Online Food Ordering System

1 Chapter 1

1.1 Definition of the Project

1.1.1 Project Purpose

- Enhancing Customer Experience: The main objective of this project is to create a user-friendly and interactive platform that allows customers to effortlessly browse through the menu, customize their food options, and conveniently place their orders from the comfort of their homes or offices. By offering a seamless and enjoyable online ordering experience, our project aims to significantly improve customer satisfaction and foster long-term loyalty.

- Quick Order Placement: Our system aims to automate the order-taking process, eliminating the need for manual order handling and minimizing the risk of errors or miscommunication. By streamlining the order placement process, customers can easily and efficiently place their orders, resulting in time savings for both the customers and the restaurant.

- Real-time Menu Updates: Our system will offer a dynamic and interactive menu that provides real-time updates, presenting customers with all available options and adjusting prices dynamically based on their selections. This feature guarantees that customers have access to accurate and up-to-date information while making their food choices.

- Efficient Order Processing: By displaying customer orders in real-time on the restaurant's screen, the system aims to enable the restaurant staff to promptly process incoming orders, minimizing delays and confusion. This efficiency helps the restaurant deliver orders faster, improving customer satisfaction and potentially increasing revenue.

- Streamlined Delivery Process: Our project aims to streamline the delivery process by implementing a delivery tracking system and providing regular status updates to customers. This feature ensures that customers are kept informed about the progress of their order and estimated delivery times, promoting transparency and building trust. By enhancing the overall delivery experience, our project seeks to improve customer satisfaction and create a stronger sense of reliability.

- Reduce Manual Workload: The project aims to automate the order taking process, relieving the restaurant staff of manual tasks and allowing them to dedicate more time to food preparation and customer service. By streamlining operations and reducing the likelihood of errors, the system optimizes efficiency and enhances overall performance.

  In summary, the project aims to develop a robust online food ordering system that elevates the customer experience, streamlines order processing for the restaurant, and enhances the efficiency of the food delivery process. By creating a comprehensive platform, the project seeks to deliver a seamless and convenient ordering experience, optimize operational workflows, and ultimately provide a positive impact on both customers and the restaurant.

1.1.2 Project Scope

Customer-Facing Website or Application: Our project involves designing and developing a user-friendly and responsive website or mobile application for customers to access our online food ordering system. The primary goal is to create a visually appealing and intuitive interface that allows customers

to effortlessly browse the menu, view item details, and make selections. We will incorporate a search functionality to enable quick item or category searches. To enhance the user experience, customers will be able to create accounts, log in, and manage their profiles. The system will also feature a secure payment gateway for seamless online transactions. Customers will have the flexibility to add items to their carts, customize selections, and specify delivery or pickup preferences. A cart summary feature will provide an overview of the selected items, quantities, and prices, allowing customers to review and modify their orders before finalizing the checkout process. To ensure smooth order placement and confirmation, we will implement user-friendly procedures, including order confirmation emails or notifications.

Interactive and Up-to-Date Menu: Our project focuses on designing and developing a dynamic menu management system that empowers restaurants to update menu items, descriptions, prices, and availability in real-time. We will create a user-friendly interface that enables restaurants to effortlessly add new items, update existing items, and manage their menu categories. The system will also support the addition of custom options or variations for menu items, such as toppings, sizes, or preferences, and will dynamically adjust prices based on these choices. To ensure seamless updates, we will implement an automated system that reflects menu changes immediately on the customer-facing website or application, providing accurate and up-to-date information for customers.

Order Management for Restaurants: Our project aims to design and develop a restaurant dashboard or interface that provides real-time visibility of incoming orders. The dashboard will offer a clear and intuitive view of order details, including itemized lists, customizations, delivery/pickup information, and payment status. We will enable restaurants to conveniently accept, reject, or modify orders as needed, with timely notifications sent to customers. To enhance order management, we will implement features to track the progress of order preparation, such as order status updates or timers. Additionally, the system will empower restaurants to generate comprehensive reports and analytics, offering insights into order volume, popular items, and customer feedback.

Real-time Order Notifications: Our project will implement a robust notification system to deliver real-time updates to both customers and restaurants. We will configure automatic email or push notifications to promptly notify customers about their orders, including order confirmation, estimated delivery times, and any updates or changes. Simultaneously, we will establish a seamless notification mechanism for restaurants, ensuring they receive instant alerts about new orders and can access relevant order details directly in their dashboard or interface.

Secure User Authentication and Data Management: Our project will prioritize the implementation of secure user authentication mechanisms to safeguard customer and restaurant login information. We will employ industry-standard practices like password hashing to ensure the confidentiality of sensitive data. Additionally, we will focus on ensuring the secure transmission of information, particularly payment details, by implementing encryption protocols. To further strengthen security, we will establish a robust database management system to securely store and manage customer profiles, order history, and restaurant information.

Delivery Tracking and Status Updates (optional): We will consider incorporating a real-time delivery tracking feature in our project if it falls within the project scope. This feature will allow customers to monitor the progress of their deliveries in real-time. We plan to integrate with third-party APIs or services to obtain accurate location data, which will be displayed to customers either on a map or through status updates. Our goal is to provide customers with timely notifications about the estimated arrival time of their orders, as well as any potential delays or changes in the delivery schedule. It's important to note that the specific implementation of these features may be subject to project requirements, budget constraints, and timeline considerations.

1.1.3 Project Constraints

- Time Constraints: The project must adhere to a specific timeline or deadline, requiring the completion of development, testing, and deployment within the allocated time frame.

- Budget Constraints: The project needs to be carefully planned and executed to stay within the assigned budget, influencing decisions related to technology choices, feature prioritization, and resource allocation.
- Technology Constraints: The project may have specific technology requirements, such as using particular programming languages, frameworks, or database systems. These constraints can be influenced by existing infrastructure, compatibility needs, or integration with other systems.
- Performance Constraints: The online food ordering system should be designed to handle a significant number of concurrent users and transactions without any performance degradation. It must ensure a seamless and responsive user experience, even during peak periods.
- Security and Privacy Constraints: The system should prioritize the security and privacy of customer data, including personal information, payment details, and order history. It should strictly adhere to industry-standard security practices to safeguard against data breaches and unauthorized access.
- Integration Constraints: The online food ordering system may require integration with existing systems or third-party services, such as payment gateways, delivery tracking services, or inventory management systems. These integrations should be effectively planned and implemented to ensure smooth data flow and seamless functionality.
- Regulatory Constraints: The project should comply with relevant legal and regulatory requirements, such as data protection laws, online payment regulations, and food safety standards. The system should be designed to handle and store data in accordance with these regulations.
- Scalability Constraints: The system should be designed to accommodate future growth and handle increasing user and transaction volumes. It should be scalable, allowing for easy expansion and the addition of new features and functionalities as the business grows.
- User Experience Constraints: The system should prioritize user experience and usability to ensure that customers find it easy and intuitive to navigate, place orders, and customize their selections. The interface should be responsive, accessible, and visually appealing across various devices and screen sizes.
- Operational Constraints: The system should be reliable, stable, and easy to maintain. It should be designed to minimize downtime, allow for regular backups and updates, and provide an effective error handling and reporting mechanism.

1.2 Actor Glossary

Customer: The customer is the primary actor in the system. They interact with the online platform, either through a website or mobile application, to browse the menu, select items, customize options, and place orders. Customers may create accounts to save their delivery addresses, track order history, and receive personalized offers. They can also make online payments and provide feedback on their ordering experience. The customer actor's actions include browsing the menu, adding items to the cart, specifying delivery or pickup preferences, making payments, and reviewing order details.

Restaurant Staff: The restaurant staff refers to the employees who handle the food preparation and order fulfillment at the restaurant. They have access to the order management interface provided by the online food ordering system. The staff can view incoming orders in real-time, update the status of orders, accept or reject orders, and communicate with customers if necessary. They are responsible for preparing the food items, ensuring the order accuracy, and packaging them for delivery or pickup. The restaurant staff actor's actions include viewing and managing incoming orders, updating order status, communicating with customers, and preparing food items.

Administrator: The administrator is an actor responsible for managing and maintaining the online food ordering system. They have privileged access to the system's backend and administrative interface. Administrators oversee the overall functioning of the system, manage user accounts, monitor system performance, handle system configurations, and resolve any technical issues that may arise. They also have the authority to add or remove restaurants from the platform, manage menu items, and generate reports. The administrator actor's actions include managing user accounts, handling system configurations, monitoring system performance, managing restaurants and menus, and resolving technical issues.

1.3 Use Cases

Browse Menu and Place Order: Description: The customer browses the menu, selects desired food items, customizes options (if available), and adds them to the cart. Actors: Customer Preconditions: Customer is logged in and the menu is available. Basic Flow: Customer selects a category or uses search functionality to explore the menu. Customer views item details, including descriptions, prices, and available options. Customer selects desired options (e.g., toppings, sizes) and specifies quantities. Customer adds items to the cart. Customer reviews the order and proceeds to checkout.

Manage Order: Description: The customer manages their order, making changes, adding or removing items, and reviewing order details. Actors: Customer, Administrator Preconditions: Customer has an existing order. Basic Flow: Customer accesses their order from the cart or order history. Customer reviews the order details, including itemized list, quantities, and prices. Customer can modify quantities, add or remove items, or update options. Customer saves the changes or proceeds to checkout. Administrator can manage the details.

Track Order Status: Description: The customer tracks the status and delivery progress of their order in real-time. Actors: Customer Preconditions: Customer has placed an order. Basic Flow: Customer accesses the order details or order history. Customer views the current status of the order (e.g., order received, order being prepared, out for delivery). If available, customer can view real-time updates on the delivery progress, including estimated arrival time. Customer may receive notifications or updates regarding any delays or changes in the order status.

Accept/Reject Order: Description: The restaurant staff receives and manages incoming orders, accepting or rejecting them based on availability and other factors. Actors: Restaurant Staff Preconditions: Order is received by the restaurant staff. Basic Flow: Restaurant staff accesses the order management interface. Staff views the details of incoming orders, including items, customer information, and delivery/pickup preferences. Staff checks the availability of items and resources to fulfill the order. Staff accepts the order and begins preparing the items. If unable to fulfill the order, staff may reject it and provide a reason if necessary.

Update Order Status: Description: The restaurant staff updates the status of an order, indicating the progress of preparation and delivery. Actors: Restaurant Staff, Administrator Preconditions: Order is being prepared or out for delivery. Basic Flow: Restaurant staff accesses the order management interface. Staff selects the order and updates its status based on the progress (e.g., preparing, out for delivery). If applicable, staff enters additional details such as the estimated delivery time or driver information. Staff may send notifications or updates to the customer regarding the status change. Administrator has permission to access updating and controlling it.

Generate Sales Report: Description: The administrator generates sales reports to analyze the performance of the online food ordering system. Actors: Administrator Preconditions: Administrator is logged in and has access to the reporting functionality. Basic Flow: Administrator accesses the reporting section of the system. Administrator selects the desired time period for the report (e.g., daily, weekly, monthly).

1.4 Project Effort Estimation

- Browse Menu and Place Order: High (5)
- Manage Order: Medium (3)
- Track Order Status: Medium (3)
- Accept/Reject Order: High (5)
- Update Order Status: Medium (3)
- Generate Sales Report: Low (1)
- TOTAL UCP: 5+3+3+5+3+1= 20

Productivity Factor: 5 person-days per Use Case Point

Estimated Person-Days = 20*5 =100

Effort Per Developer = 100/3= 33

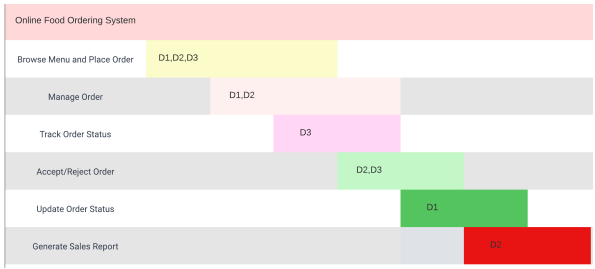

Figure 1: Gantt Chart

2 Chapter 2

2.1 Non-functional Requirements

- Performance Requirements:

Requirement 1: Response Time It is essential for the system to deliver swift response times, guaranteeing a smooth and uninterrupted user experience. The system should strive to maintain an average response time within a specified threshold, such as 2 seconds, for menu loading, item selection, and the checkout process. This requirement aims to enable customers to swiftly navigate the menu, select their desired items, and seamlessly complete their orders without facing significant delays or encountering any frustration.

Requirement 2: Scalability The system's design should be robust enough to accommodate a substantial surge in user traffic and transaction volumes without experiencing significant performance degradation. It should be capable of horizontal scalability, effectively distributing the workload across multiple servers or instances. This approach ensures that response times and system availability are maintained, even during peak usage periods. By meeting this requirement, the system can effectively handle a growing customer base and seamlessly manage high demand during busy periods while upholding optimal performance standards.

- Security and Privacy Requirements:

Requirement 1: Data Encryption To safeguard sensitive user data, such as personal information, payment details, and order history, the system must implement robust encryption mechanisms, such as SSL/TLS. It should ensure that all data transmitted between the client and server is securely encrypted, thereby preventing unauthorized access or interception. By fulfilling this requirement, the system guarantees the confidentiality and integrity of customer data during transmission, bolstering overall data security.

Requirement 2: Access Control and Authorization To maintain stringent access control, the system must incorporate robust mechanisms that restrict access to specific functionalities and data within the system. Only authorized users, including customers, restaurant staff, and administrators, should be granted access. User authentication should be implemented to verify the identity of users, while appropriate authorization levels should be enforced to limit access to sensitive information and system functionalities. By fulfilling this requirement, the system ensures that only authorized individuals can access and manipulate data, effectively mitigating the risks associated with unauthorized access and potential data breaches.

2.2 Functional Requirements

- Menu Management:

Requirement 1: Dynamic Menu Updates The system should allow restaurant staff to easily update the menu, including adding new items, modifying existing items, and marking items as unavailable when out of stock. This requirement ensures that the menu displayed to customers is always up-to-date and accurately reflects the available options and prices.

Requirement 2: Customizable Menu Options The system should support the inclusion of customizable options for menu items, such as toppings, sizes, or preferences. Customers should be able to select these options during the ordering process and see the dynamic price adjustments based on their selections. This requirement allows for a flexible and personalized ordering experience, catering to individual customer preferences and customization requests.

- Order Management:

Requirement 1: Real-time Order Processing The system should enable real-time order processing to minimize delays and ensure a quick turnaround time for customers' orders. When a customer places an order, it should be immediately visible to the restaurant staff, allowing them to promptly start preparing the items. This requirement ensures efficient order fulfillment and reduces waiting times for customers.

Requirement 2: Order Status Tracking and Notifications The system should provide order status tracking capabilities for customers, allowing them to monitor the progress of their orders. Customers should receive real-time notifications regarding order confirmation, preparation updates, and delivery status. This requirement enhances transparency and improves the overall customer experience by keeping them informed about the progress of their orders.

3 Chapter 3

3.1 Use-Case Diagram

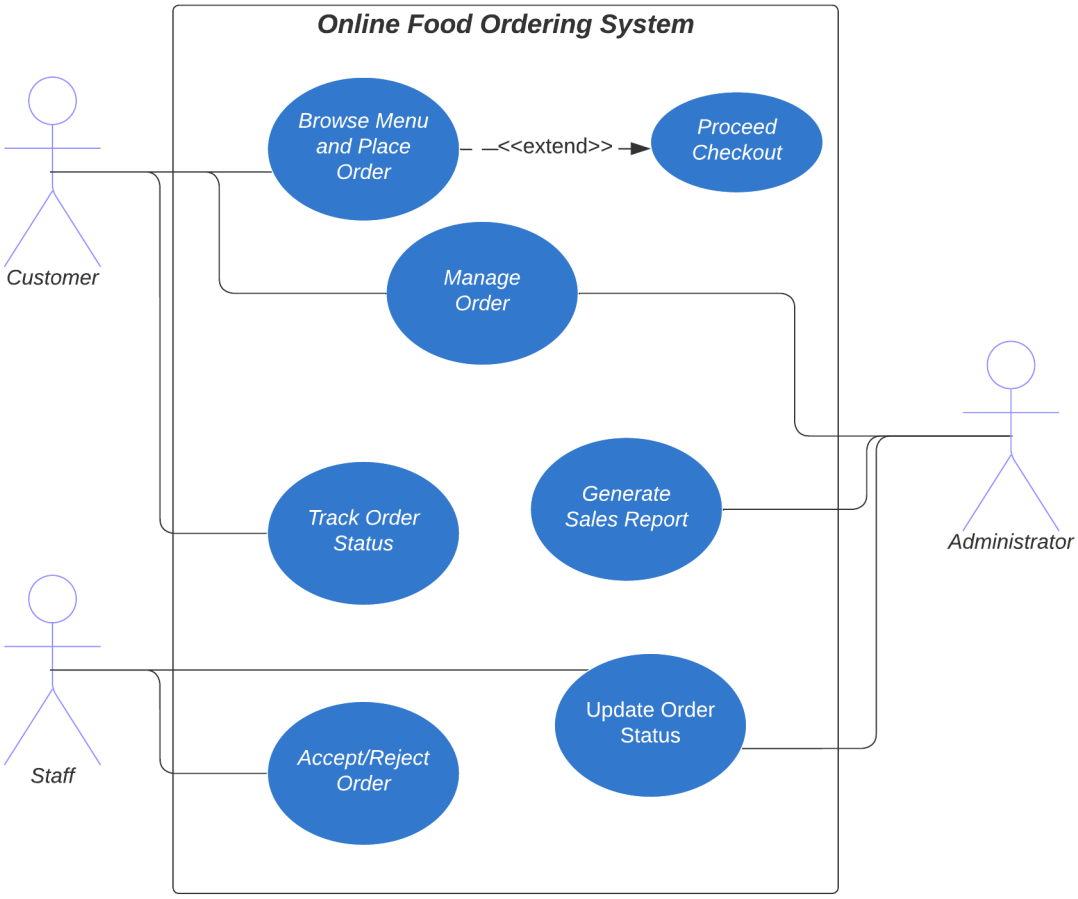

Figure 2: Use-Case Diagram

3.2 Use-Case Descriptors

Browse Menu and Place Order is conected to customer. Then it extends to proceed checkout. Manage Order use case is connected to both customer and administrator. Because administrator can see the order. Track Order Status use-case is only connected to customer so he/she can check the orders stage. Accept/Reject Order is connected to the restourant staff because they can cancel or approve the order depend on the situation. Then Update Order Status is connected to both staff and administrator. Finally the Generate Sales Report use-case only connected to administrator. And the main actors are Customer, Restaurant Staff, and Administrator.

Name: Browse Menu and Place Order | **ID:** UC001 |
İmportance Level: High

Description: The "Browse Menu and Place Order" use case allows the customer to explore the menu, select desired food items, customize options, and place an order through the online food ordering system. This use case is initiated when the customer accesses the system and selects the "Browse Menu" option.

Actors: Customer, Restaurant Staff, Administrator

Trigger: The customer accesses the online food ordering system

Relationships:
Actor: Customer
İnclude:
Extend: Proceed Checkout
Generalization:Action

Main Flow:

1. The customer accesses the online food ordering system and selects the "Browse Menu" option.
2. The system presents the customer with a categorized menu or search functionality to explore available food items.
3. The customer selects a category or uses the search functionality to view a list of food items within that category.
4. The system displays the menu items, including their names, descriptions, and prices.
5. The customer selects a specific menu item of interest.
6. The system provides detailed information about the selected item, such as ingredients, allergen information, and customer ratings.
7. If the item has customizable options (e.g., toppings, sizes), the customer selects the desired options.
8. The system updates the item's price based on the selected options.
9. The customer specifies the quantity of the item they wish to order.
10. The customer adds the selected item to their order cart.
11. The system updates the cart, displaying the added item with the selected options, quantity, and the updated total price.
12. The customer can continue browsing the menu and repeat steps 5 to 11 for additional items they wish to order.
13. Once the customer has finished selecting items, they review the contents of their order cart.
14. The customer has the option to make modifications to the order, such as updating quantities, removing items, or adding special instructions.
15. The customer proceeds to the checkout process, where they provide delivery or pickup details, select a payment method, and confirm the order.
16. The system validates the order and generates an order confirmation number.
17. The system sends an order confirmation notification to the customer via email or SMS, including the order details and estimated delivery/pickup time.

Subflows:

- Customizing Options

The customer selects a menu item that offers customizable options, such as toppings or sizes.
The system presents the available options and allows the customer to choose their preferences.
The system updates the item's price based on the selected options.

- Modifying Order in the Cart

After adding items to the cart, the customer reviews the order contents.
The customer can modify the quantity of items, remove items, or add special instructions.
The system updates the order cart accordingly, reflecting the customer's modifications.
Subflow: Searching for Specific Items
Instead of browsing categories, the customer uses the search functionality to find specific menu items.
The customer enters a search query.
The system filters the menu based on the search query and displays the relevant results.

- Handling Unavailable Items

The customer selects a menu item that is currently unavailable.
The system notifies the customer about the unavailability and suggests alternative options.
The customer can choose an alternative item or decide to proceed without that particular item.

Alternative Flows:

- If an item is unavailable: The system notifies the customer that the selected item is currently unavailable and suggests alternative options or provides a notification when it will be available again.

Figure 3: Use Case Descriptor

3.3 Activity Diagram

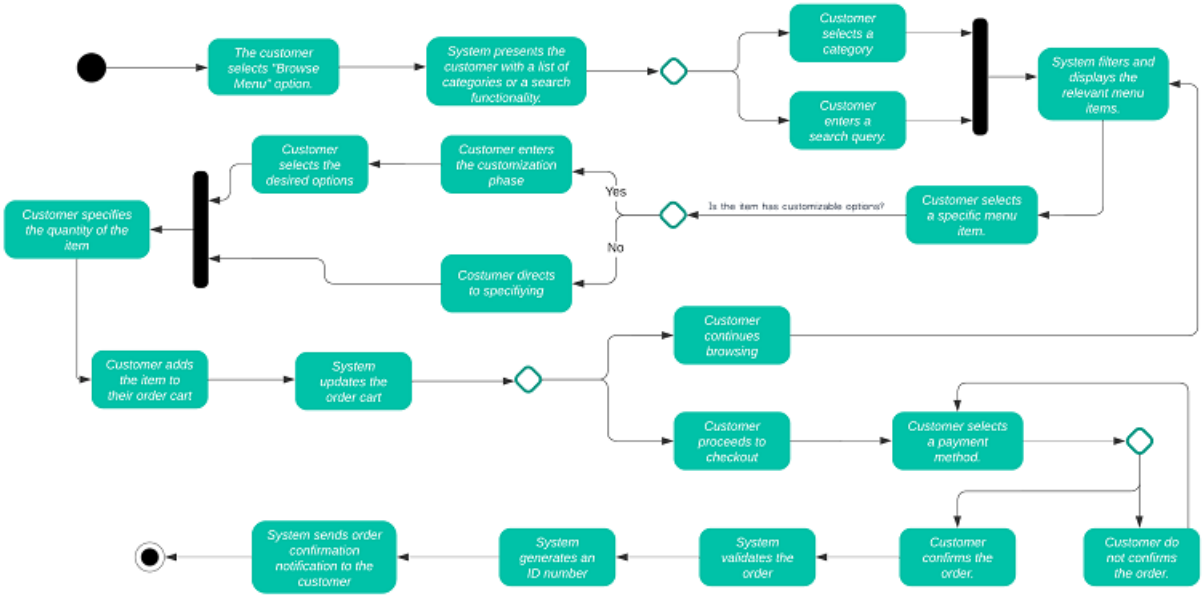

Figure 4: Activity Diagram

The diagram starts. The customer enters the menu section by selecting the "Browse Menu" option. The system presents the customer with a list of categories or a search functionality. The customer selects a category or enters a search query. The system filters and displays the relevant menu items. The customer selects a specific menu item. If the item has customizable options, the customer enters the customization phase. The customer selects the desired options, such as toppings or sizes. The system updates the item's price based on the selected options. The customer specifies the quantity of the item. The customer adds the item to their order cart. The system updates the order cart, displaying the added item, options, quantity, and the updated total price. The customer has the option to continue browsing the menu or proceed to the checkout. If the customer continues browsing, the diagram loops back to displaying menu. If the customer proceeds to checkout, they provide delivery or pickup details. The customer selects a payment method. The customer confirms the order. The system validates the order and generates an order confirmation number. The system sends an order confirmation notification to the customer. The activity diagram ends at the "End" node.

4 Chapter 4

4.1 Sequence Diagram-1

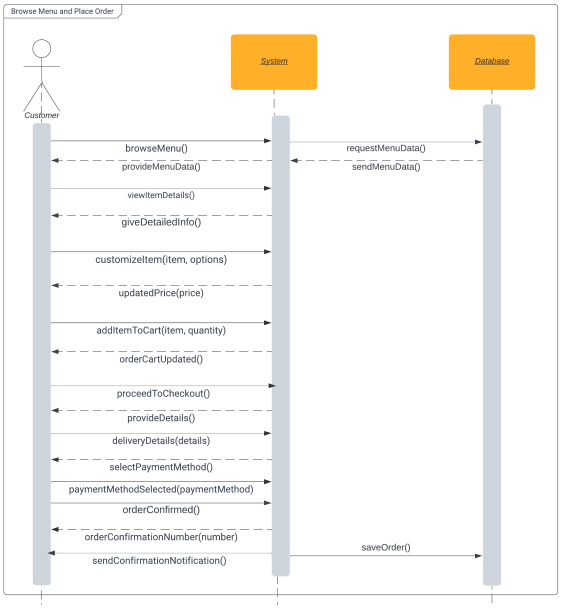

Figure 5: Sequence Diagram-1

The Customer sends a "browseMenu()" message to the Online Food Ordering System. The Online Food Ordering System receives the message and retrieves the menu data from the database. The Online Food Ordering System sends a "menuData" message back to the Customer, containing the menu information. The Customer selects a specific menu item and sends a "viewItemDetails(item)" message to the Online Food Ordering System. The Online Food Ordering System receives the message and retrieves the detailed information for the selected item. The Online Food Ordering System sends a "itemDetails(itemDetails)" message back to the Customer, containing the item details. If the item has customizable options, the Customer selects the desired options and sends a "customizeItem(item, options)" message to the Online Food Ordering System. The Online Food Ordering System receives the message, updates the item's price based on the selected options, and sends a "updatedPrice(price)" message back to the Customer. The Customer specifies the quantity of the item and adds it to their order cart by sending a "addItemToCart(item, quantity)" message to the Online Food Ordering System. The Online Food Ordering System receives the message, updates the order cart, and sends an "orderCartUpdated()" message back to the Customer. If the Customer wants to continue browsing the menu, they can repeat steps 4- If the Customer decides to proceed to checkout, they

send a "proceedToCheckout()" message to the Online Food Ordering System. The Online Food Ordering System receives the message and prompts the Customer to provide delivery or pickup details by sending a "provideDetails()" message. The Customer provides the necessary details by sending a "deliveryDetails(details)" message to the Online Food Ordering System. The Online Food Ordering System receives the details and prompts the Customer to select a payment method by sending a "selectPaymentMethod()" message. The Customer selects a payment method and sends a "paymentMethodSelected(paymentMethod)" message to the Online Food Ordering System. The Customer confirms the order by sending a "orderConfirmed()" message to the Online Food Ordering System. The Online Food Ordering System receives the confirmation and generates an order confirmation number, which it sends back to the Customer by sending an "orderConfirmationNumber(number)" message. The Online Food Ordering System also notifies the Customer of the order confirmation via email or SMS by sending a "sendConfirmationNotification()" message. Finally the system will save data to database.

4.2 Sequence Diagram-2

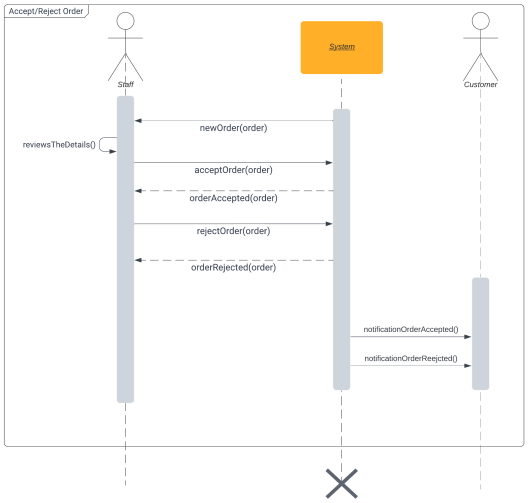

Figure 6: Sequence Diagram-2

The Customer places an order through the online food ordering system. The Order Management System receives the order and sends a "newOrder(order)" message to the Restaurant Staff. The Restaurant Staff receives the message and reviews the details of the order. If the Restaurant Staff decides to accept the order, they send an "acceptOrder(order)" message to the Order Management System. The Order Management System receives the message and updates the order status as "Accepted". The Order Management System sends an "orderAccepted(order)" message back to the Restaurant Staff. If the Restaurant Staff decides to reject the order, they send a "rejectOrder(order)" message to the Order Management System. The Order Management System receives the message and updates the order status as "Rejected". The Order Management System sends an "orderRejected(order)" message back to the Restaurant Staff. If the order is accepted, the Order Management System may send a "notification" message to the Customer to inform them that their order has been accepted. If the order is rejected, the Order Management System may send a "notification" message to the Customer to inform them that their order has been rejected.