Software Requirements Specification

for

Quick Crave

Version 1.0

Prepared by: Cache Me If You Can (Team 9)

Dnyanesh Hemendra Shinde	230390	dnyanesh23@iitk.ac.in
Nikhilesh Joshi	230700	nikhileshj23@iitk.ac.in
Kamatam Akshay Reddy	230517	areddy23@iitk.ac.in
Arihant Kumar	230189	arihantkmr23@iitk.ac.in
Kshitij Gupta	230581	kshitijg23@iitk.ac.in
Parnika Mittal	230736	mparnika23@iitk.ac.in
Rohit Somani	230871	rohitvs23@iitk.ac.in
Suvid Goel	231065	gsuvid23@iitk.ac.in
Yashaswi Raj	231191	yashaswir23@iitk.ac.in
Yatharth Dangi	231197	yatharth23@iitk.ac.in

Course: CS253

Mentor TA: *Nij Bharatkumar Padariya*

Submission Date: *January 24, 2025*

Contents

1 Introduction

1.1 Product Scope

Quick Crave is an intelligent ordering and canteen management system for the IIT Kanpur campus. It offers a unified platform for students, faculty, and staff to order food, book events, and canteen managers to manage operations with ease.

The platform enhances efficiency and streamlines the ordering procedure with functionalities such as live updates, advance meal reservations, and a custom dashboard. The software accelerates food ordering and makes it convenient while facilitating efficient operations for canteens.

For staff and students, Quick Crave provides a simple means of menu browsing, ordering, and tracking. You can also pre-order meals, saving time and averting last-minute problems.

For canteen administrators, the system streamlines activities such as monitoring orders and planning during peak periods.

Quick Crave is user-friendly and makes for a more efficient and pleasant canteen experience for all campus members. It's an intelligent solution to streamline food ordering and management into something simple and convenient.

1.2 Intended Audience and Document Overview

1.2.1 Intended Audience

This document is intended for developers, project managers, testers, and other stakeholders involved in the project. It serves as a comprehensive guide, detailing the functional and non-functional requirements, system constraints, and implementation considerations to ensure successful development and deployment.

1.2.2 Document Overview

- **Introduction:** This section provides essential information to help readers understand the SRS, such as document conventions and abbreviations. If readers are already familiar with the basic terms, they may choose to skip this part. However, it serves as a handy reference to clear up any confusion while reading the document.
- **Overview of the system:** This part offers a general description of the software system, including its functionalities, assumptions, and dependencies. It is ideal for those seeking a quick understanding of the system and forms a good foundation for the more detailed sections that follow.
- **Specific requirements:** Here, you'll find an in-depth explanation of the software's features and functions, often supported by tree diagrams. This section is invaluable for end-users, clients, and developers, serving as a guide during development and a user manual for future use.
- **Non-Functional requirements:** This section highlights the important non-functional requirements of the software. It holds particular importance for developers, guiding them in ensuring the system's performance, reliability, and other essential aspects.
- **Other Requirements:** This section captures essential requirements that do not fall under functional or non-functional categories but are critical for the system's success. It includes compliance, compatibility, security, documentation, and integration needs. This section

ensures all additional considerations are documented to guide development and meet stakeholder expectations.

- **Appendix:** This section provides supplementary information to support the main content of the SRS document. It includes the Data Dictionary, which defines the structure, format, and meaning of all data elements used in the databases, and the Group Log, which records key decisions, changes, and discussions among stakeholders during the project. This section serves as a reference for developers, testers, and other stakeholders to ensure clarity and consistency throughout the project lifecycle.

1.3 Definitions, Acronyms, and Abbreviations

- **UPI:** Unified Payments Interface, used for cashless transactions.
- **UI:** User Interface. The part of the system with which the user interacts.
- **DBMS**: Database Management System. Software used to store, retrieve, and manage data in databases for the Quick Crave system.
- **API:** Application Programming Interface – Set of protocols for communication between different parts of the Quick Crave system or with external systems.
- **OTP:** One-Time Password. Used to verify Guest Users who signed-up using their phone numbers.
- **SMS:** Short Message Service. OTPs are sent to Guest Users via SMS.
- **WebApp:** Web Application – A browser-based application providing the Quick Crave services to users, accessible via internet-enabled devices.
- **Dashboard:** A personalized interface showing order history, ongoing bookings, and account details.
- **Live Updates:** Real-time information provided to users about their orders, including preparation, estimated delivery time, and completion status.
- **Peak Periods:** High-demand times during which the canteen experiences a surge in orders.
- **Event Booking:** A feature in Quick Crave that allows users to reserve space and arrange catering for special events on campus.
- **Pre-order Meals:** The process of ordering meals in advance to ensure availability and reduce wait times.

1.4 Document Conventions

- This document follows the IEEE standards for Software Requirements Specification (SRS).
- The document is written with the **Arial** font of size 11 with single spacing and 1 inch margins.
- Important keywords are highlighted in bold font and comments are italicized
- The document is prepared using LaTeX (LaTeX) to ensure consistency in formatting, ease of maintenance, and scalability for future updates.
- Section titles are hierarchical and numbered, with clear and concise descriptions under each heading.

1.5 References and Acknowledgments

- We extend our gratitude to our TA, Nij Padariya, and our course instructor, Prof. Indranil Saha, for their invaluable guidance throughout the creation of this document and for providing us with a template for the Software Requirements Specification.

- References include initial project discussions and relevant course materials.

2 Overall Description

2.1 Product Overview

Our product, Quick Crave, is a dynamic food ordering platform designed to enhance the dining experience for the IITK community. It offers smooth interactions and improved service accessibility while streamlining the ordering procedure for meals from campus canteens and dining places.

The software supports multiple user roles, such as customers and canteen managers. Canteen managers can manage orders and reservations, adjust menus, and evaluate their performance. Customers can explore the menu, place and track orders, view their order history and manage their account.

To ensure a secure and personalized experience, Quick Crave integrates email-based account verification, accepts a variety of payment methods, and provides user-centric features to guarantee a safe and customized experience. For Guest Users, their is also the option to sign-up using their phone numbers and get verified by an OTP sent via SMS

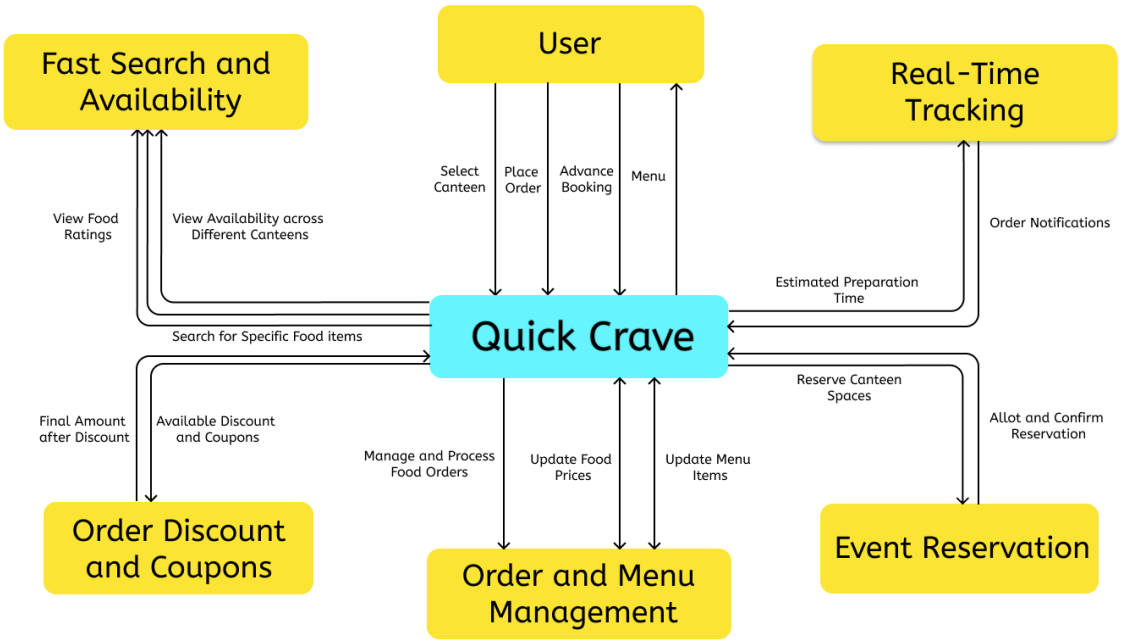


Figure 1: Overview Flowchart

2.2 Product Functionality

2.2.1 User Functionalities

- **Account Creation and Role-Based Access:**
  - Users can create accounts with email-based OTP verification.
  - Guest Users can create accounts with SMS-based OTP verification.
  - Role options include: Customer and Canteen Manager.

- **Personalized Recommendations and Menu Updates:**
  - Customers can see their frequently ordered food items from specific canteens.
  - Newly added menu items are highlighted directly on the display page
- **Advanced Search and Availability:**
  - Users can search for specific food items and view their availability across multiple canteens, along with user ratings.
- **Seamless Order Placement and Real-Time Tracking:**
  - Customers can place orders, which are processed upon approval by the canteen manager.
  - Receive real-time notifications when orders are ready for collection.
  - Track active orders with estimated preparation time updates.
- **Advance Food Preordering:**
  - Customers can prebook food items for specific times to ensure availability and timely service during peak hours.
- **Specialized Ordering and Event Reservations:**
  - Option to order custom-designed birthday cakes with preferred flavors.
  - Reserve canteen spaces for private events such as birthdays and small gatherings.
- **User Dashboard for Order Management:**
  - Access and review order history, and view estimated preparation times through an intuitive user dashboard.

2.2.2 Canteen Manager Functionalities

- **Order and Menu Management:**
  - Manage incoming orders, process approvals, and update order statuses.
  - Manage reservation requests
  - Add, update, or remove menu items, including pricing and availability.
  - Give discounts on certain dishes and issue coupons
- **Availability Scheduling:**
  - Define operational hours and set availability for specific days.

2.3 Design and Implementation Constraints

The development of Quick Crave is subject to the following constraints:

- The application must be developed using HTML, CSS, JavaScript, ReactJS, and Angular for the frontend, and Node.js with Express.js for the backend.
- Data storage will be managed using MongoDB or PostgreSQL based on data requirements.

- The system should support up to 2000 concurrent users with a response time under 3 seconds.
- Security measures such as AES-256 encryption and IIT Kanpur email authentication using students' IIT Kanpur email must be implemented.
- The system will be available from 2 PM to 2 AM, aligning with canteen operating hours.
- Deployment will be done using AWS free-tier services initially, with scalable options for production.
- The system will support mobile platforms, with the mobile application available for Android devices. Future support for iOS may be considered based on user demand and feasibility.

2.4 Assumptions and Dependencies

The successful operation of Quick Crave relies on the following assumptions and dependencies:

- Users will have reliable internet access and provide accurate information.
- Canteen Manager will regularly update menus and food availability.
- The system depends on third-party services such as email notifications and payment gateways like UPI (PhonePe , GooglePay , PayTM etc)
- The system relies on server availability and internet connectivity for real-time updates and notifications.
- Compliance with IIT Kanpur's IT policies and data security guidelines is mandatory.
- The system will adhere to relevant food safety regulations and data protection laws.

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

- The application offers a login interface catering to both canteen managers and customers, including guests and members. Users are required to register themselves and further undergo additional verification processes

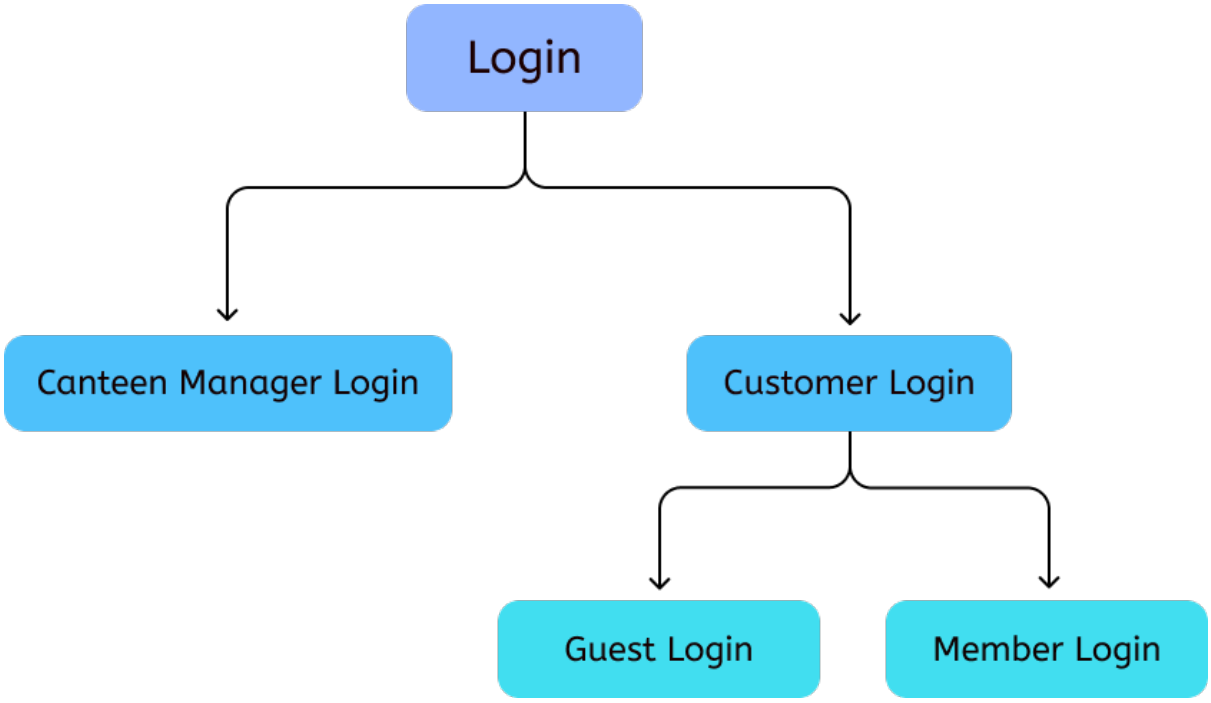


Figure 2: Login Flowchart

- The home screen for customers presents several options like Profile, Recommended and Favourites, Search Bar, Select Canteen and Current Orders options. The Profile page offers multiple options like update profile, address, wallet, coupons, history, current reservations and logout options.

- The home screen for canteen managers presents several options like Manage Order Queue, Manage Reservations, Offer Discounts, Issue Coupons. The Profile page offers multiple options to update profile, update menu, update

3.1.2 Hardware Interfaces

- **Customer Devices:** The WebApp would be available for use on devices with internet access. This will include smartphones, tablets, and laptops. The accessibility of the WebApp on mobile devices such as smartphones would make the process of ordering food hassle-free and convenient for the customer.

  Cameras on smartphones can be used to scan QR code while making UPI final payments for reservations after dining in.

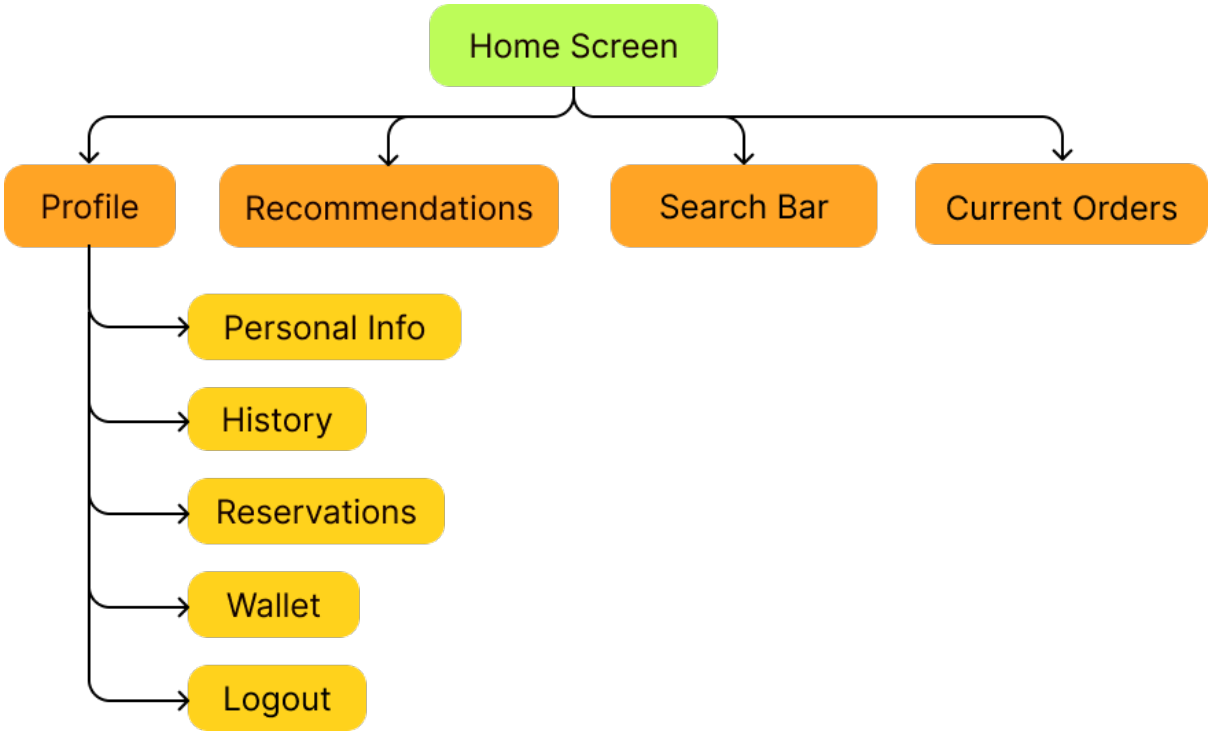


Figure 3: Customer Home Screen Overview

GPS systems of devices will be used to determine the location and finding the nearest canteens.

Microphones of the devices will be used for searching by voice.

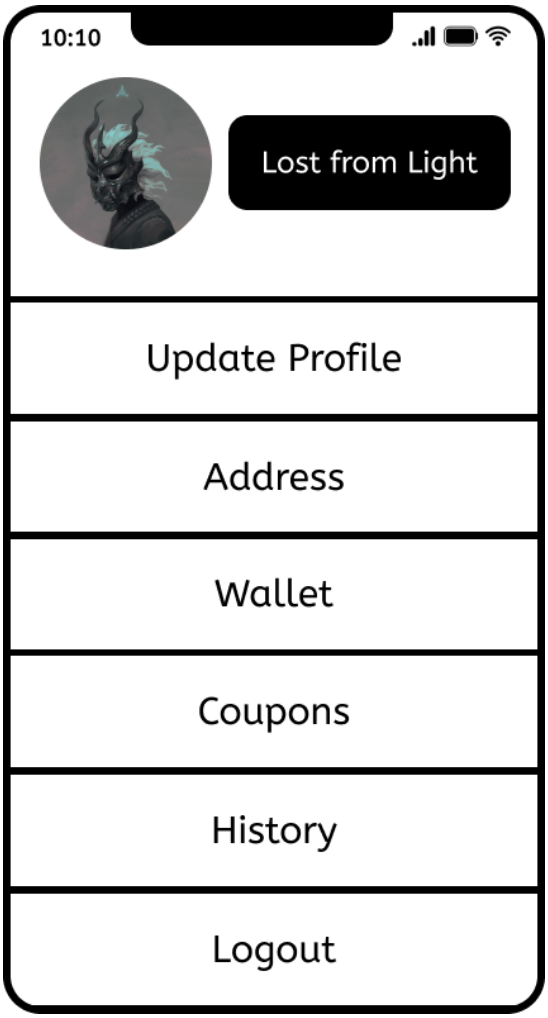
- **Canteen Manager Devices:** The WebApp would be available for use on devices with internet access. This will include smartphones, tablets, and laptops. The accessibility of the WebApp on mobile devices such as smartphones would make the process of managing order queue, reservations and updating menus hassle-free and convenient for the canteen manager.

  GPS systems of devices will be used to determine the location canteens which the will be used to suggest nearest canteens to customers.

3.1.3 Software Interfaces

- The server-side components, including the database, will be hosted in a UNIX-based operating system environment
- The client-side components must be functional on modern web browsers as well as their mobile versions, like Google Chrome, Safari, Mozilla Firefox, Microsoft Edge, and Brave
- The database management system used will be PostgreSQL or MongoDB:
  - The User database will store all users' data, including authentication details and personal information
  - The Canteen database will store the canteen manager's data including authentication information and other canteen specific details.
  - The Dishes database will store data like the unique ID of the dish, the unique ID of the owner, price, canteen where it is served, availability, discount etc.

(a) Customer Home Page



(b) Customer Profile Page

Figure 4: Customer UI

- The Payment database will store all the information related to payments
- The Order Database will store all the orders processed by system. Orders will be identified by unique IDs.

3.2 Functional Requirements

3.2.1 Account Creation

- Sign-up should provide two options: one for customers and another for canteen managers.
- It is recommended to use an IITK email address for registration.
- A verification email should be sent from our account to the registered email address.
- Upon successful verification, the user should be added to the list of registered users.
- Guest users can sign up using their phone numbers and should be verified by an OTP via SMS.
- Provide two sign-in options: Customers and Canteen Managers.

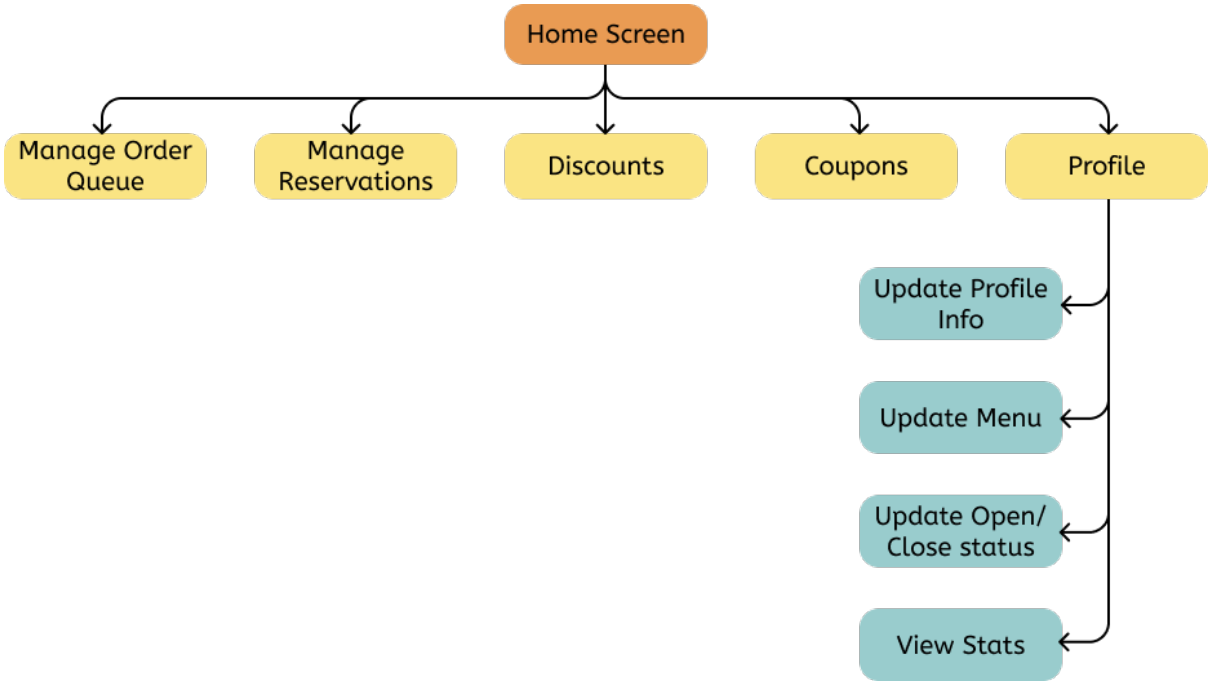


Figure 5: Canteen Manager Home Screen Overview

- Include a ”Forgot Password” option to allow users to recover their accounts.
- Offer a ”Remember Me” option to keep the user signed in for up to one month.

3.2.2 Customer Functionalities

- **Option to Select Canteen:** Users can choose the desired canteen to place their order from.
- **Order by Dish:** A search bar allows users to order by dish.
- **Option to Order from Nearest Canteen:** Users can opt to place orders from the nearest available canteen.
- **Dine-in, Takeaway, and Delivery Options:** Customers can choose between dining in, taking away their food, or opting for delivery. Dine-in further offers options to dine-in immediately and pre-order for later options.
- **Estimated Time for Preparation:** The system displays the estimated preparation time for orders.
- **Cake Reservation:** Customers can reserve cakes for birthdays or special events.
- **Canteen Reservation:** Customers can reserve canteen for celebrations and special events.
- **Rating and Review System:** Customers can rate their experiences and provide reviews for items or canteens.
- **Order History:** Users can view their previous orders for easy reordering.
- **Recommendations:** The system provides recommendations such as bestsellers and house specials.

- **Payment Methods:** Customers can pay using UPI, credit cards, debit cards, or a wallet.
- **Dietary and General Filters:** Options include Veg, Non-Veg, Jain, and other general filters.
- **Add Favorite Food Items:** Users can mark their favorite items for quick access in the future.
- **Canteen-Specific History:** Customers can view their order history specific to a particular canteen.
- **Monthly Spending Statistics:** The system provides an overview of the user's monthly spending.
- **Profile Management:** Users can manage their account details, including photo, username, and phone number.

3.2.3 Canteen Manager Functionalities

- **Update Menus:**
  - Add new dishes
  - Remove existing dishes
  - Update current availability of dishes.
  - Update prices
  - Upload photos of dishes
- **Order Notifications:** A pop-up appears on screen to notify canteen manager about new order request which can be clicked to get redirected to order queue
- **Order Queue Management:** Options to accept, reject, prioritize orders for efficient order queue management
- **Manually Notify Customers:** Option to manually notify customers.
- **Manage Reservations:** Option to accept/reject reservation requests
- **Offer Discount and Coupons:** Options to offer discounts on selected items and also issue coupons which can be redeemed
- **Monthly Statistics:**
  - Orders statistics
  - Earnings statistics
  - Identify the most ordered or most profitable dish.
- **Manage Canteen Opening Times/Days:**
  - Manage canteen open/closed status which is visible to customers
  - Option to mark closed days on a calendar.
- **Account Management:** Options to update contact details including phone number, profile photos, username, password etc.

3.3 Customer Use Case Model

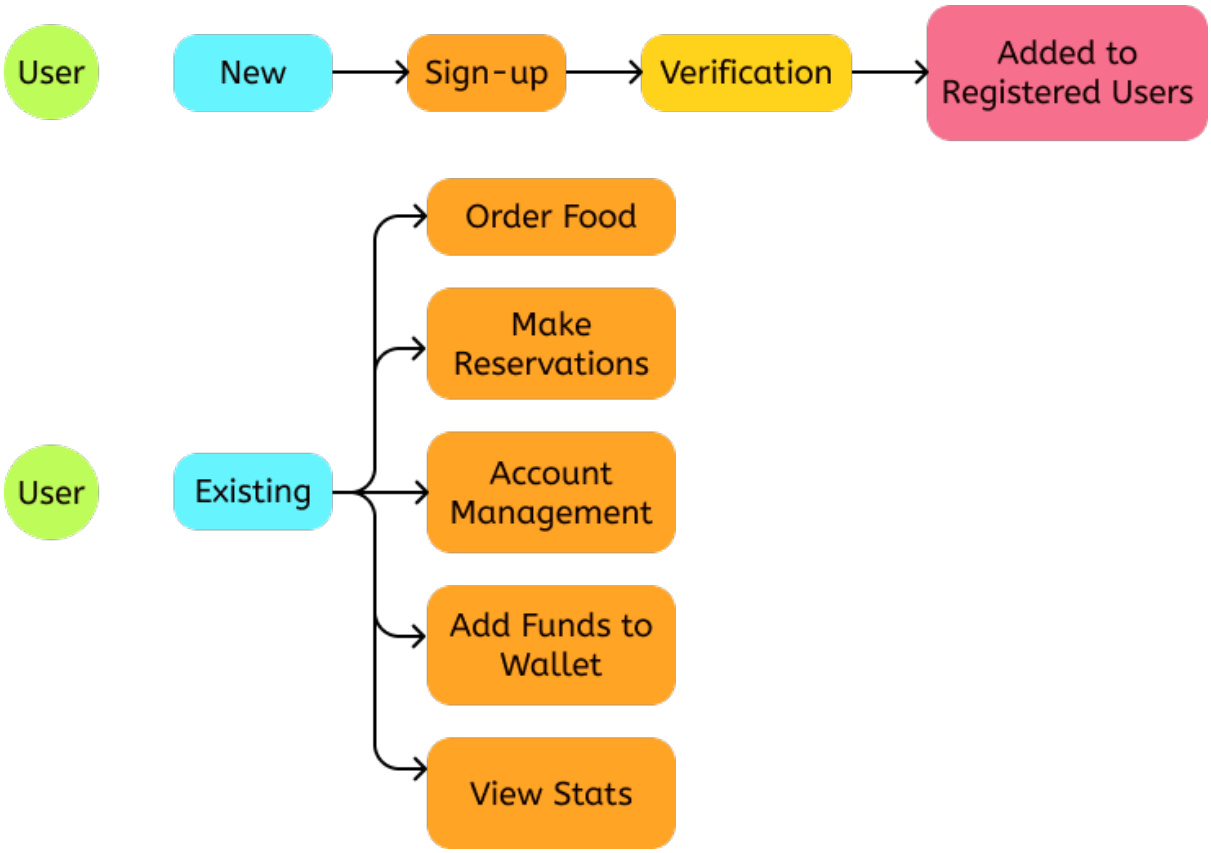


Figure 6: Customer Use Case Model

3.3.1 Customer Use Case 1: Ordering Food

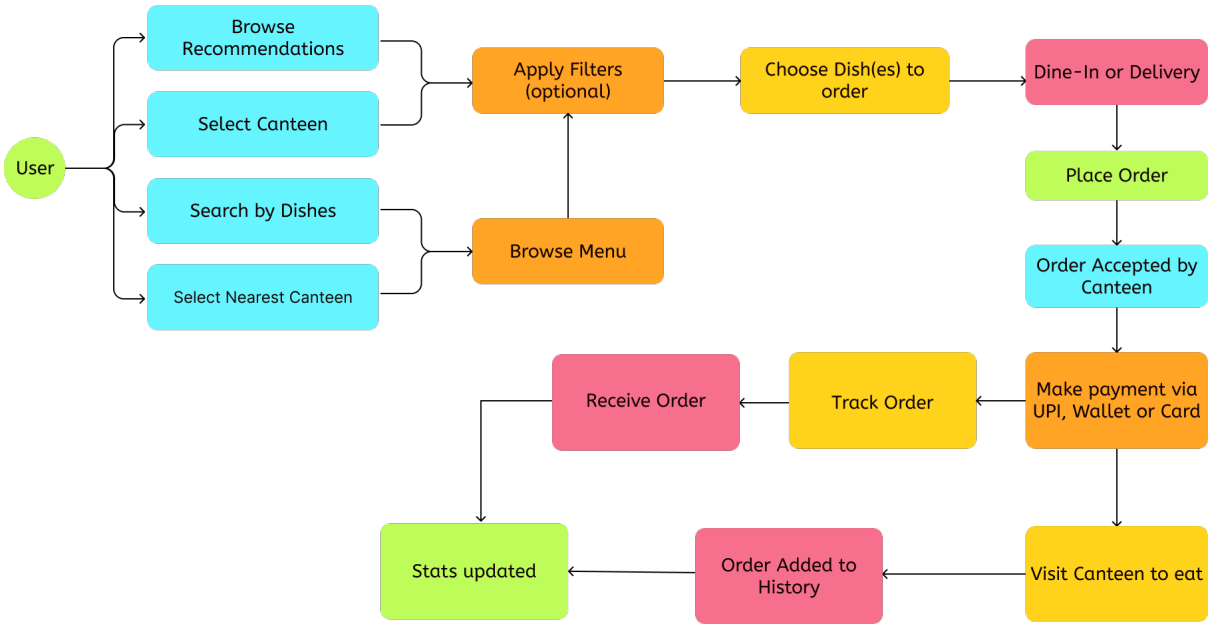


Figure 7: Customer Use Case 1

- **Purpose:** To let the customers order food from canteens across IIT Kanpur
- **Priority:** The priority of this use case is high as the WebApp is being built to serve this purpose
- **Pre-Conditions:**
  - Customer should have logged into the app and should have access to internet
  - Customer should have provided location access to the application to suggest nearest canteens and to have the food delivered to them
  - The canteen must be open and able to accept orders
  - Customer should have their payment methods (UPI, Wallet) already set up and functional
- **Post-Conditions:**
  - Payment is processed successfully, and the transaction record is updated
  - The user successfully places the order for dine-in or delivery
  - The canteen receives and fulfills the order request
  - The user receives the order or visits the canteen for dine-in
  - The order is added to the user's order history
  - The user's account statistics (like order history, rewards, or preferences) are updated
- **Actors:** The actors involved in this use case are the customers, canteen managers, canteen staff and delivery partners

3.3.2 Customer Use Case 2: Making a Reservation

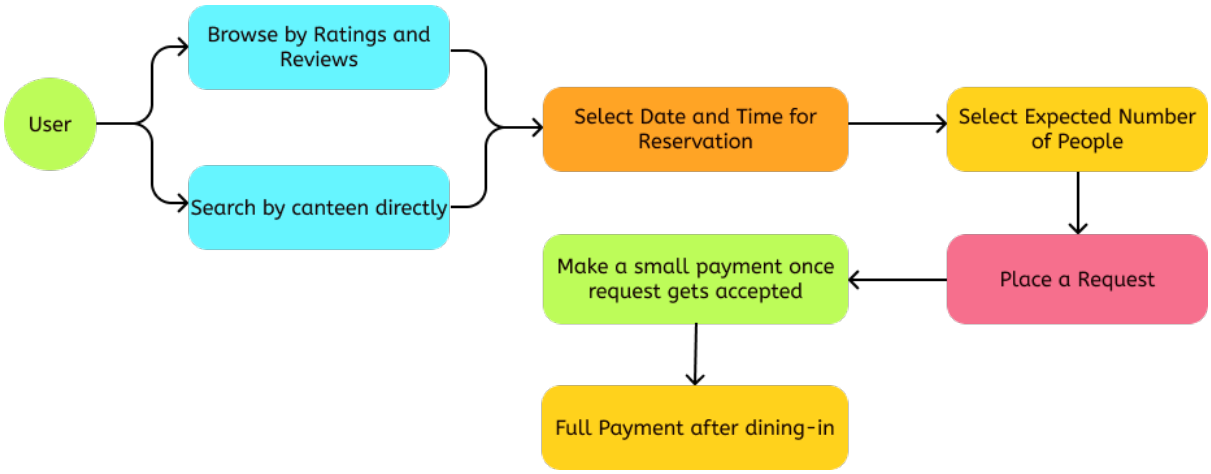


Figure 8: Customer Use Case 2

- **Purpose:** To let the customers make a reservation at a canteen in IIT Kanpur
- **Priority:** The priority of this use case is high as it provides
  - An easy and user-friendly way for customers to make dining reservations
  - The ability to browse by ratings and reviews helps customers make informed decisions, leading to greater satisfaction
- **Pre-Conditions:**
  - Customer should have logged into the app and should have access to internet
  - The canteens must have their ratings, reviews, and reservation options listed in the application
  - The canteen must be open on the day of reservation and able to cater to the reservation
  - The canteen should have available seating for the specified number of people.
  - Customer should have their payment methods (UPI, Wallet) already set up and functional
- **Post-Conditions:**
  - The canteen confirms the reservation for the specified date, time, and number of people
  - The small fee for the reservation request is processed once it is accepted
  - The reservation request is successfully placed and recorded in the system
  - The canteen's booking system is updated with the reservation details
  - The reservation details are added to the user's history for future reference
  - The user dines in at the reserved time
  - The customer makes the full payment after dining in, and the transaction record is updated
- **Actors:** The actors involved in this use case are the customers, canteen managers and canteen staff

3.3.3 Customer Use Case 3: Account Management

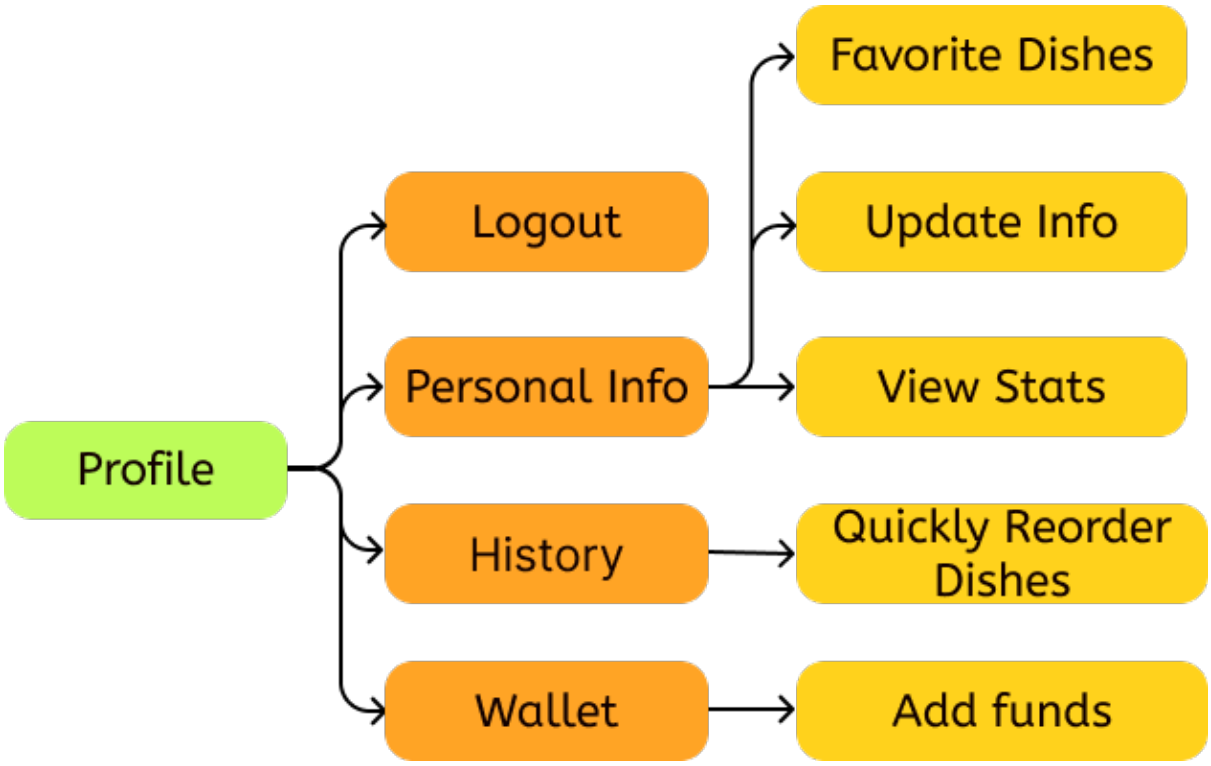


Figure 9: Customer Use Case 3

- **Purpose:** To let the customers manage their accounts on the WebApp
- **Priority:** The priority of this use case is high
  - As it serves as the central hub for user interactions, including managing personal information, viewing order history, and favorite dishes, which are critical for personalizing the experience
  - Features like ”Quickly Reorder Dishes”, ”Add Funds”, and ”Favorite Dishes” enhance user convenience and encourage repeated use of the platform
  - Options to update personal info, view stats, and track wallet funds provide transparency and control, which increases user trust and satisfaction.
- **Pre-Conditions:**
  - The customer must be logged in to access the profile features
  - Personal information, order history, favorite dishes, and wallet details must already exist in the system database
  - The wallet functionality should be linked to valid payment methods to allow fund additions
  - The user’s profile should contain enough data to display stats or manage preferences effectively
  - For ”Quickly Reorder Dishes,” previous orders must exist in the system
- **Post-Conditions:**

- Changes made to personal info or preferences are saved successfully
- The wallet reflects the added funds, and a transaction record is updated
- Any additions or changes to favorite dishes are recorded for future reference.
- All profile interactions (e.g., logout, info updates) are logged for security and analysis
- Viewing stats allows users to track orders and payments, improving transparency
- Quick reordering and favorite dishes reduce the effort for future transactions

• **Actors:** The actors involved in this use case are the customers

3.4 Canteen Manager Use Case Model

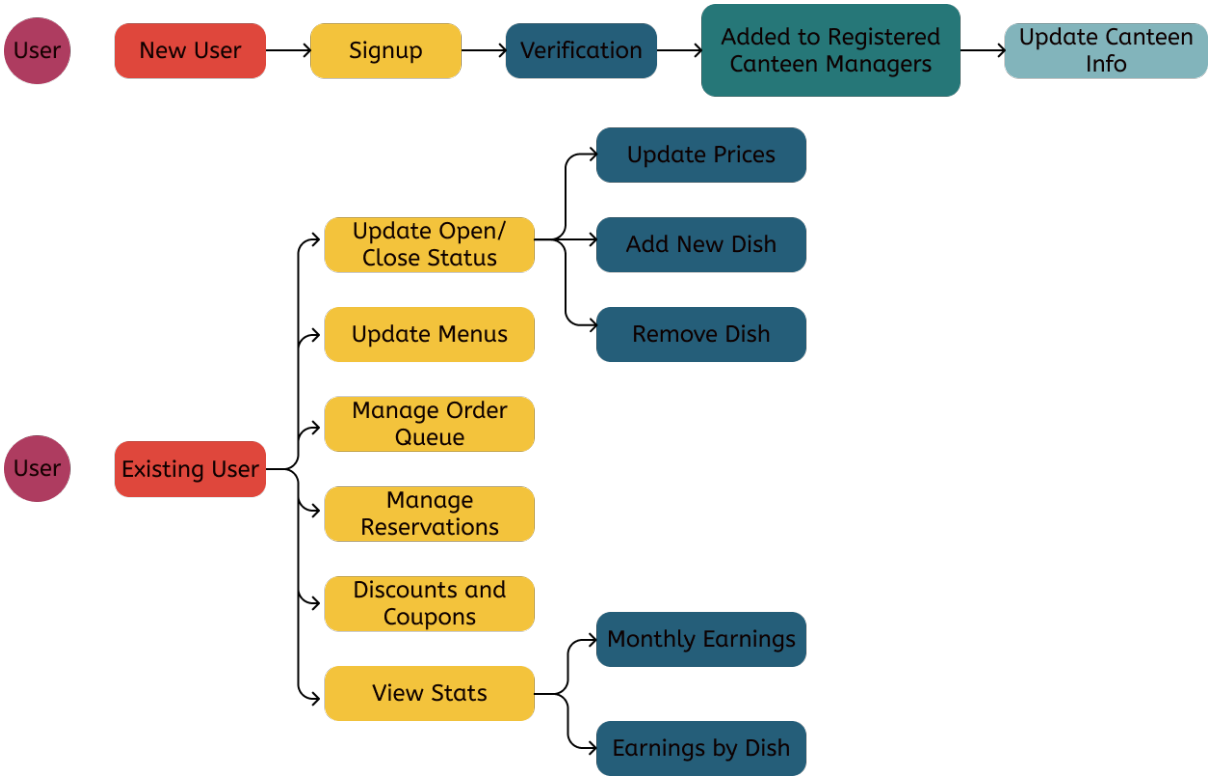


Figure 10: Canteen Manager Use Case Model

3.4.1 Canteen Manager Use Case 1: Orders and Reservations Management

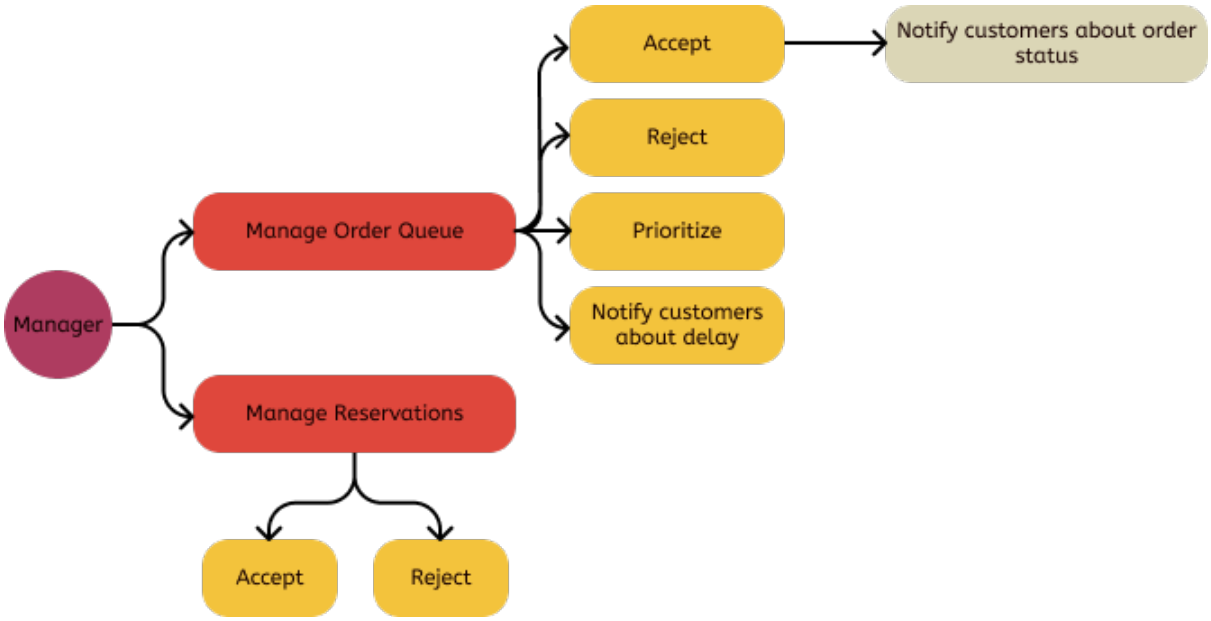


Figure 11: Canteen Manager Use Case 1

- **Purpose:** To let the canteen managers manage incoming orders and handle reservation requests
- **Priority:** The priority of this use case is high as the WebApp is being built to serve this purpose
- **Pre-Conditions:**
  - The Canteen Manager should be logged into the WebApp on a device with internet connection
  - The system must have access to real-time data about incoming orders and reservation requests
- **Post-Conditions:**
  - Orders are processed, and their statuses are updated (accepted, rejected, or prioritized)
  - Payment is processed successfully, and the transaction record is updated
  - Notifications regarding order delays or reservation acceptance/rejection are sent to customers
  - Reservations are updated in the system with the appropriate status (accepted/rejected)
- **Actors:** The actors involved in this use case are the canteen managers and canteen staff

3.4.2 Canteen Manager Use Case 2: Offer Discounts and Issue Coupons

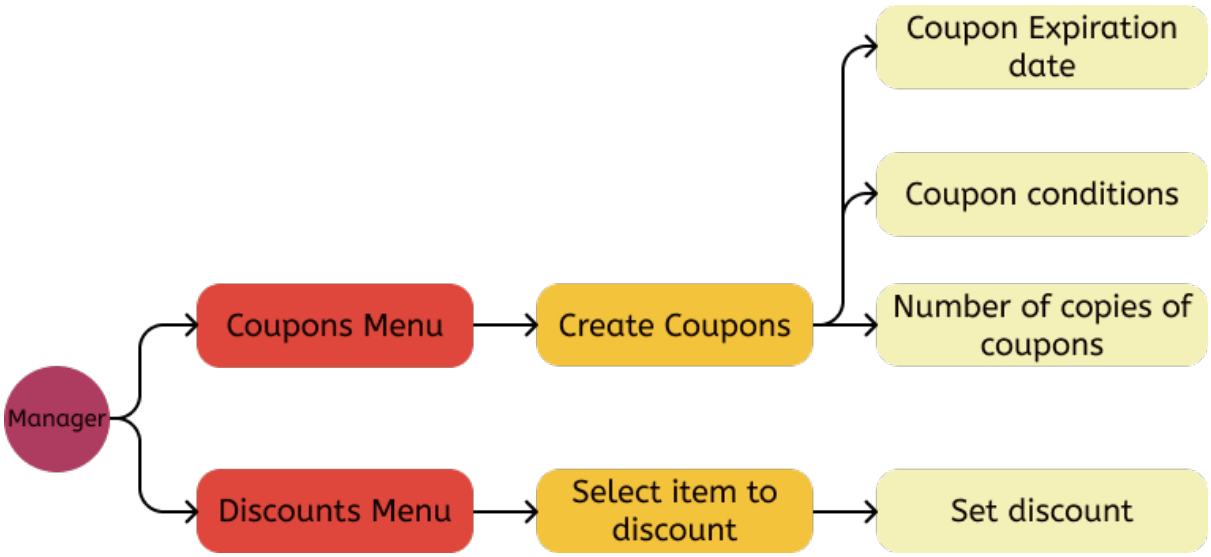


Figure 12: Canteen Manager Use Case 2

- **Purpose:** To let canteen managers define discounts on specific items to attract customers and promote sales and create and manage coupons (with conditions such as expiration date and number of copies)
- **Priority:** The priority of this use case is high as
  - Providing discounts and coupons helps attract more customers, boosting sales
  - Coupons and discounts can improve customer loyalty by giving them incentives to return
  - Offers and discounts help the business stay competitive in the market
  - A systematized approach reduces manual errors and ensures that promotions are consistently applied
- **Pre-Conditions:**
  - The Canteen Manager should be logged into the WebApp on a device with internet connection
  - Items must already be set up in the system to apply discounts or coupons to them
- **Post-Conditions:**
  - The coupon is successfully created, including all details like expiration date, conditions, and number of copies
  - Discounts are successfully applied to the selected items and reflected in the pricing system
  - Customers can view and redeem the created coupons and discounts during purchases
- **Actors:** The actors involved in this use case are the canteen managers

4 Other Non-functional Requirements

4.1 Performance Requirements

4.1.1 Response Time for User Interactions

- **Requirement:** The system shall ensure that the average response time for user actions (e.g., placing an order, making a reservation) does not exceed 3 seconds.
- **Rationale:** Users expect quick processing for complex tasks. Delays beyond 3 seconds may lead to frustration and abandonment.

4.1.2 Response Time for Simple Actions (Database Queries, UI Updates)

- **Requirement:** The system should process simple actions like fetching data and querying the database within 500 milliseconds.
- **Rationale:** Fast feedback is crucial for quick actions. Delays beyond 500 milliseconds can make the app feel sluggish.

4.1.3 Concurrent Users Handling

- **Requirement:** The system should handle up to 2000 concurrent users during peak times.
- **Rationale:** The system must support high traffic periods without performance degradation.

4.1.4 System Throughput

- **Requirement:** The system should process at least 200 orders per minute during peak times.
- **Rationale:** The system should efficiently handle large volumes of orders, especially during peak meal hours.

4.2 Safety and Security

4.2.1 User Authentication and Access Control

- **Requirement:** Users must log in securely using student credentials (e.g., email and password), with passwords hashed using secure algorithms (e.g., bcrypt). Multi-factor authentication (MFA) should be implemented for admin roles.
- **Rationale:** Ensures that only authorized users can access the app, protecting sensitive data and preventing unauthorized actions.
- **Safeguards:**
  - Passwords must be hashed.
  - Prevent brute force attacks by limiting to 5 failed attempts per 15 minutes.
- **CIA Triad:**
  - **Confidentiality:** Ensures only authorized users can access the system.

4.2.2 Data Protection and Privacy

- **Requirement:** Sensitive data (e.g., personal details, payment information) must be securely handled and protected from unauthorized access or disclosure.
- **Rationale:** Protects user privacy and ensures that sensitive information is not exposed to unauthorized parties.
- **Safeguards:**
  - Use appropriate encryption and security measures for all data in transit and storage.
  - Regularly update security protocols to adapt to evolving threats.
  - Ensure access control mechanisms limit data access to authorized individuals only.
- **CIA Triad:**
  - **Confidentiality:** Protects sensitive information.
  - **Integrity:** Ensures data remains accurate and unmodified.

4.2.3 Secure Payment Processing

- **Requirement:** Payments should be processed through secure gateways (e.g., Razorpay, PayPal), adhering to PCI-DSS standards.
- **Rationale:** Protects financial data and ensures secure transactions.
- **Safeguards:**
  - Comply with PCI-DSS regulations.
- **CIA Triad:**
  - **Confidentiality:** Prevents unauthorized access to financial data.
  - **Integrity:** Ensures payment data is accurate and unaltered.

4.2.4 Safeguards Against Data Loss

- **Requirement:** Implement regular data backups and disaster recovery plans.
- **Rationale:** Protects against data loss due to system failures or attacks.
- **Safeguards:**
  - Regular backups of critical data.
  - Disaster recovery plans for system restoration.
- **CIA Triad:**
  - **Availability:** Ensures data is accessible after a failure.

4.3 Software Quality Attributes

4.3.1 Usability

The user interface should be simple and intuitive, with core actions (like ordering or checking status) accessible in no more than two taps. The design will be based on feedback from real users to ensure it meets their needs, and adjustments will be made to improve user experience continuously.

4.3.2 Maintainability

The codebase should be well-structured, modular, and properly documented. This will ensure ease of updates, bug fixes, and adding new features. Version control will allow developers to track changes and collaborate efficiently.

4.3.3 Interoperability

The app should function smoothly on both iOS and Android, supporting the latest version of each platform. Additionally, it must be accessible as a fully functional website, ensuring users can place orders or view the menu from any browser. Using cross-platform development tools will ensure consistent behavior across mobile devices and the web, while thorough testing will verify its performance on different devices and screen sizes.

4.3.4 Robustness

The app should handle common issues like network failures or incorrect user inputs without crashing. Clear error messages will guide users through recovery steps, and built-in fallback mechanisms will allow the app to function smoothly even in unexpected situations. Regular testing will ensure the system is resilient to different failure scenarios.

4.3.5 Flexibility

The app should be adaptable to future changes, such as adding new features, supporting different payment methods, or integrating with other campus services. The system architecture should be modular to allow easy updates without disrupting existing functionality. This flexibility will be achieved by using scalable and extensible technologies, enabling quick adaptation to changing requirements or user feedback.

5 Other Requirements

5.1 Legal Requirements and Copyright

User data collected should be stored in the country's regional data center. This is to comply with data protection and privacy rules that may be enforced by the Government of the nation. Copyright of the source code and the documents must be retained by the developers of the system.

5.2 Authentication

Permission will be required for utilising OTPs for the authentication process as this will require sending automated emails to the user's email id.

Appendix

A Data Dictionary

A.1 User Database

Variable	Description	Requirements
ID	Unique ID of the user	Unique for each user
~~Personal Details~~	Personal information of the user	String
Password	Password of the user	Encrypted string
Wallet	User's wallet balance	Numeric value
Email	Email of the user	IITK Email
Fav Dish	User's favorite dish	String
Order History	List of past orders made by the user	List of orders
Current Orders	List of ongoing orders	List of orders

A.2 Canteen Database

Variable	Description	Requirements
ID	Unique ID of the canteen	Unique for each canteen
Name	Name of the canteen	String
~~Owner ID~~	ID of the owner managing the canteen	Unique for each owner
Password	Password to access the canteen database	Encrypted string
~~Menu~~	List of items available in the canteen	List of Dishes
Order	Current orders placed	List of Dishes
~~Queue~~	Current order queue	List of Dishes
~~Previous Order Details~~	Details of past orders	List of Orders
Opening Time	Daily opening time of the canteen	Time format (HH:MM)
Closing Time	Daily closing time of the canteen	Time format (HH:MM)
Contact Details	Contact information of the canteen	String

A.3 Dishes Database

Variable	Description	Requirements
ID	Unique ID of the dish	Unique for each dish
~~Owner ID~~	~~ID~~ of the owner who manages the dish	Unique for each owner
Price	Price of the dish	Numeric value
Canteen Name	Name of the canteen offering the dish	String
Availability	Availability status of the dish	Boolean
Discount	Discount offered on the dish	Numeric value (Percentage)
Photo	Image of the dish	File/Image
~~Other Miscellaneous Details~~	Additional information about the dish	String

CANTEEN ID

A.4 Payments Database

Attribute	Description	Requirements
ID	Unique ID of the payment	Unique for each payment
User ID	ID of the user who made the payment	Must be a valid ID
Ride ID	ID of the ride for which the payment was made	Must be a valid ID
Amount	Amount of the payment	Integer
Time	Time at which the payment was made	Time
Payment Method	Method used for the payment	String
Status	Whether the payment was successful or not	Boolean

A.5 Order Database

Attribute	Description	Requirements
ID	Unique ID of the order	Unique for each order
User ID	ID of the user who placed the order	Must be a valid ID
Booking Time	Time at which the order was booked	Time
List of Pair {Dish, Quantity}	List of dishes and their respective quantities	List of pairs
Total Amount	Total amount for the order	Numeric value

B Group Log

Meeting notes and activities will be maintained here

S.No	Date	Timings	Venue	Agenda
1	06/01/2024	14:00 to 16:00	RM Building	Discussed and proposed various project ideas, including: Canteen and Mess Management System, Hostel Management Application.
2	09/01/2024	10:00 to 12:00	RM Building	Finalized the project idea and brainstormed specific requirements and functionalities.
3	10/01/2024	22:00 to 23:00	L-20	Met with the professor to present the project idea and seek suggestions on potential technologies to use.
4	14/01/2024	22:00 to 23:00	Google Meet	Reviewed and discussed the SRS (Software Requirements Specification) template.
5	16/01/2024	12:00 to 14:00	RM Building	Explored different technologies to implement in the project and planned a meeting with the mentor for additional guidance.
6	20/01/2024	23:00 to 00:00	Google Meet	Consulted the Teaching Assistant (TA), divided tasks for completing the SRS document, and established work distribution.
7	23/01/2024	14:00 to 16:00	RM building	Completed the work on SRS document and reviewed all the work.