SECP3204: Software Engineering (WBL)

System Documentation (SD)

Food Ordering System at Arked Angkasa,
Universiti Teknologi Malaysia (UTM)

Version 1

Date 16/5/2023

Faculty of Computing

Prepared by: Beta

Revision Page

a. Overview

This overview provides a brief summary of the Software Requirements Specification (SRS) document being developed by Team Beta for the Food Ordering System at Arked Angkasa. The system aims to streamline the food ordering process for UTM Students by partnering with two shops: Sdap Kitchen, specializing in Nasi Lemak, and Deen Corner, offering a variety of drinks.

b. Target Audience

The target audience for the Food Ordering System at Arked Angkasa includes UTM (Universiti Teknologi Malaysia) students who reside or frequently visit Arked Angkasa. The system is designed to cater specifically to their needs and provide them with a convenient and efficient food ordering experience. The target audience can be further defined as follows:

1. **UTM Students**: The primary target audience consists of UTM students from various faculties and programs. These students may have busy schedules and limited time for meal preparation, making the Food Ordering System a valuable solution for their food requirements.
2. **UTM Staff**: UTM faculty and administrative staff can benefit from the Food Ordering System by conveniently ordering food from Sdap Kitchen and drinks from Deen Corner. The system can provide them with a time-saving solution for their meal requirements during working hours.

c. **Project Team Members**

Member Name	Role	Task	Status
MUHAMMAD DANIEL HAKIM BIN SYAHRULNIZAM	Requirements Analyst	System Features (2.2.1 - 2.2.8)	Complete
MUHAMMAD ARIFF DANISH BIN HASHNAN	Developer	User Characteristics & System Features (2.1 & 2.2)	Complete
MUHAMMAD IMAN FIRDAUS BIN BAHARUDDIN	User Experience Designer	Introduction (1.1-1.5)	Complete
MUHAMMAD SAFWAN BIN MOHD AZMI	Quality Assurance	System Features (2.2.1 - 2.2.8)	Complete
CHE MARHUMI BIN CHE AB RAHIM	Documentation Specialist	Software System Attributes, Performance and Other Requirements & Design Constraint (2.3 & 2.4)	Complete

d. **Version Control History**

Version	Primary Author(s)	Description of Version	Date Completed
1.0	Team leader 1 (Muhammad Daniel Hakim bin Syahrulnizam)	Completed Chapter 1 and 2, Section Introduction and Specific Requirements	10/06/2023

Table of Contents

1. Introduction

Online ordering and takeaway platforms have become a necessity in modern life throughout the Digital Revolution. These platforms' efficiency and convenience have revolutionized the food business by making it simple for people to order meals from their favorite restaurants. In this setting, two eateries that serve students, "Sdap Kitchen" and "DEEN CORNER," understand the value of adopting digital ordering systems to improve their offerings and broaden their clientele.

"Sdap Kitchen" and "DEEN CORNER" have made the decision to install online ordering platforms in order to satisfy the demands of tech-savvy students looking for a quick and easy manner to purchase meals. They may offer their mouthwatering menu items for online ordering and takeaway by utilizing these channels, giving students a smooth dining experience.

However, establishing the platforms is only the first step in extending access to online ordering and takeaway services. To guarantee seamless operations and give a good customer experience, vigilant system maintenance is necessary. It becomes essential to keep an eye on the system's functioning in order to quickly spot and resolve any potential problems.

The system's effectiveness is maintained through regular monitoring, which also makes it possible to make forecasts and take preventative measures to lessen undesirable results. "Sdap Kitchen" and "DEEN CORNER" can proactively fix issues by closely monitoring the system, assuring the sustainability and smooth functioning of their online ordering and takeaway platform. This consistent dedication to quality improves students' entire dining experiences and equips "Sdap Kitchen" and "DEEN CORNER" to succeed in a cutthroat food industry.

1.1 Purpose

This System Documentation (SD) is intended to give a thorough explanation of the creation of the food ordering system targeted at students and created specifically for the two

little booths known as "Sdap Kitchen" and "DEEN CORNER." The System Design Document (SDD), System Requirements Specification (SRS), and System Testing Document (STD) are important parts of the SD.

For all parties participating in the development process, including developers, testers, project managers, and stall owners, the SD is an essential reference guide. The processes for system requirements, design, and testing are made obvious, ensuring that all stakeholders are on the same page and pursuing the same goals.

The SD describes the precise features and functionalities of the Food Ordering System for the intended student audience, putting a special emphasis on easy online ordering and practical takeaway options. It takes into account user needs such as easy menu navigation, customized orders, and safe payment methods. The SD makes sure that the ordering and takeaway procedures are quick, simple to use, and catered to the tastes and requirements of the students.

The SD aids in the success of the Food Ordering System development process by offering organized and thorough documentation. Error risk is decreased, efficiency is increased, and overall project results are improved. The SD serves as a useful tool, guiding the stakeholders throughout the development lifecycle and supporting the smooth deployment of the Food Ordering System for "Sdap Kitchen" and "DEEN CORNER."

1.2 Scope

This System Documentation (SD) for the software product "Food Ordering System" is intended to give readers a comprehensive understanding of the system development process, including the System Requirements Specification (SRS), System Design Document (SDD), and System Testing Document (STD).

The "Food Ordering System" software solution will allow students to easily order and pick up their meals from the two booths known as "Sdap Kitchen" and "DEEN CORNER." Students will be able to explore menus, personalize their orders, and make safe payments

using a user-friendly interface. The software's primary purpose will be to make the ordering and takeaway processes easier; no other functionalities will be included.

The following is covered by the "Food Ordering System" software product:

1. Identification of user and system requirements, including functionality, performance, usability, reliability, and security requirements, that are unique to the demands of "Sdap Kitchen" and "DEEN CORNER" booths.

2. An agile-driven strategy will be used to create the software solution, providing for flexibility and adaptation to meet the changing needs of the students and stalls.

3. Effortless integration with their current systems and compatibility with the ordering platforms used by "Sdap Kitchen" and "DEEN CORNER" booths.

4. Adherence to software engineering best practices, such as coding standards, testing protocols, and quality assurance methods, to guarantee a solid and trustworthy solution.

5. Thorough testing to guarantee that the "Food Ordering System" satisfies all requirements and offers a seamless ordering experience using a combination of manual and automated approaches.

6. Giving the students and the stalls a thorough installation manual and user manual to help them set up and efficiently use the "Food Ordering System" software.

The System Development (SD) ensures a disciplined and organized approach to the development of the "Food Ordering System" software by documenting the system development process and its associated components. It is a useful tool for the parties engaged in its creation and implementation, allowing them to coordinate their efforts and improve the way people order meals from the "Sdap Kitchen" and "DEEN CORNER" kiosks.

1.3 Definitions, Acronyms and Abbreviation

Term	Definition
Food Ordering System	The software product being developed to give customers, especially students, a smooth and convenient ordering experience when they want to place and pick up food orders from "Sdap Kitchen" and "DEEN CORNER" kiosks.
SRS	System Requirements Specification - a document that describes in detail the requirements and functionalities of a software system being developed
SDD	System Design Document - a document that provides a detailed overview of the architectural design and structure of a software system
STD	System Testing Document - a document that outlines the testing approach, strategies, and procedures for verifying and validating the functionality and performance of a software system
UI	User Interface - refers to the graphical or visual elements of a software application or system that allow users to interact with it
UX	User Experience - encompasses the overall experience and satisfaction that a user has when interacting with a product, system, or service

1.4 References

https://www.techtarget.com/searchsoftwarequality/definition/documentation

https://sqa.mampu.gov.my/index.php/ms/contoh-penyelesaian-dokumen-krisa

https://helpjuice.com/blog/software-documentation

1.5 Overview

The Software Documentation (SD) gives an in-depth overview of the Food Ordering System software being created for the two stalls known as "Sdap Kitchen" and "DEEN CORNER." System Requirements Specification (SRS), System Design publication (SDD), and System Testing Document (STD) are the three primary sections of this publication.

The structure of the SD is as follows:

1. User Characteristics: In this section, the Food Ordering System software product's target consumers are described, with students serving as the section's main stakeholders. When it comes to ordering and takeaway services, it gives a general overview of their requirements, anticipations, and pertinent traits.

2. System Features: The functional and non-functional requirements of the software product are described in depth in this part, which is adapted specifically to the requirements of the "Sdap Kitchen" and "DEEN CORNER" booths. It covers necessary functions like menu browsing, order customization, safe payment methods, and directions for order pickup.

3. System Quality Attributes: The stability, usability, maintainability, and portability of the Food Ordering System software package are highlighted in this section. When students order and pick up their meals, these features guarantee a smooth and simple process.

4. Performance and Design Constraints: This section describes the performance specifications and design limitations that must be taken into account throughout the software product's development. It guarantees that the Food Ordering System satisfies the unique requirements of "Sdap Kitchen" and "DEEN CORNER" stalls and operates at its peak in the context of food ordering that is focused on students.

All parties engaged in the design, testing, and deployment of the Food Ordering System software product can use the SD document as a thorough reference. It offers a clear

overview of the software specifications, design guidelines, and testing procedures, assisting in the development process and ensuring the system lives up to users' expectations as well as those of the two tiny restrooms.

2. Specific Requirements

2.1 User characteristics

User requirements are very important in determining how a food ordering system is designed and operates. It is crucial to take into account the specific characteristics of the users when creating such a system, including their level of technological expertise, time constraints, mobile-oriented behavior, dietary preferences, financial constraints, convenience-seeking mindset, feedback orientation, security and privacy awareness, and propensity for social sharing. A user-friendly and effective food ordering system can be developed by taking into account these user characteristics in order to adapt to their unique demands, preferences, and behaviors, thereby improving their overall experience and satisfaction. By designing the system to reflect these characteristics, it becomes an effective tool for simplifying the ordering process and offering users a satisfying and practical way to browse menus, place orders, personalize their preferences, pay for their orders, and interact with the system.

2.1.1 Student UTM

- UTM students are likely to be familiar with technology and comfortable using digital platforms and mobile apps for various purposes, including food ordering.
- They rely on mobile devices as their primary means of accessing information and conducting online activities. Therefore, having a mobile-responsive interface or a dedicated mobile app for the food ordering system can be beneficial.
- They often have busy schedules and limited time between classes, assignments, and extracurricular activities. This system should be designed to be efficient and save time by offering quick menu browsing, easy customization options, and streamlined checkout processes.
- They usually have limited budgets. Offering affordable menu options, discounts, and loyalty programs can be appealing to this demographic.
- They often have a diverse student population with various dietary preferences and restrictions (e.g., vegetarian, vegan, halal, gluten-free). This system should provide clear and accurate information about ingredients and allow customization options to accommodate different dietary needs.

- They often enjoy socializing and ordering food with friends or in groups. This system could incorporate features that allow for group orders, splitting bills, or sharing recommendations.
- They value their experiences and often provide feedback to improve services. Implementing a feedback mechanism within the system can allow students to share their opinions, report issues, and provide suggestions for improvement.

2.1.2 Owner of Restaurant

- Restaurant owners are focused on running their businesses efficiently and maximizing profitability. They seek solutions that streamline operations, increase order volume, and improve customer satisfaction.
- While not all restaurant owners are tech-savvy, many recognize the value of technology in enhancing their operations. They are open to adopting digital solutions, such as a food ordering system, to automate processes and improve overall efficiency.
- They are responsible for creating and maintaining their menus. They require a food ordering system that allows easy management of menu items, including adding, editing, and removing dishes, specifying prices, descriptions, and customizable options.
- They need a system that effectively processes incoming orders, manages order statuses, and ensures efficient coordination between the kitchen and staff.
- They benefit from accessing comprehensive reports and analytics that provide insights into sales data, order volume, popular menu items, customer feedback, and other key metrics. This information helps them make informed business decisions and optimize their operations.
- They may seek features within the food ordering system that enable promotional activities, such as discounts, coupon codes, special offers, and loyalty programs. These features help attract customers, retain existing ones, and encourage repeat orders.
- They value reliable customer support and training resources to ensure smooth adoption and ongoing usage of the food ordering system. Timely assistance in resolving issues or addressing technical concerns is crucial for uninterrupted business operations.

2.1.3 Assistant Owner of Restaurant

- As an assistant owner, they prioritize operational efficiency and smooth order management. The food ordering system should offer streamlined processes, such as easy order placement, order tracking, and integration with the kitchen and staff for seamless communication and order fulfillment.
- They need the ability to update menus, add or remove items, set prices, and manage inventory levels within the food ordering system. This allows them to keep the menu up to date and ensure accurate availability information for customers.
- They value insights into sales performance, order volume, popular menu items, and customer preferences. The food ordering system should provide comprehensive reporting and analytics functionalities to help them make informed decisions, identify trends, and optimize your menu and operations.
- The system should allow them to easily create customized options or daily specials within the menu. This flexibility enables them to adapt to changing customer demands, offer unique dishes, and experiment with new menu items or promotions.
- They require real-time order tracking and management features to efficiently process orders, assign them to kitchen staff, track preparation and cooking times
- The system should enable customers to provide feedback, ratings, and reviews. This feedback can help them monitor customer satisfaction, identify areas for improvement, and respond promptly to customer concerns or issues.
- Access to comprehensive training resources and responsive customer support is essential. It ensures that they can effectively utilize the food ordering system, resolve any technical issues, and maximize its potential for your restaurant.

2.2 System Features

The food ordering system, from a product perspective, serves as a comprehensive solution that revolutionizes the way restaurants handle customer orders and streamline their operations. It acts as a digital platform that seamlessly connects customers, restaurant owners, kitchen staff, and delivery personnel. The system provides a user-friendly interface for customers to browse menus, place orders, and make payments conveniently. For restaurant owners, the system offers functionalities such as menu and inventory management, order processing, analytics, and integration with existing systems like accounting. It enables owners to efficiently manage their operations, track orders, and gain insights into sales performance.

The food ordering system also enhances the experience for kitchen staff by facilitating order communication and coordination, leading to improved efficiency and accuracy. Overall, the food ordering system provides a holistic solution that enhances the customer experience, streamlines restaurant operations, and improves efficiency across all stakeholders involved in the ordering and delivery process.

The system features are illustrated in **Figure 1** below. The detailed description of each module and functions is tabulated in **Table 1**.

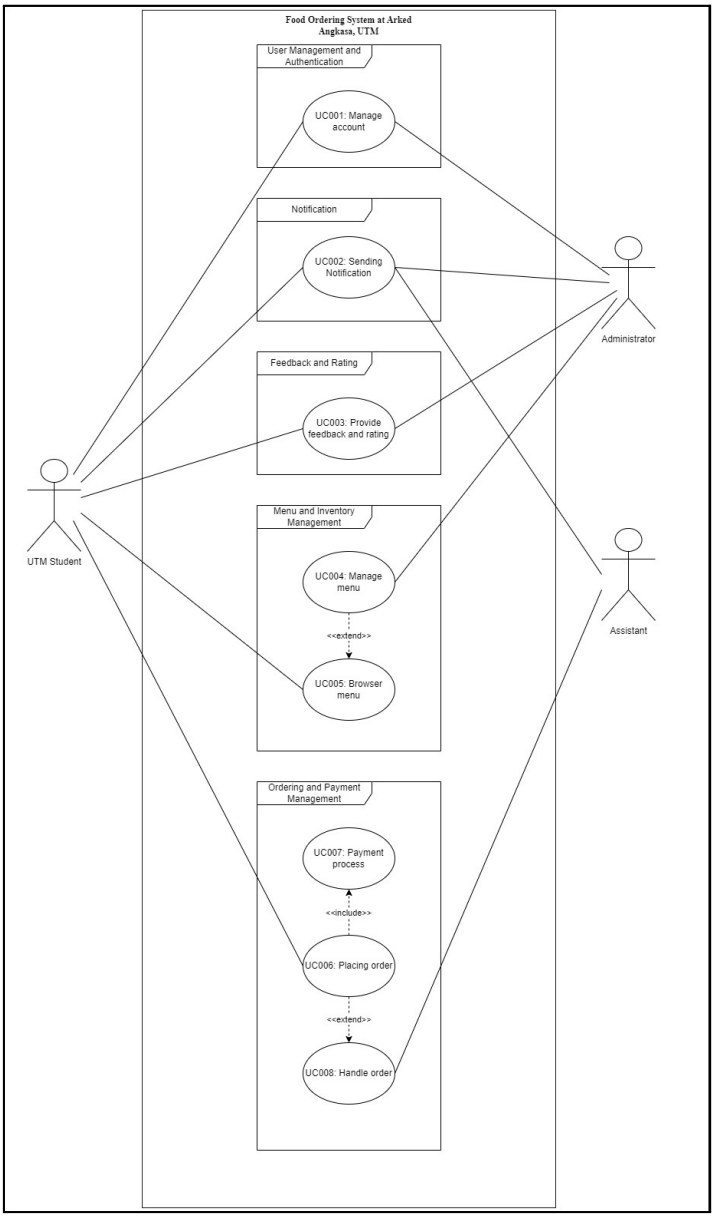


Figure 1.1 : Use Case Diagram for Food Ordering System

Table 1 : Description of Module and Functions for Food Ordering System

Module	***Function***	***Description***
User Management and Authentication Module	***UC001*** *- Manage account*	*This use case allows students to sign up as a user for the mobile application.*
Notification Module	***UC002*** *- Sending notification*	*This use case focuses on the system's ability to send notifications to the relevant parties, such as assistant staff, administrators, and customers, based on specific trigger events.*
Feedback and Rating Module	***UC003*** *- Provide feedback and rating*	*This use case involves allowing users to express their opinions, experiences, and satisfaction levels regarding a particular offering. It typically involves a feedback mechanism where users can provide their comments, suggestions, or criticisms, as well as assign a rating or score to indicate their overall satisfaction or dissatisfaction.*
Menu and Inventory Management Module	***UC004*** *- Manage menu*	*This use case focuses on the management and administration of a menu system. It involves the tasks and functionalities associated with creating, editing, and organizing menus in a user-friendly manner. This use case typically involves features such as adding new menu items, updating existing items, arranging items in categories or sections, setting prices, and controlling the availability of menu items.*
	UC005 *- Browse menu*	*This use case provides users with a simple and intuitive way to navigate and explore content or options within a system or application. The Browse menu typically displays a list of categories, subcategories, or sections that users can select to access specific content or perform certain actions. It allows users to visually browse through available choices and select the desired option with ease. The Browse menu enhances user experience by organizing information and providing a clear pathway for users to find what they are looking for within the system or application.*
Ordering and Payment Management Module	***UC006*** *- Placing order*	*This use case selects products or services they wish to purchase and initiates the order placement procedure. It involves providing relevant information, such as the desired quantity, delivery address, and any additional specifications or preferences. The system then processes the order, verifies availability, calculates the total cost, and generates an order confirmation. This use case focuses on the fundamental steps involved in placing an order, facilitating a smooth and efficient transaction between the customer and the business.*

	UC007 *- Payment process*	*This use case refers to a payment process that involves the transfer of funds from one party to another in a simple and straightforward manner. It typically begins with the initiation of a payment by the payer, who provides the necessary information such as the recipient's details and the amount to be transferred. This information is then transmitted through a secure channel to a payment service provider, which verifies the transaction and ensures the availability of funds. Once the verification is complete, the payment is authorized, and the funds are transferred from the payer's account to the recipient's account. The process is designed to be efficient, secure, and convenient, providing a seamless experience for individuals and businesses when conducting financial transactions.*
	UC008 *- Handle order*	

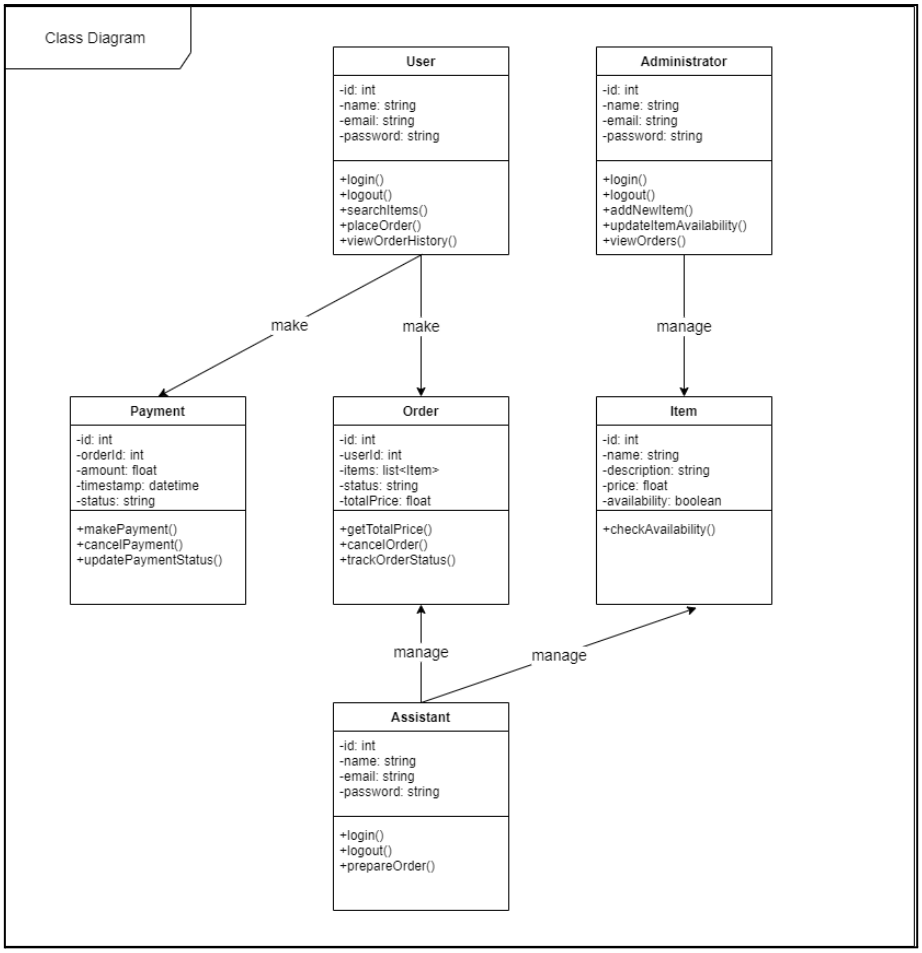


Figure 1.2 : Domain Model for Food Ordering System

2.2.1 UC001 : *Manage account*

Table 2.1.1 : Use Case Description for Manage account as UTM student

Use case : *Manage Account*
ID: UC001
Actors: UTM Student(customer)
Preconditions: Has active internet connection
Flow of events: 1. The user presses the “Sign Up” button to create an account. 2. System display registration account page 3. The user needs to fill details information for verification purpose. 4. The user presses the “submit” button. 5. If the form is invalid, the system will display errors encountered. 6. Otherwise, the system adds a user account to the database and the user enters the home page. 7. Use case end. …
Postconditions: User can log in into the system as UTM student (customer)
Postconditions: User cannot sign up into the system.
Exception flow (if any): 1. Invalid registration form 2. System display error message

Table 2.1.2 : Use Case Description for Manage account as Administrator

Use case : *Manage Account*
ID: UC001
Actors: Administrator
Preconditions: Has active internet connection
Flow of events: 1. The user presses the “Sign Up” button to create an account. 2. System display registration account page 3. The user needs to fill in detailed information for verification purposes. 4. The user presses the “submit” button. 5. If the form is invalid, the system will display errors encountered. 6. Otherwise, the system adds a user account to the database and the user enters the home page. 7. Use case end.
Postconditions: User can log in into the system as administrator
Postconditions: User cannot sign up into the system.
Exception flow (if any): 1. Invalid registration form. 2. System display error message.

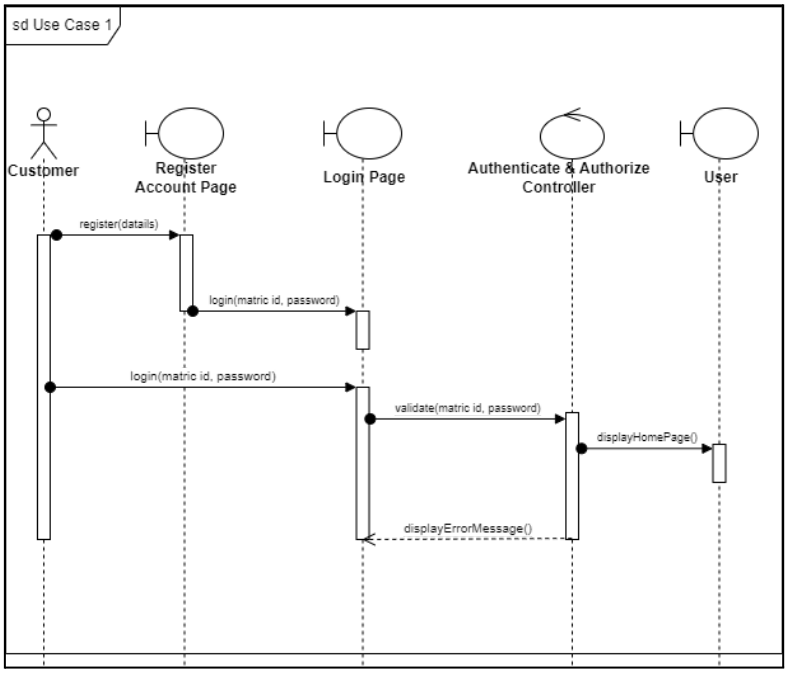


Figure 2.1.1: Sequence Diagram for Manage Account as UTM Student (customer)

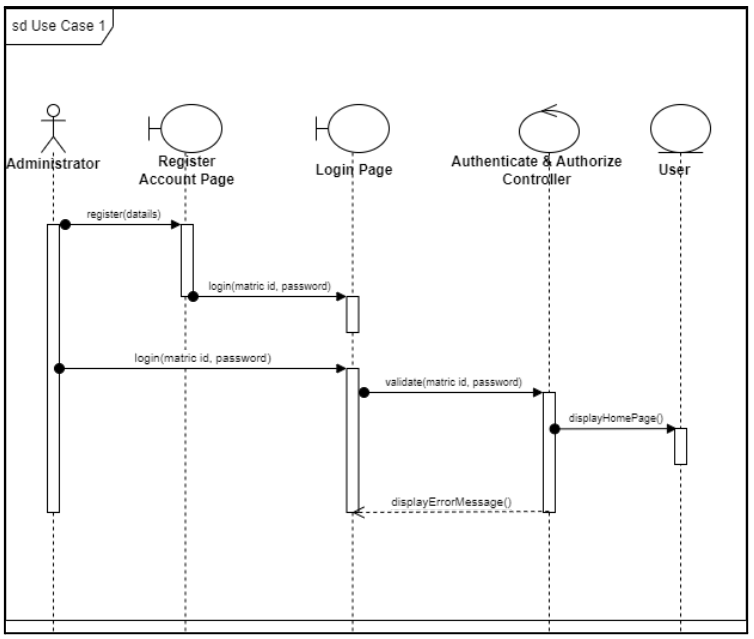


Figure 2.1.2: Sequence Diagram for Manage Account as Administrator

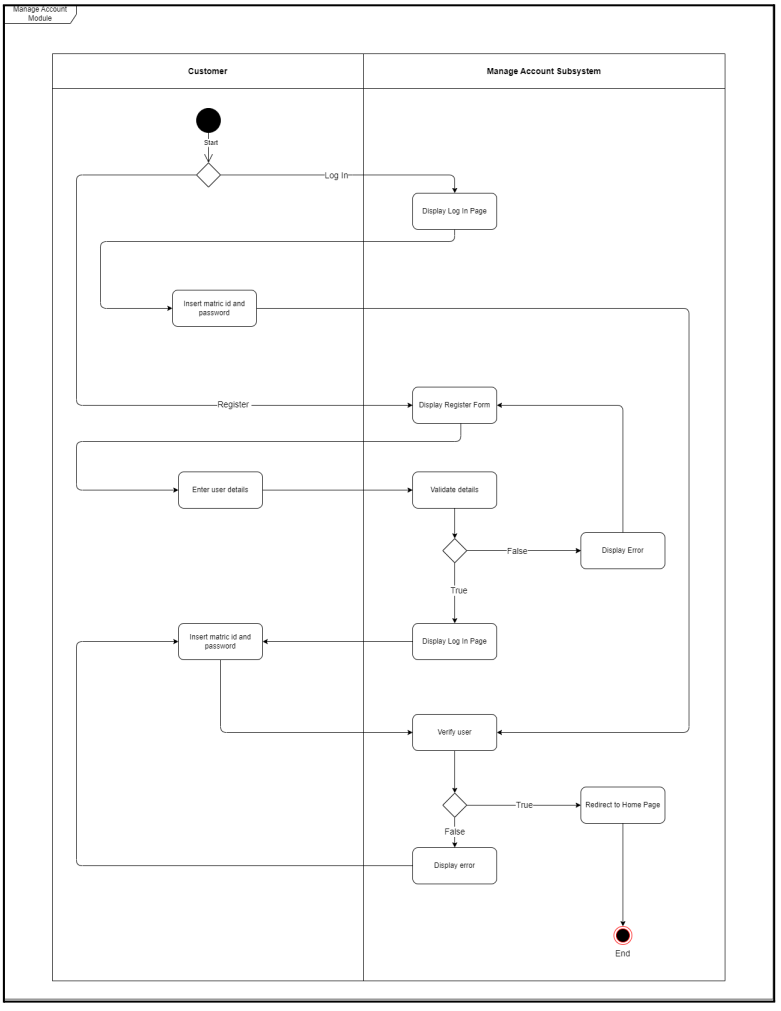


Figure 2.1.3: Activity Diagram for Manage Account as Customer (UTM Student)

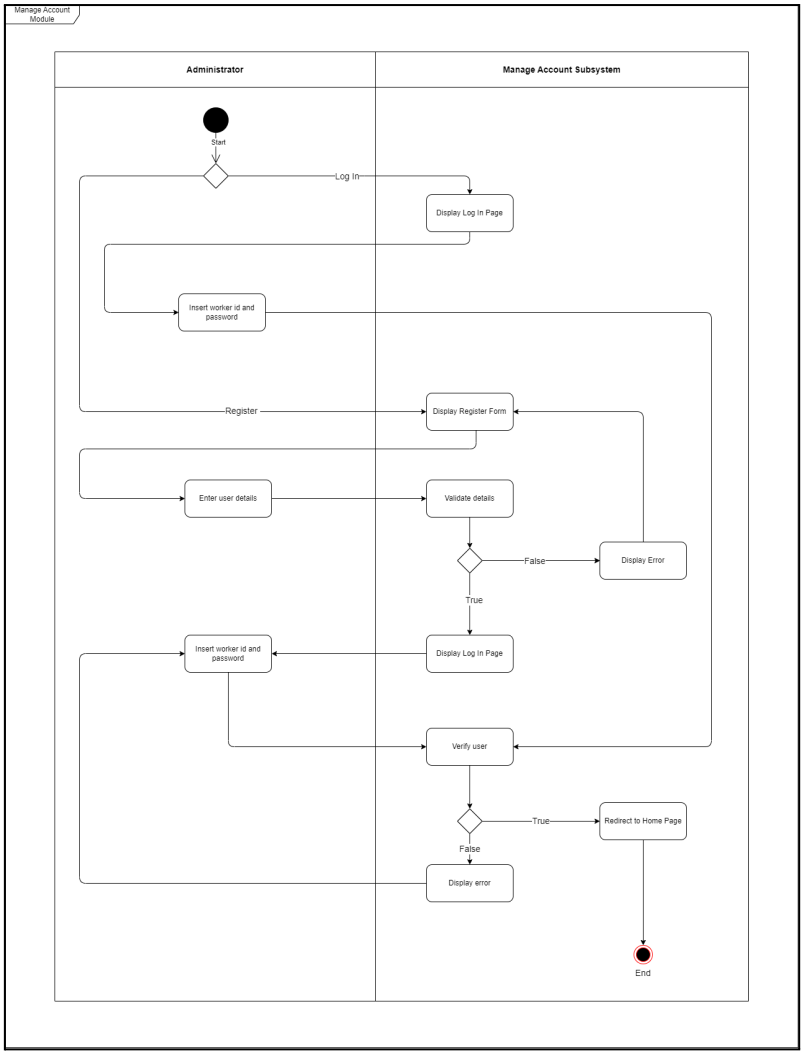


Figure 2.1.4: Activity Diagram for Manage Account as Administrator

2.2.2 UC002 : *Sending notification*

Table 2.2 : Use Case Description for Sending notification

Use case : *Sending notification*
ID: UC002
Actors: Customer (UTM Student)
Brief Preconditions: 1.The system has relevant contact information for customers, administrators, and operator staff. 2.There is a trigger event that requires sending a notification.
Precondition : Has active internet connection
Flow of events: 1.The system detects a trigger event that requires sending a notification. 2.The system identifies the recipient(s) of the notification (customer, administrator, and/or operator staff). 3.The system generates the notification content based on the trigger event. 4.The system sends the notification to the respective recipient(s) through the appropriate communication channel (email, SMS, app notification, etc.). 5.The recipient(s) receive the notification.
Postconditions: The notification is successfully sent to the recipient(s).
Postconditions: 1.The system encounters an error while sending the notification. 2.The system logs the error and retries sending the notification. 3.If the retries fail, the system notifies the technical team for resolution. 4.The flow continues, and the recipient(s) may not receive the notification immediately.

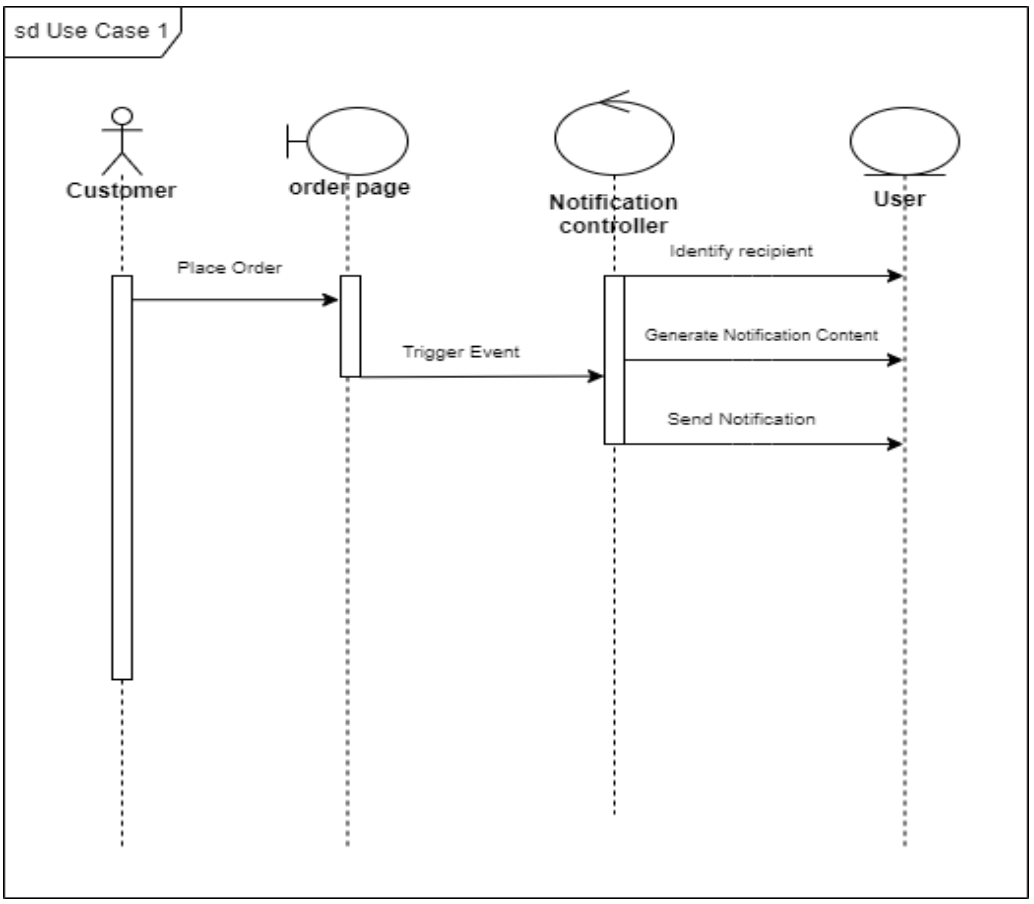
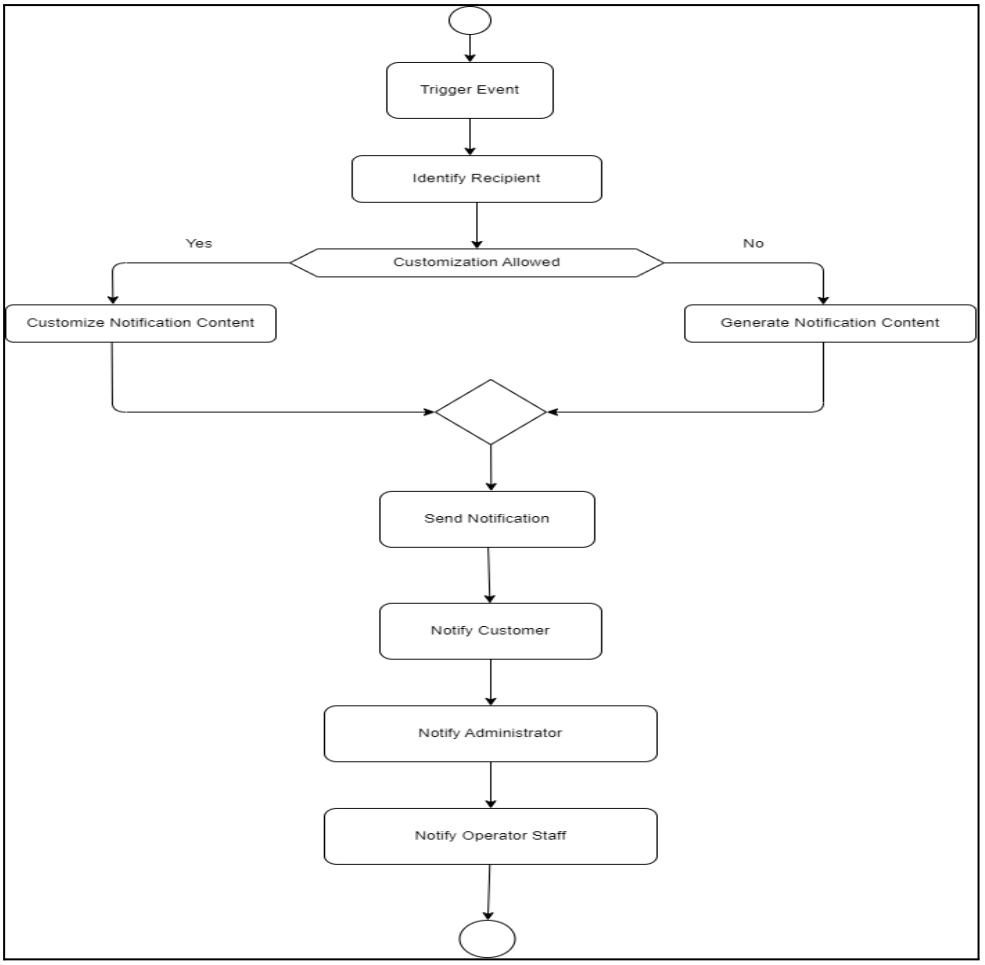


Figure 2.2.1: Sequence Diagram for Sending Notifications as Customer (UTM Student)

Figure 2.2.2 : Activity Diagram for Sending Notifications as Customer (UTM Student)



2.2.3 UC*003* : *Provide feedback and rating*

Table 2.3.1 : Use Case Description for Provide feedback and rating as Customer (UTM Student)

Use case : *Provide feedback and rating*
ID: UC003
Actors: Customer (UTM Student)
Preconditions: Has a good internet connection
Flow of events: 1. Feedback and Ratings pop up on screen when user enter the application 2. If users press the “Yes” button, it means that they are willing to share their user experience. 2.1. 2.Users need to rate stars between 0 to 5 according to a few questions that have been asked. 2.2. System records the feedback and rating from the user into the database for developer purposes. 3. If users press the “Not Really” button, it means that they are not willing to share their user experience. 4. Then, users return to their current page.
Postconditions: Return to their current page.

Table 2.3.2 : Use Case Description for Provide feedback and rating as Administrator

Use case : *Provide feedback and rating*
ID: UC003
Actors: Administrator
Preconditions: Has a good internet connection
Flow of events: 1. Feedback and Ratings pop up on screen when user enter the application. 2. If users press the “Yes” button, it means that they are willing to share their user experience. 2.1. Users need to rate stars between 0 to 5 according to a few questions that have been asked. 2.2. System records the feedback and rating from the user into the database for developer purposes. 3. If users press the “Not Really” button, it means that they are not willing to share their user experience. 4. Then, users return to their current page.
Postconditions: Return to their current page.

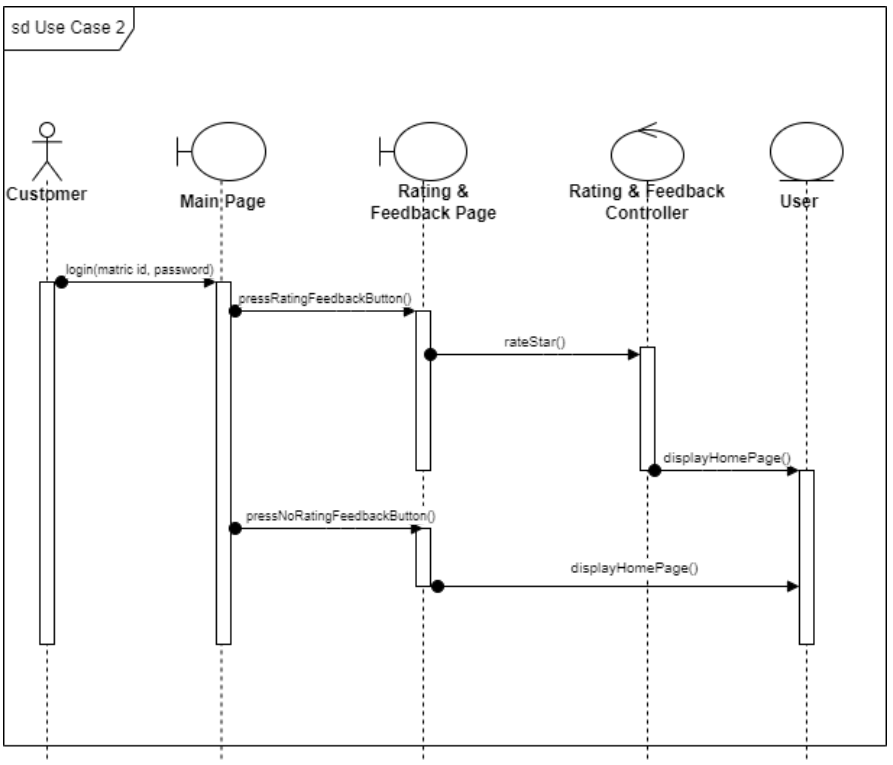


Figure 2.3.1: Sequence Diagram for Rating and Feedback as Customer (UTM Student)

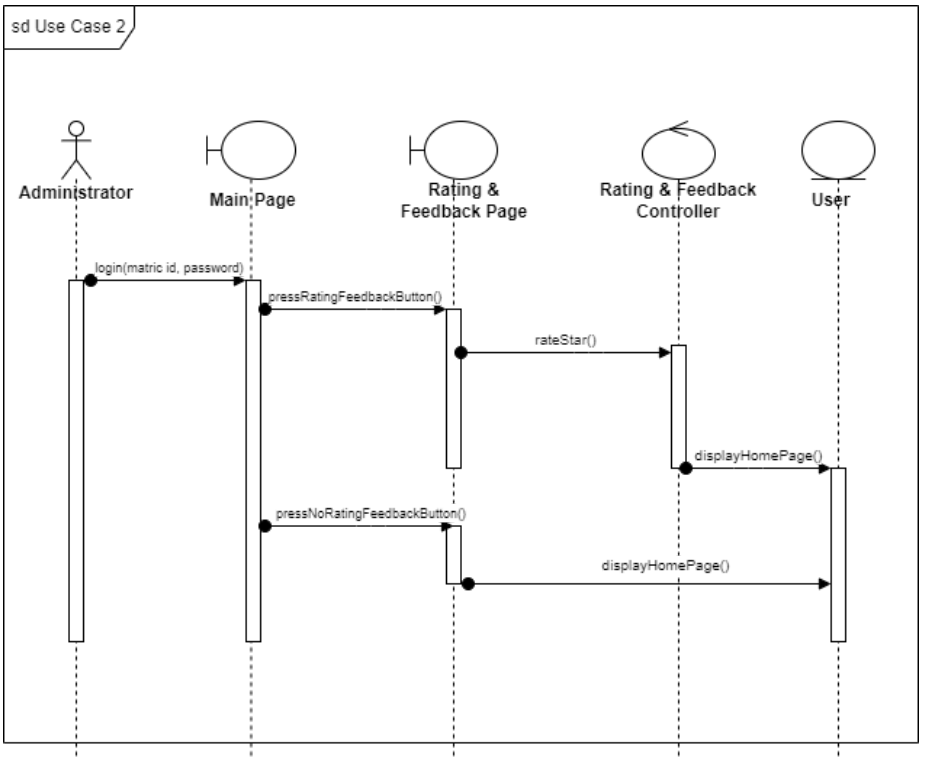


Figure 2.3.2: Sequence Diagram for Rating and Feedback as Administrator

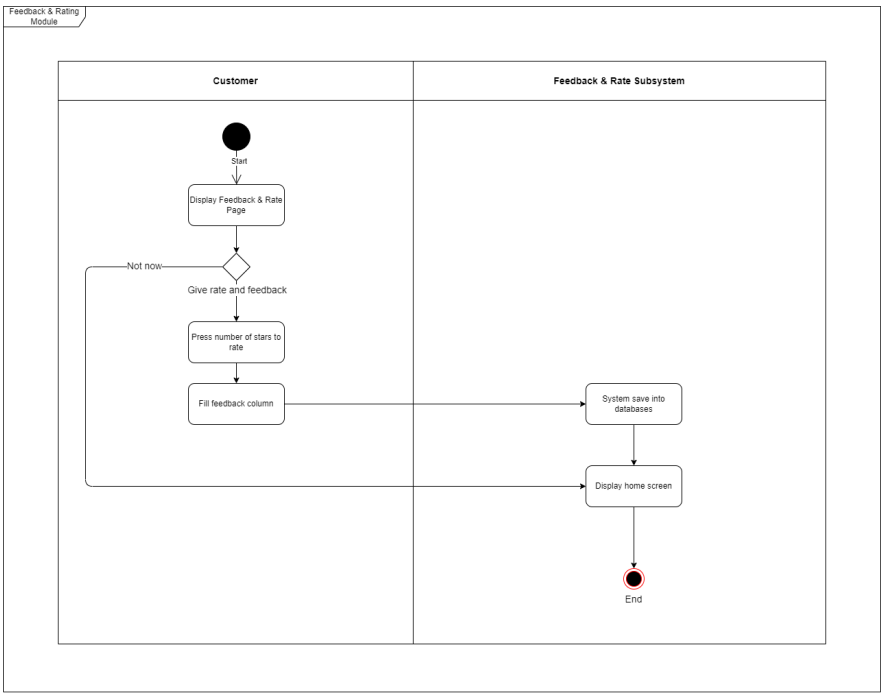


Figure 2.3.3: Activity Diagram for Feedback and Rating as Customer (UTM Student)

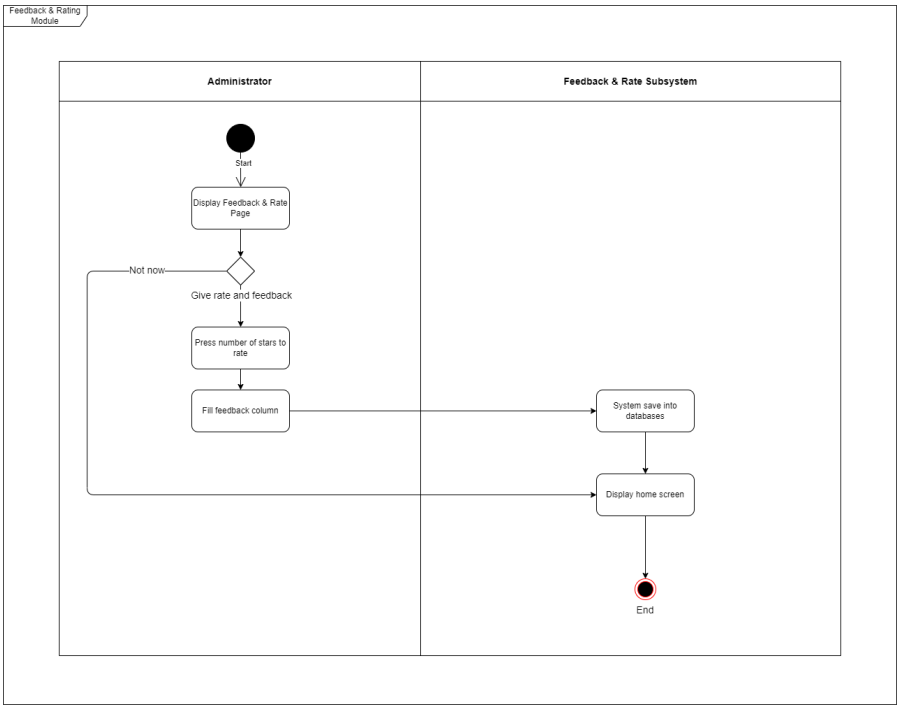


Figure 2.3.4: Activity Diagram for Feedback and Rating as Customer (UTM Student)

2.2.4 UC*004* : *Manage menu*

Table 2.4 : Use Case Description for Manage menu

Use case :*Manage menu*
ID: UC004
Actors: Administrator
Preconditions: Has a good internet connection
Flow of events: 1. Open browser menu page. 2. System displays all menu and remaining stock. 2.1 Admin tap on which menu they want to add or delete its quantity. 2.2 Next, the admin need to tap “Done” button to confirm their changes and update into the system. 3. System update into the databases.
Postconditions: Has actual number of stock remaining.

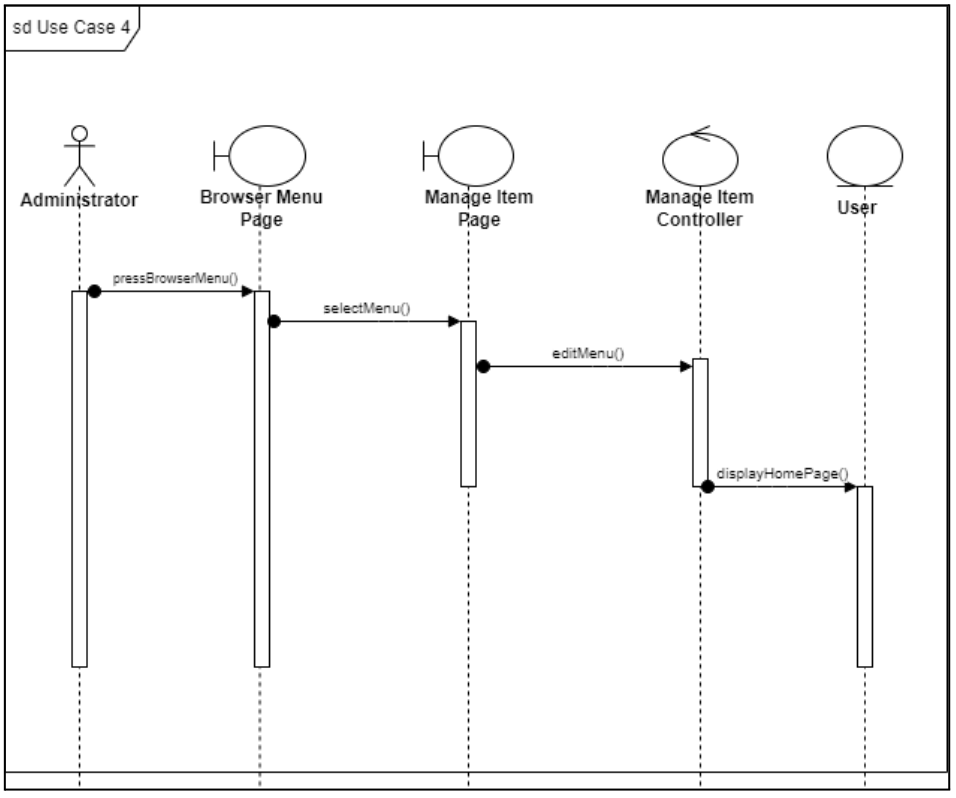


Figure 2.4.1: Sequence Diagram for Manage Menu as Administrator

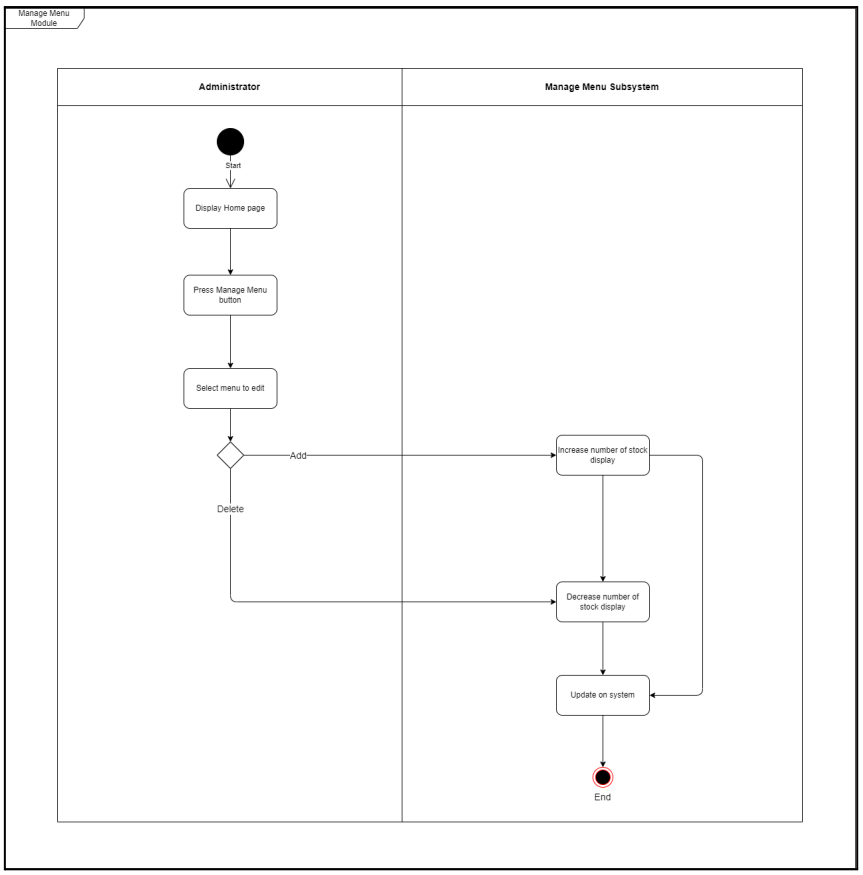


Figure 2.4.2: Activity Diagram for Manage Menu as Administrator

2.2.5 UC*005* : *Browse menu*

Table 2.5 : Use Case Description for Browse menu

Use case : *Browse menu*
ID: UC005
Actors: Customer (UTM Student)
Preconditions:Has a good internet connection
Flow of events: 1. User press the "Browser Menu" button. 2. User choose which store they want to purchase either foods or drinks. 3. User enter the Browser Menu Page. 3.1 User can scroll down until they find their foods/drinks to order. 3.2 Otherwise, user can also use the search button to find their foods/drinks to order.
Postconditions: Display available current foods and drinks.
Exception flow (if any): 1. If foods/drinks are not available, the browser menu page will display not available. 2. If the store is currently closed, it will display closed in the store page.

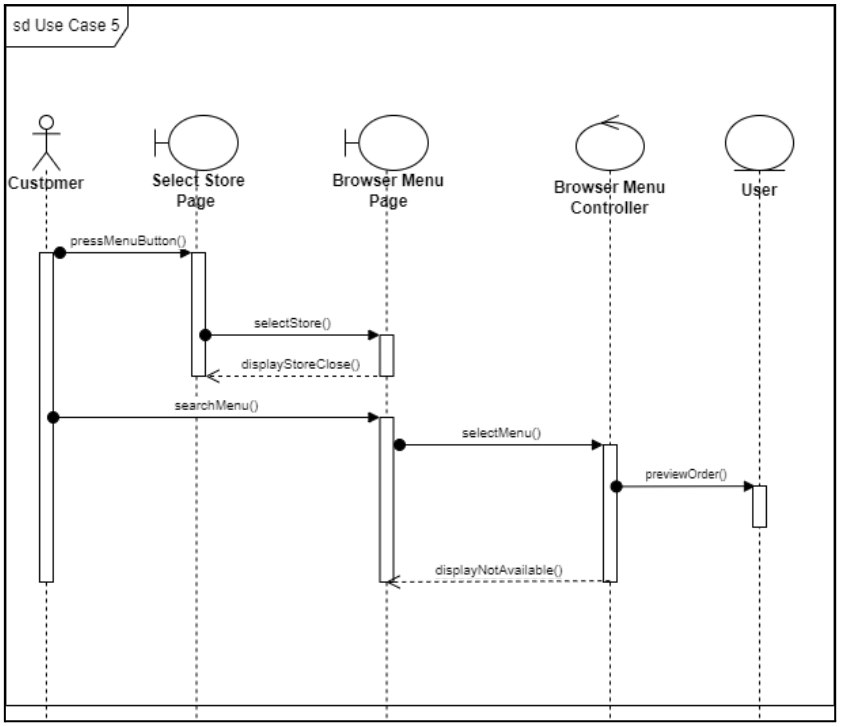


Figure 2.5.1: Sequence Diagram for Browser Menu as Customer (UTM Student)

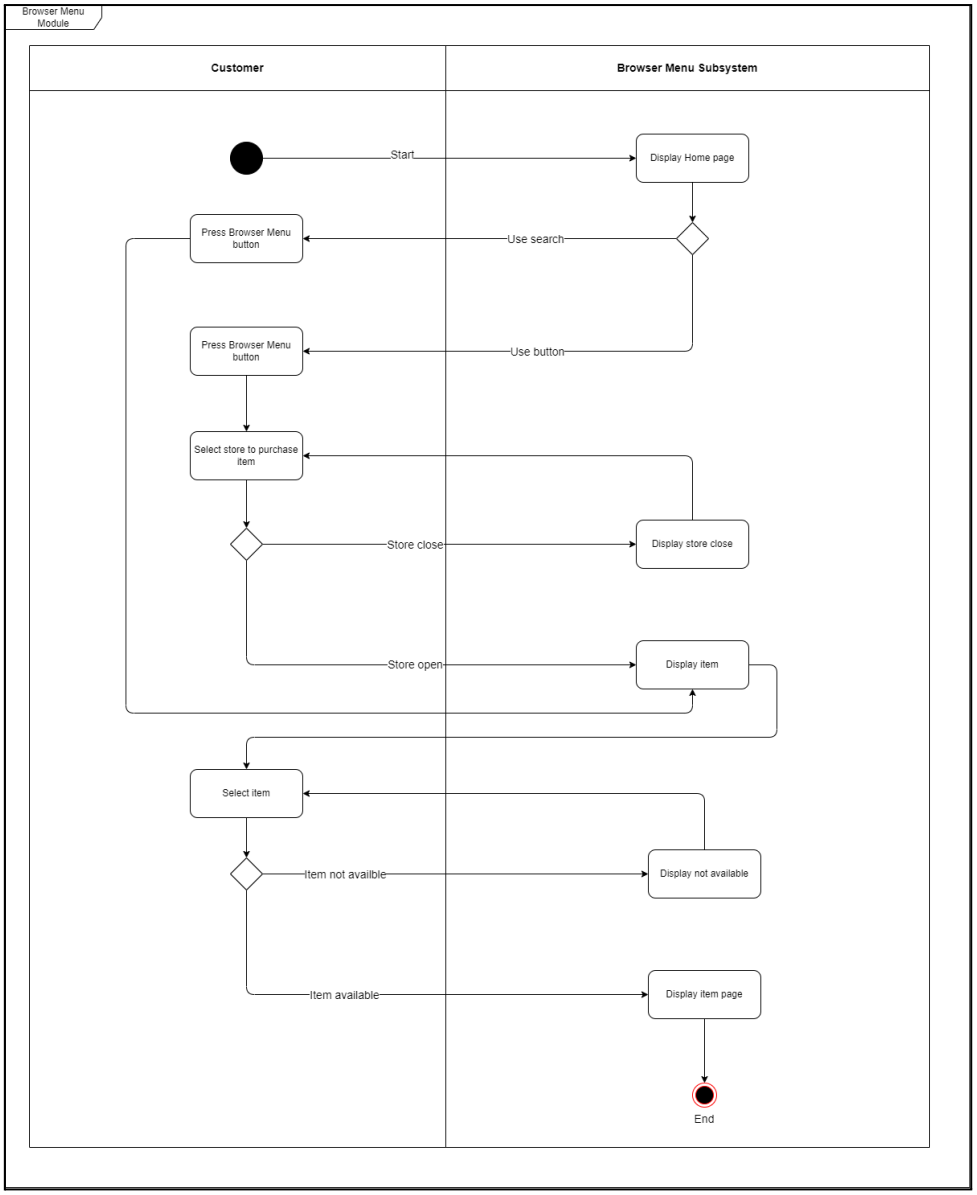


Figure 2.5.2 : Activity Diagram for Browser Menu as Customer (UTM Student)

2.2.6 UC*006* : *Placing order*

Table 2.6 : Use Case Description for Placing order

Use case : *Placing order*
ID: UC006
Actors: Customer (UTM Student)
Preconditions: 1. Has a good internet connection 2. Store is currently open.
Flow of events: 1. User enter the Browser Menu page. 2. User select item. 3. User customize order and add to cart. 4. If the user want to add more but it must be from the same store. 4.1 User select item. 4.2 User customize order and add to cart. 5. User make an order and redirect to the payment page. 6. System save the order into databases.
Postconditions: User can track the order by press “Track Order” button.

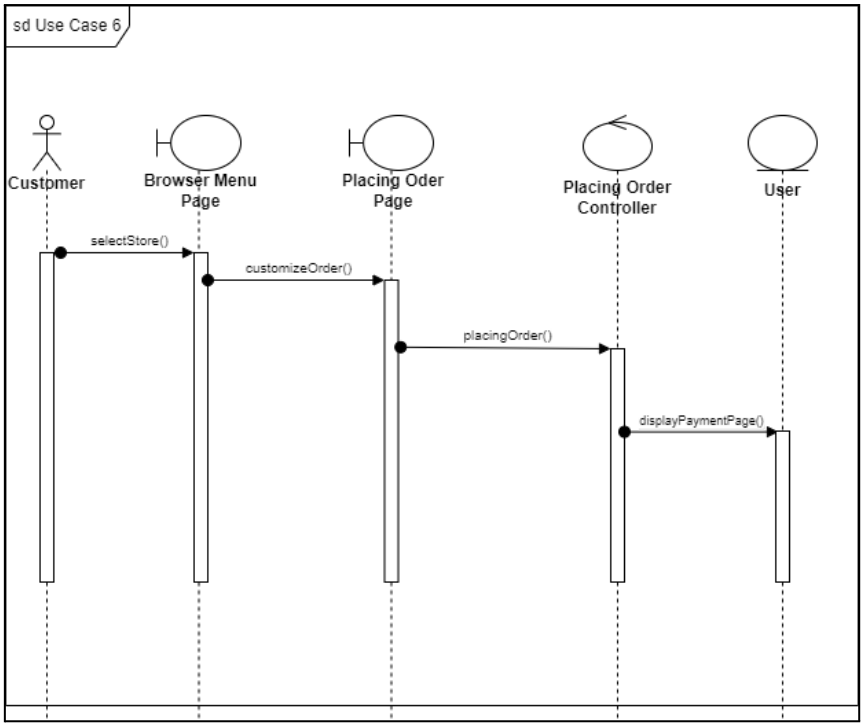


Figure 2.6.1: Sequence Diagram for Placing Order as Customer (UTM Student)

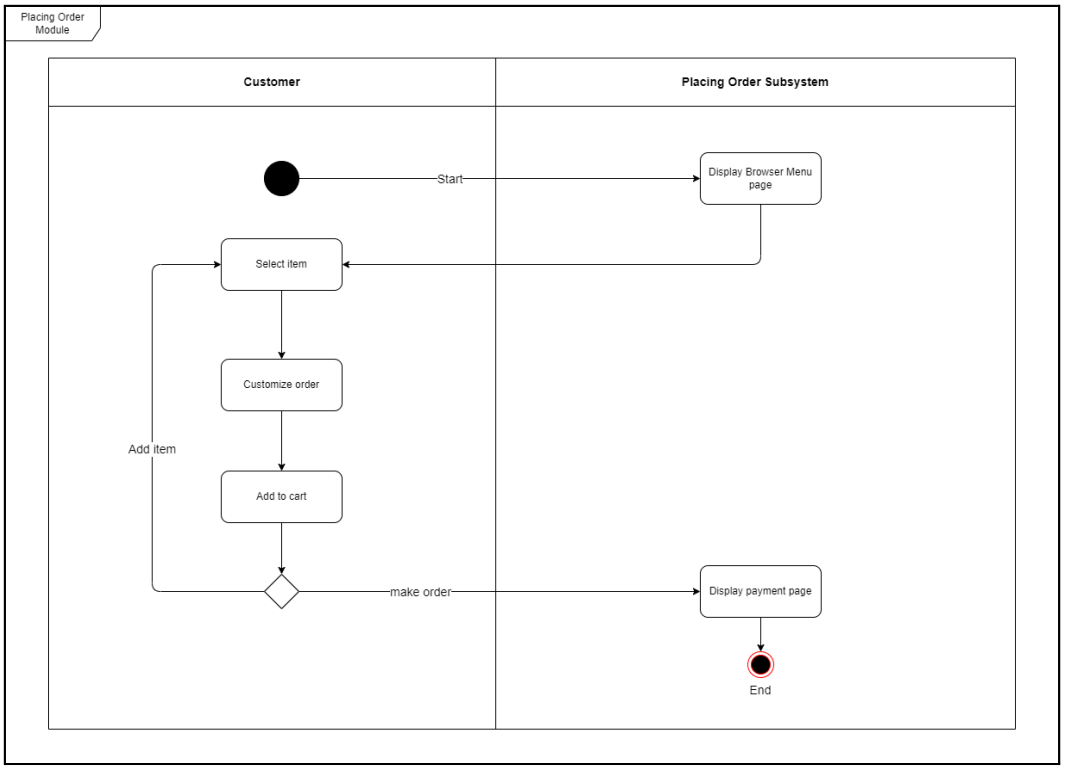


Figure 2.6.2: Activity Diagram for Placing Order as Customer (UTM Student)

2.2.7 UC*007* : *Payment process*

Table 2.7 : Use Case Description for Payment process

Use case : *Payment process*
ID: UC007
Actors: Customer (UTM Student)
Preconditions: 1.The customer has selected the items they want to purchase and added them to the cart. 2.The customer has reached the payment page.
Flow of events: 1. The customer is presented with the payment page, which displays the total amount to be paid and various payment options. 2. The customer selects a payment method from the available options, such as credit card, debit card, or mobile payment. 3. The customer reviews the payment details to ensure accuracy. 4. The customer submits the payment by clicking the "Pay" or "Confirm" button. 5. The system initiates the payment processing by communicating with the selected payment service provider.
Postconditions: 1.If the payment is successful, the customer's order is confirmed, and the details of the payment transaction are saved. 2. Customer receives order confirmation with order ID. 3. Customer payments are processed securely, and the appropriate amount is charged. 4. If a payment fails, the customer is notified of the failure and prompted to take corrective action.
Exception flow (if any): If the chosen payment method is not available or has a technical problem: 1. The system notifies the customer of the issue and provides alternative payment options, if available. 2. Customers can choose another payment method or try payment later. 3. The system logs the error for further investigation or resolution

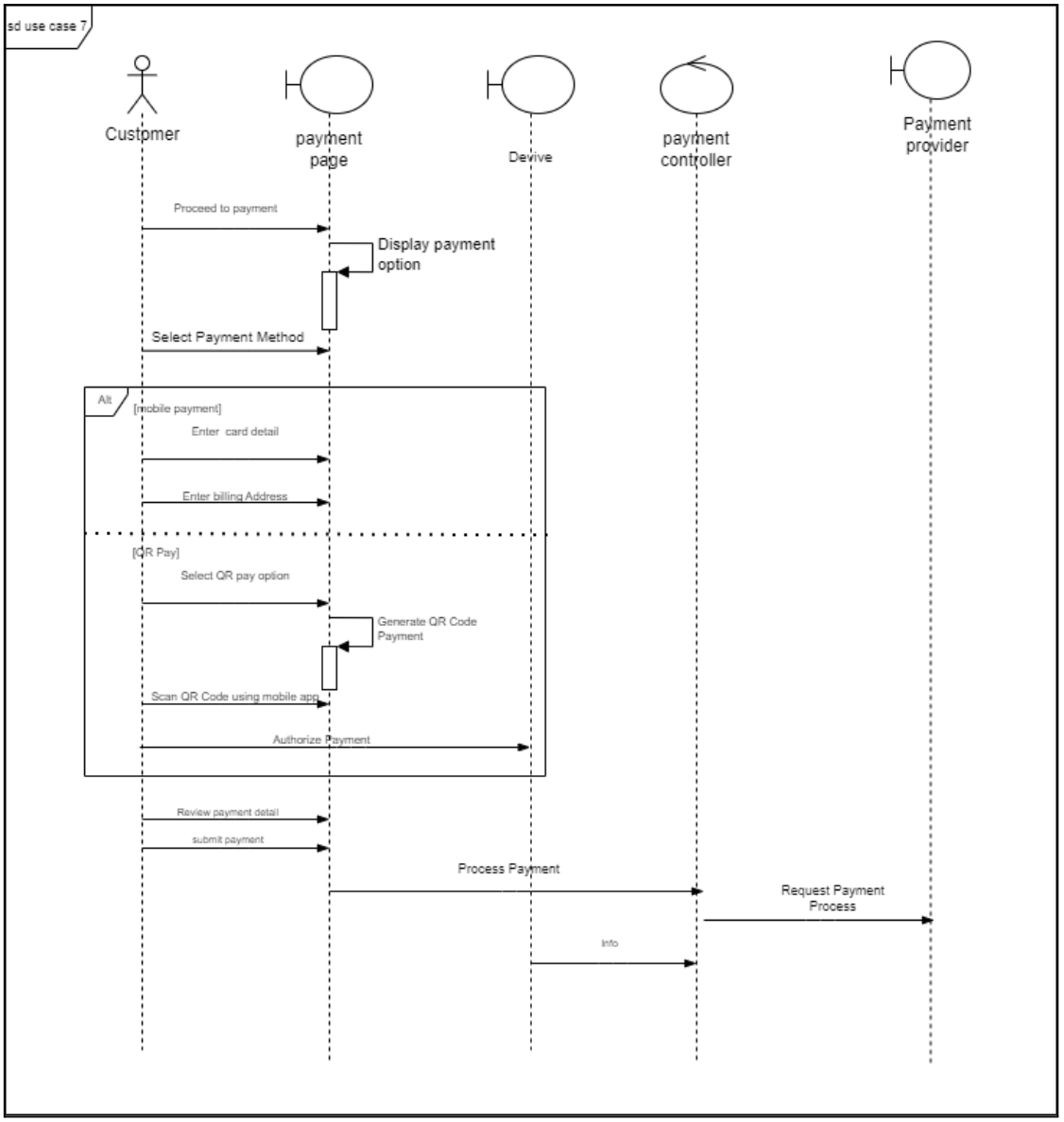


Figure 2.7.1 : Sequence Diagram for Payment Process as Customer (UTM Student)

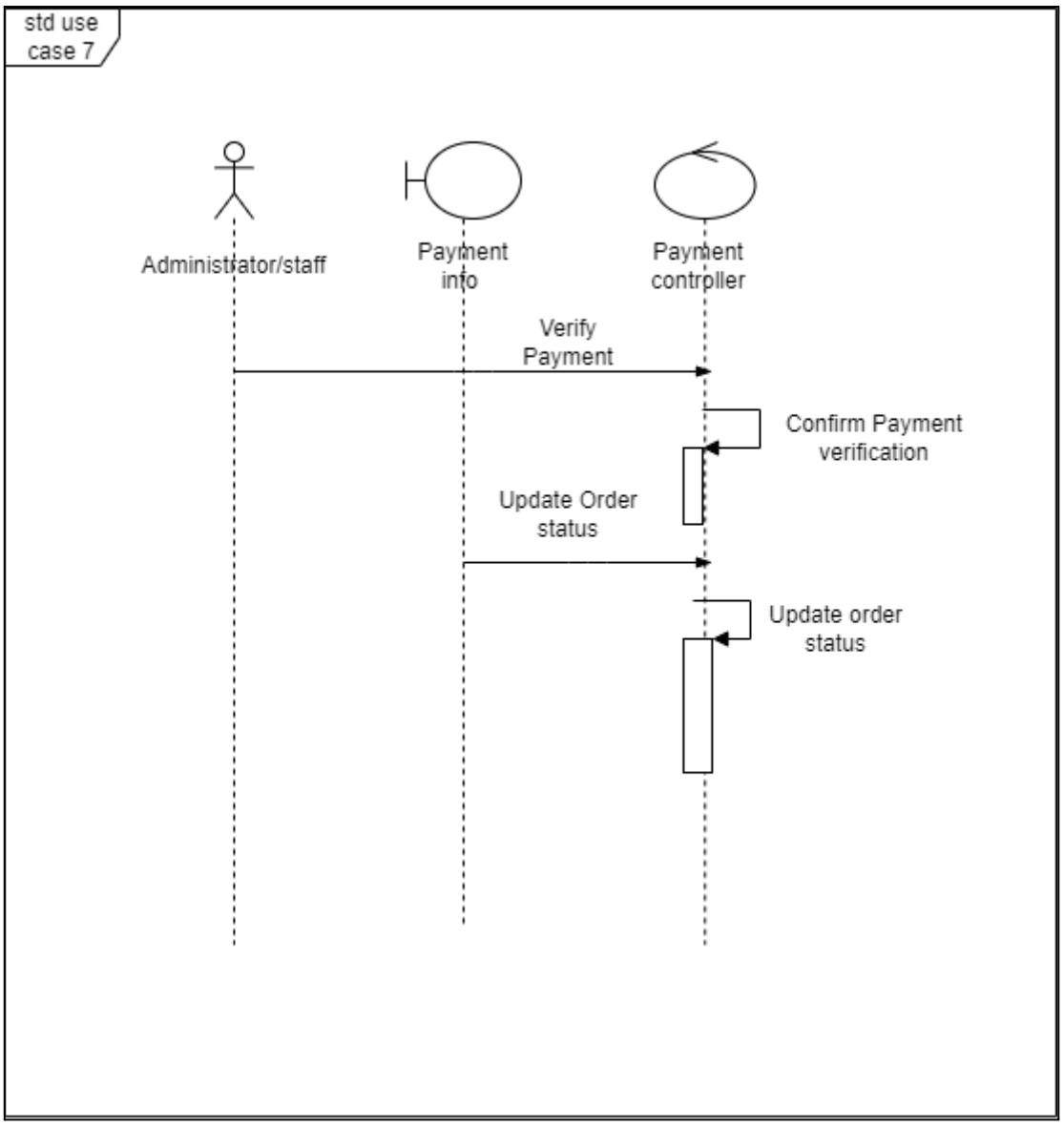


Figure 2.7.2 : Sequence Diagram for Payment Process as Administrator/Staff

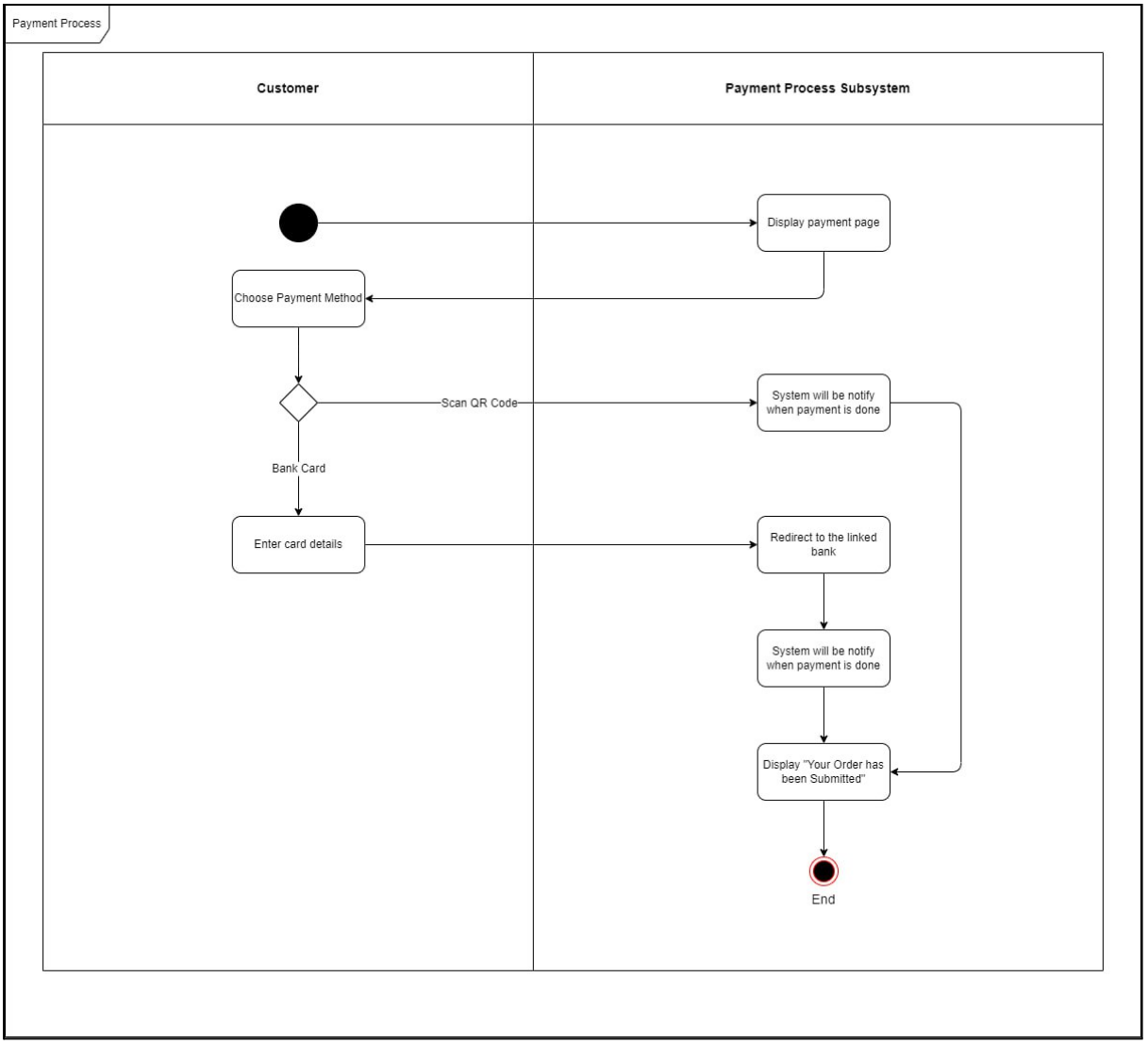


Figure 2.7.3 : Activity Diagram for Payment Process as Customer (UTM Student)

2.2.8 UC008 : Handle order

Table 2.8 : Use Case Description for Handle order

Use case : *Handle order*
ID: UC008
Actors:staff operator and Administrator
Preconditions: 1. Staff Operator/Administrator is logged into the system. 2. There is at least one pending order to be handled
Flow of events: 1. Staff Operator/Administrator accesses the order management system. 2. Staff Operator/Administrator views the list of pending orders. 3. Staff Operator/Administrator selects an order from the list. 4. System retrieves the details of the selected order. 5. Staff Operator/Administrator reviews the order details, including the customer information, items ordered, and delivery address (if applicable). 6. Staff Operator/Administrator performs the necessary actions based on the order requirements 7. Staff Operator/Administrator updates the order status and adds any relevant notes or comments. 8. System saves the updated order details in the database. 9. Staff Operator/Administrator notifies the customer about the order status, if required. 10. Staff Operator/Administrator repeats steps 3-9 for any other pending orders.
Postconditions: 1. The selected order is processed and its status is updated. 2. The order details are saved in the database. 3. The customer may be notified about the updated order status.
Alternative flow *n*: If there are no pending orders, the Staff Operator/Administrator may be notified or prompted to check again later.
Postconditions:
Exception flow (if any): If there are any issues encountered during order handling (e.g., item unavailability, delivery complications), the Staff Operator/Administrator takes appropriate actions and updates the order status accordingly. The customer may be notified about the issue and provided with alternative options or resolutions.

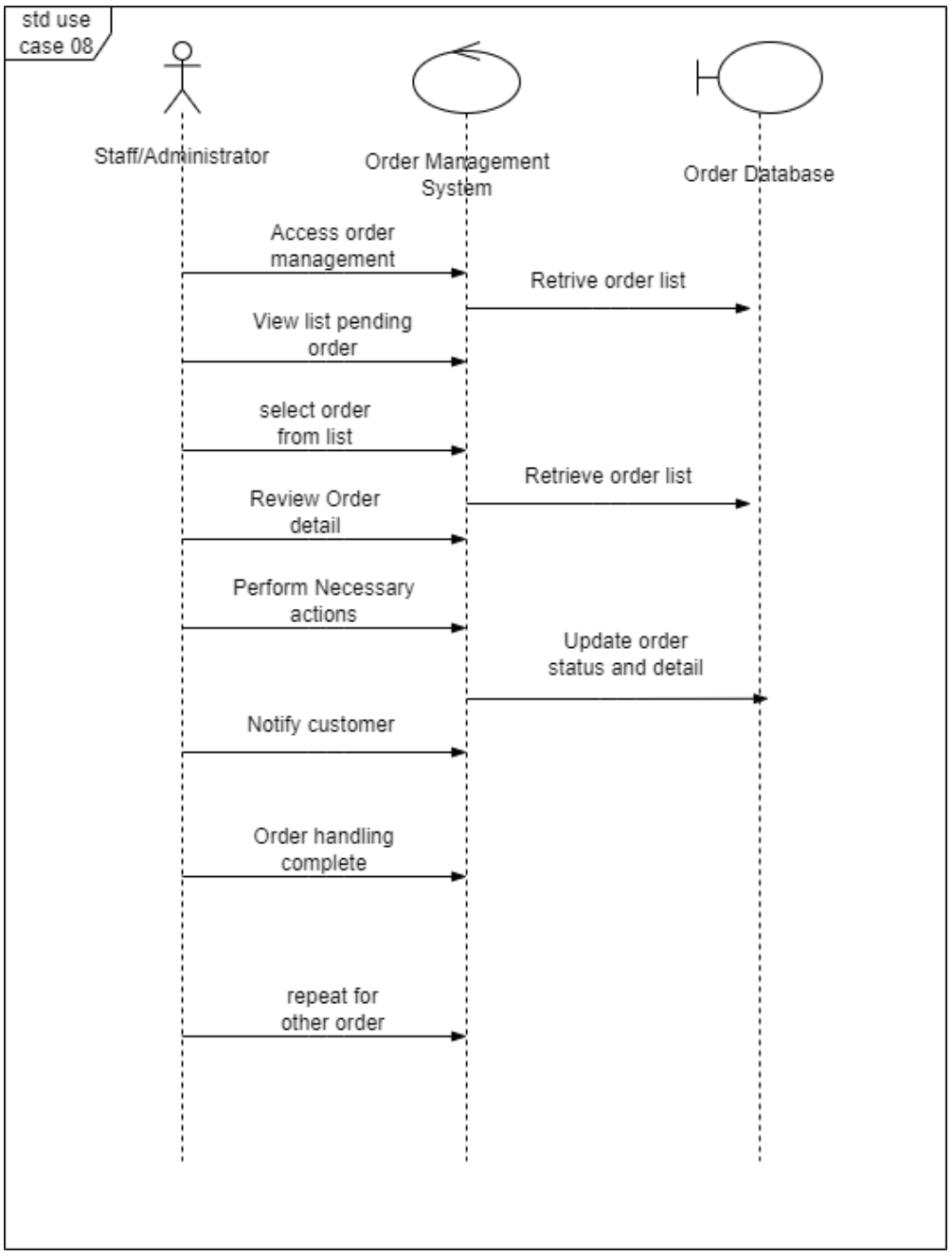


Figure 2.8.1 : Sequence Diagram for Handle Order as Administrator/Staff

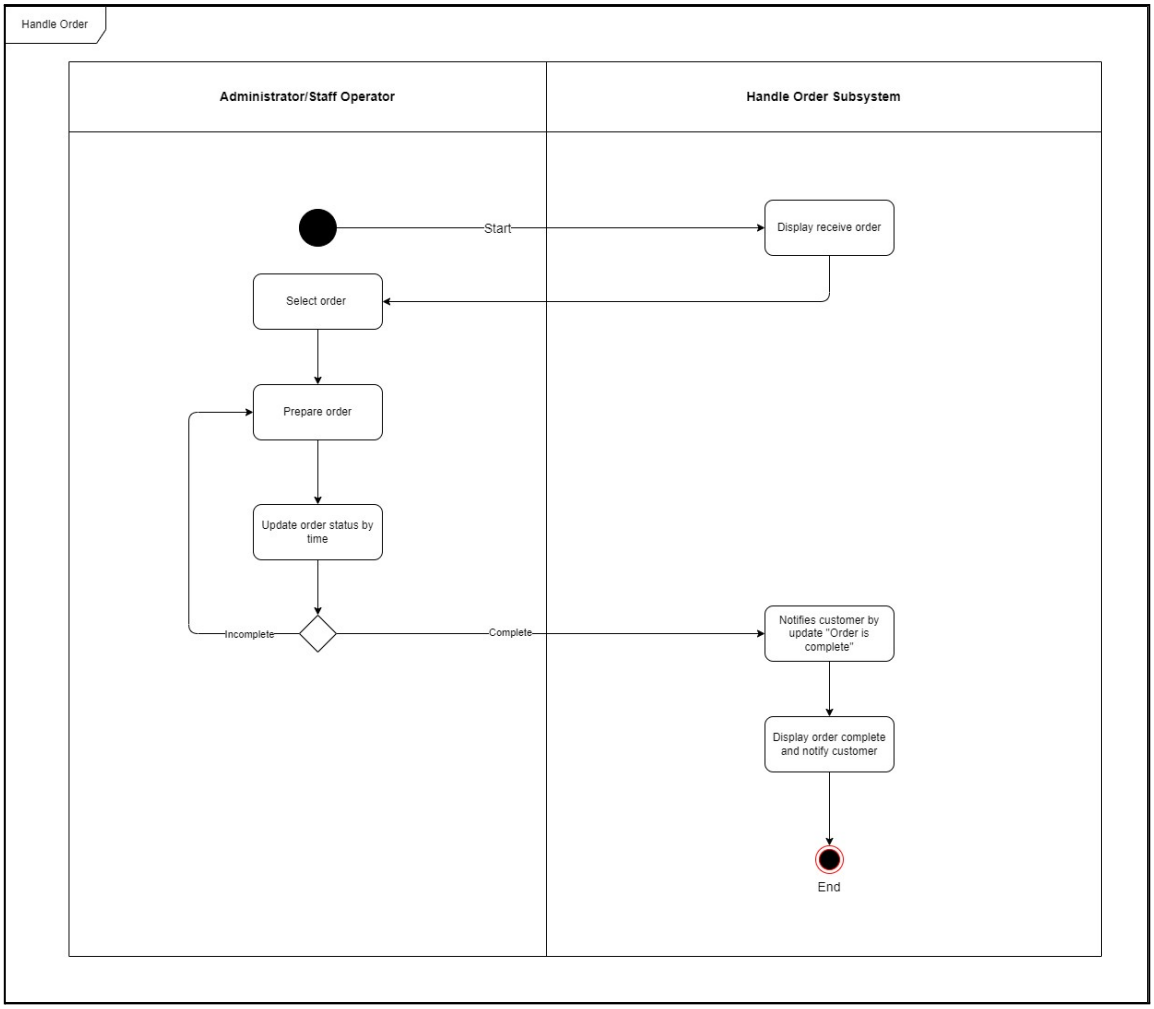


Figure 2.8.2 : Activity Diagram for Handle Order as Administrator/Staff

2.3 Software System Attributes, Performance and Other Requirements

1. Software system attributes:
    - Usability: The ordering process for meals should be simple to use and understand, with an easy-to-use interface that enables both consumers and restaurant personnel to work effectively. The system's overall usability is influenced by clear instructions, suitable feedback, and logical processes.
    - Availability: Users should always have access to the system, with little downtime required for maintenance or upgrades. High availability may be achieved by using redundancy, load balancing, and failover techniques.
    - Scalability: The system must be built to accommodate an increased user base, more orders, and more restaurant partners. The capacity to extend physical resources, such as servers and databases, as well as software components to meet growing demand without suffering performance loss is referred to as scalability.
    - Compatibility: The system should be constructed and designed to work with a variety of platforms and gadgets. It ought to support a variety of operating systems, including Windows, macOS, Linux, iOS, and Android, as well as web browsers like Chrome, Firefox, Safari, and Internet Explorer. This guarantees that more users, regardless of chosen platform or device, can access and use the system. In order to provide a consistent and smooth user experience across all platforms and screen sizes, such as desktops, laptops, tablets, and smartphones, the system must be responsive and adaptable.
    - Maintenance and support: The seamless functioning, issue repairs, and upgrades of the system depend on ongoing maintenance. A support system should be in place for the system to respond to user questions, offer assistance, and quickly fix problems.

2. Performance requirements:
    - Real-time Order Processing: For effective and fast delivery, a meal ordering system should be able to process orders in real-time. To reduce any unneeded waits or delays, improve the order processing procedure. To simplify the entire

order fulfillment process, implement systems for real-time inventory management and order status updates.

- Responsiveness: The capacity of a system to react quickly and effectively to user requests or actions is referred to as responsiveness. When navigating menus, making selections, and completing orders, customers want a quick and easy experience. A responsive system benefits from effective front-end coding techniques, server response time optimisation, and reduced network latency.
- Resource Optimization: The software and hardware resources should be used as efficiently as possible by the system. This entails effective disc I/O operations, CPU utilization, and memory management. Resource use may be decreased and system performance can be increased by using methods like object pooling, memory caching, and query optimisation.
- Network Efficiency: The system ought to be built to reduce network traffic and enhance data transfer. The amount of data payloads must be decreased, files must be compressed for quick delivery, and fewer network queries must be made for a given task. Reduce latency and boost system performance with effective network use.
- Scalability: The system must be able to handle a growing volume of orders, transactions, and concurrent users without noticeably degrading performance. The system should scale horizontally or vertically to meet the additional workload while maintaining optimal performance and response times as the user base expands or during peak hours.
- Availability: Availability describes a system's capacity to be reachable and functional by users at all times. Users might become irate and miss out on business possibilities as a result of downtime or unavailability. Redundancy techniques like load balancing and failover systems can be used to increase availability. This guarantees that the system can continue to function normally in the event that one server or component fails. Regular monitoring and preventative upkeep also aid in finding and fixing problems that could reduce availability.

3. Other Requirements:
    - User-friendly Interface: Customers should be able to simply explore menus, choose goods, customize orders, and make payments using the system's straightforward and user-friendly interface. The user interface needs to work well on a variety of screens and devices and be responsive.
    - Security: User data, such as personal data, payment information, and order history, should be secured by the system. Protect sensitive data by putting in place mechanisms like encryption, secure communication protocols (like HTTPS), and secure storage procedures. Mechanisms for user authentication and permission have to be in place to stop unapproved access to the system.
    - Menu and Inventory Management: Restaurant owners or administrators should be able to modify menus, adjust item availability, establish pricing, and handle inventory management using the system's user-friendly interface. This covers features including introducing new products, changing descriptions, describing dietary restrictions, and controlling ingredient stock levels.
    - Order Tracking and Status Updates: Customers should be able to check the progress of their orders in real-time via the system. This may incorporate functions like real-time delivery driver tracking, order confirmation alerts, and projected delivery timings. Customers are more satisfied when they receive regular information on the order status, covering the preparation, cooking, and delivery phases.
    - Payment Integration: To make online transactions easier, the system should interact with well-known, secure payment gateways. Customers are able to conveniently make payments using credit cards, digital wallets, or other preferred payment methods thanks to this. Refund processing and other payment-related difficulties should be handled efficiently and securely.
    - Customer Feedback and Ratings: Customers should be able to submit feedback, ratings, and reviews on their ordering experience, food quality, and delivery services through the system. This input can assist preserve customer satisfaction, promote continual improvement, and improve overall service quality.
    - Reporting and Analytics: For restaurant owners or administrators to gain insights and receive useful information, the system should offer reporting and

analytics capabilities. In doing so, information on order volume, sales trends, top sellers, client preferences, and other crucial indicators will be generated. Analytics tools can help with decision-making, marketing plans, and streamlining business processes.

2.4 Design Constraints

Design Constraints for the Food Ordering System:

1. Platform Compatibility:
    - Responsive Design: The user interface needs to be responsively developed to accommodate various screen sizes and resolutions. This guarantees that the system's design, functionality, and content are properly displayed and accessible across a range of gadgets, including desktop computers, smartphones, and tablets. Regardless of the device they are using, consumers may simply engage with the system thanks to responsive design.
    - Cross-Browser Compatibility: Popular online browsers like Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge should all work with the system. It is easier to make sure that a system works correctly and consistently across several browser platforms by designing and testing it on a variety of browser systems. To offer a consistent user experience, compatibility concerns such as layout inconsistencies or functionality failures should be rectified.
    - Operating System Compatibility: Different operating systems, including Windows, macOS, and Linux, should work with the system. This includes making sure that the system runs properly on these platforms, that any platform-specific requirements are taken into account, and that all relevant dependencies are present. To find and fix any compatibility problems, each target operating system should undergo compatibility testing.
2. Organizational Standards:
    - The organization's standards and policies established by the university must be followed by the system. This involves adhering to design principles, branding rules, and other pertinent policies.

3. Performance and Scalability:
    - The system must be able to handle a sizable user base and a big number of orders. To ensure the system's performance and scalability, designing for scalability, effective data management, load balancing, and caching methods are crucial concerns.
4. Usability and User Experience:
    - A food ordering system must have an interface that is simple to use and intuitive. Users should find it simple to explore menus, alter orders, follow deliveries, and submit payments. Critical design restrictions include taking into account user experience design principles, conducting usability tests, and incorporating user feedback.