
Software Requirements Specification

for

Cafeteria Ordering System

Prepared by Thanh Nguyen

Eric Martinez

Loc Nguyen

Ryan Pham

12/11/2024

Table of the Content

1. Introduction	1
1.1 Purpose	1
1.2 Scope	1
1.3 Definitions, acronyms, and abbreviations	2
Definitions:	2
Acronyms and abbreviations:	2
1.4 References	3
1.5 Overview	3
2. Overall Description	4
2.1 Product Perspective	4
2.2 Product Features (FEs)	6
2.3 User Classes and Characteristics	9
2.4 Operating Environment	12
2.4.1 Hardware Requirements	12
2.4.2 Software Requirements	12
2.4.3 Network Environment	13
2.4.4 User Environment	13
2.4.5 External Dependencies	13
2.4.6 Security and Compliance	13
2.5 Design and Implementation Constraints	14
2.5.1 Corporate/Regulatory Policies	14
2.5.2 Hardware limitations	14
2.5.3 Security Considerations	14
2.5.4 Language Requirements	15
2.5.5 Testing and Quality Assurance	15
2.6 User Documentation	15
2.6.1 User Manuals	15
2.7 Assumptions and Dependencies	17
3. Use-case diagram with Use-case descriptions (UCs)	18
3.1 Use-case Diagram	18
3.2 Use-case Description(s)	19
4. External Interface Requirements	55
4.1 User Interfaces	55
4.2 Hardware Interfaces	60
4.3 Software Interfaces	61
4.4 Communications Interfaces	63

5. Other Nonfunctional Requirements	65
5.1 Performance Requirements	65
5.2 Safety Requirements	65
5.3 Security Requirements	65
5.4 Software Quality Attributes	66
6. Other Requirements	67
7. Functional Requirements (FRs)	68
7.1 FE-1: Order meals from the cafeteria menu for pickup or delivery.	68
7.1.1 FR-1: System displays menu of available items	68
7.1.2 FR-2: Add food item from menu	68
7.1.3 FR-3: Remove food item from active order	69
7.1.4 FR-4: Select order pickup option	69
7.1.5 FR-5: Select order delivery option	70
7.1.6 FR-6: Select delivery time	70
7.1.7 FR-7: Patron chooses delivery location	70
7.2 FE-2: Order meals from local restaurants for delivery.	71
7.2.1 FR-8: Allowing Patron to specify a menu date	71
7.2.2 FR-9: Allowing Patron the Patron to choose to order for pickup or delivery.	72
7.2.3 FR-10: Managing local-restaurant orders	72
7.3 FE-3: Create, view, modify, and delete meal service subscriptions.	73
7.3.1 FR-11: Create Meal Service Subscription	73
7.3.2 FR-12: Subscription Length Selection	74
7.3.3 FR-13: Delivery Frequency Options	74
7.3.4 FR-14: Input Field Validation	75
7.3.5 FR-15: Database Management	75
7.3.6 FR-16: Payment Processing	75
7.3.7 FR-17: Confirmation Output	76
7.3.8 FR-18: View Meal Service Subscription	76
7.3.9 FR-19: Subscription Summary Display	77
7.3.10 FR-20: Delete Meal Service Subscription	77
7.3.11 FR-21: Validate Subscription Eligibility for Deletion	78
7.3.12 FR-22: Update Subscription Status to Void	78
7.3.13 FR-23: Generate Void Confirmation Notification	79
7.3.14 FR-24: Modify Meal Service Subscription	79
7.3.15 FR-25: Modify Subscription Length	80
7.3.16 FR-26: Modify Delivery Frequency	81
7.3.17 FR-27: Address and Contact Information Update	81
7.3.18 FR-28: Modify Meal Plan Selection	81
7.3.19 FR-29: Validate and Confirm Modification Inputs	82
7.3.20 FR-30: Database Update for Modified Subscription	82

7.3.21 FR-31: Payment Adjustment Processing	83
7.3.22 FR-32: Modification Confirmation Output	83
7.4 FE-4: Register various meal payment methods.	84
7.4.1 FR-33: Registering for payment by payroll deduction	84
7.4.2 FR-34: Registering for payment by credit card	84
7.4.3 FR-35: Registering for payment by debit card	85
7.4.4 FR-36: Registering for payment by Digital Wallet	85
7.5 FE-5: Manage cafeteria menus (create, view, modify, delete).	86
7.5.1 FR-37: Create cafeteria menus	86
7.5.2 FR-38: View cafeteria menu	87
7.5.3 FR-39: Modify cafeteria menus	87
7.5.4 FR-40: Delete cafeteria menus	88
7.6 FE-6: Order custom meals that are not on the menu.	89
7.6.1 FR-41: Order custom meals not on the menu	89
7.7 FE-7: Request and collect feedback from customers.	90
7.7.1 FR-42: Request and collect customer feedback	90
7.8 FE-8: Contact support	91
7.8.1 FR-43: Customer Support	91
7.8.2 FR-44: Support Request Form	92
7.9 FE-9: Track order status.	93
7.9.1 FR-45: Track order status	93
7.10 FE-10: Manage favorite items.	94
7.10.1 FR-46: Save Favorite items.	94
7.10.2 FR-47: View Favorite items.	94
7.10.3 FR-48: Remove Favorite items.	95
TABLE 2: Traceability Matrix (Use-cases & FRs).	96
8. Functional Modeling (DFDs)	97
8.1 Context Diagram (Level 0)	97
8.2 Level 1	98
8.2.1 Ordering System, Order Tracking	98
8.2.2 Account Management	99
8.2.3 Cafeteria Management	99
8.2.4 Feedback System	100
8.2.5 Contact Support	100
8.3 Level 2	101
8.3.1 Process 1.0 - Placing menu order	101
8.3.2 Process 2.0 - Fulfill menu order	102
8.3.3 Process 3.0 - View menu	102
8.3.4 Process 4.0 - Manage payment method	103
8.3.5 Process 5.0 - Manage subscriptions	104
8.3.6 Process 6.0 - Track order status	104

8.3.7 Process 7.0 - Request feedback	105
8.3.8 Process 8.0 - Process feedback	105
8.3.9 Process 9.0 - Manage favorite items	106
8.3.10 Process 10.0 - Manage cafeteria menus	107
8.3.11 Process 11.0 - Manage cafeteria inventory	108
8.3.12 Process 12.0 - Update cafeteria inventory	108
8.3.13 Process 13.0 - Contact support	109
8.3.14 Process 14.0 - Problem with order	110
8.3.15 Process 15.0 - Process payment	110
8.4 Level 3	111
8.4.1 Process 1.1 - Placing cafeteria order	111
8.4.2 Process 10.1 - Create cafeteria menu	112
TABLE 3: Traceability Matrix (FRs and DFD Processes)	113
9. Class Analysis Modeling	115
9.1 Initial Class Diagram (ICs)	115
9.1.1 Order a meal	115
9.1.2 Manage meal service subscriptions	116
9.1.3 Register payment method	116
9.1.4 Manage cafeteria menus	117
9.1.5 Order custom meals not on the menu	117
9.1.6 Request and collect feedback	118
9.1.7 Contact support	118
9.1.8 Track meal order status	119
9.1.9 Manage favorite items	120
TABLE 4: Traceability Matrix (FRs and Initial Classes)	121
9.2 Modified Class Diagram (MCs)	123
9.2.1 Order a meal	123
9.2.2 Manage meal service subscriptions	123
9.2.3 Register payment method	124
9.2.4 Manage cafeteria menus	124
9.2.5 Order custom meals not on the menu	125
9.2.6 Request and collect feedback	125
9.2.7 Contact support	126
9.2.8 Track meal order status	126
9.2.9 Manage favorite items	127
TABLE 5: Traceability Matrix (Initial Classes (IC) and modified Classes (MC))	128

1. Introduction

1.1 Purpose

Software Requirements Specification (SRS) aims to define the functional and non-functional requirements for the Cafeteria Ordering System (COS). The SRS describes the intended software behaviors to guide the development, testing, and planning activities to achieve the business objectives.

Intended Audience:

- product owners
- developers
- testers
- project managers
- clients
- stakeholders

1.2 Scope

The COS is a web-based application that will allow employees to order lunch meals from the cafeteria and participating local restaurants.

COS will enable employees to pick up their meals or have them delivered to a designated location. It will provide employees the option to customize their food or request certain items that are not on the menu. It will also offer an option for meal service subscriptions and support various preferred payment methods.

COS will limit access to only employees on the Process Impact campus in Clackamas, Oregon. It will also not allow the ability to deliver any special requested items. This is because some food

items from the cafeteria can't be delivered, so a smaller subset of the menu will be shown for menu items that are eligible for delivery.

Benefits, objectives, and goals are available in the Cafeteria Ordering System Vision and Scope document.

1.3 Definitions, acronyms, and abbreviations

Definitions:

1. **SRS (Software Requirements Specification):** This document explains exactly what the software needs to do. It will list all the features and requirements so that everyone involved knows what the final product should look like and how it should work.
2. **V&S (Vision and Scope):** This document will give an overview view of the project and explain why the software is being created, as well as identify what problems the software will solve, who will use it, and what the software will and won't do.

Acronyms and abbreviations:

1. **SC:** Success Criteria
2. **BO:** Business Objectives
3. **RI:** Risk Items
4. **AS:** Assumptions
5. **DE:** Dependencies
6. **FE:** Features
7. **LI:** Limitations
8. **COS:** Cafeteria Ordering System
9. **SLA:** Service Level Agreement
10. **PCI DDS:** Payment Card Industry Data Security Standard
11. **OE:** Operating Environment
12. **HDR:** Hardware Requirements
13. **SWR:** Software Requirements
14. **NTE:** Network Environment

- 15. **HI:** Hardware Interfaces
- 16. **SI:** Software Interfaces
- 17. **SEC:** Security Requirements
- 18. **SAF:** Safety Requirements
- 19. **PER:** Performance Requirements
- 20. **SQA:** Software Quality Attributes
- 21. **CI:** Communications Interfaces
- 22. **CO:** Constraints
- 23. **OTH:** Other Requirements
- 24. **UMAN:** User Manuals
- 25. **USR:** User Environment
- 26. **EXT:** External Dependencies
- 27. **SCC:** Security and Compliance

1.4 References

- 1. "About Us." *PCI Security Standards*, www.pcisecuritystandards.org/. Accessed 6 Oct. 2024.
- 2. Wiegers, Karl, and Joy Beatty. *Software Requirements*. 3rd ed., *Microsoft Press*, 2013. pp. 575-595.
- 3. "Video 23 - The Software Requirements Specification." *YouTube*, uploaded by Enfocus Solutions, 14 Feb. 2012, youtu.be/_XTQjKhh6hQ?si=8GTkLTI_6I3iRrNL.

1.5 Overview

The software requirement specifications document will be organized into the following sections; overall description, use-case diagram, external interface requirements, other nonfunctional requirements, other requirements. The second section, overall description will detail everything needed to understand what product is being built. This includes the product features, user classes, design methods, and assumptions. The third section, use-case diagram with use case description, will contain the various interactions in the system. Section four, external interface requirements, will elaborate on the GUI, tools, libraries, licenses, and communications interactions by the product.

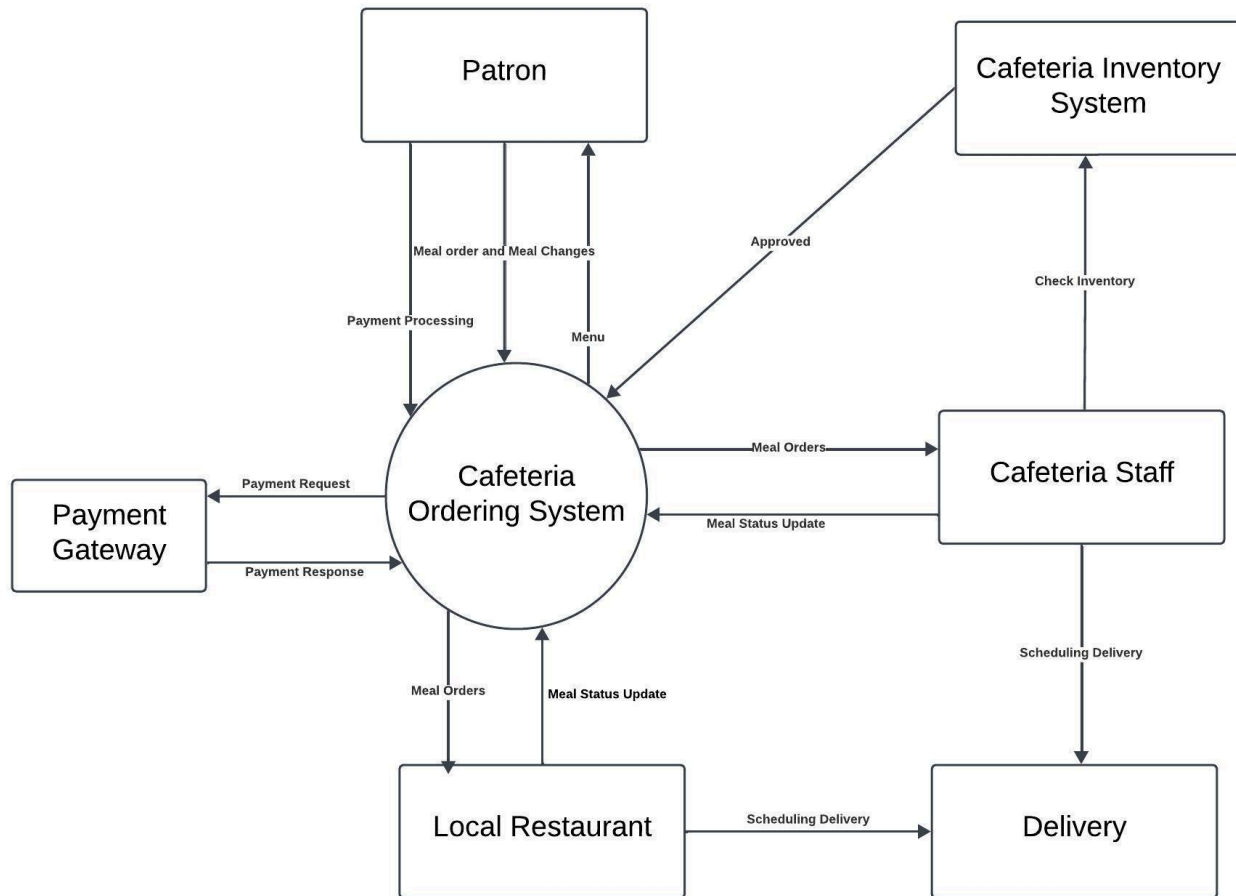
2. Overall Description

2.1 Product Perspective

Cafeteria Ordering System (COS) is a self-contained software designed for Process Impact employees in Oregon. It helps reduce the 90 minutes typically spent on lunch by allowing online meal orders from the company cafeteria or local restaurants, with delivery to the campus.

The following are the main features that are included in COS:

- Time-saving: COS cuts down on the time employees spend ordering and picking up meals.
- Real-time inventory: COS interacts with the Cafeteria Inventory system to track the available items and ensure that employees have better chances of getting their desired meal selections.
- Integration with local restaurants: Employees gain access to a broader range of food choices.
- Secure payment processing: The system is integrated with a Payment Gateway to securely process payments and support multiple payment methods.

Figure 2.1 Context Diagram for Cafeteria Ordering System

2.2 Product Features (FEs)

The Cafeteria Ordering System includes various major features that allow users to easily order meals and manage their subscriptions. These features are grouped into the following categories:

2.2.1 Ordering System

2.2.1.1 Description

Verified patrons from the institution can order meals from the cafeteria using the COS for pickup or delivery and can keep track of their orders by being able to check the status of their orders for updates. Priority = High.

- FE-1: Order meals from the cafeteria menu for pickup or delivery.
- FE-2: Order meals from local restaurants for delivery.
- FE-9: Track order status

2.2.2 Subscription Management:

2.2.2.1 Description

Users are able to manage their meal plans by being able to set up, change, or cancel meal subscriptions. Priority = Medium.

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

2.2.3 Payment Options

2.2.3.1 Description

There are multiple payment methods available for patrons to pay for their meals. Users will be able to add and manage multiple payment methods to pay for their meals. Priority = high.

- FE-4: Register various meal payment methods.

2.2.4 Menu and Custom orders:

2.2.4.1 Description

The COS will allow easy access to create, edit, and remove items from the cafeteria menu. Patrons will be able to customize their orders and save their favorite items to reorder in the future. Priority = Medium

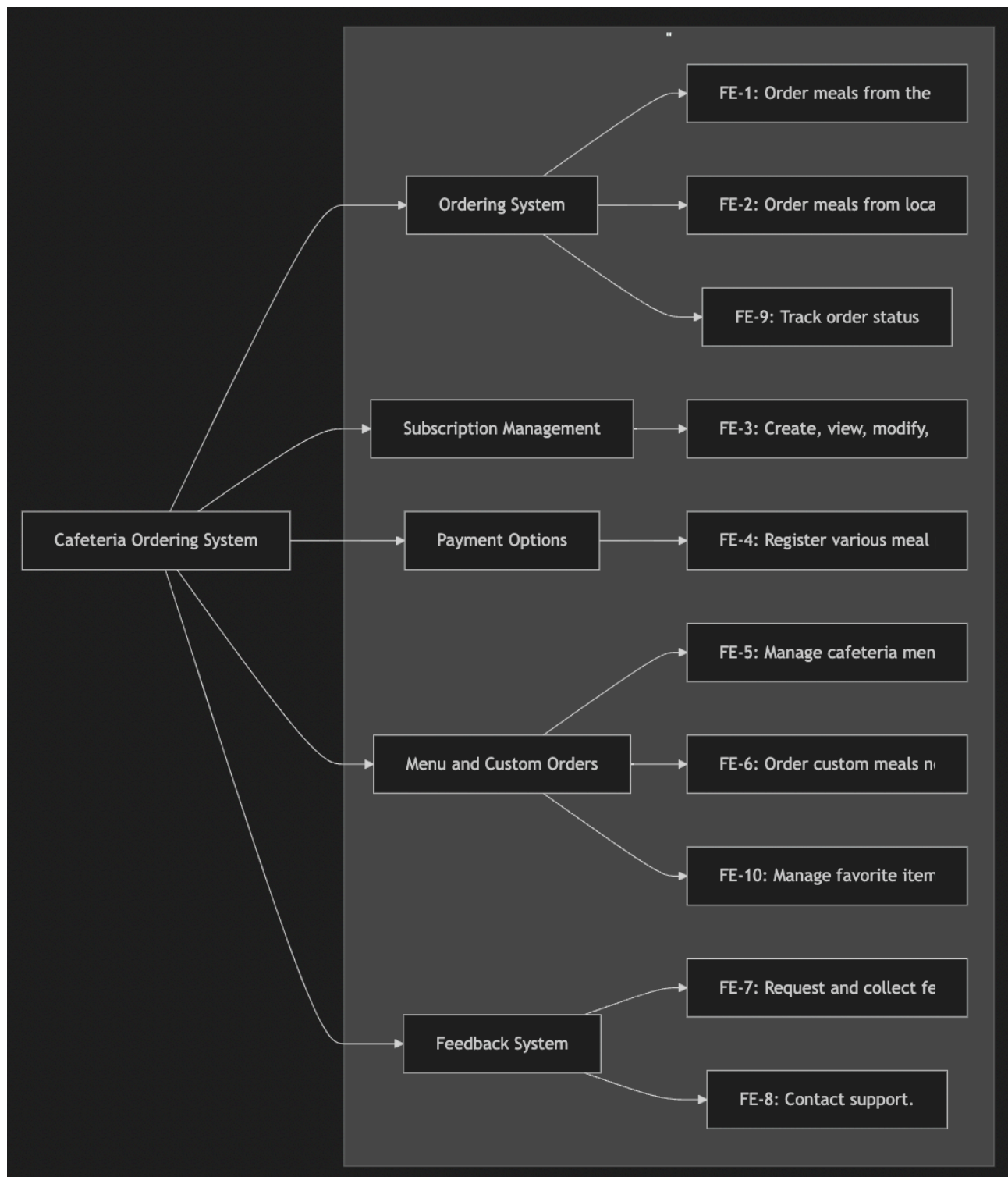
- FE-5: Manage cafeteria menus (create, view, modify, delete).
- FE-6: Order custom meals not on the menu.
- FE-10: Manage favorite items.

2.2.5 Feedback System:

2.2.5.1 Description

Patrons will be able to provide feedback and rate their meals and will be able to contact support for help if their order encounters any issues

- FE-7: Request and collect feedback from customers.
- FE-8: Contact support.

Figure 2-2: Diagram for major features the product contains.

2.3 User Classes and Characteristics

Table 2-1: User classes for the Cafeteria Ordering System

Name	Number	Description
Patron (favored)	All Process Impact employees located at the main campus in Clackamas, Oregon	Patrons are the target audience of the ordering system. Patrons are working professionals and have basic internet and computer knowledge. Each patron will use the system to peruse the menu, schedule food orders, register payment options, and manage meal service subscriptions. Patrons will receive their order by choosing the pickup or delivery options. They will select food options by choosing from the cafeteria menu or from a participating local restaurant menu. Patrons will use the system to create, view, modify, or delete their meal service subscriptions. Patrons need to have the option to register for meal payment options. A patron will use the system 1-5 times a week.

Name	Number	Description
Cafeteria Staff (favored)	20	Cafeteria staff will use the system to manage the order queue. Staff members have experience using traditional point-of-sale systems for tendering transactions. A staff member will fulfill orders by preparing, packing, staging food for pick up or delivery to a designated location. Staff members will need to search orders in the system to help locate details of a customer's order and the pickup/delivery status of the order. Staff members will use the system every time there is an order to fulfill. It could range from 10 to the number of Process Impact employees.
Cafeteria Manager	1-3	The cafeteria manager will use the system to perform administrative tasks relating to maintaining cafeteria menu options, ensuring a stocked inventory, auditing SLAs on order fulfillment, and collecting customer feedback. Cafeteria managers have proficient skills in operating computers, customer service satisfaction, and restaurant

Name	Number	Description
		process operations. The cafeteria manager will use the system on a daily basis.
Local restaurant manager	1-3	The local restaurant manager will not directly use COS. The manager will use their own online order system to manage (i.e., search, view, update) orders received from COS. They will see the details of the COS order, such as food item, quantity, contact information, pick up/delivery date and time.
Local restaurant staff	1-10	The local restaurant staff will not directly use COS. The staff will use their own online order system to view and fulfill orders received from COS. They will see the details of the COS order, such as food item, quantity, contact information, pick up/delivery date and time.

2.4 Operating Environment

2.4.1 Hardware Requirements

HDR-01: The COS shall operate on a server with the minimum specifications of Dual-core CPU, 8GB RAM, and 100GB SSD storage.

HDR-02: The COS shall be accessible from any client device with internet access, including smartphones, tablets, and computers, as a minimum requirement; and from client devices with a minimum of 4 GB RAM and modern web browsers as a recommended requirement.

2.4.2 Software Requirements

SWR-01: The COS shall operate on servers running Ubuntu 20.04 or higher, or CentOS 7 or higher.

SWR-02: The COS shall be accessible from client devices running Windows 10 or higher, macOS Mojave or higher, and the latest stable versions of Linux distributions.

SWR-03: The COS shall operate on web servers using either Nginx or Apache.

SWR-04: The COS shall utilize database management systems PostgreSQL or MySQL, specifically the latest stable versions.

SWR-05: The COS shall be developed using backend frameworks Node.js, Ruby on Rails, Python (Django/Flask) or similar.

SWR-06: The COS shall be developed using frontend technologies React.js, Angular, or similar.

2.4.3 Network Environment

NTE-01: The COS shall require a minimum internet connectivity of 1 Mbps download speed for the user.

NTE-02: The COS shall require load balancing for handling high traffic loads.

NTE-03: The COS shall permit user access from the corporate intranet.

2.4.4 User Environment

USR-01: The COS shall operate correctly with the following web browsers: Windows Internet Explorer versions 7, 8, and 9; Firefox versions 12 through 26; Google Chrome (all versions); and Apple Safari versions 4.0 through 8.0.

2.4.5 External Dependencies

EXT-01: The COS shall integrate with payment processing services such as Stripe, PayPal, or similar platforms.

EXT-02: The COS shall leverage map services to allow location tracking

2.4.6 Security and Compliance

SCC-01: The COS shall adhere to web security protocols.

SCC-02: The COS shall adhere to data compliance standards, including PCI DSS for payment processing and GDPR for user data protection.

2.5 Design and Implementation Constraints

2.5.1 Corporate/Regulatory Policies

CO-1: The Cafeteria Ordering System must be designed to follow all legal rules and regulations, which may involve:

- User Data Protection
- Payment Security Standards
- Accessibility Guidelines
- Allergen Information Regulations

2.5.2 Hardware limitations

CO-2: The application software should be able to work on a variety of mobile devices without the concern for compatibility regarding the phone model and operating system.

- Timing Requirements
 - The system should provide order status information
 - The application on both web and mobile devices must be in sync and responsive
- Memory Requirements
 - The app should be designed to work well on phones and tablets that do not have a lot of memory
 - It should be able to run smoothly on older or less powerful devices

2.5.3 Security Considerations

CO-3: The goal of these considerations is to secure the safety and privacy of user information. To achieve this, the system's design should incorporate:

- User Authentication
- Data Encryption
- Secure Payment Processing

2.5.4 Language Requirements

CO-4: The application should support multiple languages for accessibility, so all users can navigate the app and place orders easily.

2.5.5 Testing and Quality Assurance

CO-5: Develop a process for feedback collection to make sure that the app satisfies user needs and function correctly across all the different devices. Some techniques for user feedback can include but are not limited to:

- User Reviews
- User Ratings
- Surveys and Questionnaires
- App Feedback Feature

2.6 User Documentation

2.6.1 User Manuals

UMAN-1 : For Patrons (Employees)

- How to access the system
- Instructions for placing and managing orders (pickup and delivery).
- Instructions for managing subscriptions (creating, viewing, modifying, and deleting).
- Instructions for managing payment options.
- **Delivery Format:**
 - PDF: Available for download and print
 - Web-based: Accessible directly within the COS platform.

UMAN-2 : For Cafeteria Staff

- How to access and manage orders.
- Instructions for updating and managing the cafeteria inventory system.

- Guide to scheduling or managing delivery operations.
- **Delivery Format:**
 - PDF: Available for download and print
 - Web-based: Accessible directly within the COS platform.

UMAN-3 : For Local Restaurant Staff

- Instructions for managing orders received from COS.
- Guide to scheduling or managing delivery operations.
- **Delivery Format:**
 - PDF: Available for download and print
 - Web-based: Accessible directly within the COS platform.

UMAN-4 : Online Help

It will include searchable topics, such as:

- Manage Your Account.
- Troubleshoot order issues.
- Manage Subscriptions.
- Security Settings.
- Contact Customer Service.
- **Delivery Format:**
 - Web-based: Integrated within the COS user interface for real-time and contextual help.

UMAN-5 : Tutorial Videos

- How to place an order from the cafeteria or local restaurants.
- How to manage subscriptions.
- How to track delivery and pickup status.
- **Delivery Format:**
 - Web-based: Hosted on the COS platform or Youtube.

2.7 Assumptions and Dependencies

AS-1: The database will be able to record and handle all data transactions.

AS-2: Database servers have backups and will always be online.

AS-3: The system will use a digital transaction system to process financial transactions.

AS-4: All devices (computers and printers) will have hardwired ethernet connections.

AS-5: Staff will be able to deliver orders within 15 minutes of the requested delivery time.

DE-1: Digital transaction system will be online in our region.

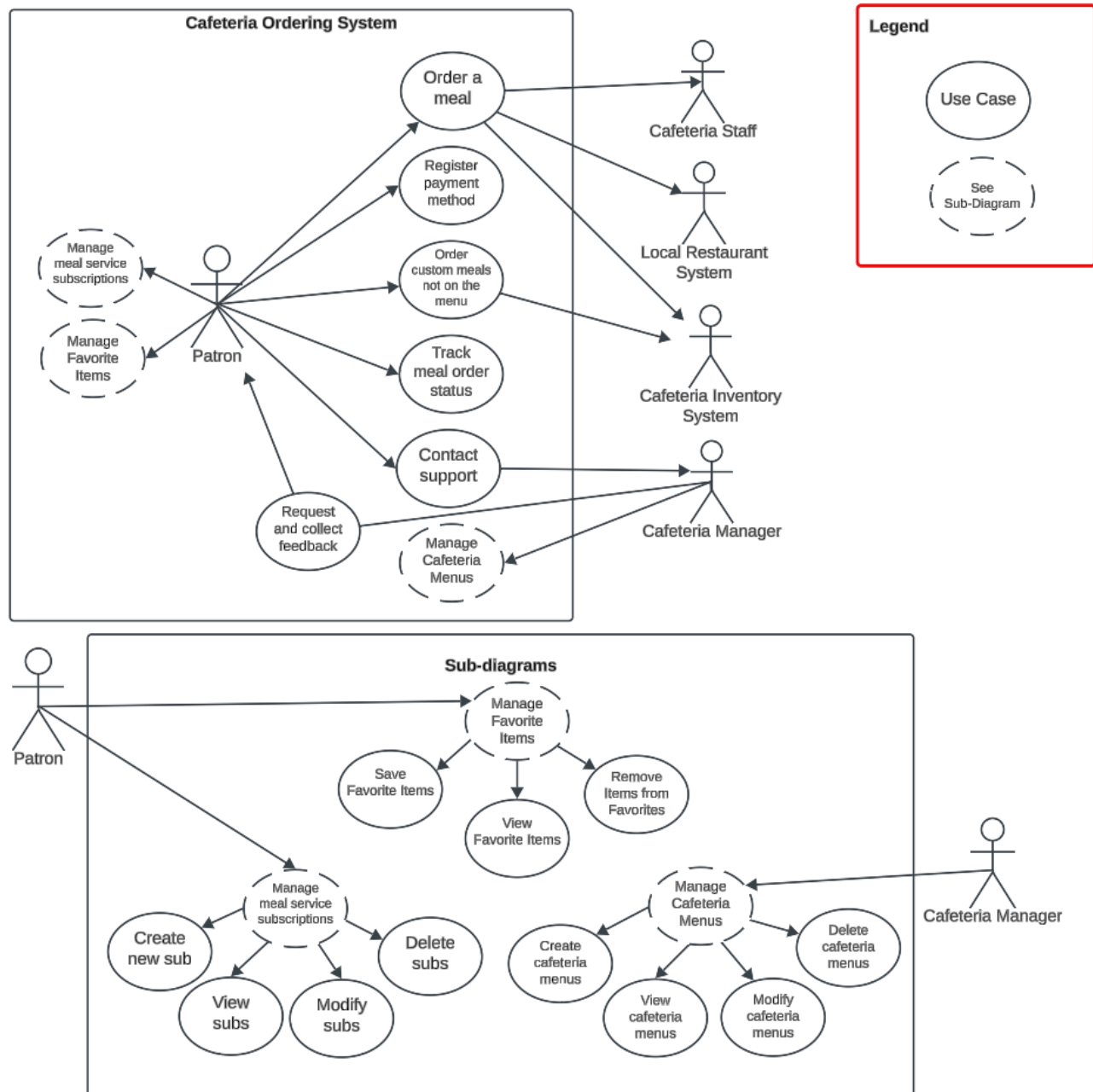
DE-2: The Local Restaurant System is always online.

DE-3: Feedback using 3rd party system will always be online.

3. Use-case diagram with Use-case descriptions (UCs)

3.1 Use-case Diagram

Figure 3-1: A use case diagram of the Cafeteria Ordering System.



3.2 Use-case Description(s)

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

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

Use Case ID:	UC-1		
Use Case Name:	Order a meal		
<u>Related Features</u>	FE-1		
Created By:	Thanh Nguyen	Last Updated By:	10/3/2024
Date Created:	10/3/2024	Last Revision Date:	
Actors:	Patron, Cafeteria Inventory System, Restaurant		
Description:	Patron accesses COS via intranet or home internet, views menu, selects food items and places an order for a meal to be delivered to a designated location within 15 minutes of the desired delivery time.		
Preconditions:	1. Patron is logged into COS. 2. Patron is registered for meal payments by payroll deduction.		
Postconditions:	1. Meal order is stored in COS with status of “accepted”. 2. Cafeteria Inventory System is updated to reflect the items in this order. 3. Delivery capacity for requested time window is updated to reflect this delivery request.		
Normal Flow:	1.0 Order a meal 1. Patron asks to view menu for specified date. 2. System displays menu of available food items. 3. Patron selects one or more food items. 4. Patron indicates ordering is complete. 5. System displays user-selected items, quantity, and price. 6. Patron confirms meal order or requests to modify meal order (back to step 3) 7. System displays available delivery times for specified date. 8. Patron selects delivery time and specifies delivery location. 9. Patron selects payment method. 10. System confirms acceptance of the order.		

Use Case ID:	UC-1
	11. System sends Patron confirmation email with order details, price, and delivery or pickup instructions. 12. System stores order, notifies Cafeteria Staff, and sends food item information to Cafeteria Inventory System.
Alternative Flows: Alternative Flow 1 – Multiple meals order Alternative Flow 2 – Pickup order Alternative Flow 3 – Restaurant order	1.1 Order multiple meals (branch after step 4) 1. Patron asks to order another meal. 2. Return to step 2. 1.2 Order meal for pickup (at step 8) 1. Patron selects pickup order option. 2. System displays available pickup times. 3. Patron selects pickup time. 4. Continue at step 9. 1.3 Order meal from local restaurant (at step 1) 1. Patron asks to view restaurant menu for a specified date. 2. Continue with step 2-11. 3. System stores order and sends order details to local restaurant's online system.
Exceptions:	1.0.E.1 Current time is after order cutoff time (at step 1) 1. System informs Patron it is too late to place an order for today. 2a. Patron cancels the meal order. 2b. System terminate use case. 3a. Patron requests to select another date. 3b. System restarts use case. 1.0.E.2 No delivery times left (at step 1) 1. System informs Patron that no delivery times are available for the meal date. 2a. Patron cancels the meal order. 2b. System terminates the use case.

Use Case ID:	UC-1
	3. Patron requests to pick the order up at the cafeteria (skip steps 7-8) 1.0.E.3 Payment is denied (at step 12) 1. System informs Patron that payment was not accepted, and order is not placed. 2a. Patron cancels meal order. 2b. System terminates use case. 3a. Patron requests to select another payment method. 3b. System restarted use case. 1.3.E.4 Order failed to send to local restaurant (at step 3) 1. System informs Patron that order failed to send to local restaurant, the order is not placed, and they are not charged. 2a. Patron cancels meal order. 2b. System terminates use case. 3. Patron repeats payment method selection.
Includes:	View Menu, Select payment option, Set delivery options
Frequency of Use:	Approximately 400 users, average of one usage per day
Special Requirements:	1. Patron shall be able to cancel the meal order at any time prior to confirming the order. 2. Patron shall be able to view all meals ordered within the last 12 months.
Assumptions:	Local restaurant is an online system that has bi-directional communication with COS.

Use Case ID:	UC-2		
Use Case Name:	Register payment method		
<u>Related Features</u>	FE-4		
Created By:	Thanh Nguyen	Last Updated By:	10/4/2024
Date Created:	10/4/2024	Last Revision Date:	
Actors:	Patron, Digital Wallet Vendor, Credit/debit Verification Service		
Description:	A Patron accesses the COS from the corporate intranet or from home and registers a payment method to pay for future meal orders.		
Preconditions:	Patron is logged into COS.		
Postconditions:	1. Payment method is stored in COS. 2. Patron is registered for meal payments.		
Normal Flow:	1.0 Register payment method with payroll deductions 1. Patron asks to view the payment registration screen. 2. System displays payment options. 3. Patron selects payment option for payroll deduction. 4. System displays user input fields for employee id. 5. Patron provides employee id. 6. Patron completes registration. 7. System confirms registration is successful.		
Alternative Flows: Alternative Flow 1 – Register with credit/debit card	1.1 Register payment method with credit/debit card (at step 3) 1. Patron selects payment option for credit/debit card. 2. System displays user input fields for credit/debit card number, expiration date, and card verification code (CVC). 3. Patron provides credit/debit card number, expiration date, and CVC. 3. Continue at step 6.		
Alternative Flow 2 –	1.2 Register payment method with digital wallet (at step 3)		

Use Case ID:	UC-2
Register with digital wallet	1. Patron selects payment option for digital wallet. 2. System displays user input fields for digital wallet vendor and username. 3. Patron provides vendor and username. 4. System sends vendor an authentication request. 5. Patron completes authentication process. 6. System confirms digital wallet authentication. 7. Continue at step 6.
Exceptions:	1.0.E.1 Employee ID not found (at step 7) 1. System informs Patron that his employee ID was not found and that registration is not complete. 2a. Patron cancels registration. 2b. System terminates use case. 3a. Patron provides employee id again. 3b. System continues at 1.0-Step 6. 1.1.E.1 Invalid credit/debit card number and expiration date (at 1.0-Step 7) 1. System informs Patron credit/debit number and expiration date combination is invalid and that registration is not complete. 2a. Patron cancels registration. 2b. System terminates use case. 3a. Patron provides credit/debit card number and expiration date again. 3b. System continues at 1.0-Step 6. 1.2.E.1 Failed authentication with digital wallet vendor (at step 6) 1. System informs Patron authentication was unsuccessful with digital wallet vendor and that registration is not complete. 2a. Patron cancels registration. 2b. System terminates use case. 3a. Patron requests to reauthenticate.

Use Case ID:	UC-2
	3b. System restarts at step 3.
Frequency of Use:	On demand
Assumptions:	Credit/debit cards are issued in the US.

Use Case ID:	UC-3		
Use Case Name:	Track meal order status		
<u>Related Features</u>	FE-9		
Created By:	Thanh Nguyen	Last Updated By:	10/5/2024
Date Created:	10/5/2024	Last Revision Date:	
Actors:	Patron		
Description:	A Patron accesses the COS from the corporate intranet or from home and tracks the status of his meal order.		
Preconditions:	1. Patron is logged into COS. 2. Patron has submitted a meal order.		
Postconditions:	System displays the order status for the meal order in question.		
Normal Flow:	1.0 Track meal order status 1. Patron asks to view meal order history. 2. System displays order number, order date, order description, order status, and meal provider for previous meal orders.		
Exceptions:			
Frequency of Use:	On demand		

Use Case ID:	UC-4		
Use Case Name:	Create meal service subscriptions		
<u>Related Features</u>	FE-3		
Created By:	Eric Martinez	Last Updated By:	Eric Martinez
Date Created:	9/20/2024	Last Revision Date:	9/25/2024
Actors:	Patron		
Description:	Patrons can create a new meal service subscription by selecting a meal plan, choosing a subscription length, and setting up delivery preferences. The COS system will validate the patron's input, process the subscription request, and send a confirmation.		
Trigger:	The patron initiates the process by selecting the "Create Subscription" option in the meal service interface.		
Preconditions:	1. The patron has a registered account. 2. The patron is logged into COS. 3. The patron has a default payment method.		
Postconditions:	1. The patron's meal subscription is successfully created. 2. The subscription details are stored in the COS system. 3. Cafeteria Inventory System validates the availability of this meal. 4. The patron receives a confirmation email.		
Normal Flow:	1. Patron selects "Manage Meal Subscription". 2. System prompts the patron to create, view, remove, and modify meal subscriptions. 3. Patron creates a meal plan from in-stock items. 4. System prompts for subscription length and delivery preferences. 5. Patron provides subscription length and delivery preferences. 6. System validates the input and processes the subscription request. 7. Patron receives a confirmation of the subscription creation.		
Alternative Flows:	5a. Invalid Subscription Length:		

[Alternative Flow 1 – Not in Network]	If the patron selects an invalid subscription length, the system displays an error message, and the patron can choose a valid length.
Exceptions:	6a. Payment Failure: If payment fails, the system displays a message indicating payment failure, and the patron is prompted to retry or update payment information.
Includes:	NONE
Frequency of Use:	On-demand
Special Requirements:	The system should comply with standards for secure payment processing.
Assumptions:	The patron understands how to navigate the system's interface and select a meal plan.
Notes and Issues:	None

Use Case ID:	UC-5		
Use Case Name:	Create Cafeteria Menus		
<u>Related Features</u>	FE-5		
Created By:	Loc Nguyen	Last Updated By:	Loc Nguyen
Date Created:	10/02/2024	Last Revision Date:	10/05/2024
Actors:	Cafeteria Manager		
Description:	The cafeteria manager creates new cafeteria menus by entering relevant details such as menu items, prices, and availability.		
Trigger:	The cafeteria manager selects the "Create Menu" option from the system.		
Preconditions:	1. The cafeteria manager is logged into the system. 2. The cafeteria inventory is updated with available items.		
Postconditions:	1. A new menu is created and available for patrons to view and order from.		
Normal Flow:	1. The cafeteria manager logs into the system. 2. The manager selects the "Create Menu" option. 3. The system prompts the manager to input the menu details, including item names, descriptions, and prices. 4. The manager submits the new menu for review. 5. The system saves the menu and confirms the creation to the manager.		
Alternative Flows: [Alternative Flow 1 – Not in Network]	1.1 Missing Required Fields (Step 3) 1. If required fields are missing (e.g., item name or price), the system displays an error message, prompting the manager to complete all fields. 2. The system returns to the form, allowing the manager to correct the errors. 3. The flow resumes at step 5 1.2 Menu Creation is Cancelled: 1. At any point during the menu creation process, the cafeteria manager can choose to cancel the operation.		

Use Case ID:	UC-5
	2. The system asks for confirmation of the cancellation. 3. The manager confirms the cancellation. 4. The system cancels the operation and no menu is created.
Exceptions:	E-1: System Error 1. If the system is unable to save the menu due to a technical issue, it displays an error message and logs the issue. 2. The requester is prompted to try again later.
Includes:	None
Frequency of Use:	As needed, when new menu items are introduced.
Special Requirements:	None
Assumptions:	1. The cafeteria manager has appropriate permissions and access to modify menus. 2. The cafeteria inventory system is updated and accurate.
Notes and Issues:	None.

Use Case ID:	UC-6		
Use Case Name:	View Cafeteria Menus		
<u>Related Features</u>	FE-5		
Created By:	Loc Nguyen	Last Updated By:	Loc Nguyen
Date Created:	10/02/2024	Last Revision Date:	10/05/2024
Actors:	Cafeteria Manager		
Description:	Users can view the cafeteria menus, including item names, descriptions, prices, and availability.		
Trigger:	A user selects the option to view the available cafeteria menus.		
Preconditions:	1. The users are logged into the system. 2. At least one menu exists in the system.		
Postconditions:	1. The user views the cafeteria menu		
Normal Flow:	1. The user logs into the system. 2. The user selects the "View Menu" option. 3. The system displays the list of menus with item details. (descriptions, prices, availability).		
Alternative Flows: [Alternative Flow 1 – Not in Network]	1.1 No Menu Available (Step 3) 1. If no menus are currently available, the system displays a message: " No menus available at this time." 2. The user returns to the previous page or tries again later.		
Exceptions:	E-1: System Error 1. If the system is unable to retrieve the menus due to a technical issue (e.g., database error), it displays an error message and logs the issue. 2. The user is prompted to try again later.		
Includes:	None		

Use Case ID:	UC-6
Frequency of Use:	Multiple times per day by patrons and cafeteria managers to review the menus.
Special Requirements:	The menus should load quickly, especially when accessed during peak hours.
Assumptions:	The displayed menus are up-to-date and reflect the current offerings.
Notes and Issues:	None

Use Case ID:	UC-7		
Use Case Name:	Modify Cafeteria Menus		
<u>Related Features</u>	FE-5		
Created By:	Loc Nguyen	Last Updated By:	Loc Nguyen
Date Created:	10/03/2024	Last Revision Date:	10/05/2024
Actors:	Cafeteria Manager		
Description:	The cafeteria manager modifies existing menus by updating menu items, prices, and availability based on changes in inventory or seasonal offerings.		
Trigger:	The cafeteria manager selects the option to modify an existing menu.		
Preconditions:	1. The cafeteria manager is logged into the system with permission. 2. A menu exists and is available for editing.		
Postconditions:	1. The selected menu is updated with new item details 2. Patrons can view the modified menu and place orders based on the updated offerings.		
Normal Flow:	1. The cafeteria manager logs into the system. 2. The manager selects the "Modify Menu" option. 3. The system displays the current menu for editing. 4. The manager makes the necessary modifications. 5. The manager submits the changes. 6. The system updates the menu and confirms the changes to the manager.		
Alternative Flows: [Alternative Flow 1 – Not in Network]	1.1 Menu modification is canceled (Step 3) 1. At any point during the modification process, the cafeteria manager can cancel the operation. 2. The system asks for confirmation of the cancellation. 3. The manager confirms the cancellation. 4. The system cancels the changes and keeps the original menu intact.		

Use Case ID:	UC-7
	1.2 Menu item not found 1. If the manager tries to modify a menu item that no longer exists, the system displays a message saying, "Menu item not found." 2. The manager can search for another item or cancel the modification.
Exceptions:	E-1 System Error 1. If the system is unable to save the modifications due to a technical issue, it displays an error message and logs the issue. 2. The cafeteria manager is prompted to try again later.
Includes:	None
Frequency of Use:	As needed based on inventory or seasonal changes.
Special Requirements:	The system should prevent conflicting menu items (e.g., duplicate items or incorrect prices).
Assumptions:	The cafeteria manager has permission to modify the menu.
Notes and Issues:	None

Use Case ID:	UC-8		
Use Case Name:	Delete Cafeteria Menus		
<u>Related Features</u>	FE-5		
Created By:	Loc Nguyen	Last Updated By:	Loc Nguyen
Date Created:	10/03/2024	Last Revision Date:	10/05/2024
Actors:	Cafeteria Manager		
Description:	The cafeteria manager deletes an outdated or unnecessary menu from the system.		
Trigger:	The cafeteria manager selects the "Delete Menu" option.		
Preconditions:	1. The cafeteria manager is logged into the system. 2. A menu exists and is available for deletion.		
Postconditions:	1. The cafeteria menu is removed from the system.		
Normal Flow:	1. The cafeteria manager logs into the system. 2. The manager selects the "Delete Menu" option. 3. The system prompts the manager for confirmation. 4. The manager confirms the deletion. 5. The system deletes the menu and confirms the action to the manager.		
Alternative Flows: [Alternative Flow 1 – Not in Network]	1.1 Menu deletion is canceled (Step 4) 1. When prompted to delete a menu, the cafeteria manager can cancel the deletion. 2. The system asks for confirmation of the cancellation. 3. The cafeteria manager confirms the cancellation. 4. The system retains the menu, and no deletion occurs. 1.2 Menu in use 1. If the menu being deleted is currently in use, the system displays a message saying, "Menu cannot be deleted because it is currently in use."		

	2. The cafeteria manager is prompted to try again.
Exceptions:	E-1: System Error 1. If the system is unable to delete the menu due to a technical issue, it displays an error message and logs the issue. 2. The cafeteria manager is prompted to try again later.
Includes:	None
Frequency of Use:	As needed or when an outdated menu needs to be removed.
Special Requirements:	The system should confirm deletion to prevent accidental removal of important menus.
Assumptions:	The cafeteria manager has appropriate permissions to delete the menu.
Notes and Issues:	None

Use Case ID:	UC-9		
Use Case Name:	View meal service subscriptions		
<u>Related Features</u>	FE-3		
Created By:	Eric Martinez	Last Updated By:	Eric Martinez
Date Created:	9/20/2024	Last Revision Date:	9/25/2024
Actors:	Patron		
Description:	Patrons can view meal service subscriptions by selecting “Manage Meal Subscriptions”. The COS system will display the meal subscriptions.		
Trigger:	The patron initiates the process by selecting the "View Meal Subscription" option in the meal service interface.		
Preconditions:	1. The patron has a registered account. 2. The patron is logged into COS.		
Postconditions:	1. The subscription details are stored in the COS system. 2. Meal Subscription will be displayed from the patron account.		
Normal Flow:	1. Patron selects "Manage Meal Subscription". 2. System prompts the patron to create, view, remove, and modify meal subscriptions. 3. Patron selects ‘view meal subscriptions’. 4. System displays previous subscription length, delivery preferences and menu items in the meal(s).		
Alternative Flows: [Alternative Flow 1 – Not in Network]	4a. Nothing to Display: If a patron has never created a meal subscription, nothing will be displayed.		
Exceptions:	NONE		
Includes:	NONE		
Frequency of Use:	On-demand		

Special Requirements:	NONE
Assumptions:	The patron understands how to navigate the system's interface and select a meal plan.
Notes and Issues:	None

Use Case ID:	UC-10		
Use Case Name:	Delete meal service subscriptions		
<u>Related Features</u>	FE-3		
Created By:	Eric Martinez	Last Updated By:	Eric Martinez
Date Created:	9/26/2024	Last Revision Date:	9/30/2024
Actors:	Patron		
Description:	Patrons can delete meal service subscriptions by selecting “Manage Meal Subscriptions”. The COS system will display the meal subscriptions.		
Trigger:	The patron initiates the process by selecting the "Delete Meal Subscription" option in the meal service interface.		
Preconditions:	1. The patron has a registered account. 2. The patron is logged into COS.		
Postconditions:	1. The subscription details are stored in the COS system. 2. Meal Subscription will be deleted from the patron account.		
Normal Flow:	1. Patron selects "Manage Meal Subscription". 2. System prompts the patron to create, view, remove, and modify meal subscriptions. 3. Patron selects ‘delete meal subscriptions’. 4. System displays previous meal subscriptions. 5. Patron selected meal subscription to delete. 6. System deletes meal subscription and send confirmation email.		
Alternative Flows: [Alternative Flow 1 – Not in Network]	4a. Nothing to Display: If a patron has never created a meal subscription, nothing will be displayed.		
Exceptions:	6a. System Error on Delete: The system was unable to delete the meal subscription.		

Includes:	NONE
Frequency of Use:	On-demand
Special Requirements:	NONE
Assumptions:	The patron understands how to navigate the system's interface and select a meal plan.
Notes and Issues:	None

Use Case ID:	UC-11		
Use Case Name:	Modify meal service subscriptions		
<u>Related Features</u>	Feature 3		
Created By:	Eric Martinez	Last Updated By:	Eric Martinez
Date Created:	9/26/2024	Last Revision Date:	9/30/2024
Actors:	Patron		
Description:	Patrons can modify meal service subscriptions by selecting “Manage Meal Subscriptions”. The COS system will display the meal subscriptions to modify.		
Trigger:	The patron initiates the process by selecting the "Manage Meal Subscription" option in the meal service interface.		
Preconditions:	1. The patron has a registered account. 2. The patron is logged into COS.		
Postconditions:	1. The subscription details are stored in the COS system. 2. Meal Subscription will successfully be modified for the patron account.		
Normal Flow:	1. Patron selects "Manage Meal Subscription". 2. System prompts the patron to create, view, remove, and modify meal subscriptions. 3. Patron selects ‘modify meal subscriptions’. 4. System displays previous meal subscriptions. 5. Patron selects a meal subscription to modify and submits. 6. System confirms the modification.		
Alternative Flows: [Alternative Flow 1 – Not in Network]	4a. Nothing to Display: If a patron has never created a meal subscription, nothing will be displayed.		
Exceptions:	6a. System Cannot Make Medication: New or existing items in meal subscriptions are no longer in stock.		

Includes:	NONE
Frequency of Use:	On-demand
Special Requirements:	NONE
Assumptions:	The patron understands how to navigate the system's interface and select a meal plan.
Notes and Issues:	None

Use Case ID:	UC-12		
Use Case Name:	Order custom meal		
<u>Related Features</u>	Feature 6		
Created By:	Eric Martinez	Last Updated By:	Eric Martinez
Date Created:	9/29/2024	Last Revision Date:	10/05/2024
Actors:	Patron		
Description:	Patrons can create a customized meal order by selecting items that are not typically contained in a meal. The COS processes the customization request and generates the meal order.		
Trigger:	The patron initiates the process by selecting the "Customize Meal" option on an existing meal.		
Preconditions:	1. The patron has a registered account. 2. The patron is logged into COS.		
Postconditions:	1. The custom meal order is successfully created. 2. The order is sent to the cafeteria kitchen for preparation. 3. The patron receives an order confirmation.		
Normal Flow:	1. Patron selects a meal. 2. Patron selects "Customize Meal" on the meal selected. 3. System prompts the patron to select/deselect items to customize meal. 4. Patron selects items from the list of available options. 5. System processes the selected ingredients, CIS checks availability, and validates the custom order. 6. Patron reviews the order and confirms the customization. 7. System sends the order to the kitchen and provides confirmation to the patron.		
Alternative Flows: [Alternative Flow 1 – Not	4a. Ingredient Not Available:		

in Network]	If an ingredient selected by the patron is not available, the system will prompt the patron to choose an alternative ingredient or modify the order.
Exceptions:	NONE
Includes:	NONE
Frequency of Use:	On-demand
Special Requirements:	NONE
Assumptions:	The patron understands how to navigate the system's interface and select ingredients for custom meals.
Notes and Issues:	None

Use Case ID:	UC-13		
Use Case Name:	Save Favorite Items		
<u>Related Features</u>	FE-10		
Created By:	Loc Nguyen	Last Updated By:	Loc Nguyen
Date Created:	10/05/2024	Last Revision Date:	10/05/2024
Actors:	Patron		
Description:	The patron can save specific items from previous orders as favorites.		
Trigger:	The patron selects an item from a past order and chooses the option to save it as a favorite.		
Preconditions:	1. The patron is logged into the system. 2. The patron has at least one past order to select items from.		
Postconditions:	1. The selected item is saved to the patron's list of favorite items.		
Normal Flow:	1. The patron logs into the system and navigates to past orders. 2. The patron selects an item from a previous order and clicks "Save to Favorites." 3. The system confirms that the item has been successfully added to the favorites list.		
Alternative Flows: [Alternative Flow 1 – Not in Network]	1.1 Duplicate Item (Step 2) 1. If the patron attempts to save an item that is already in the favorites list, the system will display a message saying, "Item already exists in favorites." 2. The patron can either choose a different item or return to the main menu.		
Exceptions:	E-1 System Error 1. If the system cannot save the favorite item due to a technical issue, it displays an error message and prompts the patron to try again later.		
Includes:	None		

Frequency of Use:	When patrons wish to save items for future orders.
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

Use Case ID:	UC-14		
Use Case Name:	View Favorite Items		
<u>Related Features</u>	FE-10		
Created By:	Loc Nguyen	Last Updated By:	Loc Nguyen
Date Created:	10/05/2024	Last Revision Date:	10/06/2024
Actors:	Patron		
Description:	The patron can view their list of favorite items to make quick meal selections.		
Trigger:	The patron selects the "View Favorite Items" option from the system.		
Preconditions:	1. The patron is logged into the system. 2. The patron has saved at least one item to their favorites list.		
Postconditions:	1. The system displays the list of favorite items.		
Normal Flow:	1. The patron logs into the system and selects "Manage Favorite Items." 2. The patron selects "View Favorite Items" to see all saved favorite meals. 3. The system retrieves and displays the list of favorite items.		
Alternative Flows: [Alternative Flow 1 – Not in Network]	1.1 No Favorites to Display 1. If there are no favorite items saved, the system displays a message stating "No favorite items saved." 2. The patron can return to the menu or save a new favorite item.		
Exceptions:	None		
Includes:	None		
Frequency of Use:	When patrons wish to view their favorite meals for quick reordering.		
Special Requirements:	None		
Assumptions:	The patron has saved favorite items previously.		
Notes and Issues:	None		

Use Case ID:	UC-15		
Use Case Name:	Remove Favorite Items		
<u>Related Features</u>	FE-10		
Created By:	Loc Nguyen	Last Updated By:	Loc Nguyen
Date Created:	10/05/2024	Last Revision Date:	10/06/2024
Actors:	Patron		
Description:	The patron can remove their list of favorite items to make quick meal selections.		
Trigger:	The patron selects an item from the favorites list and removes it.		
Preconditions:	1. The patron is logged into the system. 2. The patron has saved items to their favorites list.		
Postconditions:	1. The selected item is removed from the list of favorite items.		
Normal Flow:	1. The patron logs into the system and selects "Manage Favorite Items." 2. The patron selects "View Favorite Items" to see all saved favorite meals. 3. The patron selects an item and clicks "Remove from Favorites." 4. The system confirms that the item has been successfully removed from the list.		
Alternative Flows: [Alternative Flow 1 – Not in Network]	1.1 No Favorites to Remove 1. If there are no items to remove, the system displays a message saying "No items to remove from favorites." 2. The patron can return to the menu or save a new favorite item.		
Exceptions:	If the system fails to remove the item, it will display an error message and ask the patron to try again later.		
Includes:	None		
Frequency of Use:	When patrons wish to update their favorite meals for quick reordering.		
Special Requirements:	None		

Assumptions:	The patron has saved favorite items previously.
Notes and Issues:	None

Use Case ID:	UC-16		
Use Case Name:	Request and Collect Feedback		
<u>Related Features</u>	FE-7		
Created By:	Ryan Pham	Last Updated By:	Ryan Pham
Date Created:	10/04/2024	Last Revision Date:	10/04/2024
Actors:	Patron (Employees that use the COS)		
Description:	The system collects information and feedback from customers who used the COS after finishing their orders which can help improve the COS system.		
Trigger:	The customer successfully receives their order		
Preconditions:	1. Patron completed their order using the COS 2. The order is delivered or picked up by the customer		
Postconditions:	1. Feedback is gathered and saved. 2. Feedback will be sent to the Cafeteria/Restaurant for review		
Normal Flow:	1. Patron picked up the order. 2. COS sends a feedback request. 3. Patron provides feedback. 4. Feedback is stored in the system. 5. Feedback is shared with the Cafeteria Manager.		
Alternative Flows: [Alternative Flow 1 – Not in Network]	1. The cafeteria staff successfully delivered the order to the patron. 2. COS sends a feedback request. 3. Patron provides feedback. 4. Feedback is stored in the system. 5. Feedback is shared with the Restaurant Manager.		
Exceptions:	The patron ignores the feedback request.		
Includes:	None		
Frequency of Use:	Only if there is a problem with the service		

Special Requirements:	None
Assumptions:	Customers have a valid reason for contacting support regarding their order.
Notes and Issues:	None

Use Case ID:	UC-17		
Use Case Name:	Contact support		
Related Features	FE-8		
Created By:	Ryan Pham	Last Updated By:	Ryan Pham
Date Created:	10/04/2024	Last Revision Date:	10/04/2024
Actors:	Patron (Employees that use the COS), Cafeteria Manager, Restaurant Manager		
Description:	This use case allows the patron to contact support when they have an issue with their order.		
Trigger:	The patron experiences an issue with their order		
Preconditions:	1. The patron ordered their meal. 2. An issue occurs for the other before/after receiving their order		
Postconditions:	1. A support ticket is created. 2. The issue is sent to the Cafeteria/Restaurant Manager for a resolution.		
Normal Flow:	1. Patron encounters an issue with their order. 2. Patron selects "Contact Support." 3. Patron describes the issue. 4. COS creates a support ticket. 5. Patron receives confirmation. 6. The cafeteria/Restaurant Manager staff resolves the issue.		
Alternative Flows: [Alternative Flow 1 – Not in Network]	None		
Exceptions:	1. The customer cannot contact support due to technical issues. 2. The issue is escalated if unresolved.		
Includes:	None		
Frequency of Use:	Only if there is a problem with the service		
Special Requirements:	None		

Assumptions:	Customers have a valid reason for contacting support regarding their order.
Notes and Issues:	None

4. External Interface Requirements

4.1 User Interfaces

UI-1: The GUI must be modern, smooth, and intuitive, following current design standards.

UI-2: Every screen should include clearly accessible settings, home, and help buttons for easy navigation.

UI-3: A title at the top of each screen should describe the current page or task the user is performing.

UI-4: Most buttons will navigate users to the next screen, except for multi-select radio buttons in the "Order a Meal" section.

UI-5: The "Order a Meal" section must allow users to select multiple options, and the next button will lead to the checkout page.

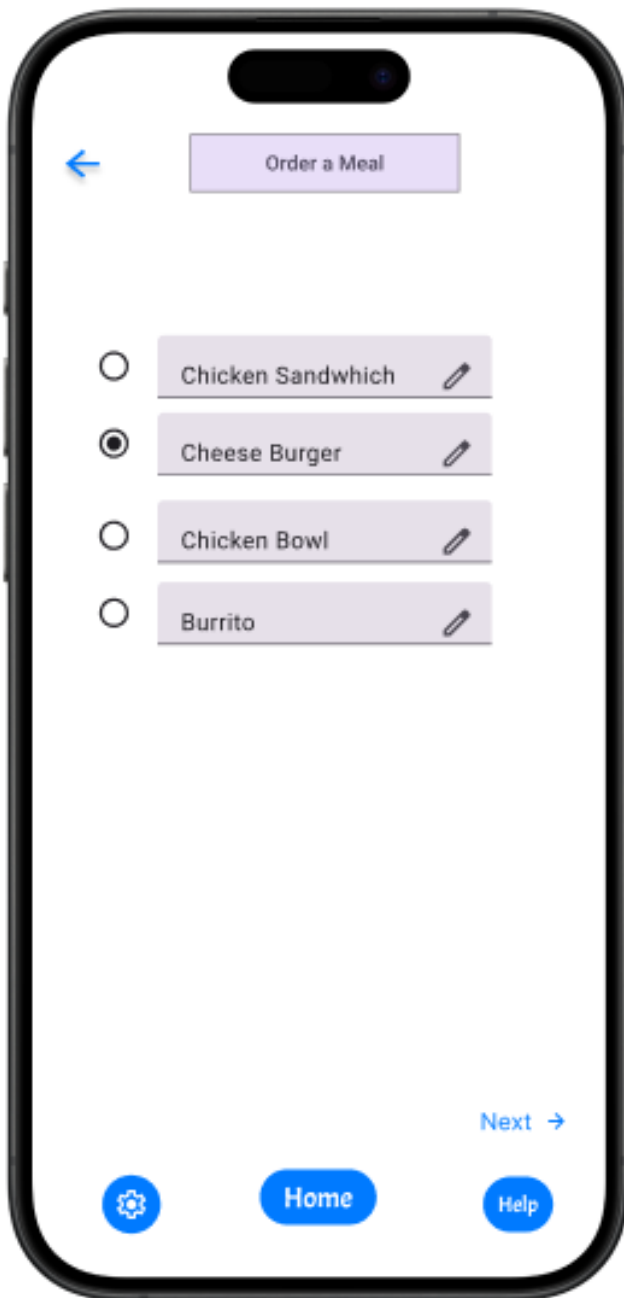
UI-6: Each meal should have an edit icon that directs users to a page where they can add or remove items that are currently in stock.

UI-7: If a user selects an out-of-stock item, an error message must notify the user of the stock issue.

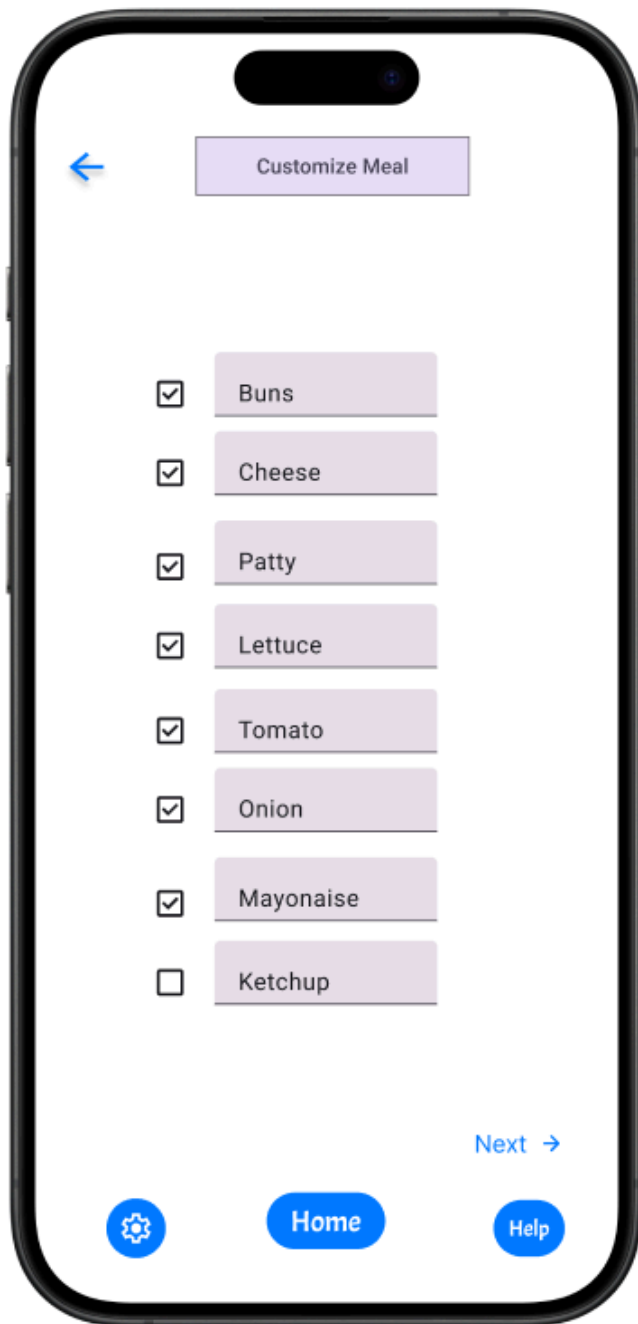
Homepage:



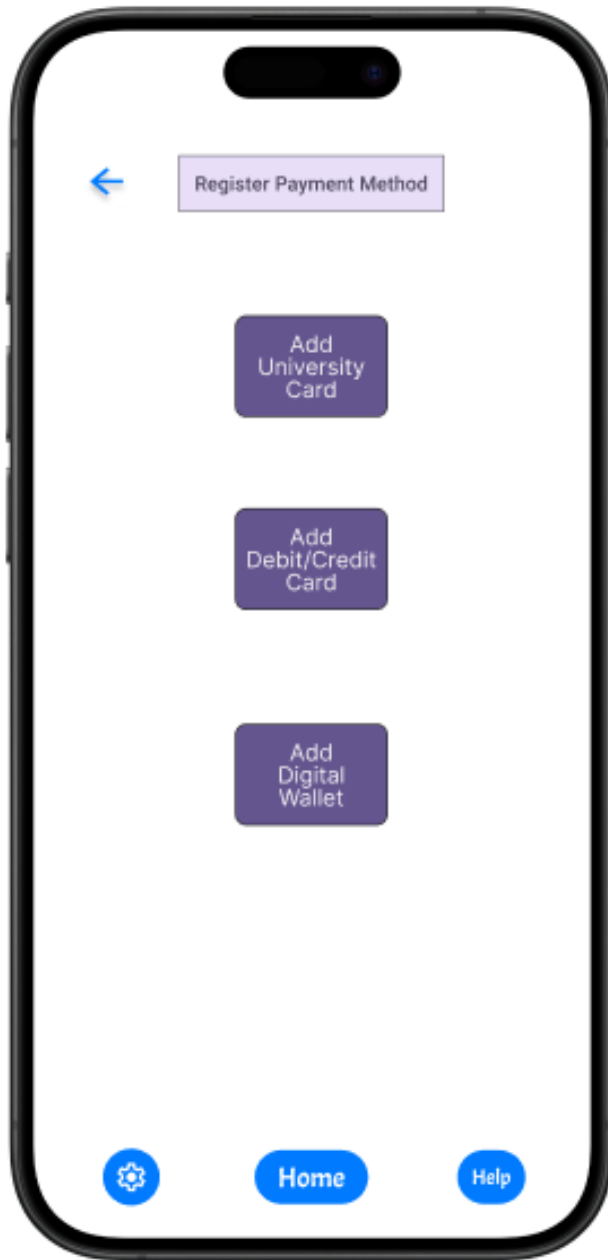
Order a meal screen:



Customize meal screen:



Register a payment method screen:



4.2 Hardware Interfaces

HI-1: Receipt Printers

- **HI-1.1:** The COS transmits order details to receipt printers via USB or network protocols (e.g., Ethernet) for printing.

HI-2: Kitchen Display Systems

- **HI-2.1:** The COS sends order details in real time to kitchen display systems via local wired or wireless networks.
- **HI-2.2:** The COS updates the kitchen display system when an order is modified or canceled.
- **HI-2.3:** The COS marks an order completed when kitchen staff confirm order preparation on the kitchen display system.

HI-3: User Devices

- **HI-3.1:** The COS is accessible via various user devices, including smartphones, tablets, and computers, through web browsers (supporting HTML5 and CSS3).
- **HI-3.2:** The COS is compatible with iOS, Android, Windows, and macOS devices.
- **HI-3.3:** The COS sends notifications about order status updates to the user's device directly.

HI-4: Delivery Tracking Devices

- **HI 4.1: GPS-Enabled Devices:** The system will interface with GPS-enabled mobile devices or delivery tracking hardware, allowing patrons and cafeteria staff to track the status and location of the meal delivery in real-time.

HI-5: Point-of-Sale Terminals (POS)

- **HI-5.1:** The COS shall interface with POS terminals through Ethernet or Wi-Fi to synchronize orders and payment data in real-time. After successful order completion, the COS shall send payment and order confirmation to the POS terminal.

4.3 Software Interfaces

SI-1: Cafeteria Inventory System

- **SI-1.1:** The COS shall request food item quantities from the Cafeteria Inventory System to determine if requested food item is available.
- **SI-1.2:** The COS shall transmit food item quantities to the Cafeteria Inventory System through a programmatic interface.
- **SI-1.3:** When the Cafeteria Inventory System notifies COS that a food item is unavailable, COS shall remove the food item from the menu for that day.

SI-2: Payroll System

- **SI-2.1:** The COS shall transmit registration and payment transactions details to the Payroll System through a programmatic interface.
- **SI-2.2:** The COS shall transmit registration details to the Payroll System to complete payment method registration for payroll deduction.
- **SI-2.3:** When Payroll System notifies COS that employee id is not found, COS shall relay notification to Patron.

- **SI-2.4:** The COS shall transmit payment deduction amount to Payroll System to complete order meal transaction.

SI-3: Payment Processing System (Credit/Debit Card)

- **SI-3.1:** The COS shall transmit payment method details to the Payment Processing System through a programmatic interface.
- **SI-3.2:** When Payment Processing System notifies COS for incomplete payment transaction, COS relays payment failure notification to Patron.

SI-4: Payment Processing System (Digital Wallet)

- **SI-4.1:** The COS shall transmit a user authentication request to Payment Processing System through a programmatic interface.
- **SI-4.2:** The COS shall transmit payment amount to the Payment Processing System to complete the order meal transaction.
- **SI-4.3:** When Payment Processing System notifies COS for failure to authenticate user, COS shall notify user of failure and allow mitigation steps.

SI-5: Local Restaurant System

- **SI-5.1:** The COS shall transmit meal orders to the Local Restaurant System through a programmatic interface.
- **SI-5.2:** Local Restaurant System shall notify COS when order is received, fulfilled, or canceled.

- **SI-5.3:** COS shall request for available menu items from Local Restaurant System to provide customer in-stock selections.

4.4 Communications Interfaces

- **CI-1:** The COS will send order confirmations and feedback to customers via email
- **CI-2:** The COS will send order confirmations, order status, and feedback requests via text messages
- **CI-3:** The system will allow all patrons and staff to interact with the COS using a website
- **CI-4:** Patrons that use the COS will be able to submit feedback forms
- **CI-5:** The COS will store data collected such as order history, processed payments, and feedback responses from the user

Communication Functions	Description	Communication Standards (How it Talks)	Security & Encryption Issues (How it's protected)	Data Transfer Rates (How fast is it?)
Email Notifications	The system will send order confirmations and feedback requests to customers via Email.	Simple Mail Transfer Protocol (SMTP)	Will be secured with SSL and TLS for encrypting data	The system will send a notification immediately after the order is processed
SMS Notifications	The system will send order confirmations, order status, and feedback requests via text messages	Use external services such as Twilio's SMS API for sending messages	SMS encryption (Service offered by Twilio) and OAuth for user authorization	Notifications will be sent based on the status of the order in Real-time, such as: 1. Order is placed

				2. Order is being prepared 3. Order is Ready for pickup/Delivered 4. Requesting feedback
Web Browser Access	This will allow all patrons and staff to interact with the system through a website	Use website communication HTTP/HTTPS	Secured with HTTPS will encrypt information sent between a web browser and a website	The website will be immediately responsive to user interaction
Feedback Forms	Users of the COS can submit feedback through forms on the website	Use website communication HTTP/HTTPS	HTTPS will be used for encryption	Forms will be submitted instantly and processed directly to the database to be synced up.
Database Communication	The system will be able to store information collected from the user, such as order history, processed payments, feedback responses	SQL (Relational Structured Query Language)	Will be secured with Secure Sockets Layer (SSL) encryption to protect the data from being viewed by external parties or internet service providers	Ideal database communication should be fast and synchronize instantly with system changes

5. Other Nonfunctional Requirements

5.1 Performance Requirements

PER-1: System must accommodate 400 users and support a maximum of 150 concurrent users during peak hours 9:00 am to 11:00 am.

PER-2: Website pages should load in under 5 seconds.

PER-3: System should allow a maximum of 10 seconds for responses from external systems.

PER-4: Page should respond with system messages to the user in under 2 seconds.

5.2 Safety Requirements

SAF-1: Data Security: The COS must implement encryption protocols for all communications between user devices and servers to prevent unauthorized access to sensitive data (e.g. payment information)

SAF-2: Multi-Factor Authentication (MFA): The COS must require MFA to prevent unauthorized access to the system.

SAF-3: Payment Safety: The system shall offer secure payment options, such as tokenized card data, to minimize the risk of information being exposed during transactions.

5.3 Security Requirements

SEC-1: Identity authentication must be enforced through secure login credentials, including multi-factor authentication (MFA), to prevent account compromise by malicious users.

SEC-2: All personal information and payment data must be encrypted both in transit and at rest, utilizing TLS (Transport Layer Security) and AES (Advanced Encryption Standard).

SEC-3: The system must comply with the Payment Card Industry Data Security Standard (PCI DSS) for payment processing security.

SEC-4: The product must adhere to the California Consumer Privacy Act (CCPA) regulations, ensuring user privacy protection, as it operates in the state of California.

SEC-5: Regular security audits and vulnerability assessments must be conducted to ensure ongoing compliance with industry security certifications and to identify potential risks.

5.4 Software Quality Attributes

5.4.1 Description

This section highlights the quality characteristics of the product that will be important for customers and developers:

1. Adaptability

- **SQA-1:** The system can easily adapt to changes systematically
- **SQA-2:** COS will make it easy to add or update menu items.

2. Availability

- **SQA-3:** The system should be 100% up and running during working hours for employees
- **SQA-4:** Patrons can access it at any time of the day

3. Correctness

- **SQA-5:** The orders placed on the COS must be processed with precision so that the items are charged correctly for each order.

4. Flexibility

- **SQA-6:** Patrons should be able to update or remove items in their cart
- **SQA-7:** Customers will be able to customize their meals.

5. Interoperability (APIs)

- **SQA-8:** The COS must be able to communicate with other external systems, such as databases for inventory, Stripe for payment processing, and Twilio for SMS notifications.

6. Maintainability

- **SQA-9:** The system's backend should be designed to be scalable and easy to use for developers to maintain and integrate new features easily.

7. Portability

- **SQA-10:** The COS should be accessible to a variety of devices and isn't limited to one

8. Reliability

- **SQA-11:** It is an expectation for this system to be reliable and work almost 100% of the time for employees of the institution to receive their food without any trouble.

9. Reusability

- **SQA-12:** Parts of the COS system, especially the backend for inventory management and payment processing, should be well designed to be reusable for future innovations.

10. Robustness

- **SQA-13:** The system should be able to function and operate smoothly without any trouble when there is a lot of user traffic around lunchtime.

11. Testability

- **SQA-14:** Developers should be able to conduct Unit tests as they develop features for the COS and debug any errors they encounter immediately

12. Usability

- **SQA-15:** The software interface should be easy to use and navigate, enabling users to place orders on the COS without any help.

6. Other Requirements

NONE

7. Functional Requirements (FRs)

7.1 FE-1: Order meals from the cafeteria menu for pickup or delivery.

7.1.1 FR-1: System displays menu of available items

A. **Introduction/Functionality:** *The COS shall allow the Patron to view a menu of available food items for a given date.*

B. **Traced:**

- UC-01 (Order a meal)
- UC-06 (View Cafeteria Menus)

C. **Inputs:**

- Desired menu date

D. **Processing:**

Step 1: System retrieves menu from Cafeteria Inventory System for given date.

Step 2: System displays available food selections for Patron to select.

E. **Outputs:**

- Display available food selections that are available in the Cafeteria Inventory System for a given date.

F. **Error Handling:**

1. If the specified date is less than today, the system shall display a warning message and ask the Patron to select the current date or a future date.

7.1.2 FR-2: Add food item from menu

A. **Introduction/Functionality:** *The COS shall allow the Patron to add food items from a menu to an active order.*

B. **Traced:** UC-01 (Order a meal)

C. **Inputs:** Desired food item selections

D. **Processing:**

Step 1: Patron selects food items.

Step 2: System adds selected food items to the active order.

E. **Outputs:** Active order reflects current food items selections

F. **Error Handling:**

1. If the Patron selects a food item whose quantity is greater than available quantity, disable the option to add more items of that food item.

7.1.3 FR-3: Remove food item from active order

A. **Introduction/Functionality:** *The COS shall allow the Patron to remove food items from an active order.*

B. **Traced:** UC-01 (Order a meal)

C. **Inputs:** Desired food item to remove from order

D. **Processing:**

Step 1: Patron selects food items for removal.

Step 2: System removes selected food items from the active order.

E. **Outputs:** Active order reflects current food items selections.

F. **Error Handling:**

1. If the quantity of a food item in an active order is less than zero, remove the item completely from the active order.

7.1.4 FR-4: Select order pickup option

A. **Introduction/Functionality:** *The COS shall allow the Patron to choose to order for pickup.*

B. **Traced:** UC-01 (Alternative Flow - Pickup Order)

C. **Inputs:** Fulfillment method (pickup)

D. **Processing:**

Step 1: Patron selects order for pickup.

Step 2: COS System sets the fulfillment method to pickup.

Step 3: Cafeteria Staff would receive order and prepare it.

E. **Outputs:** Display fulfillment method.

F. **Error Handling:** None

7.1.5 FR-5: Select order delivery option

- A. **Introduction/Functionality:** *The COS shall allow the Patron to choose to order for delivery.*
- B. **Traced:** UC-01
- C. **Inputs:**
 - Fulfillment method (delivery)
 - Delivery date
- D. **Processing:**
 - Step 1:** Patron selects order for delivery.
 - Step 2:** COS System retrieves available delivery times for delivery date.
- E. **Outputs:** Display available delivery times.
- F. **Error Handling:**
 - 1. If there are no available delivery times for delivery date, the COS shall warn the Patron there are no delivery times available for the specified date and provide alternatives (1) select a new date or (2) select pickup option.

7.1.6 FR-6: Select delivery time

- A. **Introduction/Functionality:** *The COS shall allow the Patron to select a delivery time.*
- B. **Traced:** UC-01
- C. **Inputs:** Delivery time
- D. **Processing:**
 - Step 1:** Patron selects available delivery time.
 - Step 2:** System sets delivery time to Patron's time selection.
- E. **Outputs:** Display empty "Delivery Location" input field.
- F. **Error Handling:** None

7.1.7 FR-7: Patron chooses delivery location

- A. **Introduction/Functionality:** *The COS shall allow the Patron to choose a delivery location.*
- B. **Traced:** UC-01

C. **Inputs:** Delivery location

D. **Processing:**

Step 1: Patron provides delivery location.

Step 2: System sets the delivery location to the Patron's desired location.

E. **Outputs:** Display delivery location.

F. **Error Handling:** None

7.2 FE-2: Order meals from local restaurants for delivery.

7.2.1 FR-8: Allowing Patron to specify a menu date

A. **Introduction/Functionality:** *The system shall allow the Patron to specify a menu date when placing a meal order from the local restaurant.*

B. **Traced:** UC-01 (Alternative Flow 3 – Restaurant order)

C. **Inputs:**

- Selected date
- Current date
- Available menu items

D. **Processing:**

Step 1: Allow the Patron to specify a date for ordering.

Step 2: If the specified date is greater than or equal to the current date, display available menu items.

Step 3: If the specified date is in the past, display a warning message prompting the Patron to select the current or a future date.

E. **Outputs:**

- List of available food selections for the chosen date
- Warning message if the date is in the past

F. **Error Handling:**

1. If the specified date is less than today, display a message "Please select a current or future date."

7.2.2 FR-9: Allowing Patron the Patron to choose to order for pickup or delivery.

- A. **Introduction/Functionality:** *The system shall allow the Patron to choose between pick up or delivery options.*
- B. **Traced:** UC-01 (Alternative Flow 3 – Restaurant order)
- C. **Inputs:**
- Select Pick up/ Delivery option.
 - Select available date and time.
- D. **Processing:**
- Step 1:** If delivery is selected and no delivery times are available, display alternatives: (1) select a new date or (2) switch to pickup.
- Step 2:** If delivery times are available, prompt the Patron to provide a delivery location.
- E. **Outputs:** Confirmation order type.
- F. **Error Handling:**
1. If no delivery times are available, display: “No delivery slots available for the selected date. Please choose a new date or select a pickup.”

7.2.3 FR-10: Managing local-restaurant orders

- A. **Introduction/Functionality:** *The system shall allow the Patron to add or remove items before confirming.*
- B. **Traced:** UC-01 (Alternative Flow 3 – Restaurant order)
- C. **Inputs:** Select menu items.
- D. **Processing:**
- Step 1:** Allow the Patron to add items to their order.
- Step 2:** Allow the Patron to remove items from their order.
- Step 3:** Update order summary
- E. **Outputs:** Confirmation message upon successfully updating order.
- F. **Error Handling:**

1. If an item cannot be added or removed, display a message “Unable to modify your order at this time, please try again later.”

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

7.3.1 FR-11: Create Meal Service Subscription

- A. **Introduction/Functionality:** The system enables patrons to create a meal plan, with all details
- B. **Traced:** UC-04: Create Meal Service Subscription
- C. **Inputs:**
 - Meal from menu
 - Subscription length
 - Delivery frequency
 - Delivery address
 - Payment information
 - Patron's account ID
- D. **Processing:**
 - Step 1:** Systems display available meal plans
 - Step 2:** Provide subscription lengths from predefined options.
 - Step 3:** Provide delivery frequency options (some plans may have specific delivery constraints).
 - Step 4:** Validate input fields
 - Step 5:** Generate a unique subscription ID.
 - Step 6:** Trigger a payment processing request; if successful, confirm the subscription creation.
 - Step 7:** Update the patron's account with the new subscription.
- E. **Outputs:**
 - A confirmation message with subscription ID and subscription details
 - An email notification is sent to the patron with the subscription ID and subscription details.
 - Subscription details are stored in the database for future reference and order fulfillment.

F. Error Handling:

Input Validation Errors:

If a field is missing or incorrect, display an error message

Payment Processing Errors:

If payment authorization fails, display an error message

System Errors:

If any backend services (inventory, payment gateway, database) are unavailable, display a message indicating the issue.

7.3.2 FR-12: Subscription Length Selection

- A. **Introduction/Functionality:** Provide predefined subscription lengths for their meal plan.
- B. **Traced:** UC-04: Create Meal Service Subscription
- C. **Inputs:** Meal plan
- D. **Processing:** The system displays available subscription lengths based on selected meal plans and validates user selection.
- E. **Outputs:** Display subscription length
- F. **Error Handling:** If an invalid length is selected, display an error message prompting the patron to choose a valid option.

7.3.3 FR-13: Delivery Frequency Options

- A. **Introduction/Functionality:** Provides delivery frequency options based on the selected meal plan.
- B. **Traced:** UC-04: Create Meal Service Subscription
- C. **Inputs:** Selected meal plan.
- D. **Processing:** The system presents delivery frequency options based on chosen meal plan's constraints.
- E. **Outputs:** Display of valid delivery frequency options.

- F. **Error Handling:** If no delivery options are available, inform the patron and suggest an alternative meal plan.

7.3.4 FR-14: Input Field Validation

- A. **Introduction/Functionality:** Validates all input fields for correctness and completeness.
- B. **Traced:** UC-04: Create Meal Service Subscription
- C. **Inputs:** Delivery address, payment information, subscription length.
- D. **Processing:** The system checks each field for required information, format, and consistency with selected options.
- E. **Outputs:** Validation results indicating success or specific errors.
- F. **Error Handling:** Highlight erroneous fields with specific messages; prompt the patron to correct issues.

7.3.5 FR-15: Database Management

- A. **Introduction/Functionality:** Manages the creation and updating of subscription records in the database.
- B. **Traced:** UC-04: Create Meal Service Subscription
- C. **Inputs:** Meal plan details, subscription length, delivery frequency, patron information.
- D. **Processing:** The system creates a new record in the subscription database and updates the patron's account.
- E. **Outputs:** Confirmation of record creation and account update.
- F. **Error Handling:** If database operations fail, log the error and notify support staff; display a user-friendly error message.

7.3.6 FR-16: Payment Processing

- A. **Introduction/Functionality:** Processes the payment for the subscription upon successful details confirmation.
- B. **Traced:** UC-04: Create Meal Service Subscription

- C. **Inputs:** Payment information and total subscription cost.
- D. **Processing:** The system initiates a payment request and verifies authorization with the payment gateway.
- E. **Outputs:** Payment confirmation or failure notification.
- F. **Error Handling:** Payment failure, display a message

7.3.7 FR-17: Confirmation Output

- A. **Introduction/Functionality:** Displays a confirmation of the subscription to the patron and sends an email notification.
- B. **Traced:** UC-04: Create Meal Service Subscription
- C. **Inputs:** Subscription details including ID, meal plan, length, and delivery schedule.
- D. **Processing:** The system generates a confirmation message and formats an email with the subscription details.
- E. **Outputs:** On-screen confirmation and email notification sent to the patron.
- F. **Error Handling:** If email sending fails, log the error and inform the user that the confirmation was displayed but not emailed.

7.3.8 FR-18: View Meal Service Subscription

- A. **Introduction/Functionality:** The system displays patrons to view details of their meal service subscriptions.
- B. **Traced:** UC-09
- C. **Inputs:** Patron's account ID, Subscription ID
- D. **Processing:**
 - Step 1:** Retrieve active and inactive subscriptions associated with PatronID.
- E. **Outputs:** A list of subscriptions
- F. **Error Handling:**
 - No Subscriptions Found:**
 - If no subscriptions are found, display a message.

System Errors:

If the system cannot retrieve data due to system error, display an error message

7.3.9 FR-19: Subscription Summary Display

- A. **Introduction/Functionality:** Provides a summarized view of all active and inactive subscriptions for the patron, enabling them to see key details at a glance.
- B. **Traced:** UC-09 View Meal Service Subscription
- C. **Inputs:** Patron's account I
- D. **Processing:**
 - Step 1:** Retrieve all subscriptions associated with the patron's account.
 - Step 2:** Display a summary for each subscription, including:
 - Meal plan name
 - Subscription length
 - Delivery frequency
 - Status (active or inactive)
- E. **Outputs:** A list view of all subscriptions with summarized details for each.
- F. **Error Handling:** If no subscriptions are found, display a message

7.3.10 FR-20: Delete Meal Service Subscription

- A. **Introduction/Functionality:** System can delete an existing meal service subscription.
- B. **Traced:** UC-10
- C. **Inputs:** Patron's account ID, Subscription ID
- D. **Processing:**
 - Step 1:** Verify the patron's account ID and the provided subscription ID
 - Step 2:** Provide overview of the subscription to be canceled.
 - Step 3:** Update the subscription status to "VOID"
 - Step 4:** Trigger a cancellation confirmation message
- E. **Outputs:**

A confirmation message displayed on screen.

An email notification sent confirming the cancellation.

F. Error Handling:

Invalid Subscription ID:

If the subscription ID is not found display an error message.

Ineligible for Cancellation:

If the subscription is ineligible for deletion display message

System Errors:

If the system encounters issue during the deletion proces display error message

7.3.11 FR-21: Validate Subscription Eligibility for Deletion

A. **Introduction/Functionality:** Ensures that the subscription selected for deletion is eligible, such as being in an active or renewable status.

B. **Traced:** UC-10

C. **Inputs:** Patron's account ID, Subscription ID

D. **Processing:**

Step 1: Verify that the subscription ID is valid and associated with the patron's account.

Step 2: Check the subscription status to confirm that it can be deleted (e.g., active or renewable).

E. **Outputs:** Validation success or an error message if the subscription cannot be deleted

F. **Error Handling:**

If the subscription ID is invalid, display an error accessed

If the subscription status is ineligible for deletion, display error

7.3.12 FR-22: Update Subscription Status to Void

A. **Introduction/Functionality:** Updates the subscription status in the database to cancel future orders

- B. **Traced:** UC-10 Delete
- C. **Inputs:** Patron's account ID, Subscription ID
- D. **Processing:** Change the subscription's status to "VOID" in the subscription database.
- E. **Outputs:** Successful update of subscription status to "VOID"
- F. **Error Handling:** If the database update fails, display an error message

7.3.13 FR-23: Generate Void Confirmation Notification

- A. **Introduction/Functionality:** Provides confirmation of cancellation to the patron both on-screen and through email.
- B. **Traced:** UC-10 Delete
- C. **Inputs:** Subscription details (ID, meal plan, final service date)
- D. **Processing:**
 - Step 1:** Display an on-screen confirmation message that the subscription has been canceled, including relevant details.
 - Step 2:** Send an email notification to the patron summarizing the canceled subscription details.
- E. **Outputs:** On-screen confirmation and email notification sent to the patron.
- F. **Error Handling:** If the email notification fails, display the error

7.3.14 FR-24: Modify Meal Service Subscription

- A. **Introduction/Functionality** The system enables patrons to modify details of their existing meal service subscription
- B. **Traced:** UC-11
- C. **Inputs:** Subscription ID, Updated meal plan (optional), Updated subscription length (optional), Updated delivery frequency (optional), Updated delivery address (optional), Patron's account ID (optional), Payment information (optional),
- D. **Processing**
 - Step 1:** Retrieve the existing subscription details from subscription ID.

Step 2: Display current subscription details

Step 3: Validate modified fields

Step 4: Update the subscription in the database with the new details.

Step 5: Calculate cost and send payment request.

Step 6: Display a confirmation message

Step 7: Send email with the updated subscription details

E. Outputs:

On-screen confirmation message with updated subscription details

An email notification sent summarizing modifications

Updated subscription details stored in the database

F. Error Handling

Input Validation Errors:

Missing field or an updated value incorrect display error message

Payment Processing Errors:

If payment authorization fails display an error message

System Errors:

If any backend services (database, payment gateway) are unavailable, display a message

7.3.15 FR-25: Modify Subscription Length

A. Introduction/Functionality

B. Traced: UC-11

C. Inputs: Subscription ID, predefined subscription length options

D. Processing:

Step 1: Retrieve the available subscription lengths and display them based on the current meal plan.

Step 2: Validate the new subscription length

E. Outputs: Confirmation of the selected subscription length

F. Error Handling: For invalid length selected, display an error message

7.3.16 FR-26: Modify Delivery Frequency

A. **Introduction/Functionality**

B. **Traced:** UC-11

C. **Inputs:** Subscription ID, selected meal plan

D. **Processing:**

Step 1: Display available delivery frequency options compatible with the chosen meal plan.

Step 2: Validate the selected delivery frequency.

E. **Output:** Confirmation of the updated delivery frequency stored for processing.

F. **Error Handling:** If no delivery frequency options are available, display error message

7.3.17 FR-27: Address and Contact Information Update

A. **Introduction/Functionality:** System to update the delivery address and contact details of subscription.

B. **Traced:** UC-11

C. **Inputs:** Subscription ID, new delivery address, and contact information

D. **Processing:**

Step 1: Validate delivery area requirements.

Step 2: Validate delivery area requirements.

Step 3: Update the contact information if modified

E. **Outputs:** Confirmation of the updated address and contact information.

F. **Error Handling:** If the address format is invalid, highlight field and display error message

7.3.18 FR-28: Modify Meal Plan Selection

A. **Introduction/Functionality:** Enables patrons to switch to a different meal plan within their existing subscription.

B. **Traced:** UC-11

C. **Inputs:** Subscription ID, available meal plan options

D. **Processing:**

Step 1: Display meal plans available for the current subscription.

Step 2: Validate the selected meal plan based on the patron's preferences and system constraints.

E. **Outputs:** Confirmation of the new meal plan choice.

F. **Error Handling:** If the selected meal plan is unavailable, display an error message and prompt the patron to select a different option.

7.3.19 FR-29: Validate and Confirm Modification Inputs

A. **Introduction/Functionality:** Checks that all updated fields are valid and complete.

B. **Traced:** UC-11

C. **Inputs:** Updated field value(s)

D. **Processing:** Validate each field based on format and availability

E. **Outputs:** Validation success or error messages for incorrect fields.

F. **Error Handling:** If any field is invalid, highlight the field and display error message

7.3.20 FR-30: Database Update for Modified Subscription

A. **Introduction/Functionality:** Manages updating the subscription record in the database with the new details provided by the patron.

B. **Traced:** UC-11

C. **Inputs:** Updated subscription details

D. **Processing:** Update the subscription record in the database with all modified details.

E. **Outputs:** Confirmation of the successful update of the subscription record.

- F. **Error Handling:** If database operations fail, log the error and notify support staff; display a user-friendly error message to the patron.

7.3.21 FR-31: Payment Adjustment Processing

- A. **Introduction/Functionality:** Processes payment adjustments if the modifications result in a price change for the subscription.
- B. **Traced:** UC-11
- C. **Inputs:** Payment information, updated total subscription cost
- D. **Processing:**
 - Step 1:** Calculate any adjustments to the total cost based on the modifications.
 - Step 2:** Initiate a payment authorization request for additional charges, if necessary.
- E. **Outputs:** Payment confirmation or notification of price adjustment completion.
- F. **Error Handling:** If payment authorization fails, display a message indicating the issue and offer retry options.

7.3.22 FR-32: Modification Confirmation Output

- A. **Introduction/Functionality:** Displays a summary of the updated subscription to the patron and sends an email notification with the modified details.
- B. **Traced:** UC-11
- C. **Inputs:** Subscription details, including ID, updated meal plan, length, delivery schedule, and cost (if adjusted)
- D. **Processing:**
 - Step 1:** Generate a confirmation message with the updated subscription details.
 - Step 2:** Format and send an email notification with the modified subscription information.
- E. **Outputs:** On-screen confirmation and email notification sent to the patron.
- F. **Error Handling:** If the email fails to send, log the error

7.4 FE-4: Register various meal payment methods.

7.4.1 FR-33: Registering for payment by payroll deduction

- A. **Introduction/Functionality:** The system shall allow Patrons to register their payroll deduction as a payment method
- B. **Traced:** UC-2
- C. **Inputs:** Prompt the Patron to enter their employee ID. Validate the employee ID against the payroll database. Confirm successful registration or display an error if the ID is not found.
- D. **Processing:**
 - Step 1:** Confirmation message upon successful registration
 - Step 2:** Error message if registration fails due to missing employee ID
- E. **Outputs:** Show the cafeteria menu with the item name and prices.
- F. **Error Handling:** If the employee ID is not found, display: “Failed to register for payroll deduction. Employee ID not found.”

7.4.2 FR-34: Registering for payment by credit card

- A. **Introduction/Functionality:** The system shall allow Patrons to register a credit card as a payment method.
- B. **Traced:** UC-2
- C. **Inputs:**
 - Credit card number
 - Expiry date
 - CVV
- D. **Processing:**
 - Step 1:** Allow the Patron to enter their credit card information.
 - Step 2:** Validate the credit card details using a payment gateway.
 - Step 3:** Confirm registration or display an error if validation fails.
- E. **Outputs:**
 - Confirmation message upon successful registration

- Error message if credit card validation fails.
- F. **Error Handling:** If credit card validation fails, display: "Failed to register credit card payment."

7.4.3 FR-35: Registering for payment by debit card

- A. **Introduction/Functionality:** The system shall allow Patrons to register a debit card as a payment method.
- B. **Traced:** UC-2
- C. **Inputs:**
- Debit card number
 - Expiry date
 - CVV
- D. **Processing:**
- Step 1:** Allow the Patron to enter their debit card information.
- Step 2:** Validate the debit card details using a payment gateway.
- Step 3:** Confirm registration or display an error if validation fails.
- E. **Outputs:**
- Confirmation message upon successful registration
 - Error message if credit card validation fails
- F. **Error Handling:** If debit card validation fails, display: "Failed to register credit card payment."

7.4.4 FR-36: Registering for payment by Digital Wallet

- A. **Introduction/Functionality:** The COS shall allow Patrons to register a digital wallet as a payment method.
- B. **Traced:** UC-2
- C. **Inputs:**
- Digital wallet service name.

- Wallet account credentials.

D. Processing:

Step 1: Allow the Patron to enter their digital wallet credentials.

Step 2: Authenticate the credentials with the digital wallet provider.

Step 3: Confirm registration or display an error if authentication fails.

E. Outputs:

- Confirmation message upon successful registration
- Error message if digital wallet authentication fails

F. Error Handling: If authentication fails, display “{digital wallet service name} fails to authenticate.”

7.5 FE-5: Manage cafeteria menus (create, view, modify, delete).

7.5.1 FR-37: Create cafeteria menus

A. Introduction/Functionality: The system shall allow cafeteria staff to create new menu items.

B. Traced: UC-5

C. Inputs:

- Admin Credentials.
- Menu item name/ID, price.

D. Processing:

Step 1: Admin logs into the system.

Step 2: Allow staff to enter details for the new menu items.

Step 3: Validate all inputs.

Step 4: Save the new menu items.

Step 5: Confirm new menu creation.

E. Outputs: Confirmation message: “New menu created successfully.”

F. Error Handling:

- If required fields are missing, display a message
- If unable to save, display a message

7.5.2 FR-38: View cafeteria menu

- A. **Introduction/Functionality:** The system shall allow cafeteria staff to view all existing menu items.
- B. **Traced:** UC-6
- C. **Inputs:** Admin Credentials.
- D. **Processing:**
 - Step 1:** Admin logs into the system.
 - Step 2:** Retrieve all menu items.
 - Step 3:** Display the list of menu items.
- E. **Outputs:** List of all current menu items with details (name, price, availability.)
- F. **Error Handling:** If unable to fetch the menu, display a message “Unable to retrieve menu at this time. Please try again.”

7.5.3 FR-39: Modify cafeteria menus

- A. **Introduction/Functionality:** The system shall allow cafeteria staff to modify all existing menu
- B. **Traced:** UC-7
- C. **Inputs:**
 - Admin Credentials.
 - Menu item name/ID
 - Update details (name, ingredients, price, availability.)
- D. **Processing:**
 - Step 1:** Admin logs into the system.
 - Step 2:** Allow staff to edit details of menu items.
 - Step 3:** Validate all updates.
 - Step 4:** Save all the changes.
- E. **Outputs: Confirmation message:** “ Menu item updated successfully.”
- F. **Error Handling:** If unable to save changes, display a message “Failed to update menu items.”

7.5.4 FR-40: Delete cafeteria menus

A. **Introduction/Functionality:** The system shall allow cafeteria staff to delete old or outdated menus from the system.

B. **Traced:** UC-8

C. **Inputs:**

- Menu Selection
- Admin Credentials

D. **Processing:**

Step 1: Admin logs into the system

Step 2: Admin selects the “Delete menu” option from the system

Step 3: The system confirms the delete request with the Admin

Step 4: Admin accepts and confirms the deletion

Step 5: The system will delete the menu and output a message to confirm that the action is completed

E. **Outputs:**

- Deleted action conforms to the Admin/User
- Updated menu list without the old menu

F. **Error Handling:**

- If the menu can’t be deleted: “Unable to delete menu at this time. Please try again later.”
- If the Admin cancels the deletion, the system will not change the existing menu selections.

7.6 FE-6: Order custom meals that are not on the menu.

7.6.1 FR-41: Order custom meals not on the menu

A. **Introduction/Functionality:** The system shall allow the patrons to order custom meals that are not available on the menu.

B. **Traced:** UC-12

C. **Inputs:**

- List of custom ingredients.
- Special instructions.
- Selected date for order.
- Pickup or delivery option.

D. **Processing:**

Step 1: Allow the patrons to enter custom ingredients and meal preferences.

Step 2: COS to check the availability of specified ingredients.

Step 3: If all ingredients are available, display a confirmation summary with custom meal details.

Step 4: Allow the patrons to select a date and choose the options for pickup or delivery.

Step 5: Confirm the order and update the system with the custom meal request.

E. **Outputs:**

- Confirmation message displaying custom meal details.
- Notification sent to the kitchen staff.

F. **Error Handling:**

- If any of the ingredients are unavailable, display a message “Some of the ingredients are not available. Please adjust your custom meal selection.”
- If selected date is in the past, display a message “Please choose a current or future date for your custom meal order.”

7.7 FE-7: Request and collect feedback from customers.

7.7.1 FR-42: Request and collect customer feedback

A. **Introduction/Functionality:** The system shall automatically request and collect customer feedback after their order has been delivered or picked up.

B. **Traced:** UC-16

C. **Inputs:**

- Order ID
- Customer ID
- Delivery/pickup confirmation status
- Customer rating
- Written feedback
- Service type (delivery or pickup)
- Menu items ordered

D. **Processing:**

Step 1: Confirm order pickup or delivery completion

Step 2: Wait 15 minutes after order completion

Step 3: Send feedback requests to customers through email

Step 4: Send feedback requests to customers through text

Step 5: Receive and check customer feedback

Step 6: Save feedback in records

Step 7: Alert manager about new feedback

Step 8: Update customer's feedback history

E. **Outputs:**

- Feedback request notification to the customer
- Confirmation message that feedback was received
- Feedback report to management
- Updated customer feedback history in the database

F. **Error Handling:**

- Submission Errors: If feedback submission fails, display "Unable to submit feedback now. Please try again later."
- Authentication Errors: If the patron is not logged in or the session has expired, prompt the patron to log in before viewing meal subscriptions.

7.8 FE-8: Contact support

7.8.1 FR-43: Customer Support

A. **Introduction/Functionality:** The system shall provide customers with direct phone access to support staff and maintain records of support calls.

B. **Traced:** UC-17: Contact Support

C. **Inputs:**

- Customer phone number
- Customer ID
- Order ID
- Call timestamp
- Cafeteria manager ID
- Support case reason

D. **Processing:**

Step 1: The system displays the support phone number to the customer

Step 2: System routes call to the available cafeteria manager

Step 3: The system creates a support case record for the call

Step 4: System logs call details and duration

Step 5: System records support resolution in the database

Step 6: System sends confirmation email of call resolution

Step 7: System updates customer support history

E. **Outputs:**

- Support case number
- Call Confirmation
- Summary of call and resolution

- A support case is stored in the database
- Updated customer support history

F. Error Handling:

7.8.2 FR-44: Support Request Form

A. Introduction/Functionality: The system will help customers report problems with their orders and get help from our support team.

B. Traced: UC-17

C. Inputs:

- Order number
- Type of problem
- Description of what went wrong
- How they want us to contact them back

D. Processing:

Step 1: Show the customer our help request form

Step 2: Make sure they fill out all the needed information

Step 3: Create a ticket number for the problem

Step 4: Save their help request in our system

Step 5: Send their request to the cafeteria manager

Step 6: Cafeteria Manager to send response back to customer.

Step 7: Send the customer their ticket number

E. Outputs:

- Message with their support ticket number
- An email confirming their request
- Updates about their problem
- Final message when the problem is fixed

F. Error Handling:

- System Errors: If the system is down, "We are experiencing some technical issues. Please call the support number for immediate assistance."

- Request Error: If the support request doesn't go through, "We are experiencing some technical issues. Please call the support number for immediate assistance."

7.9 FE-9: Track order status.

7.9.1 FR-45: Track order status

A. **Introduction/Functionality:** The system shall enable Patrons to track the status of their current and previous orders.

B. **Traced:** UC-3

C. **Inputs:**

- Order dates
- Order ID
- Delivery/Pickup type.

D. **Processing:**

Step 1: Display the order history table for the Patron, sorted by date in descending order.

Step 2: Retrieve and display the status of each order (Accepted, In Progress, Out for Delivery, Ready for Pickup).

Step 3: Allow the Patron to select any previous order to view its details.

Step 4: Update the order status in real-time based on current processing updates.

E. **Outputs:**

- Order history table with status updates
- Detailed order information upon selection
- Real-time updates of order status.

F. **Error Handling:**

- If the order history fails to load, display: "Unable to retrieve order history. Please refresh the page."

- If real-time status updates fail, display: "Order status is temporarily unavailable. Please try again later."

7.10 FE-10: Manage favorite items.

7.10.1 FR-46: Save Favorite items.

- A. **Introduction/Functionality:** The system shall allow the patrons to save specific items from past orders for quick reordering.
- B. **Trace:** UC-13
- C. **Inputs:** Item name/ID
- D. **Processing:**
 - Step 1:** Allow the patron to select an item from their past orders.
 - Step 2:** Check if the selected item already exists in the patron's favorites.
 - Step 3:** Save the item to favorites list.
 - Step 4:** Confirm the successful addition.
- E. **Outputs:**
 - Confirmation message: "Items added to your favorites successfully."
 - If item already exists, display a message "Item already exists in the favorites."
- F. **Error Handling:**
 - If item is already in the favorites, display a message "Item already exists in the favorites."
 - If unable to save, display a message "Unable to save favorites item. Please try again later."

7.10.2 FR-47: View Favorite items.

- A. **Introduction/Functionality:** The system shall allow the patrons to view their favorite items for quick reordering.
- B. **Traced:** UC-14

- C. **Inputs:** The patron logs into their account.
- D. **Processing:**
 - Step 1:** Retrieve the list of favorite items for the patron.
 - Step 2:** Display the list of items with details (name, price, last ordered date.)
- E. **Outputs:**
 - List of favorite items displayed.
 - If the list is empty, display a message “No favorite items saved.”
- F. **Error Handling:** If unable to load favorites, display a message “Unable to load favorites. Please try again later.”

7.10.3 FR-48: Remove Favorite items.

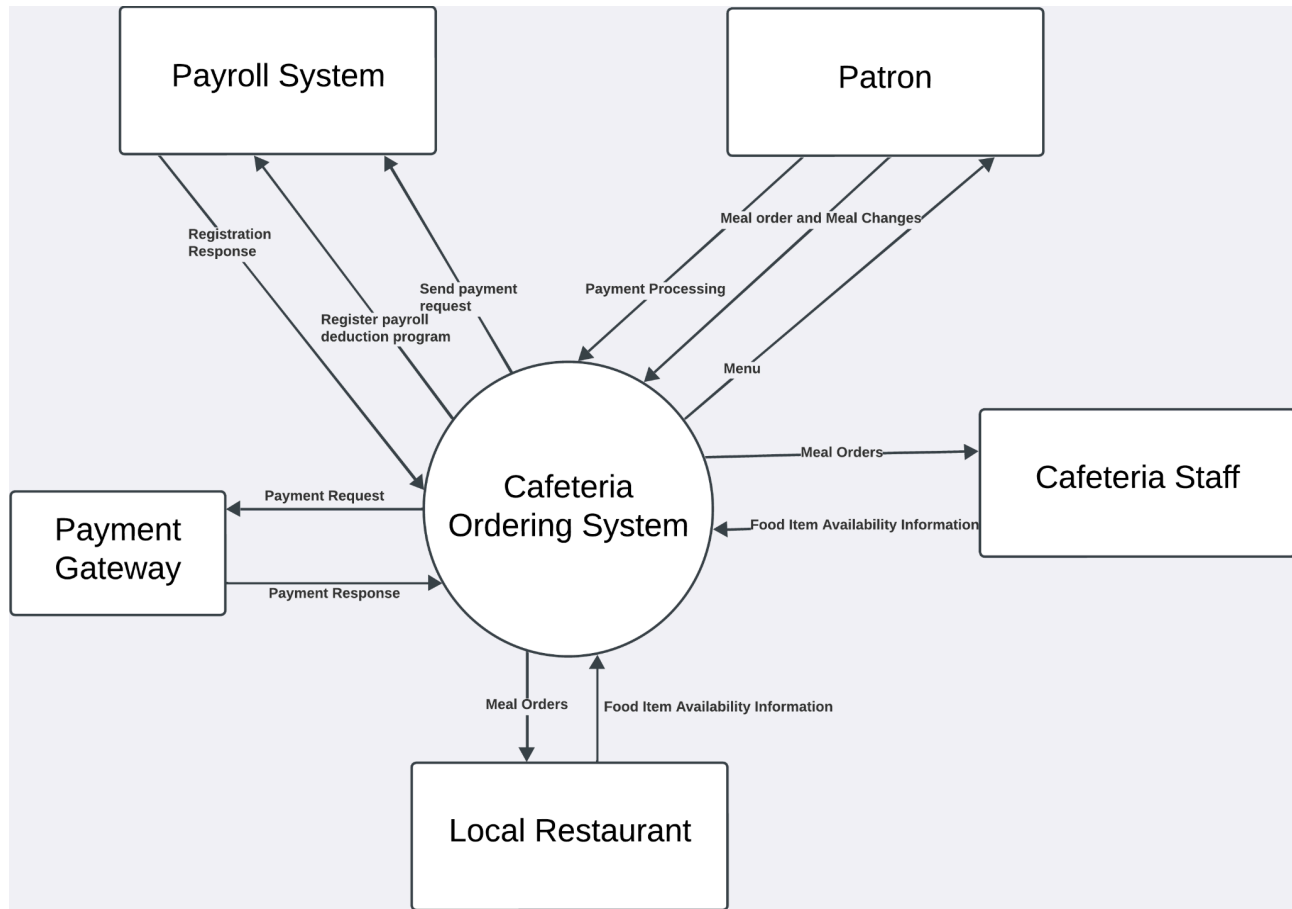
- A. **Introduction/Functionality:** The system shall allow the patrons to remove items from their favorites list.
- B. **Traced:** UC- 15
- C. **Inputs:** Item name/ID
- D. **Processing:**
 - Step 1:** Allow the patron to select an item to delete.
 - Step 2:** Confirm the deletion before proceeding.
 - Step 3:** Remove item from the list.
- E. **Outputs:**
 - Confirmation message: “Item removed from your favorites.”
 - If the list is empty, display a message “No item to remove.”
- F. **Error Handling:**
 - If unable to remove, display a message “Unable to remove favorite item. Please try again later.”

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

	Related FRs
UC-01	FR-1, FR-2, FR-3, FR-4, FR-5, FR-6, FR-7, FR-8, FR-9, FR-10
UC-02	FR-33, FR-34, FR-35, FR-36
UC-03	FR-45
UC-04	FR-11, FR-12, FR-13, FR-14, FR-15, FR-16, FR-17
UC-05	FR-37
UC-06	FR-1, FR-38
UC-07	FR-39
UC-08	FR-40
UC-09	FR-18, FR-19,
UC-10	FR-20, FR-21, FR-22, FR-23
UC-11	FR-24, FR-25, FR-26, FR-27, FR-28, FR-29, FR-30, FR-31, FR-32
UC-12	FR-41
UC-13	FR-46
UC-14	FR-47
UC-15	FR-48
UC-16	FR-42
UC-17	FR-43, FR-44

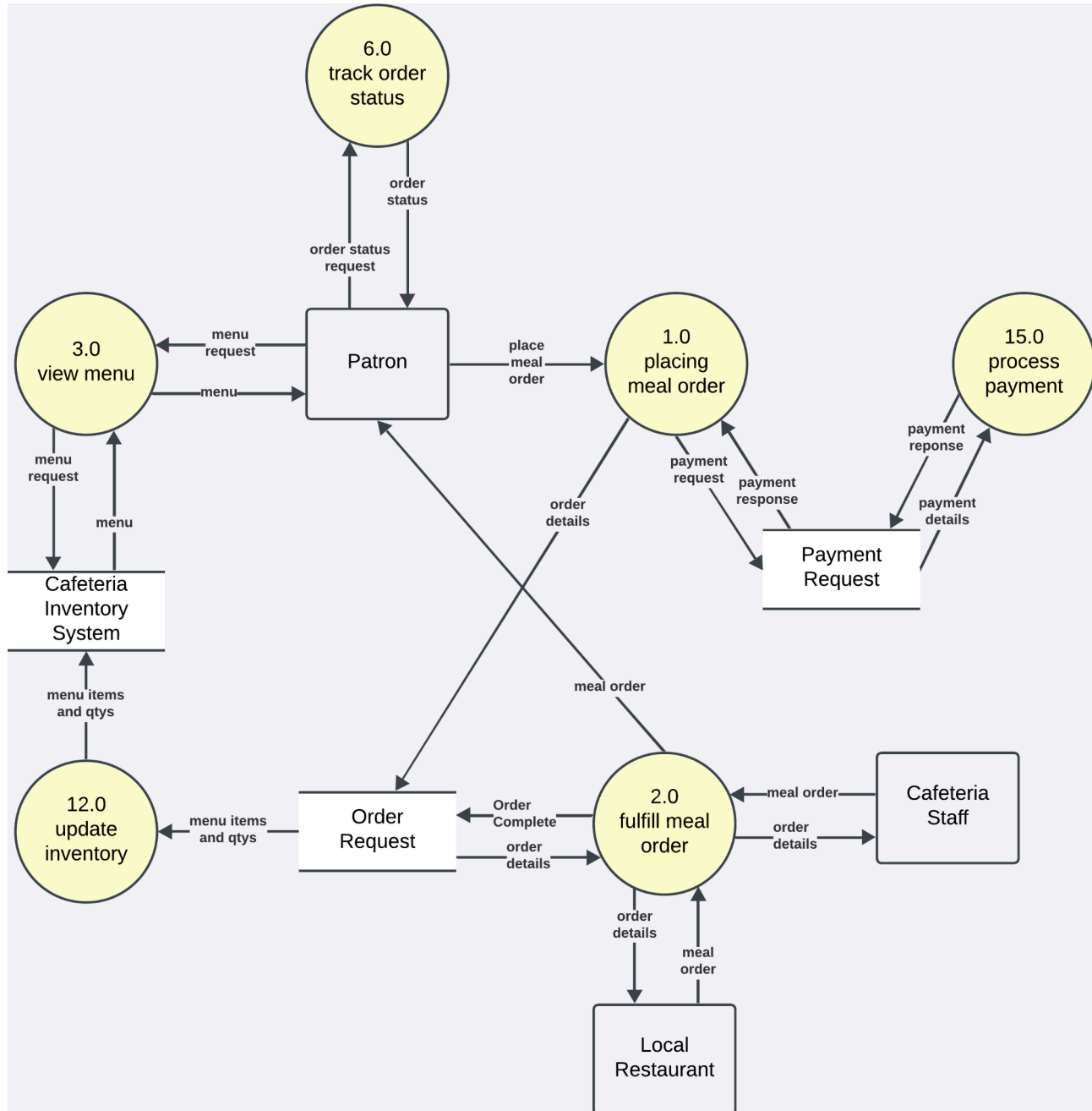
8. Functional Modeling (DFDs)

8.1 Context Diagram (Level 0)

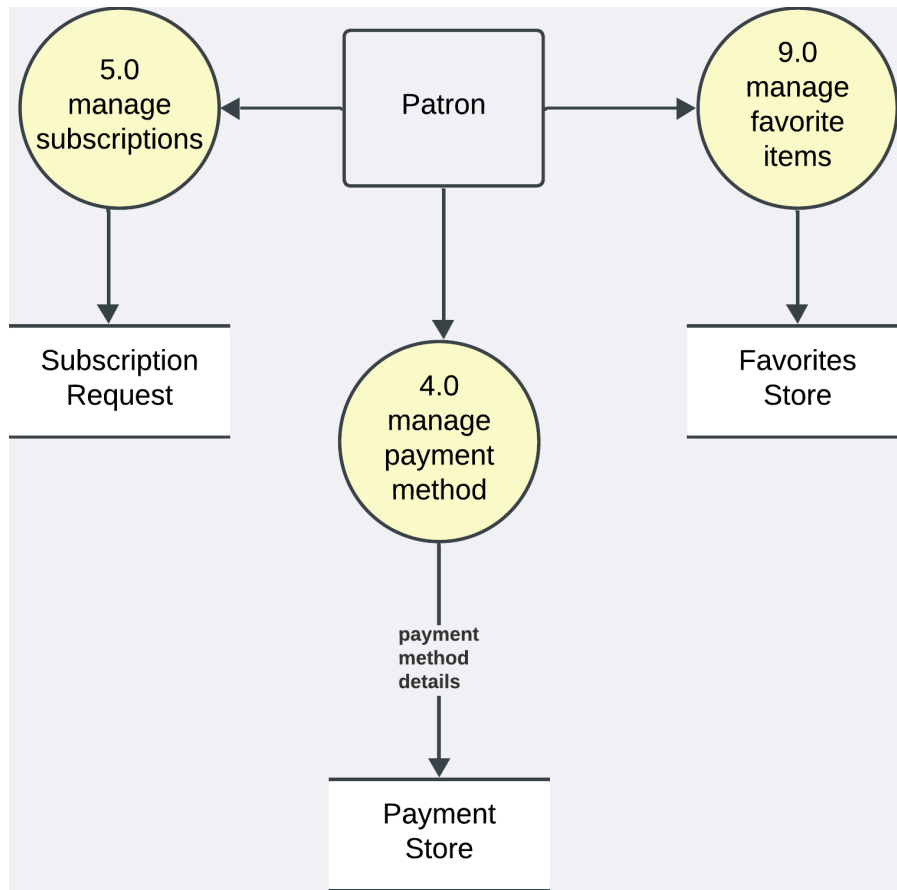


8.2 Level 1

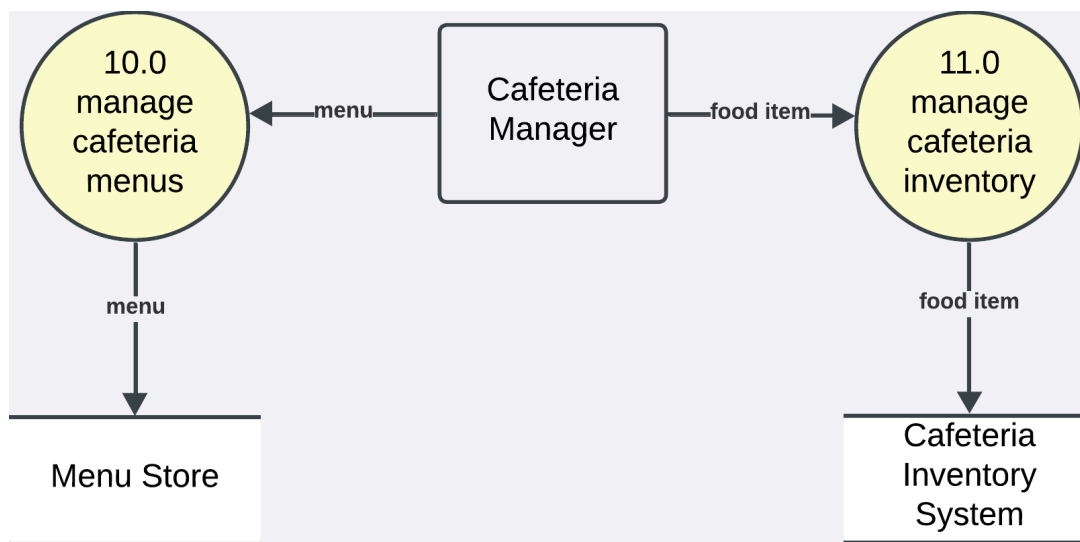
8.2.1 Ordering System, Order Tracking



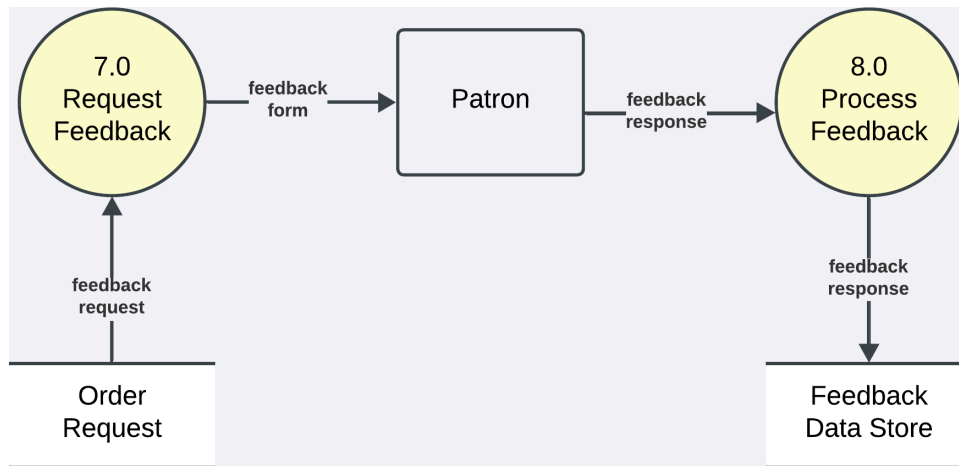
8.2.2 Account Management



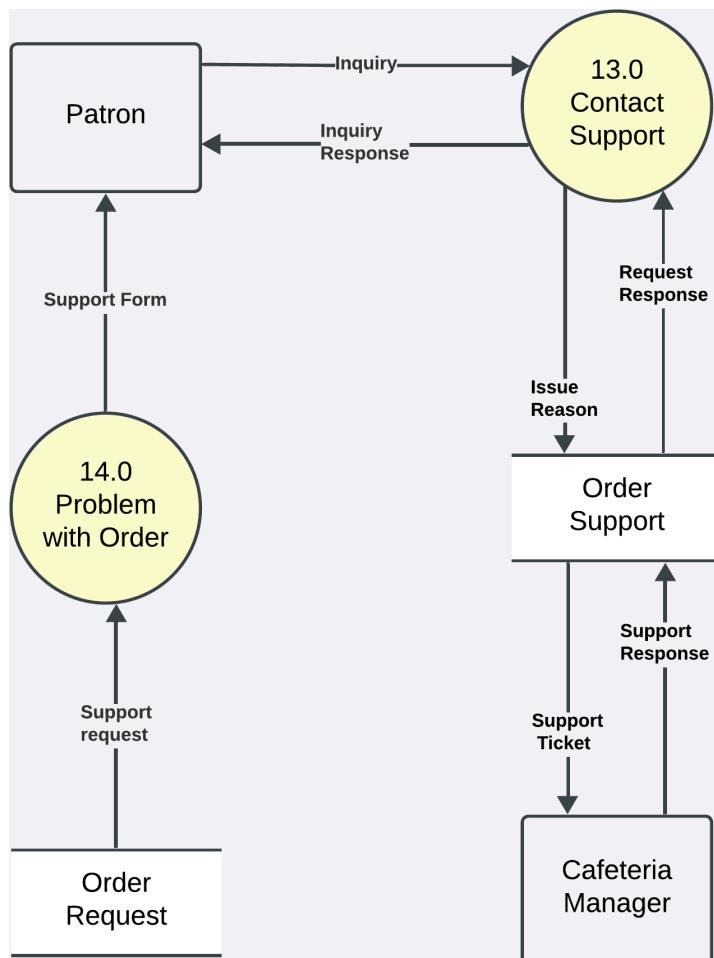
8.2.3 Cafeteria Management



8.2.4 Feedback System

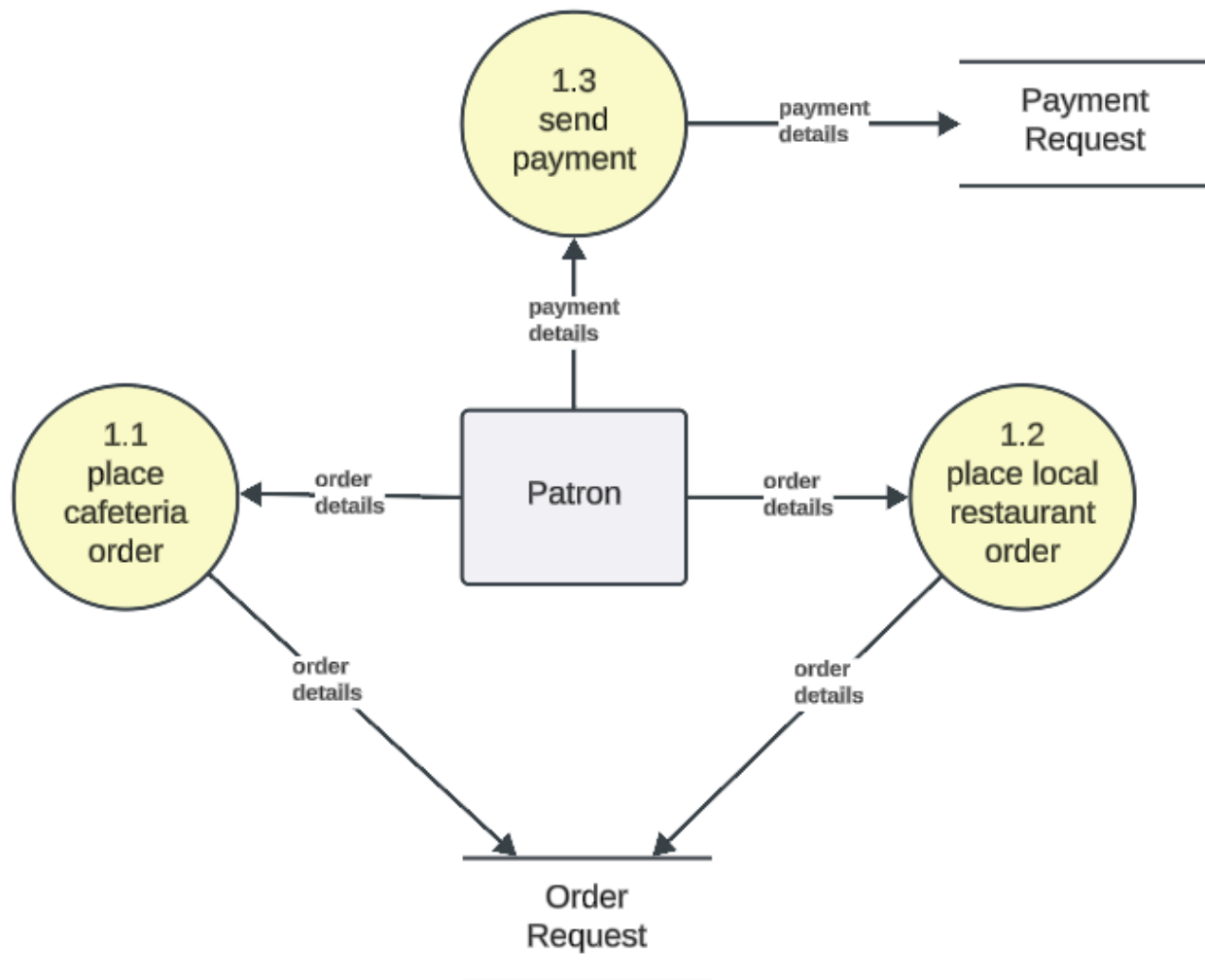


8.2.5 Contact Support

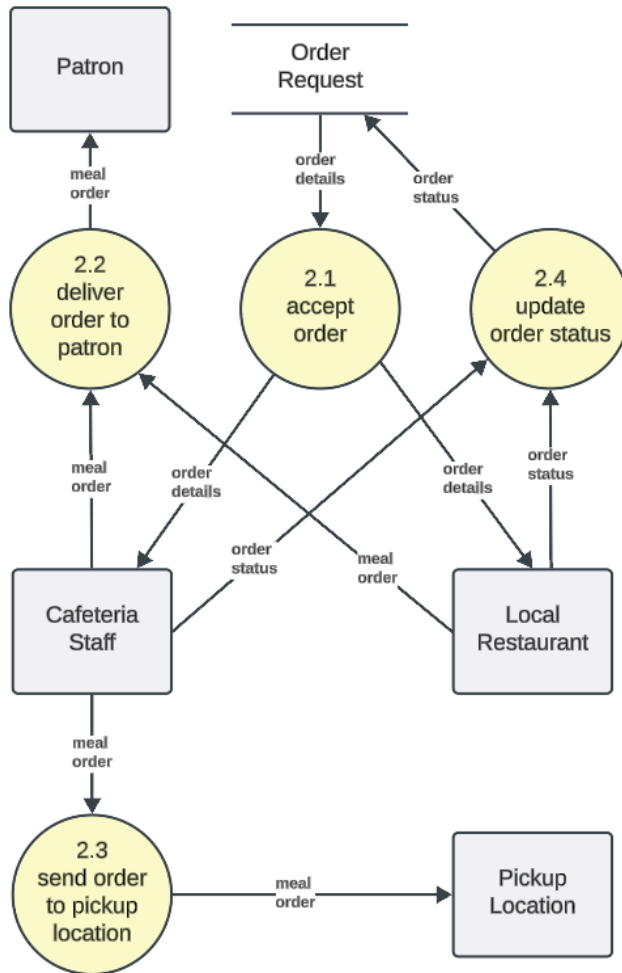


8.3 Level 2

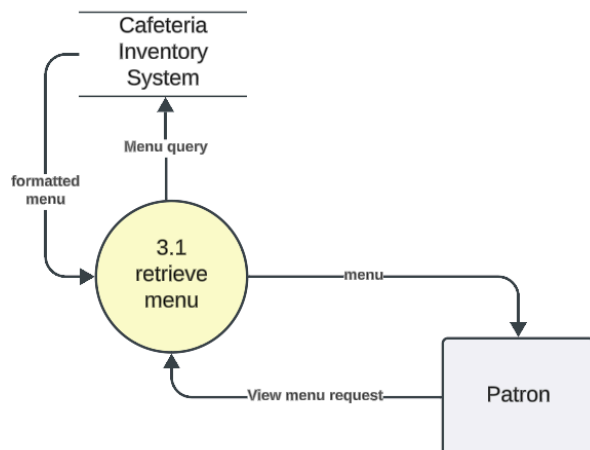
8.3.1 Process 1.0 - Placing menu order



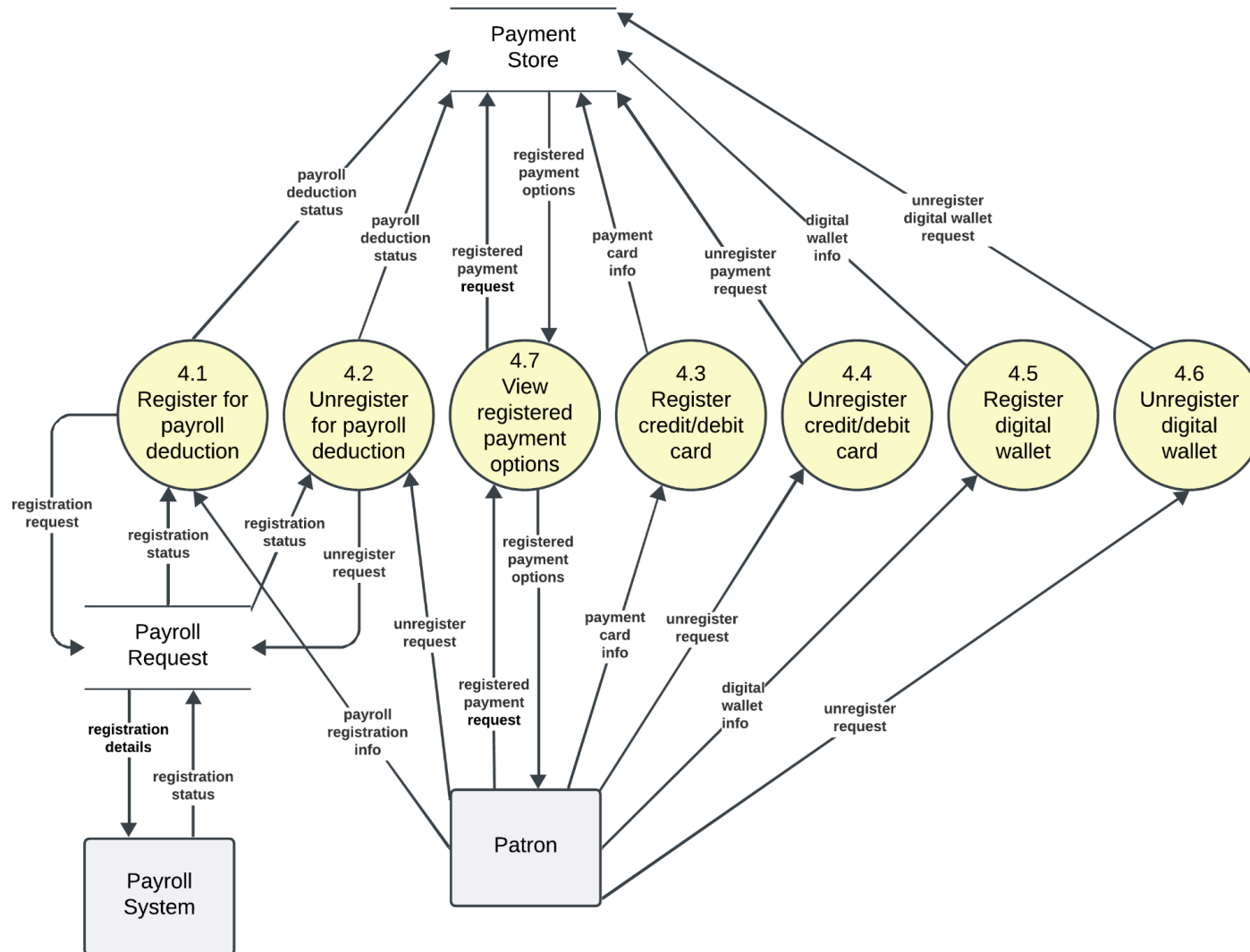
8.3.2 Process 2.0 - Fulfill menu order



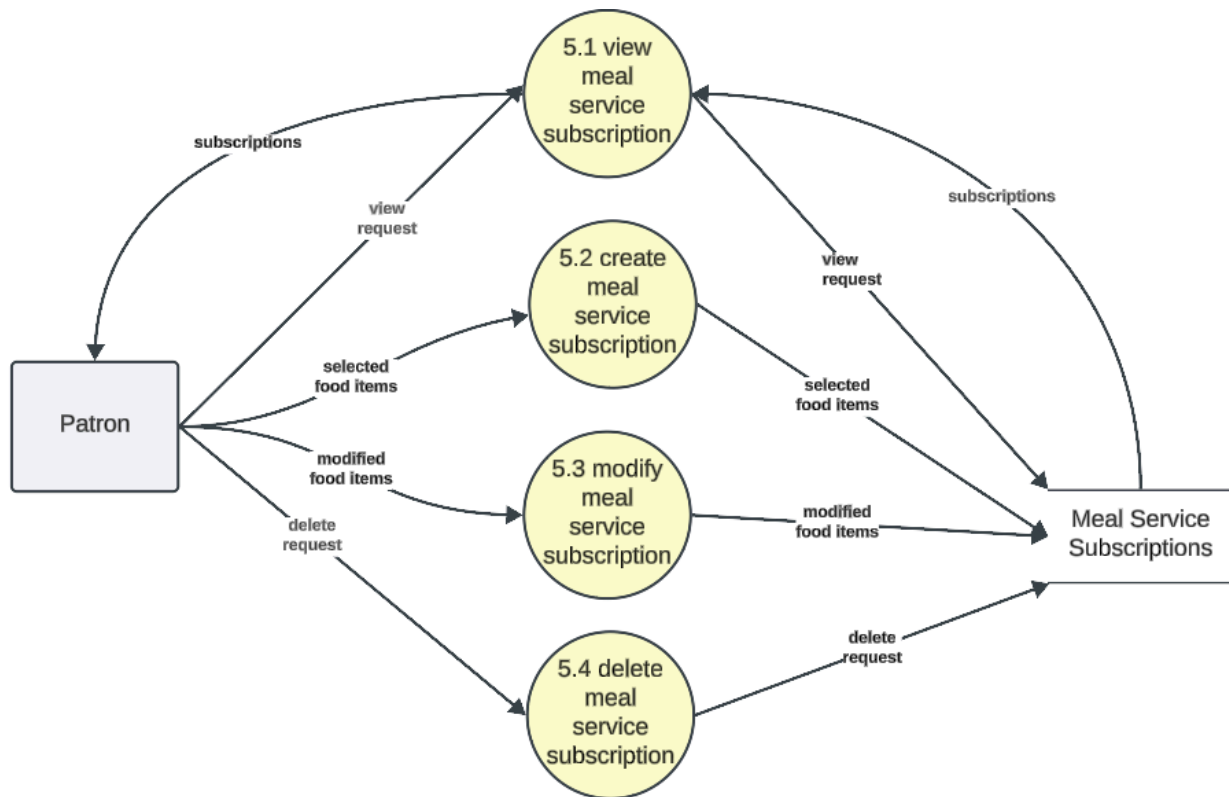
8.3.3 Process 3.0 - View menu



8.3.4 Process 4.0 - Manage payment method

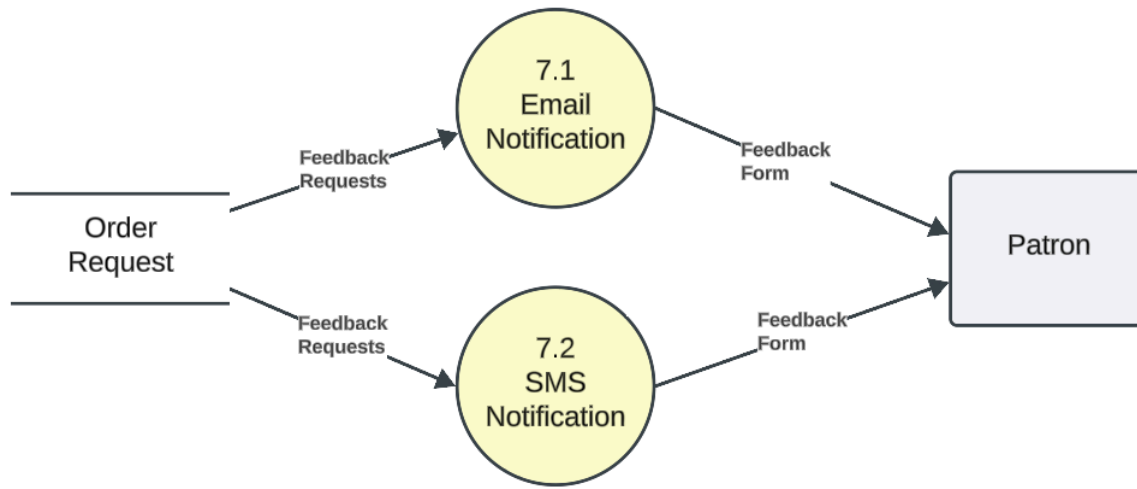
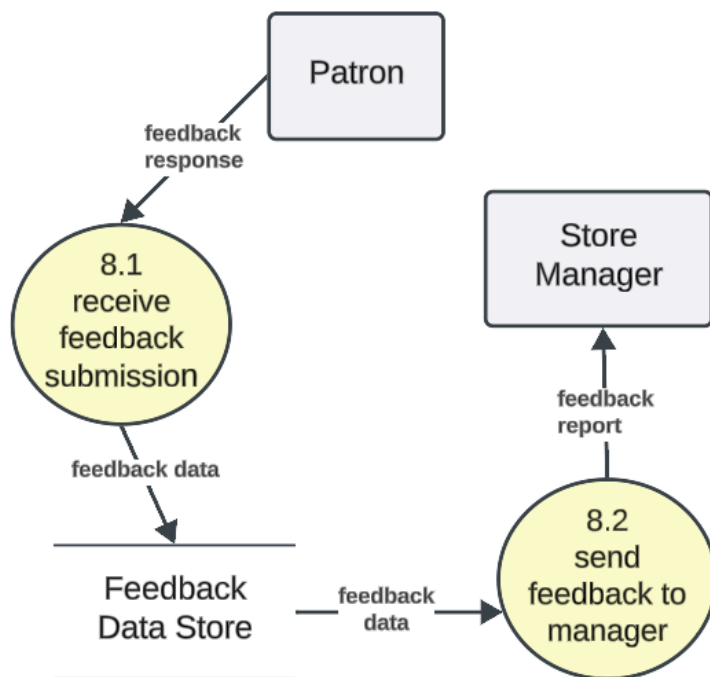


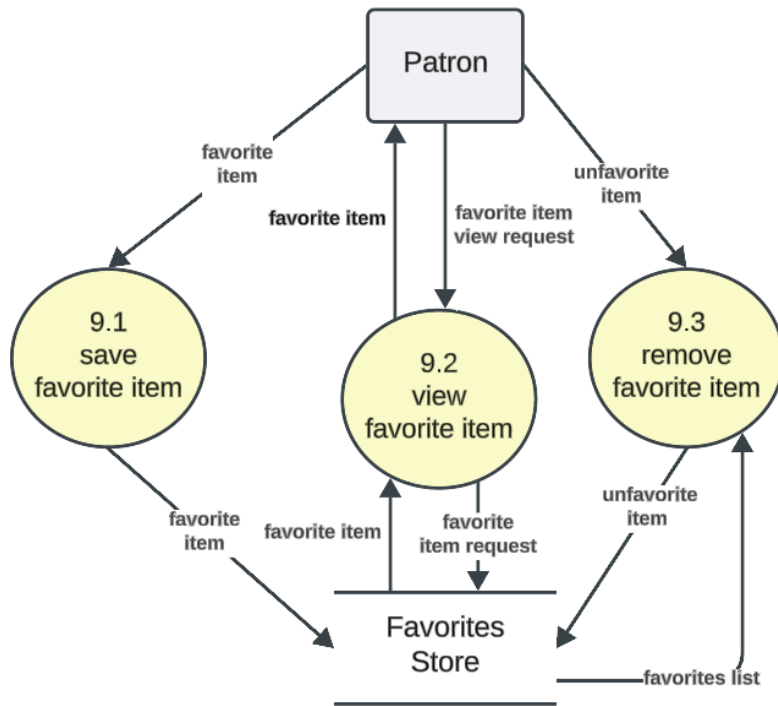
8.3.5 Process 5.0 - Manage subscriptions

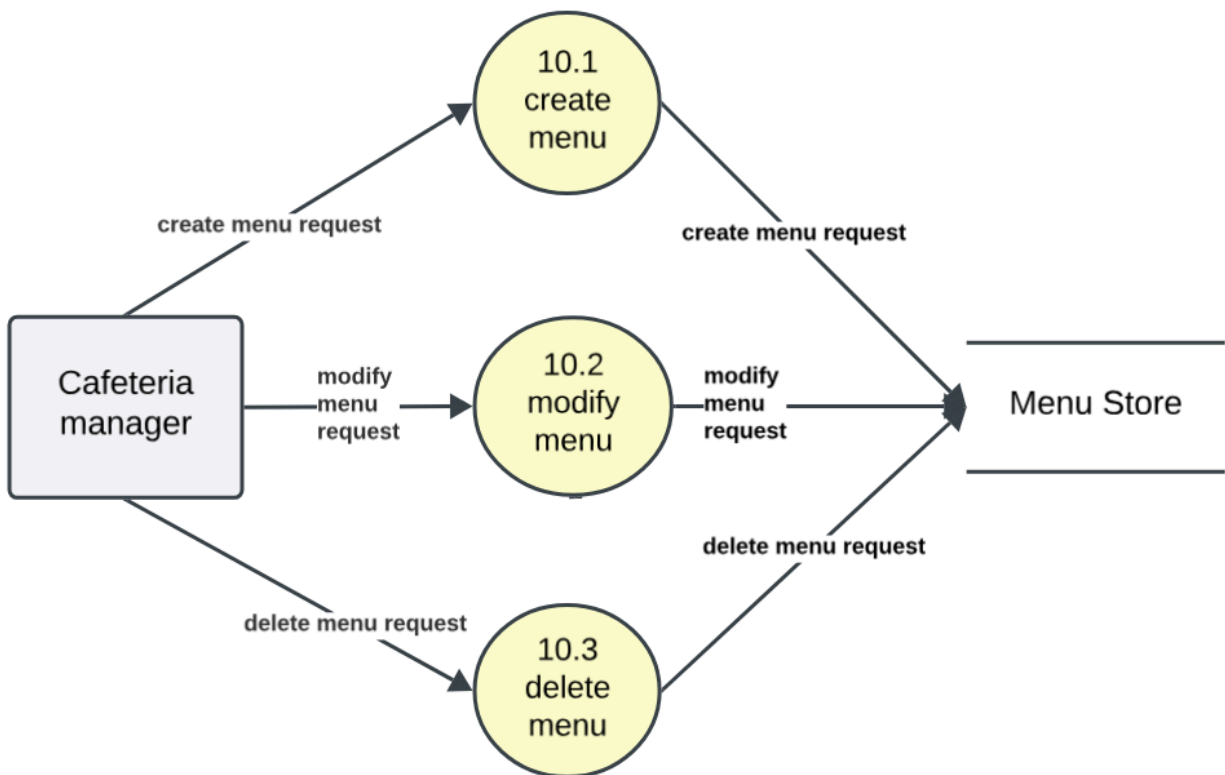


8.3.6 Process 6.0 - Track order status

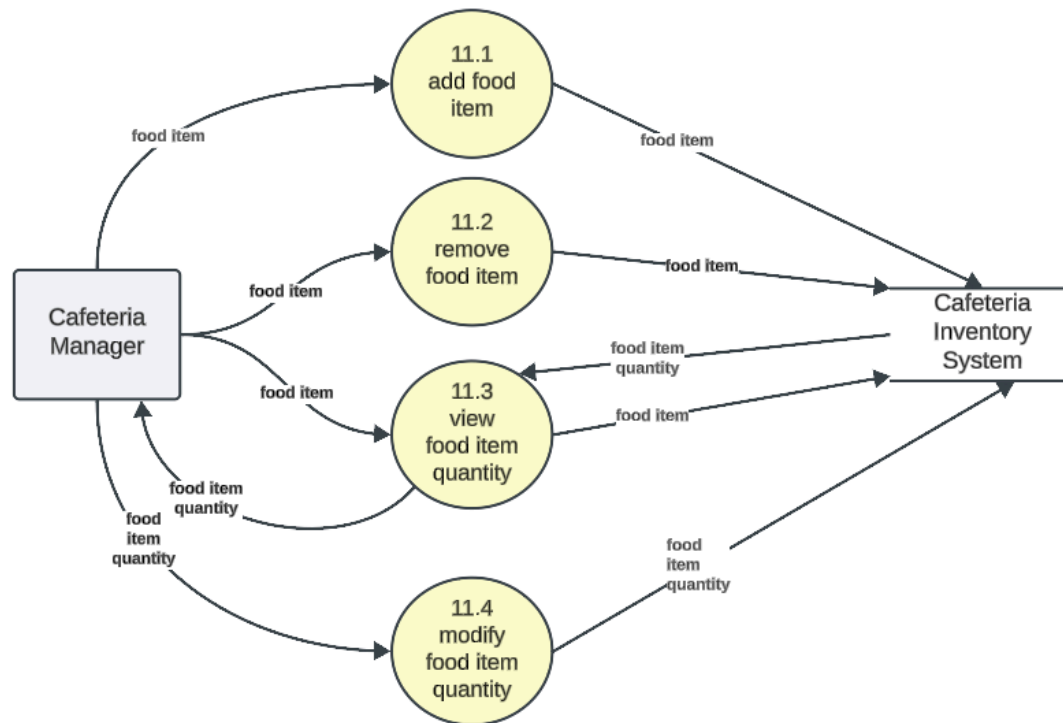


8.3.7 Process 7.0 - Request feedback**8.3.8 Process 8.0 - Process feedback**

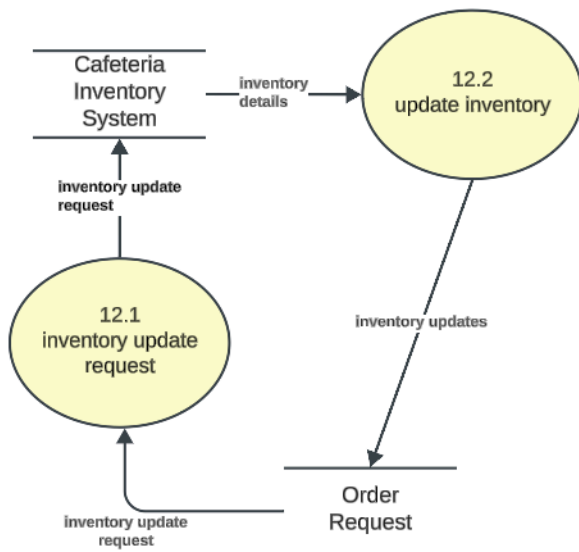
8.3.9 Process 9.0 - Manage favorite items

8.3.10 Process 10.0 - Manage cafeteria menus

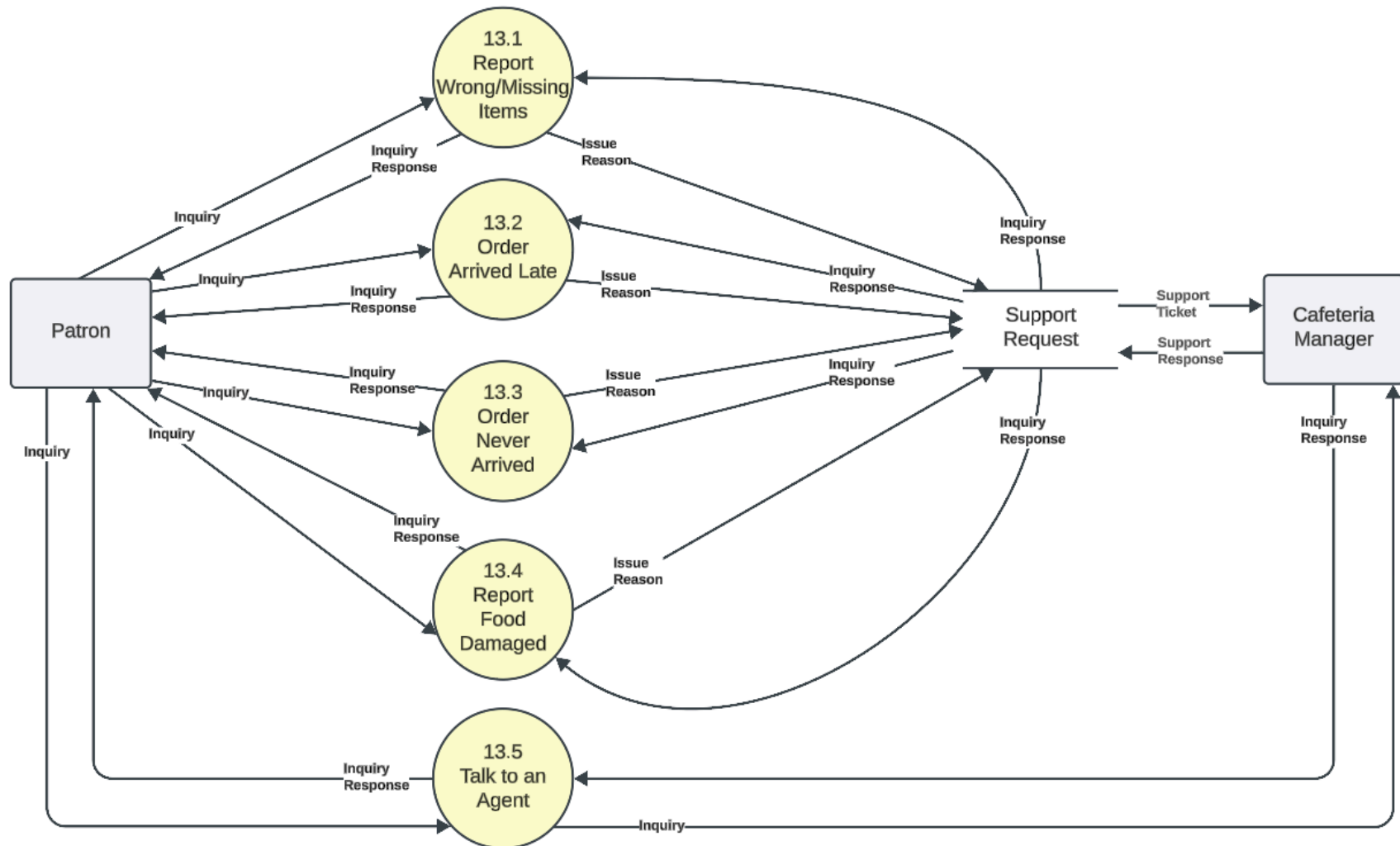
8.3.11 Process 11.0 - Manage cafeteria inventory



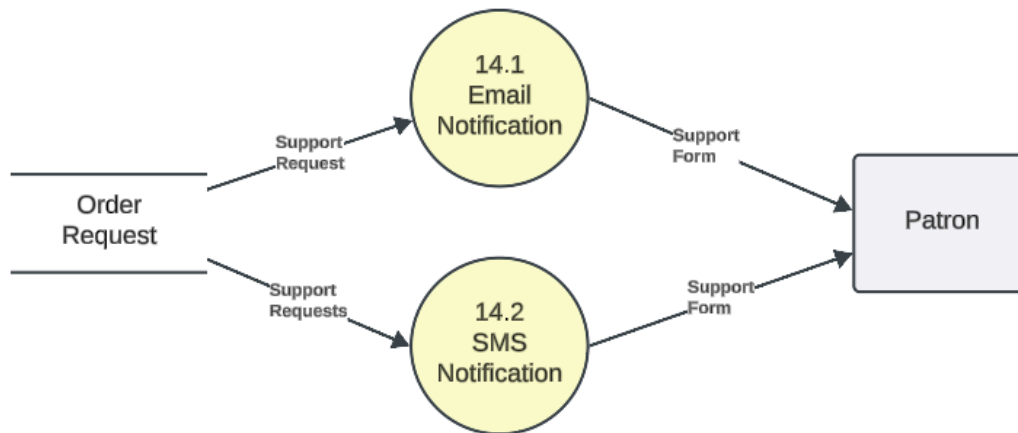
8.3.12 Process 12.0 - Update cafeteria inventory



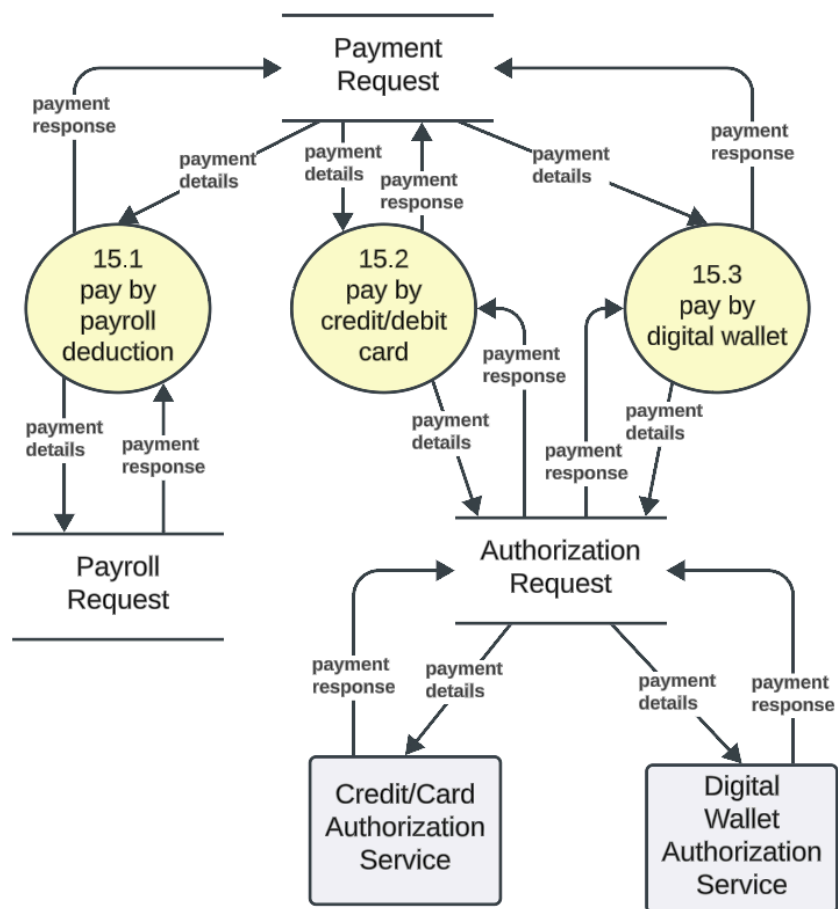
8.3.13 Process 13.0 - Contact support



8.3.14 Process 14.0 - Problem with order

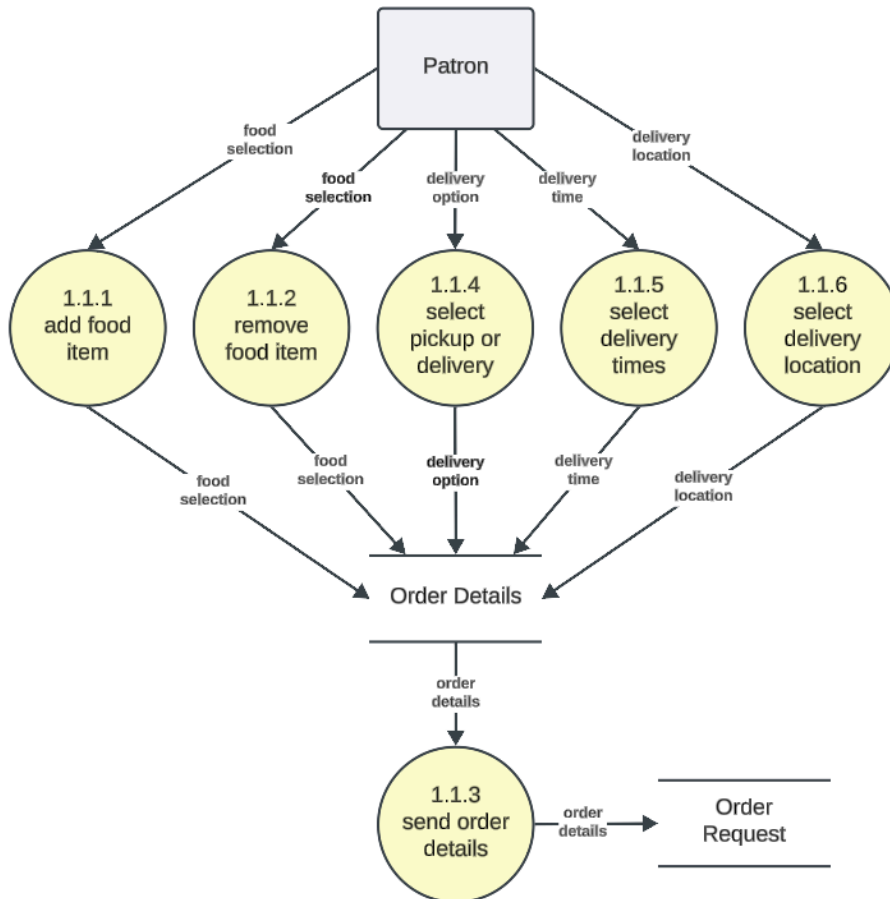


8.3.15 Process 15.0 - Process payment



8.4 Level 3

8.4.1 Process 1.1 - Placing cafeteria order



8.4.2 Process 10.1 - Create cafeteria menu

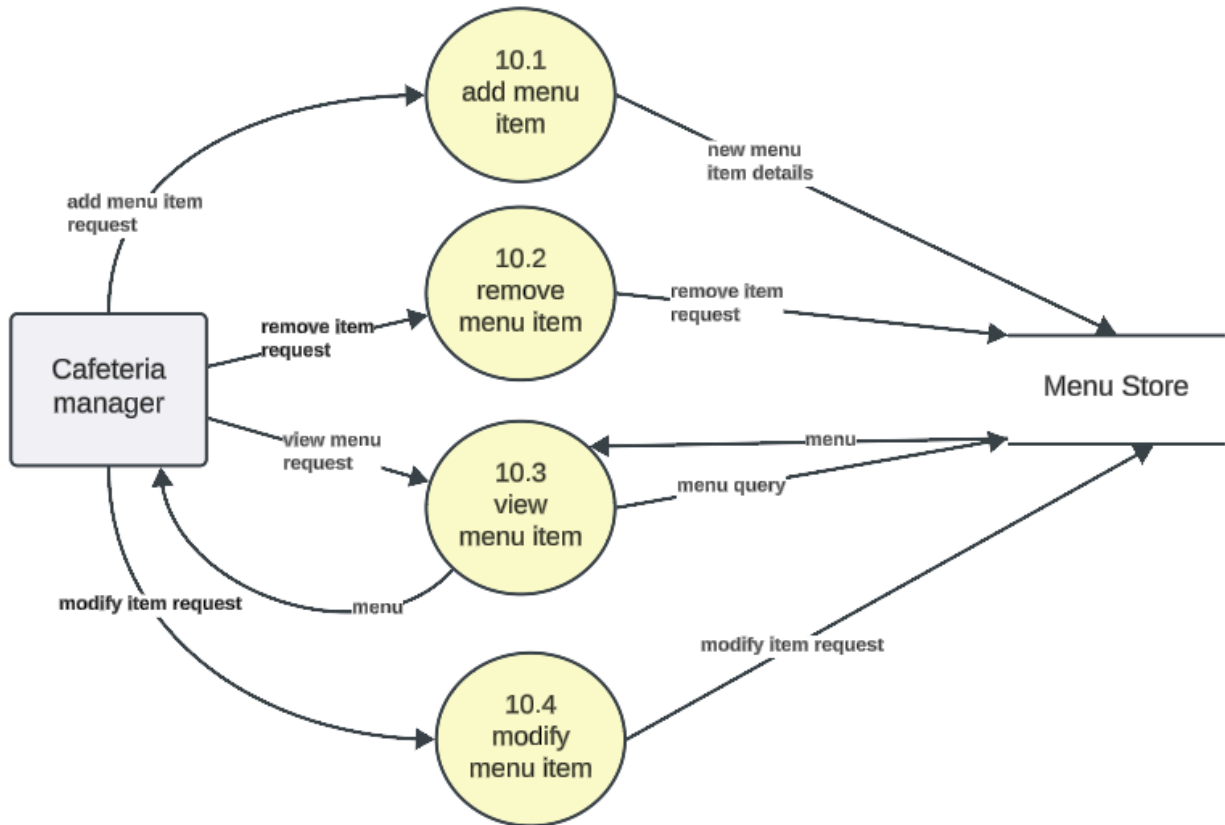


TABLE 3: Traceability Matrix (FRs and DFD Processes)

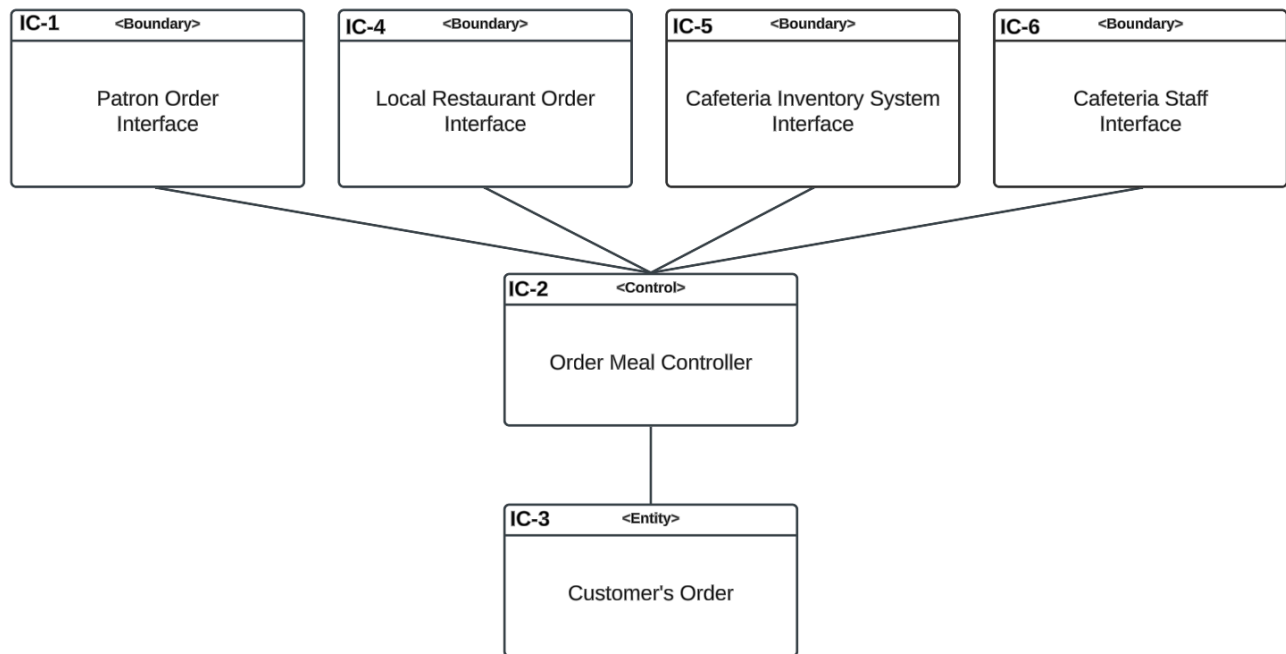
	All Related Processes in DFD
FR-1	3.0
FR-2	1.1.1
FR-3	1.1.2
FR-4	1.1.4
FR-5	1.1.4
FR-6	1.1.5
FR-7	1.1.6
FR-8	1.2
FR-9	1.2
FR-10	1.2
FR-11	5.2
FR-12	5.2, 5.3
FR-13	5.2, 5.3
FR-14	5.2
FR-15	5.2
FR-16	5.2
FR-17	5.2
FR-18	5.1
FR-19	5.1
FR-20	5.4
FR-21	5.4
FR-22	5.3
FR-23	5.4
FR-24	5.3
FR-25	5.3

FR-26	5.3
FR-27	5.3
FR-28	5.3
FR-29	5.3
FR-30	5.3
FR-31	5.3
FR-32	5.3
FR-33	4.1
FR-34	4.3
FR-35	4.3
FR-36	4.5
FR-37	10.0
FR-38	3.0
FR-39	10.0
FR-40	10.0
FR-41	1.1
FR-42	7.1, 7.2
FR-43	13.5
FR-44	13.1, 13.2, 13.3, 13.4
FR-45	6.0, 6.1
FR-46	9.1
FR-47	9.2
FR-48	9.3

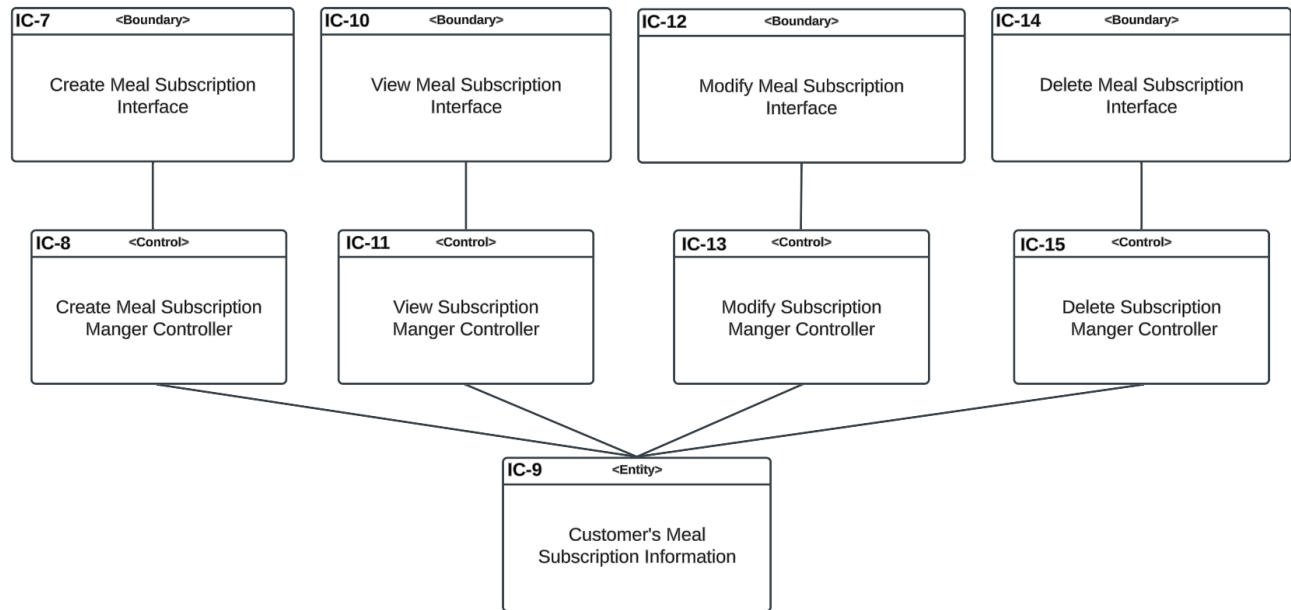
9. Class Analysis Modeling

9.1 Initial Class Diagram (ICs)

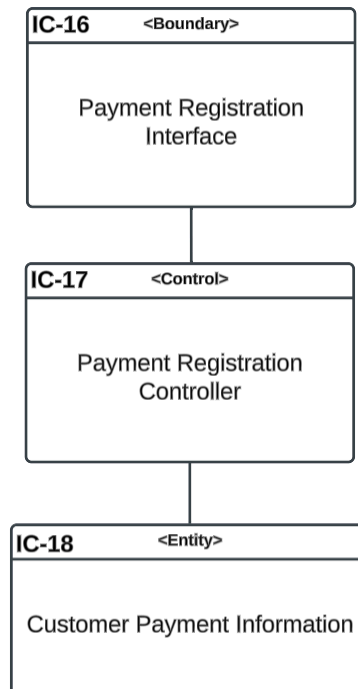
9.1.1 Order a meal



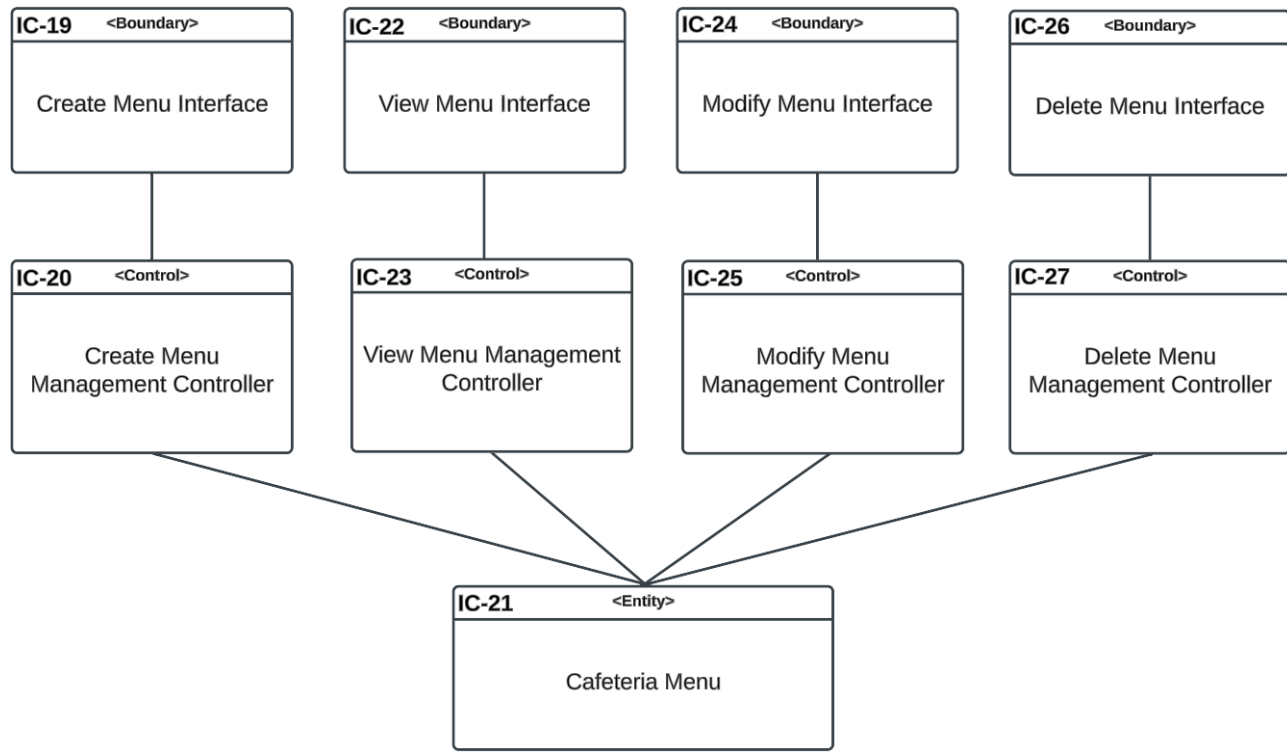
9.1.2 Manage meal service subscriptions



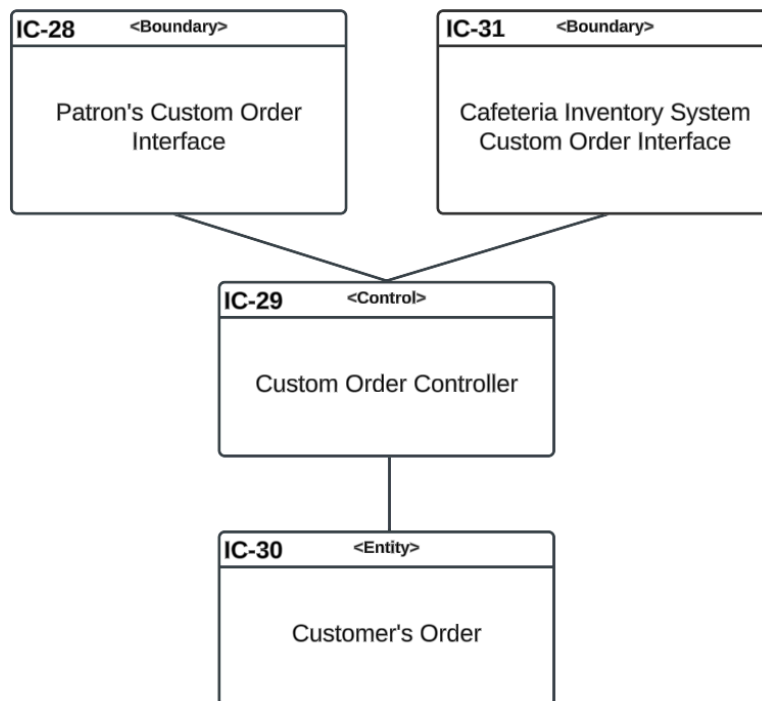
9.1.3 Register payment method



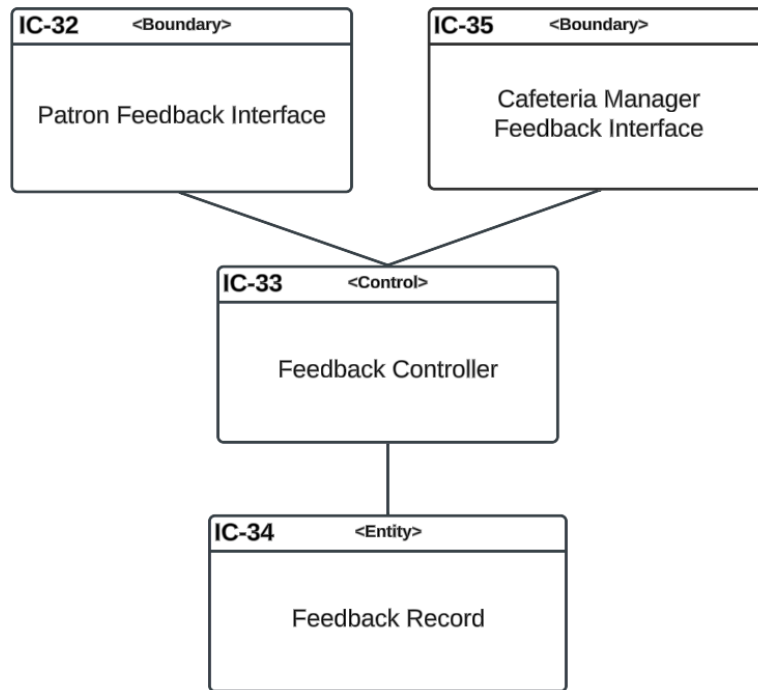
9.1.4 Manage cafeteria menus



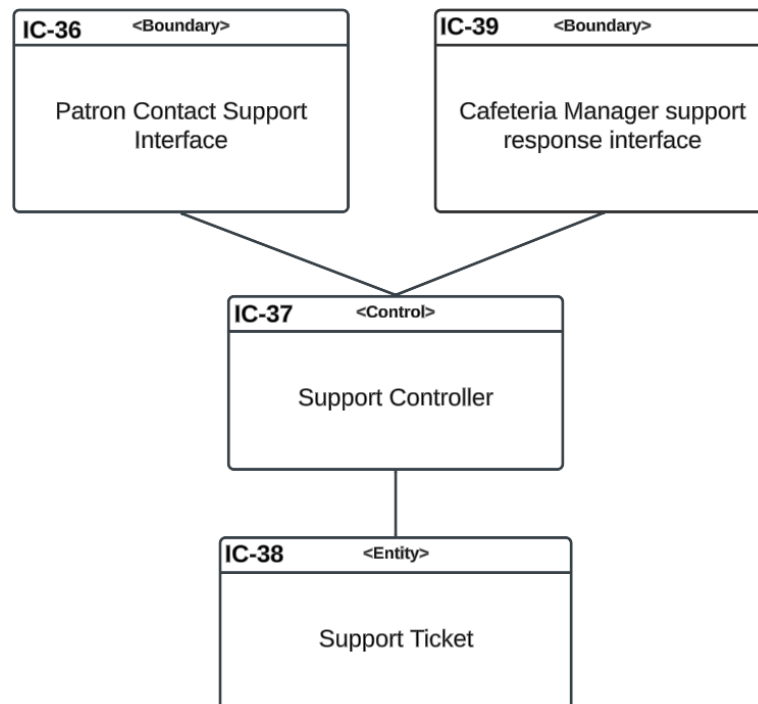
9.1.5 Order custom meals not on the menu



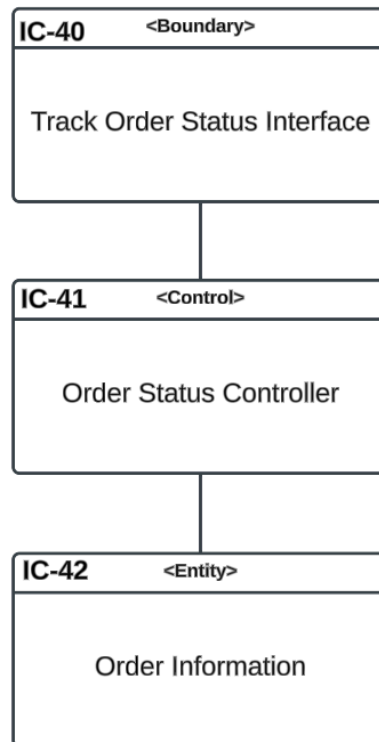
9.1.6 Request and collect feedback



9.1.7 Contact support



9.1.8 Track meal order status



9.1.9 Manage favorite items

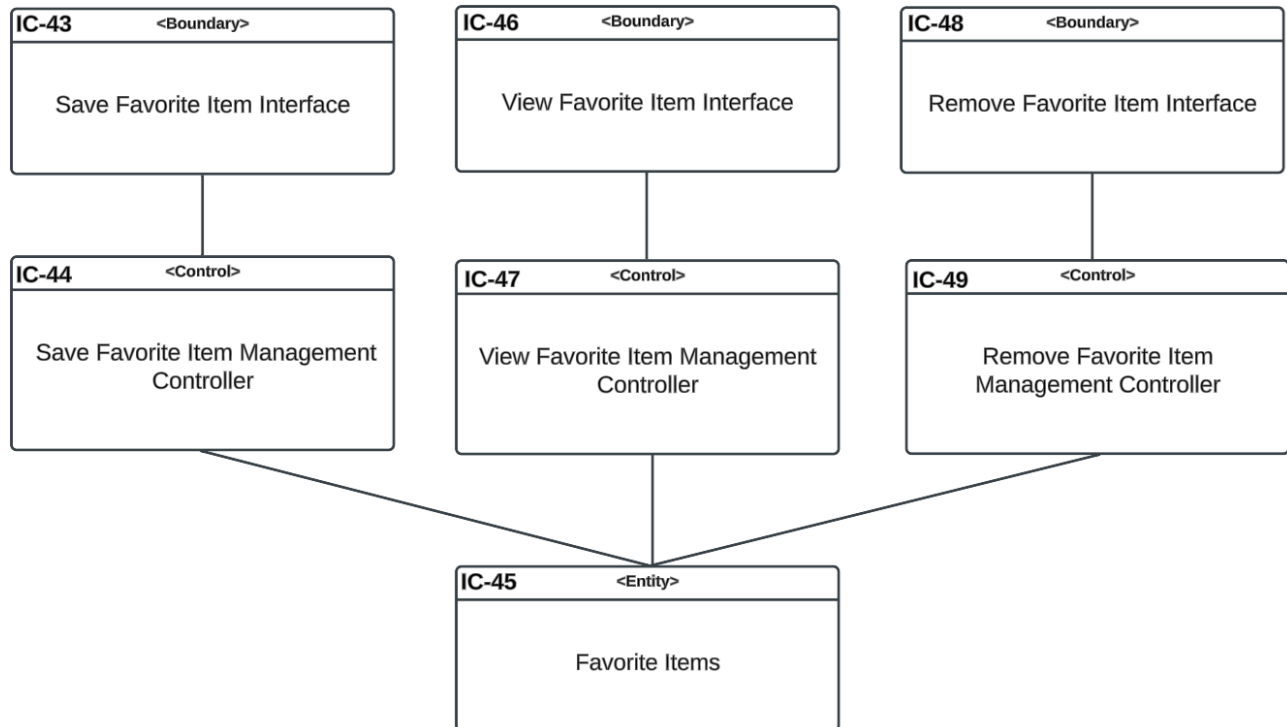


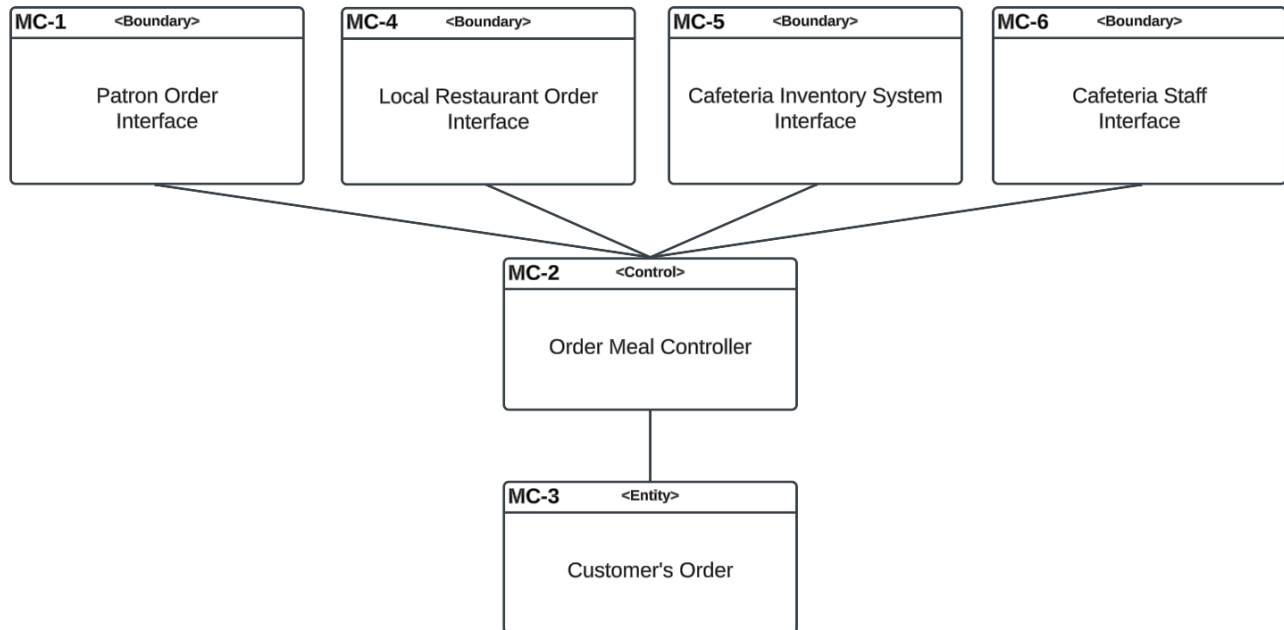
TABLE 4: Traceability Matrix (FRs and Initial Classes)

	Related ICs
FR-1	IC-1, IC-2, IC-3
FR-2	IC-1, IC-2, IC-3
FR-3	IC-1, IC-2, IC-3
FR-4	IC-1, IC-2, IC-3, IC-6
FR-5	IC-1, IC-2, IC-3, IC5
FR-6	IC-1, IC-2, IC-3
FR-7	IC-1, IC-2, IC-3
FR-8	IC-4, IC-2, IC-3
FR-9	IC-4, IC-2, IC-3
FR-10	IC-4, IC-2, IC-3
FR-11	IC-7, IC-8, IC-9
FR-12	IC-7, IC-8, IC-9
FR-13	IC-7, IC-8, IC-9
FR-14	IC-7, IC-8, IC9
FR-15	IC-7, IC-8, IC9
FR-16	IC-7, IC-8, IC9
FR-17	IC-7, IC-8, IC-9
FR-18	IC-9, IC-11, IC-10
FR-19	IC-9, IC-11, IC-10
FR-20	IC-9, IC-14, IC-15
FR-21	IC-9, IC-14, IC-15
FR-22	IC-9, IC-14, IC-15
FR-23	IC-9, IC-14, IC-15
FR-24	IC-9, IC12, IC-13
FR-25	IC-9, IC12, IC-13

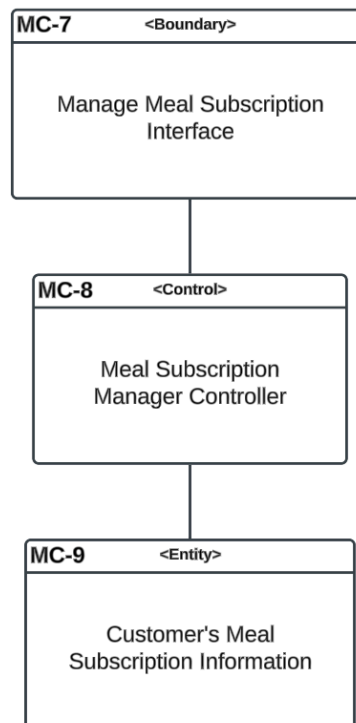
	Related ICs
FR-26	IC-9, IC12, IC-13
FR-27	IC-9, IC12, IC-13
FR-28	IC-9, IC12, IC-13
FR-29	IC-9, IC12, IC-13
FR-30	IC-9, IC12, IC-13
FR-31	IC-9, IC12, IC-13
FR-32	IC-9, IC12, IC-13
FR-33	IC-16, IC-17, IC-18
FR-34	IC-16, IC-17, IC-18
FR-35	IC-16, IC-17, IC-18
FR-36	IC-16, IC-17, IC-18
FR-37	IC-21, IC-19, IC-20
FR-38	IC-21, IC-22, IC-23
FR-39	IC-21, IC-24, IC-25
FR-40	IC-21, IC-26, IC-27
FR-41	IC-29, IC-29, IC-30, IC-31
FR-42	IC-29, IC-29, IC-30, IC-31
FR-43	IC-36, IC-37, IC-38, IC-39
FR-44	IC-36, IC-37, IC-38, IC-39
FR-45	IC-40, IC-41, IC-42
FR-46	IC-43, IC-44, IC-45, IC-46, IC-47, IC-48, IC49
FR-47	IC-43, IC-44, IC-45, IC-46, IC-47, IC-48, IC49
FR-48	IC-43, IC-44, IC-45, IC-46, IC-47, IC-48, IC49

9.2 Modified Class Diagram (MCs)

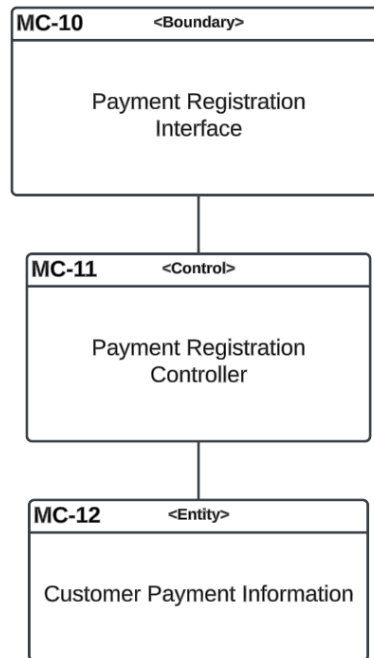
9.2.1 Order a meal



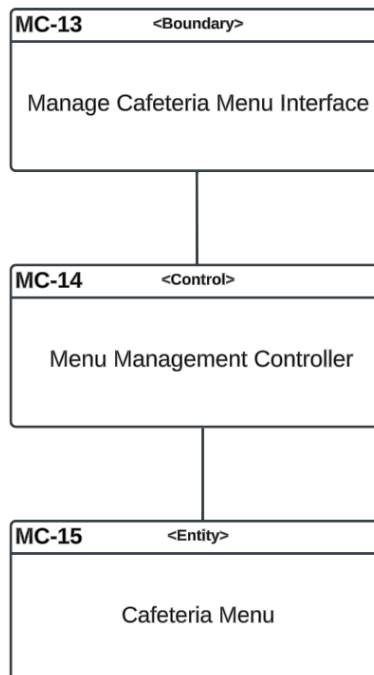
9.2.2 Manage meal service subscriptions



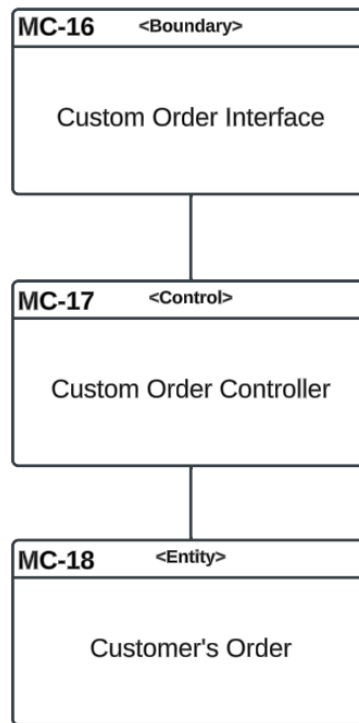
9.2.3 Register payment method



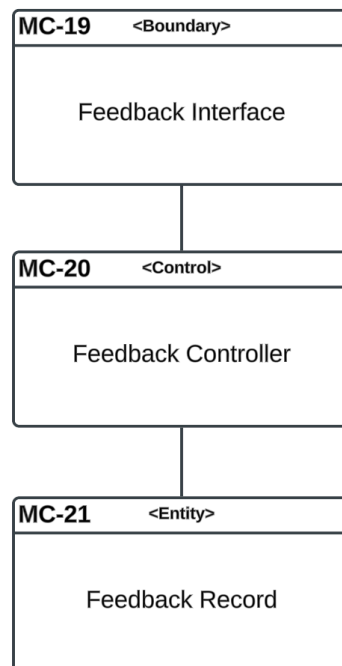
9.2.4 Manage cafeteria menus



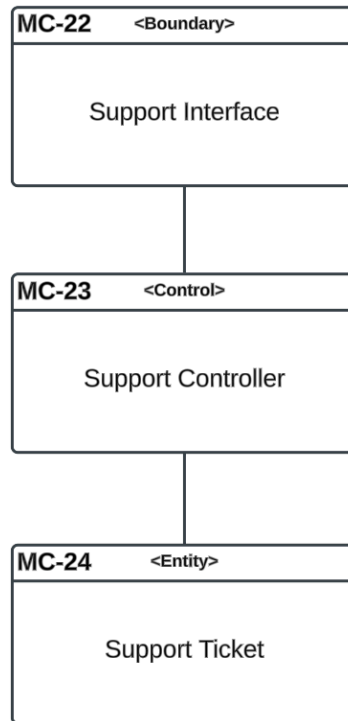
9.2.5 Order custom meals not on the menu



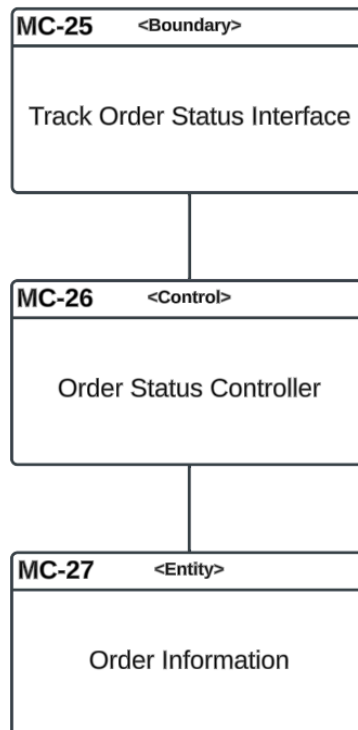
9.2.6 Request and collect feedback



9.2.7 Contact support



9.2.8 Track meal order status



9.2.9 Manage favorite items

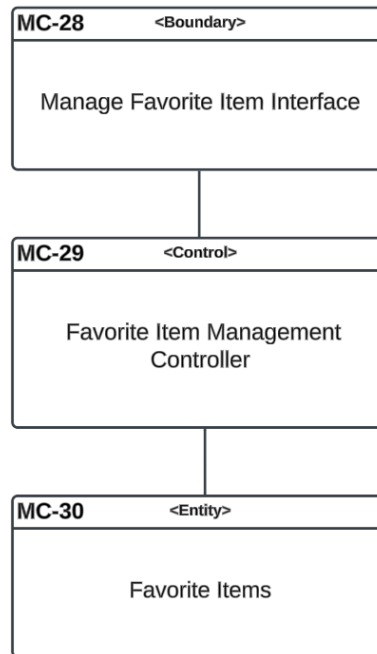


TABLE 5: Traceability Matrix (Initial Classes (IC) and modified Classes (MC))

	Related MCs
IC-1	MC-1
IC-2	MC-2
IC-3	MC-3
IC-4	MC-4
IC-5	MC-5
IC-6	MC-6
IC-7	MC-7
IC-8	MC-8
IC-9	MC-9
IC-10	MC-7
IC-11	MC-8
IC-12	MC-7
IC-13	MC-8
IC-14	MC-7
IC-15	MC-8
IC-16	MC-10
IC-17	MC-11
IC-18	MC-12
IC-19	MC-13
IC-20	MC-14
IC-21	MC-15
IC-22	MC-13
IC-23	MC-14

	Related MCs
IC-24	MC-13
IC-25	MC-14
IC-26	MC-13
IC-27	MC-14
IC-28	MC-16
IC-29	MC-17
IC-30	MC-18
IC-31	MC-16
IC-32	MC-19
IC-33	MC-20
IC-34	MC-21
IC-35	MC-19
IC-36	MC-22
IC-37	MC-23
IC-38	MC-24
IC-39	MC-22
IC-40	MC-25
IC-41	MC-26
IC-42	MC-27
IC-43	MC-28
IC-44	MC-29
IC-45	MC-30
IC-46	MC-28
IC-47	MC-29
IC-48	MC-28
IC-49	MC-29