8	MC1, MC9, MC16
IC9	MC1, MC9, MC19
IC10	MC1, MC18
IC11	MC1, MC2, MC12
IC12	MC1, MC3, MC12
IC13	MC1, MC4, MC13
IC14	MC1, MC5, MC14, MC21
IC15	MC1, MC6, MC14, MC22
IC16	MC1, MC7, MC15
IC17	MC8, MC16
IC18	MC9, MC16
IC19	MC10, MC17

IC20	MC11, MC18
IC21	MC2, MC3, MC12, MC19
IC22	MC4, MC 13, MC20
IC23	MC5, MC14, MC21
IC24	MC6, MC14, MC22,
IC25	MC7, MC15, MC23,
IC26	MC8, MC9, MC16, MC24
IC27	MC10, MC17, MC25
IC28	MC11, MC18, MC26
IC29	MC25
IC30	MC26