
Design Document

for

Credit Snap

Version 1.0

Prepared by

Group 6 :

Group Name: *Bug Dealers*

Kasavajjala Sai Shreyas	240530	ksshreyas24@iitk.ac.in
Gade V S S R K Tejas Reddy	240390	tejasreddy24@iitk.ac.in
Palagiri Sathish Reddy	240717	psreddy24@iitk.ac.in
Kondreddy Sai Haneesh Reddy	240551	haneesh24@iitk.ac.in
Yerraiahgari Ram Charan Goud	241212	ramcharang24@iitk.ac.in
Golla Lekha Harshaa	240402	lekhahg24@iitk.ac.in
Yash Raj	241202	yashr24@iitk.ac.in
Korada Jayavarahan	240554	jkorada24@iitk.ac.in
Boppudi Sai Chaitanya	240283	boppudis24@iitk.ac.in
Panchaneni Ashwin Rao	240727	ashwinr24@iitk.ac.in

Course: CS253

Mentor TA: George T L

Date: February 6, 2026

CONTENTS

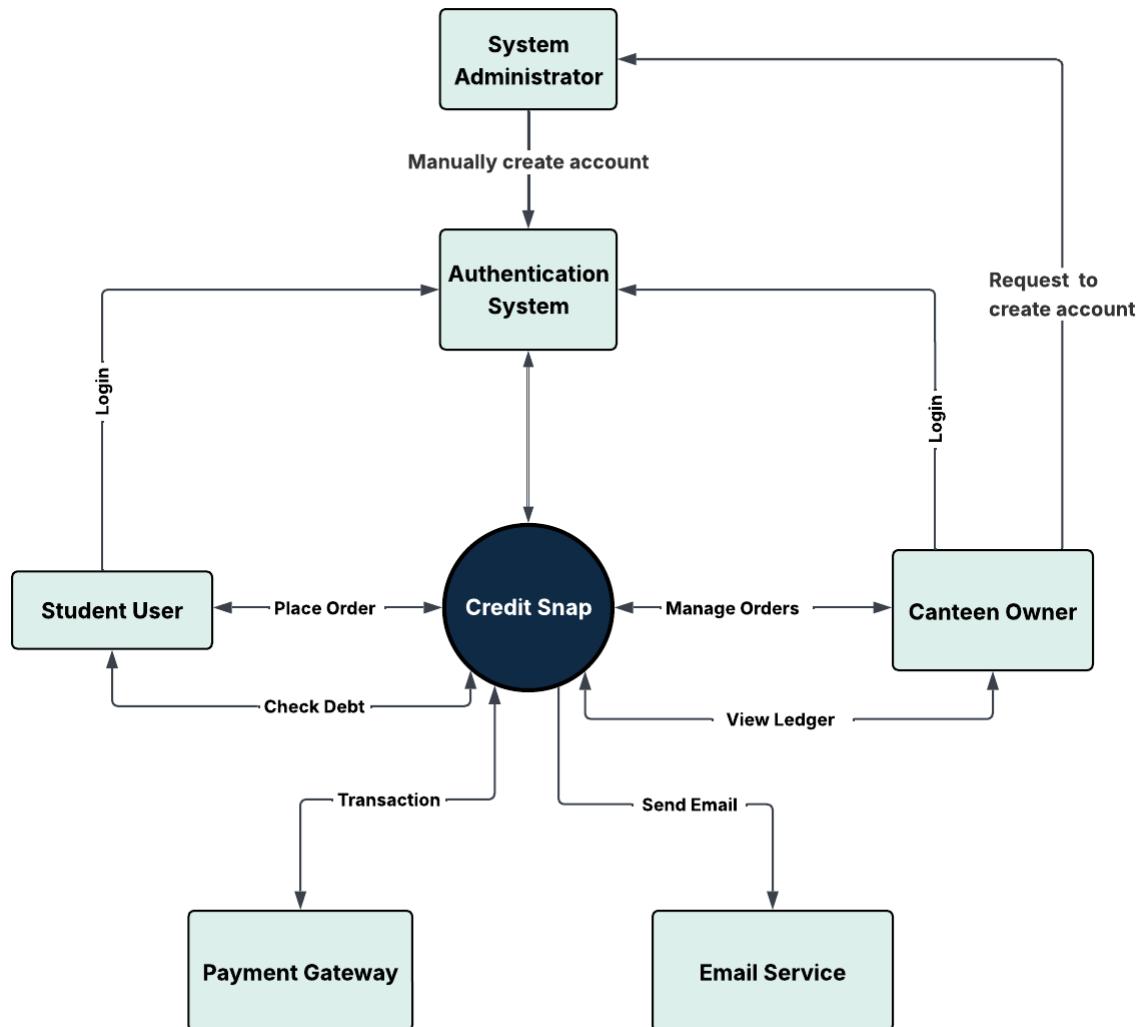
CONTENTS	ii
REVISIONS	iii
1 CONTEXT DESIGN	1
1.1 CONTEXT MODEL	1
1.2 HUMAN INTERFACE DESIGN	3
2 ARCHITECTURE DESIGN	14
3 OBJECT-ORIENTED DESIGN	17
3.1 USE CASE DIAGRAM	17
3.2 CLASS DIAGRAM	22
3.3 SEQUENCE DIAGRAM	23
3.4 STATE DIAGRAM	31
4 PROJECT PLAN	35
APPENDIX A – GROUP LOG	36

REVISIONS

Version	Primary Author(s)	Description of Version	Date Completed
v1.0	Kasavajjala Sai Shreyas Gade V S S R K Tejas Reddy Palagiri Sathish Reddy Kondreddy Sai Haneesh Reddy Yerraiahgari Ram Charan Goud Golla Lekha Harshaa Yash Raj Korada Jayavarthan Boppudi Sai Chaitanya Panchaneni Ashwin Rao	The first version of the Design Document	06/02/26

1 Context Design

1.1 Context Model



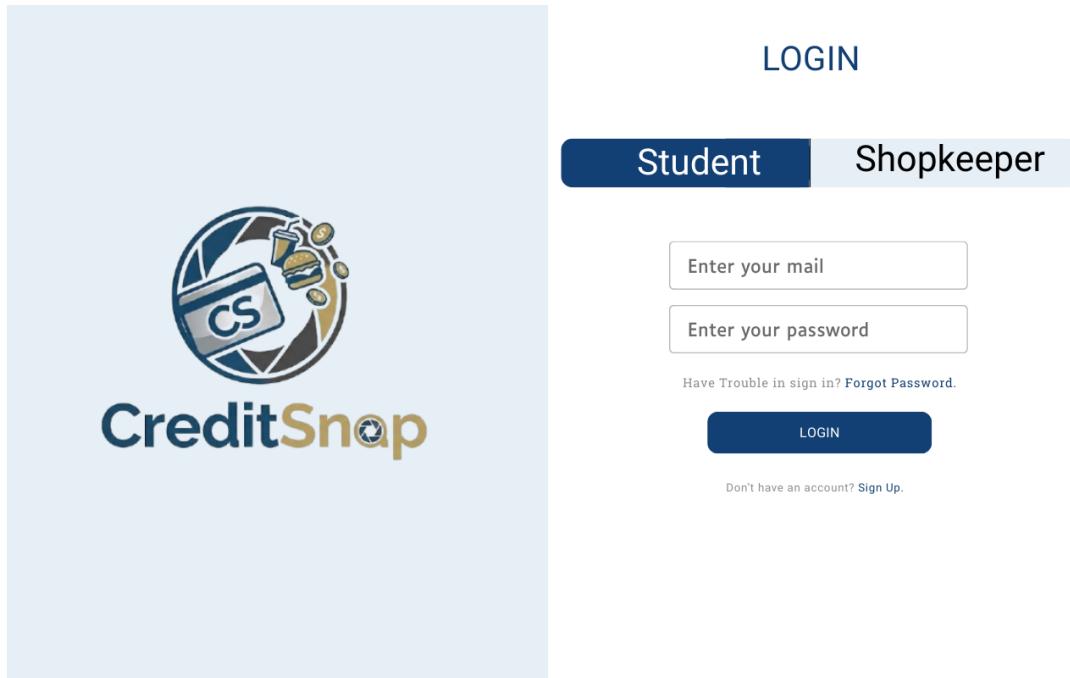
- **System Administrator**: The System Administrator acts as the security gatekeeper, responsible for manually provisioning Canteen Owner accounts. This ensures strict adherence to the system's invite-only policy, preventing unauthorized public sign-ups.
- **Authentication System**: This component handles identity verification for all users, processing "Login" requests before granting access. It also serves as the secure repository where the administrator stores manually created owner credentials.
- **Student User**: Representing the primary consumer, the Student User interacts with the system to place food orders and check debt status. They initiate key data flows, including authentication and financial transactions via the Payment Gateway.
- **Canteen Owner**: The Canteen Owner manages cafeteria operations, specifically handling incoming orders and viewing the digital ledger. Unable to self-register,

they must request account creation from the System Administrator before accessing the system.

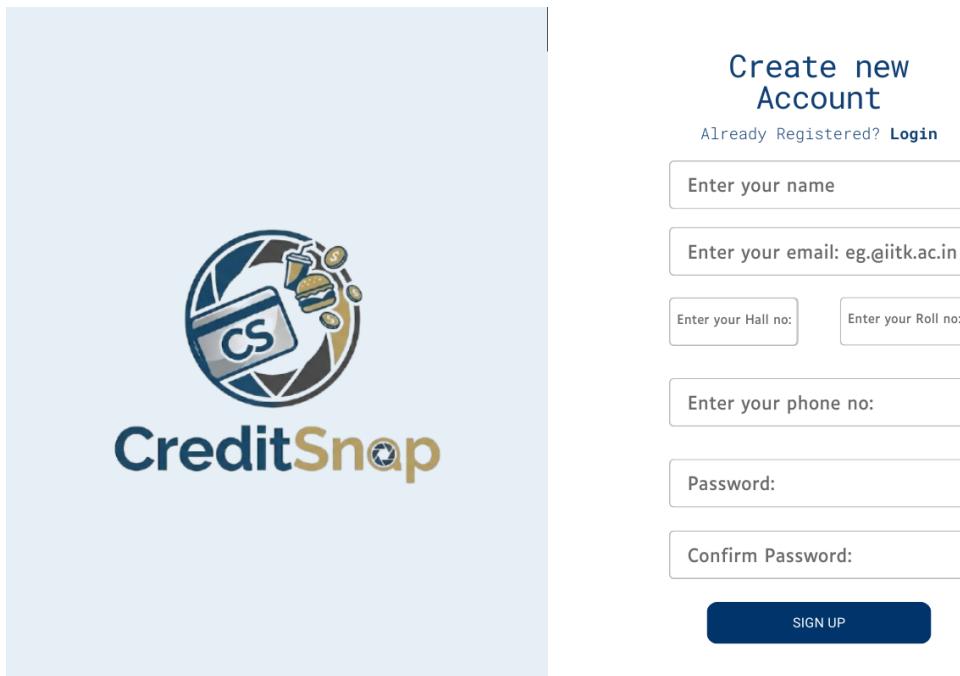
- **Payment Gateway:** This external service is dedicated to securely processing financial transactions initiated by students. It handles the actual money transfer when a student clears their debt, keeping sensitive financial logic separate from the core app.
- **Email Service:** Triggered by the core system, this service delivers automated notifications such as order confirmations and verification links. It ensures that users receive real-time updates on critical events without manual intervention.

1.2 Human Interface Design

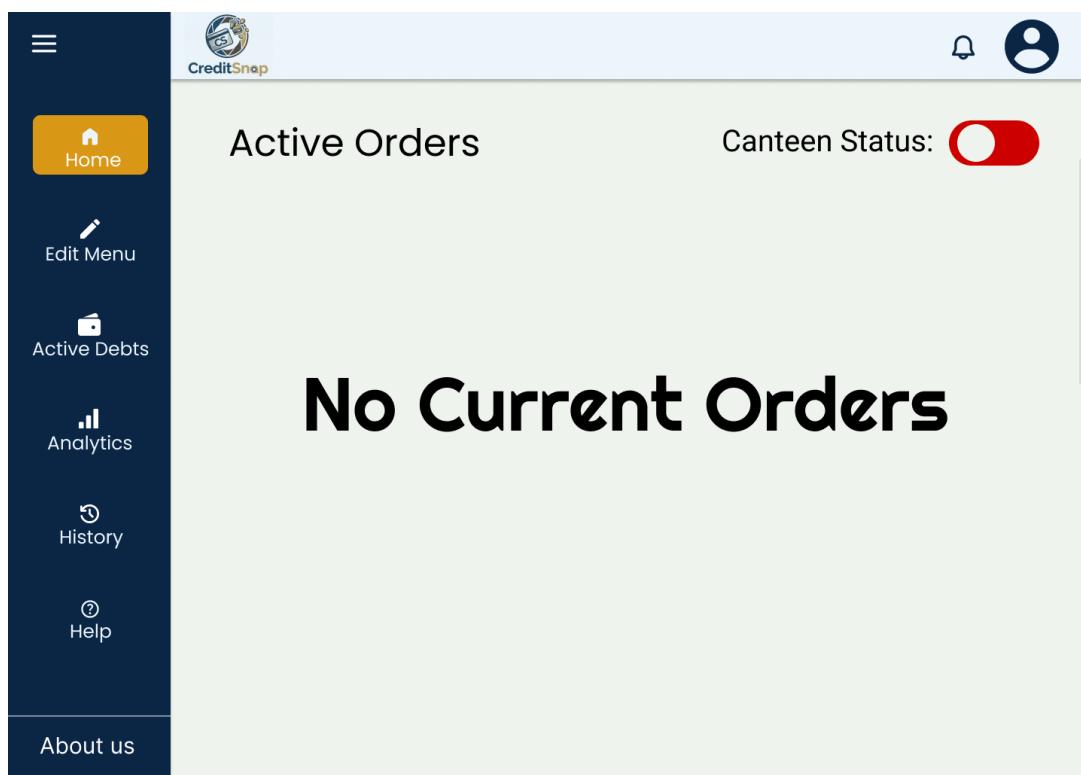
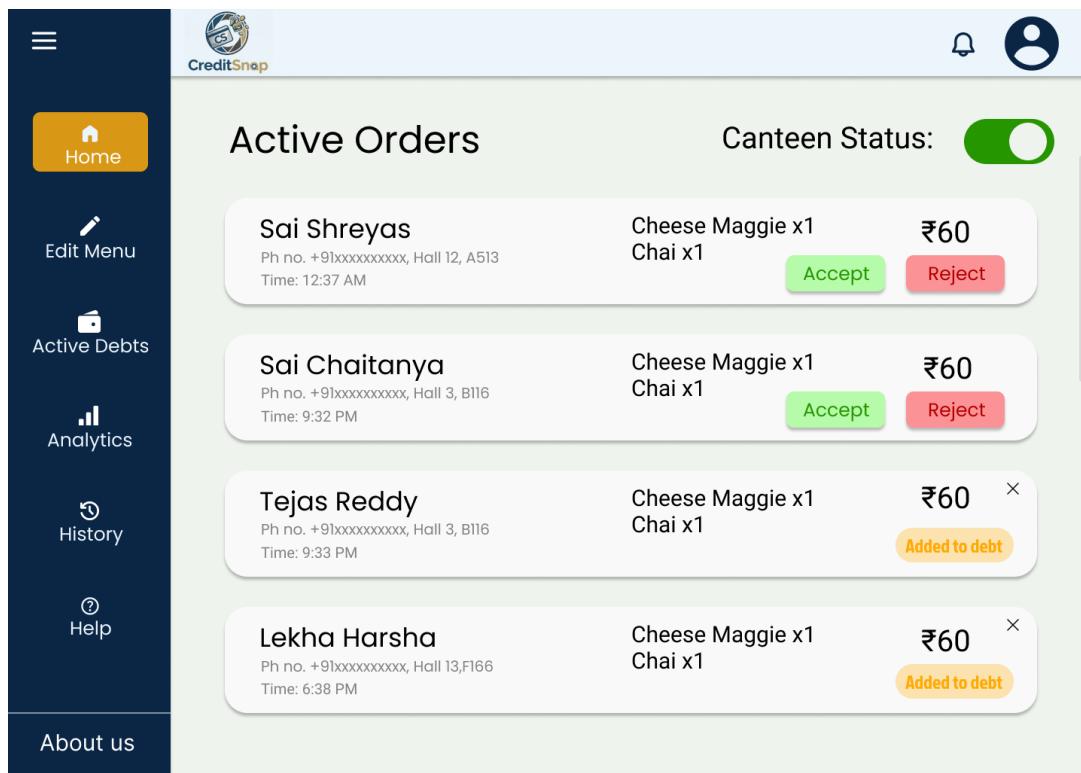
1.2.1 Login Page :



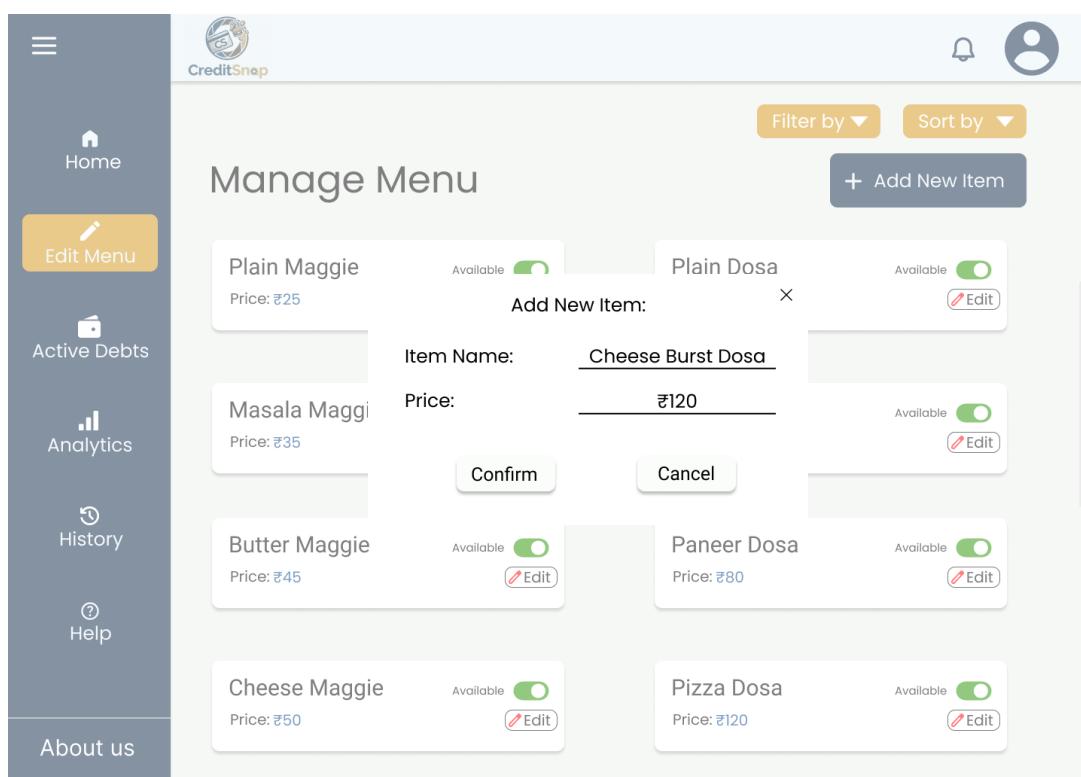
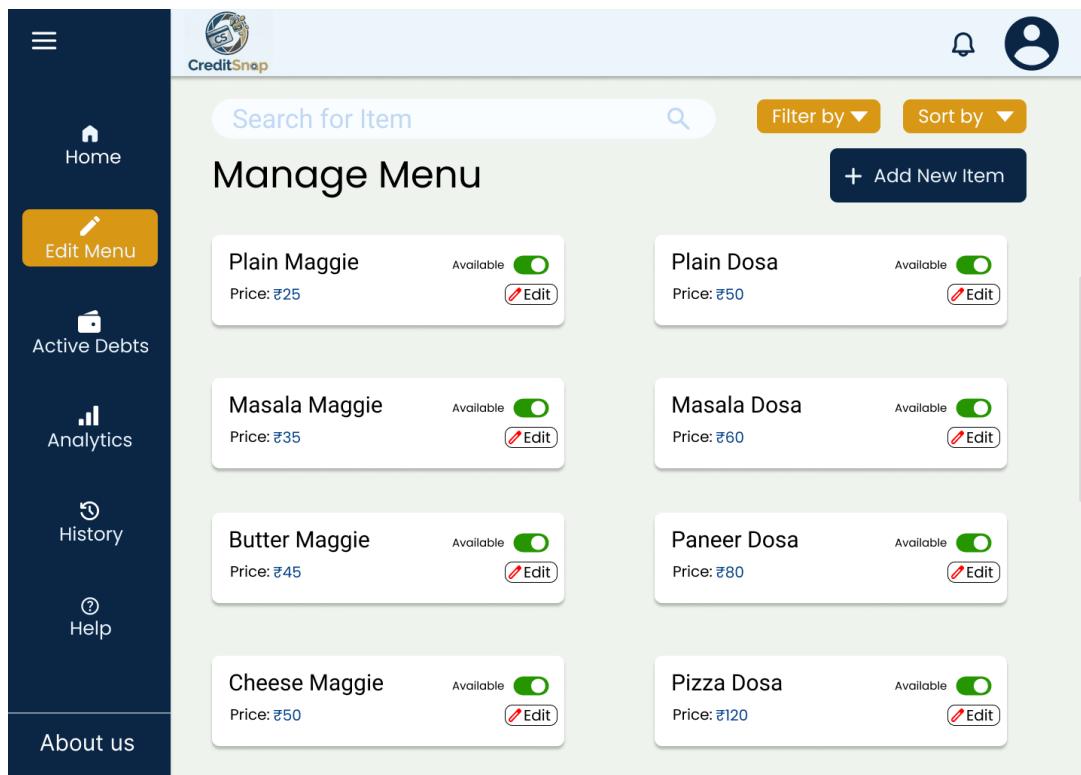
1.2.2 Sign Up Page:



1.2.3 Owner's Home Page :



1.2.4 Owner's Edit Menu Page :



1.2.5 Owner's Active Debts Page :

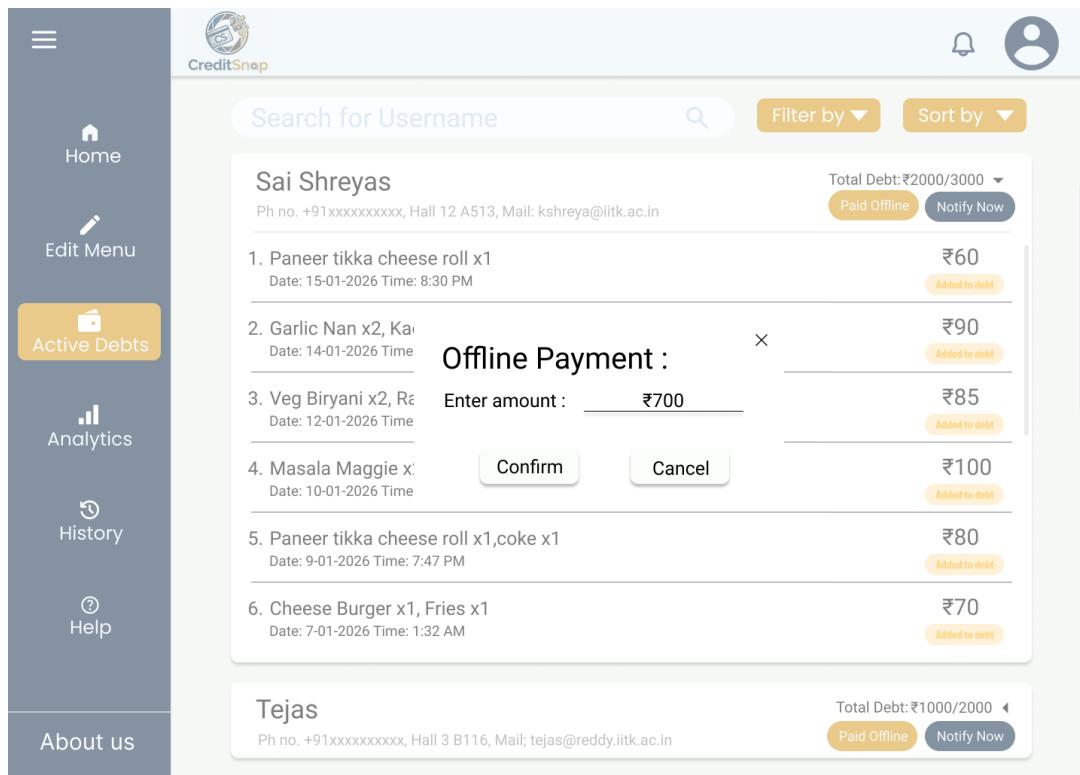
The screenshot shows the 'Active Debts' page for an owner. On the left is a dark sidebar with icons for Home, Edit Menu, Active Debts (which is orange), Analytics, History, Help, and About us. The main area has a header with the CreditSnap logo, a search bar, and filter/sort buttons. Below is a list of users:

- Sai Shreyas**: Total Debt: ₹2000/3000. Paid Offline, Notify Now.
- Tejas**: Total Debt: ₹1000/2000. Paid Offline, Notify Now.
- Sai Chaitanya**: Total Debt: ₹2200/3000. Paid Offline, Notify Now.
- Lekha Harsha**: Total Debt: ₹4500/5000. Paid Offline, Notify Now.
- Haneesh**: Total Debt: ₹2000/3000. Paid Offline, Notify Now.
- Ram Charan**: Total Debt: ₹3000/5000. Paid Offline, Notify Now.

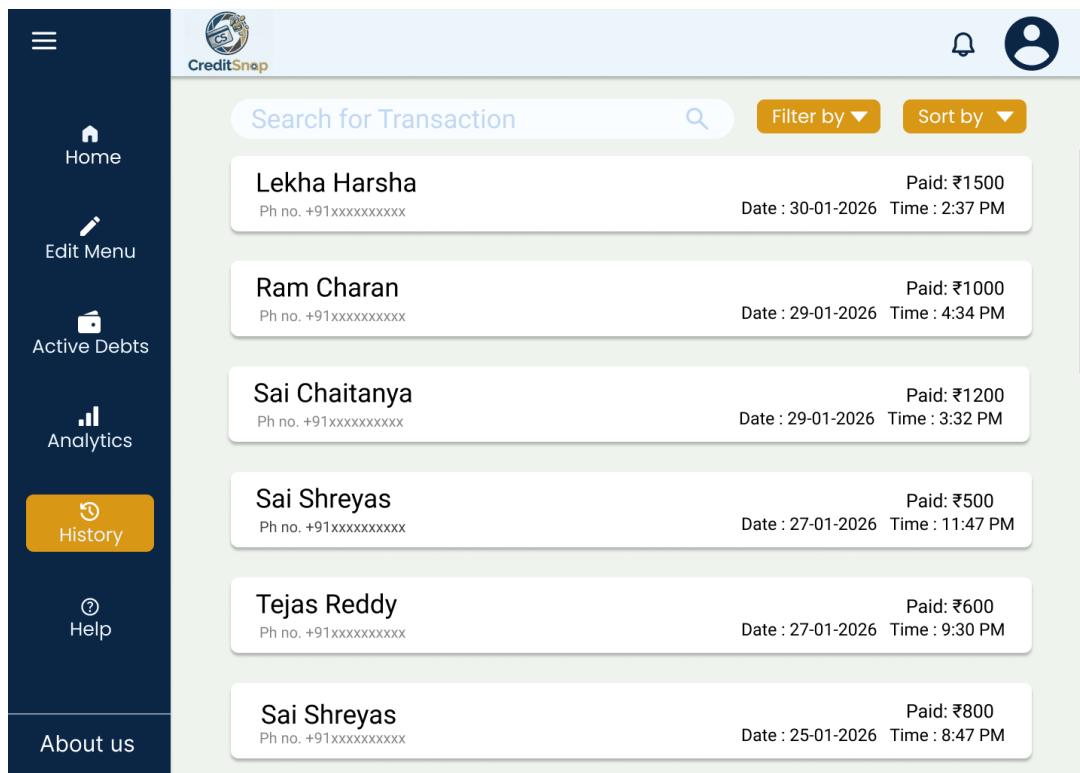
This screenshot shows the same interface but focuses on the debts of a specific user, Sai Shreyas. The sidebar and header are identical. The main area shows a detailed list of her debts:

Item Description	Amount	Status
1. Paneer tikka cheese roll x1 Date: 15-01-2026 Time: 8:30 PM	₹60	Added to debt
2. Garlic Nan x2, Kadai Paneer x1 Date: 14-01-2026 Time: 7:56 PM	₹90	Added to debt
3. Veg Biryani x2, Raita x1 Date: 12-01-2026 Time: 9:51 PM	₹85	Added to debt
4. Masala Maggie x2, Fries x1 Date: 10-01-2026 Time: 11:57 PM	₹100	Added to debt
5. Paneer tikka cheese roll x1, coke x1 Date: 9-01-2026 Time: 7:47 PM	₹80	Added to debt
6. Cheese Burger x1, Fries x1 Date: 7-01-2026 Time: 1:32 AM	₹70	Added to debt

Below this, there is a summary for Tejas: Total Debt: ₹1000/2000, Paid Offline, Notify Now.



1.2.6 Owner's History Page :



1.2.7 Owner's Analytics Page :

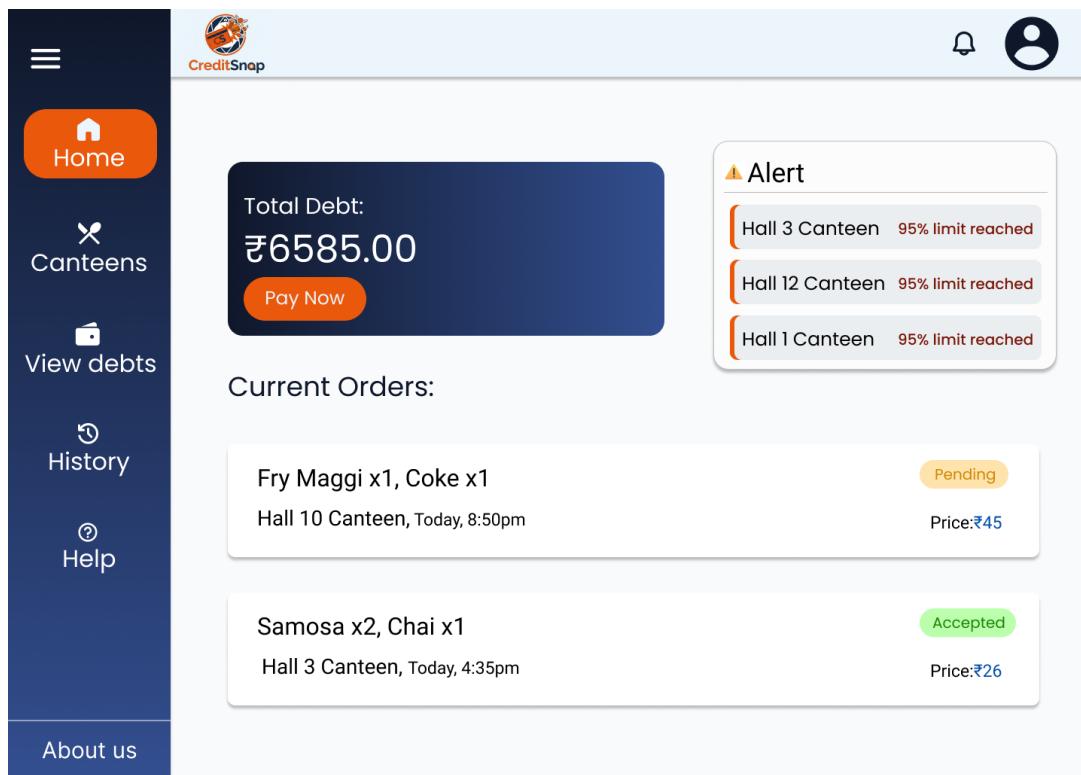


1.2.8 Owner's Help Page :

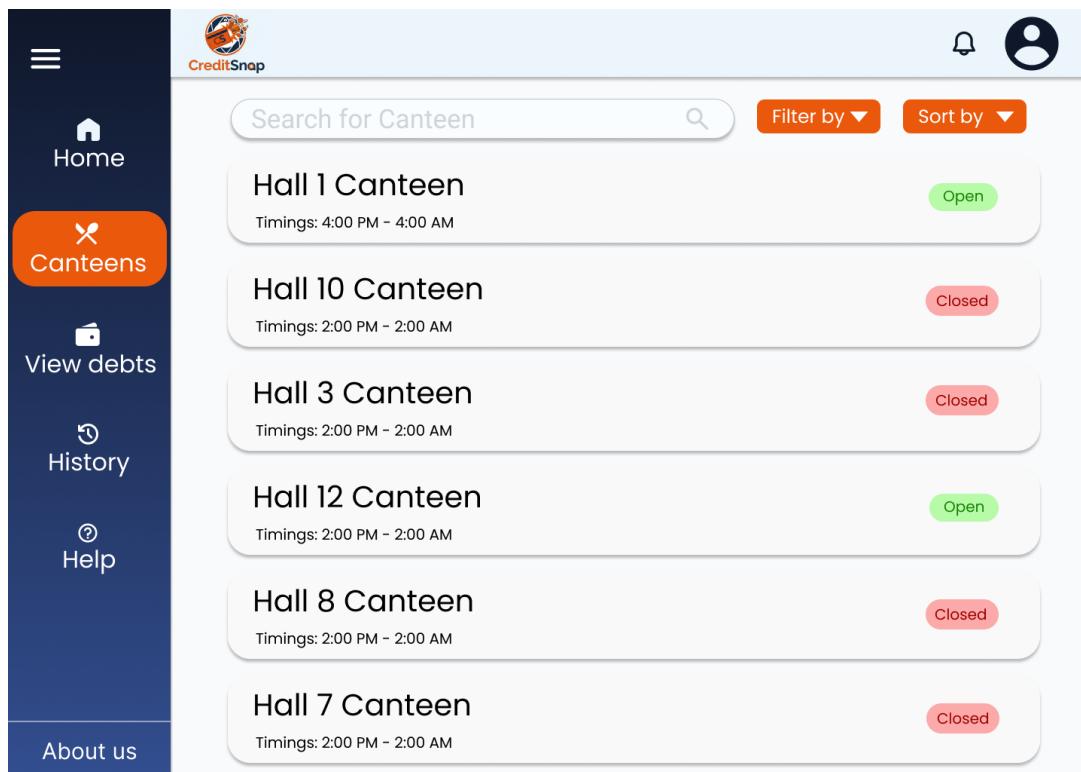
The page includes a sidebar with navigation links and a main content area for frequently asked questions (FAQs):

- FAQs Section:**
 - How to edit menu:** A detailed description explaining how to change item availability and delete items.
 - Payment related issues:**
 - Understand what is Credit:**
 - Updating Personal Information:**
 - How to clear Debt manually:**
- Contact Us:** An email address for queries: CreditSnap@gmail.com

1.2.9 Student's Home Page :



1.2.10 Student's Canteens Page :



Hall 1 Canteen

Search for Item Filter by ▾ Sort by ▾

Plain Maggie Price: ₹25	- 1 +	Plain Dosa Price: ₹50	- 2 +
Cheese Maggie Price: ₹50	Order Now	Masala Maggie Price: ₹35	Order Now
Chicken Biryani Price: ₹120	Order Now	Veg Biryani Price: ₹60	Order Now

Plain Maggie x1, Plain Dosa x2
₹125 **Check Out**

Hall 1 Canteen

Filter by ▾ Sort by ▾

Search for Items

S.No	Food items	Quantity	Cost
1.	Plain Maggie	- 1 +	₹25
2.	Plain Dosa	- 2 +	₹100
		Total	₹125

Place Debt Request

1.2.11 Student's View Debts Page :

The screenshot shows the 'View debts' section of the app. On the left is a sidebar with icons for Home, Canteens, View debts (highlighted in orange), History, Help, and About us. The main area displays a list of canteens with their current debt amounts and a 'Clear Debt' button.

Canteen	Debt	Action
Hall 10 Canteen	Debt:3915/5000	<button>Clear Debt</button>
Hall 1 Canteen	Debt:1180/10000	<button>Clear Debt</button>
Hall 12 Canteen	Debt:875/6000	<button>Clear Debt</button>
Hall 6 Canteen	Debt:530/5000	<button>Clear Debt</button>
Hall 3 Canteen	Debt:85/3000	<button>Clear Debt</button>

The screenshot shows the 'View debts' section of the app. The sidebar is identical to the previous screenshot. The main area shows a summary for Hall 10 Canteen and a detailed history of transactions.

Date	Time	Paid/Debt	Amount
Dec 20, 2025	5:30pm	Paid Online	-₹500
Dec 19, 2025	7:35pm	Paid Offline	-₹1000
Dec 16, 2025	10:30pm	Debt taken	+₹350
Dec 16, 2025	5:50pm	Debt taken	+₹200

Total Paid: ₹6500

1.2.12 Student's History Page :

The screenshot shows the 'History' section of the student dashboard. On the left is a dark sidebar with navigation links: Home, Canteens, View debts, History (highlighted in orange), Help, and About us. The main area displays a list of four recent orders:

- Samosa x2, Chai x1** - Accepted, Price: ₹26. Located in Hall 3 Canteen, Today, 4:35pm.
- Chicken Biryani x2, Masala Dosa x4** - Rejected. Located in Hall 10 Canteen, Yesterday, 11:50pm.
- Chicken Macaroni x2, Coke x2** - Accepted, Price: ₹160. Located in Hall 10 Canteen, Yesterday, 8:20pm.
- Matar Mushroom x1, Butter Roti x4, Sprit x1** - Accepted, Price: ₹100. Located in Hall 7 Canteen, 20-12-2025, 10:30pm.

At the top right are icons for a bell and user profile. At the top center are search, filter, and sort buttons.

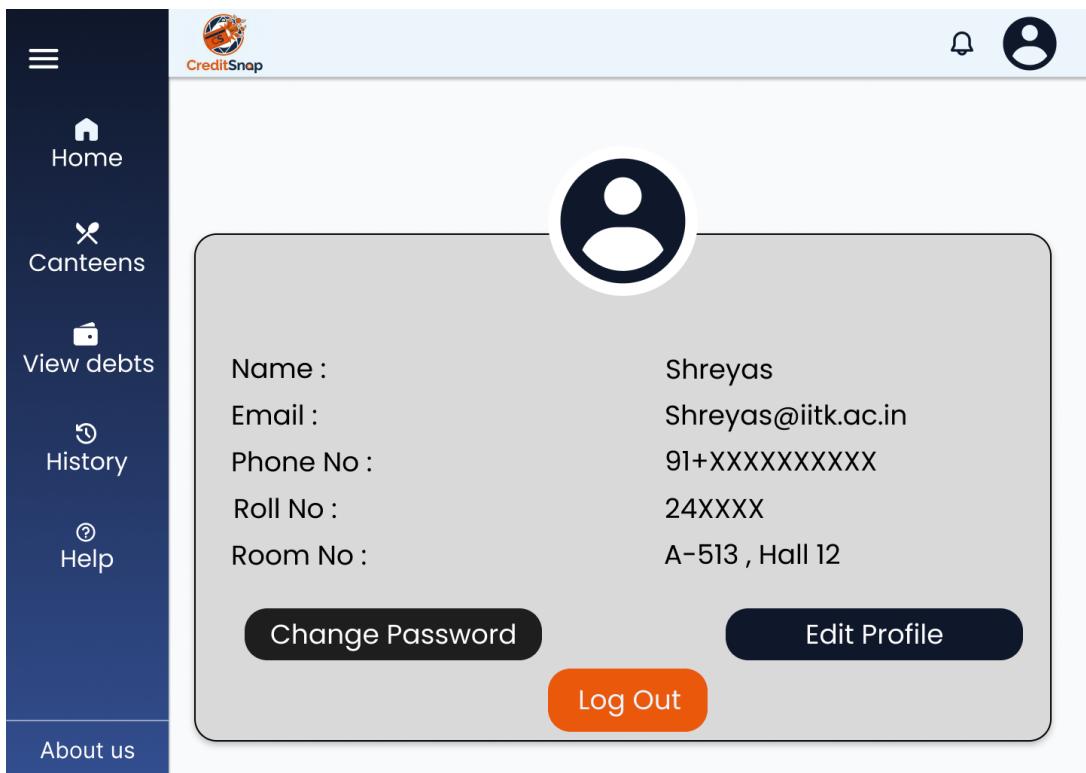
1.2.13 Student's Help Page :

The screenshot shows the 'Help' section of the student dashboard. The sidebar includes: Home, Canteens, View debts, History, Help (highlighted in orange), and About us. The main content is titled 'FAQs' and contains the following expandable sections:

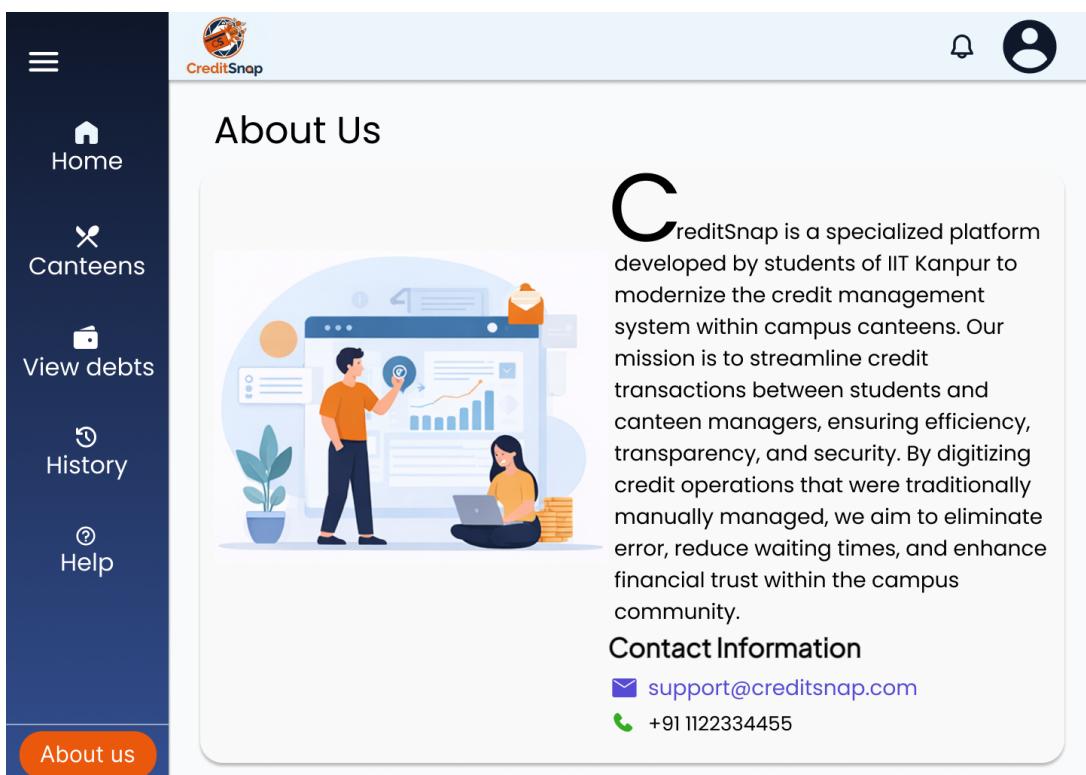
- My order was rejected** (with a downward arrow icon): Your order has been rejected by the canteen owner due to some reason. Therefore your debt balance remains unchanged. For any clarification on why the order was canceled, Please reach out to the canteen owner, or you can continue with your next order.
- Payment Related Issues** (with a right-pointing arrow icon)
- Can i pay my debt offline** (with a right-pointing arrow icon)
- Updating Personal Information** (with a right-pointing arrow icon)
- How to pay Debts** (with a right-pointing arrow icon)

Below the FAQs is a 'Contact Us' section with an email address: CreditSnap@gmail.com.

1.2.14 Student's Profile Page :

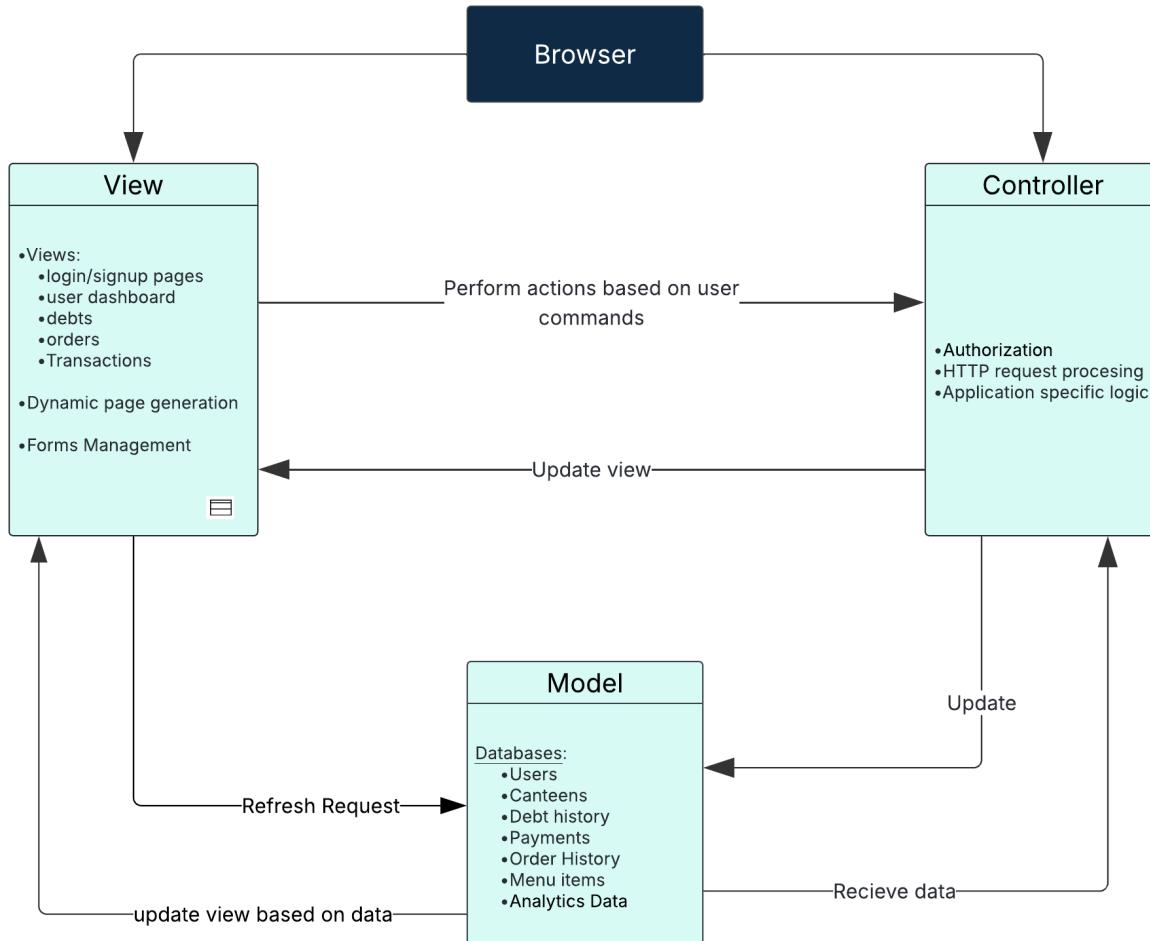


1.2.15 About Us Page :



2 Architecture Design

The overall architecture of the Credit Snap website follows the Model-View-Controller pattern.



Model-View-Controller :

The architecture follows the Model View Controller pattern, which organizes the application into three logical components: **Model**, **View**, and **Controller**. Each of these components is built to handle specific development aspects of an application.

Model :

- The Model in our Credit Snap system is responsible for managing the database using the **MongoDB** database management system. It manages tasks related to storing student profiles, canteen menus, order records, and the credit ledger.
- The Model receives information about new orders, payment updates, and menu changes from the Controller. It ensures that critical data rules are followed, such as checking if a student has crossed their **Debt Ceiling** (limit) or ensuring that closed transactions are stored safely.

View :

- The View component defines the user interface for our Credit Snap website, incorporating **React.js** to build a responsive design. It handles the presentation layer, showcasing features like the **Student Dashboard**, **Live Menu**, **Active Orders**, and **Debt Status**.
- The View acts like the presentation layer of the web application as it provides an interface for Students and Canteen Owners to interact with the system. It displays real-time updates, such as showing “Order Accepted” or “Limit Reached,” based on the data it receives.

Controller :

- It acts as an intermediate between the Model and View. It utilizes Node.js and Express.js to manage the application logic and flow. The Controller receives data regarding user interactions from the View and sends relevant information to the Model for database operations.
- The Controller acts like a logical segment between the view and model; it determines processes and manages the flow of the data. It handles core logic such as **User Authentication** (Login/Signup), processing **Order Requests**, and managing **Debt Calculations**.

This architecture should meet the following non-functional requirements:

- **Performance :** The MERN stack provides a lightweight, asynchronous, and event-driven architecture which ensures high performance and responsiveness, meeting the requirement of page loads under 3 seconds. The use of Node.js enables non-blocking I/O operations, allowing the system to efficiently handle concurrent active users without server timeouts. Additionally, MongoDB's document-oriented structure enables fast data retrieval for live menus and transaction histories.
- **Security & Privacy :** Security is a top priority as the platform handles sensitive student financial data. Our architecture implements robust security measures by strictly separating the Controller logic from the View. The Controller enforces Role-Based Access Control (RBAC) using JWT, ensuring that students cannot access Canteen Owner endpoints. Furthermore, the Model layer enforces the Immutable Ledger Policy, architecturally preventing the deletion of accepted transaction records to ensure financial integrity.
- **Maintainability:** The strict separation of the User Interface (Frontend) from the Server-Side (Backend) logic ensures high maintainability. By using the MVC pattern, developers can modify the React.js frontend to improve the user experience without accidentally breaking the core Business Logic (like debt calculations) residing in the Controller and Model layers. This organized structure simplifies debugging and future feature additions.
- **Scalability:** The system is designed to handle the growing number of transactions and student profiles over the semester. The backend (Node.js with Express.js)

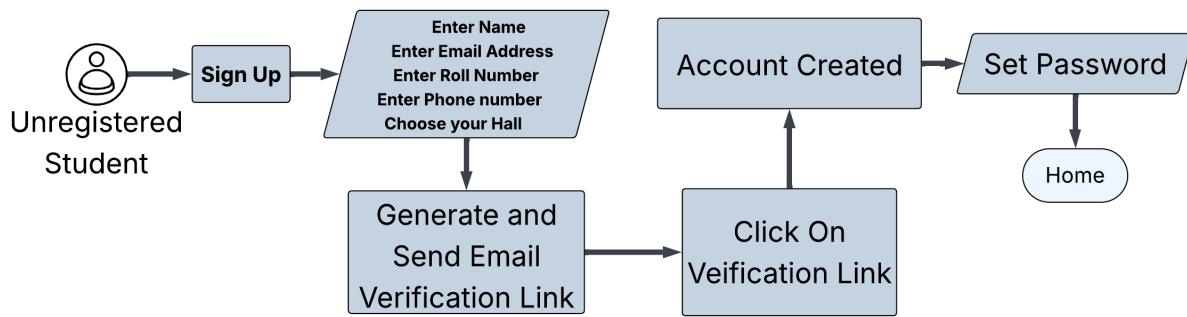
and the database (MongoDB) can be scaled to manage large datasets of debt ledgers and user profiles without impacting query speeds or report generation times, ensuring the system remains stable even as the user base expands.

- **Portability:** The use of React.js for the frontend ensures that the application is responsive and functions flawlessly across all major platforms and devices, including smartphones, tablets, and laptops. This fulfills the requirement for students to access credit facilities from anywhere on campus using standard web browsers.
- **Adaptability & Flexibility:** The architecture is designed to support the dynamic nature of a canteen environment. The integration between the View and the Controller allows for real-time synchronization. For instance, when a Canteen Owner marks an item as “Unavailable” in the dashboard, the change is immediately reflected in the database and propagated to the Student’s view without requiring a full page refresh, preventing overselling.

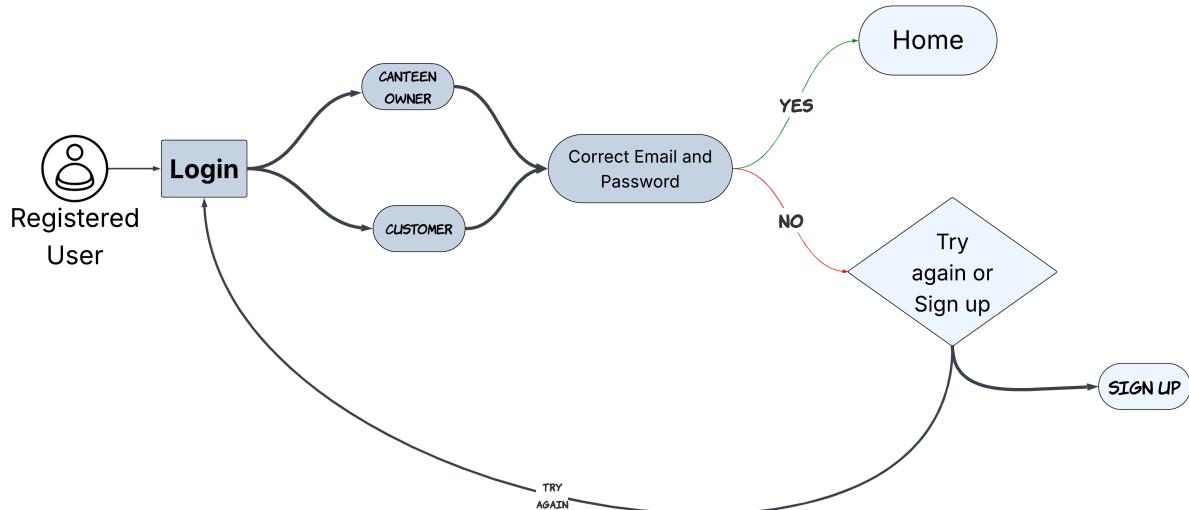
3 Object Oriented Design

3.1 Use Case Diagrams

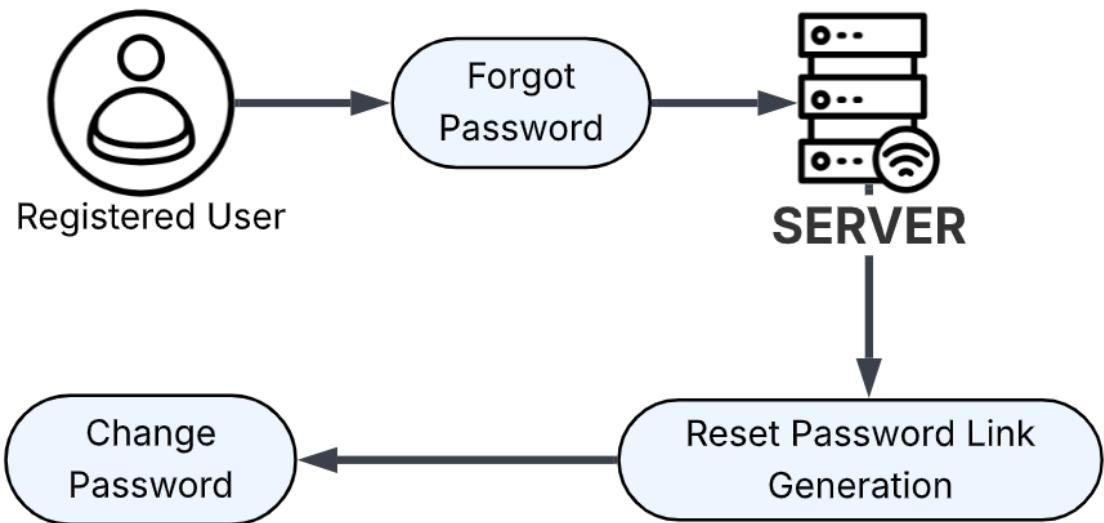
3.1.1 Use Case #1 : Student registration



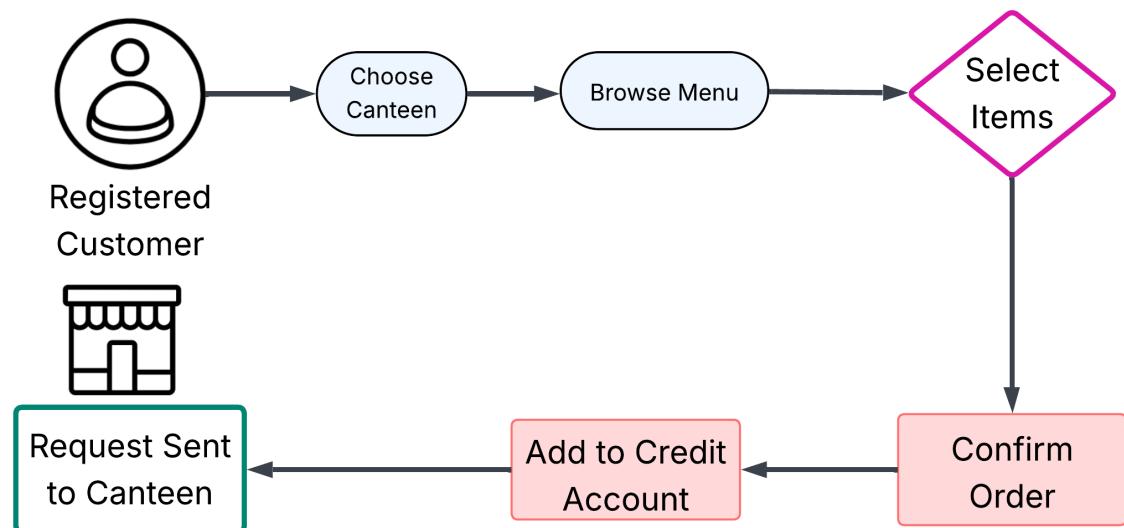
3.1.2 Use Case #2 : Registered user login



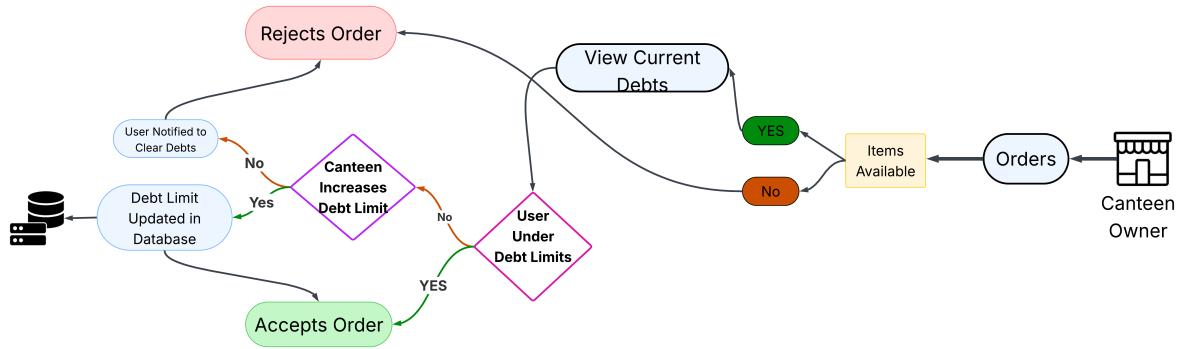
3.1.3 Use Case #3 : Forgot password



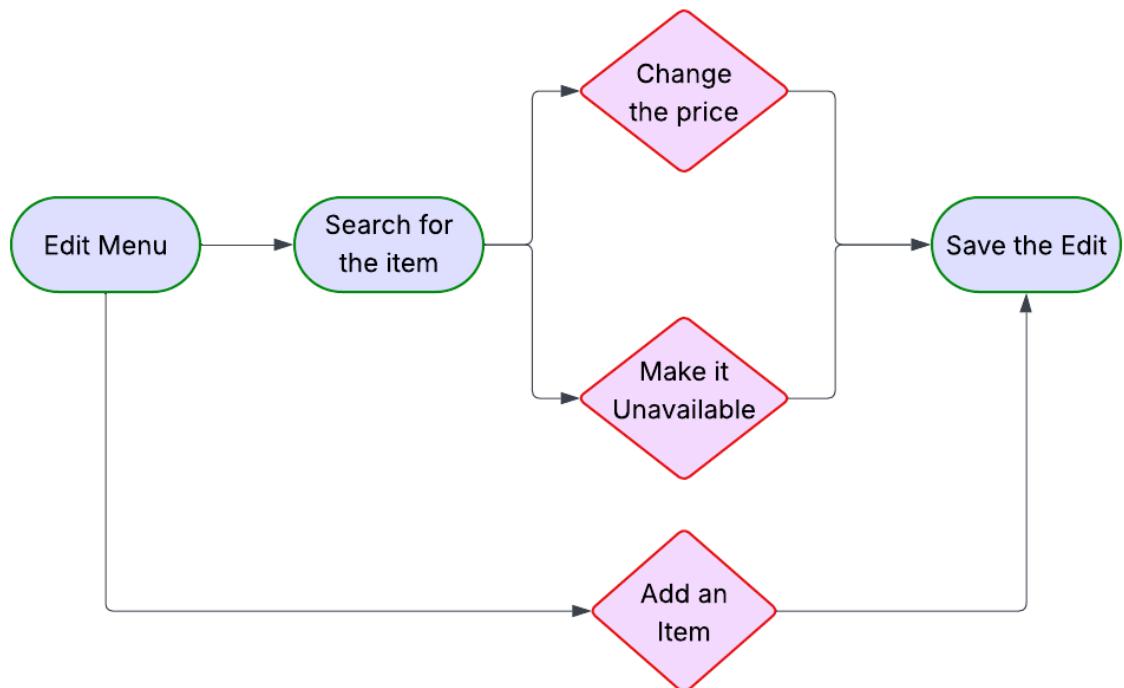
3.1.4 Use Case #4 :Student placing order



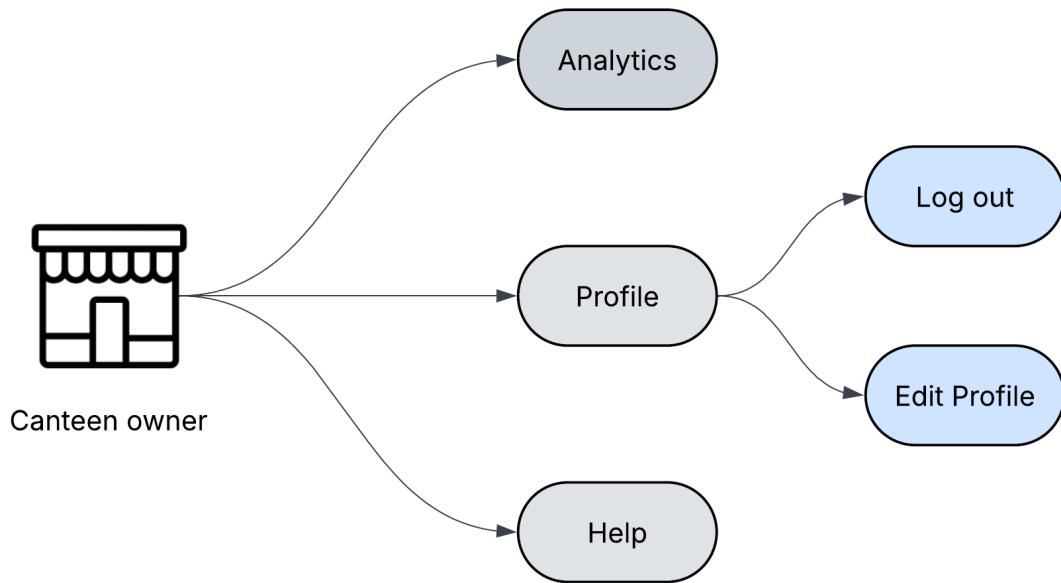
3.1.5 Use Case #5 :Managing order (Canteen owner)



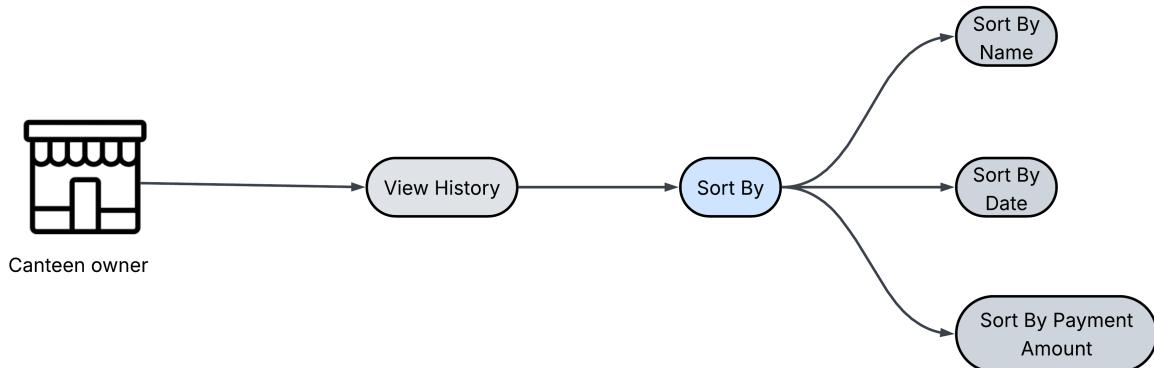
3.1.6 Use Case #6 : Edit menu (Canteen owner)



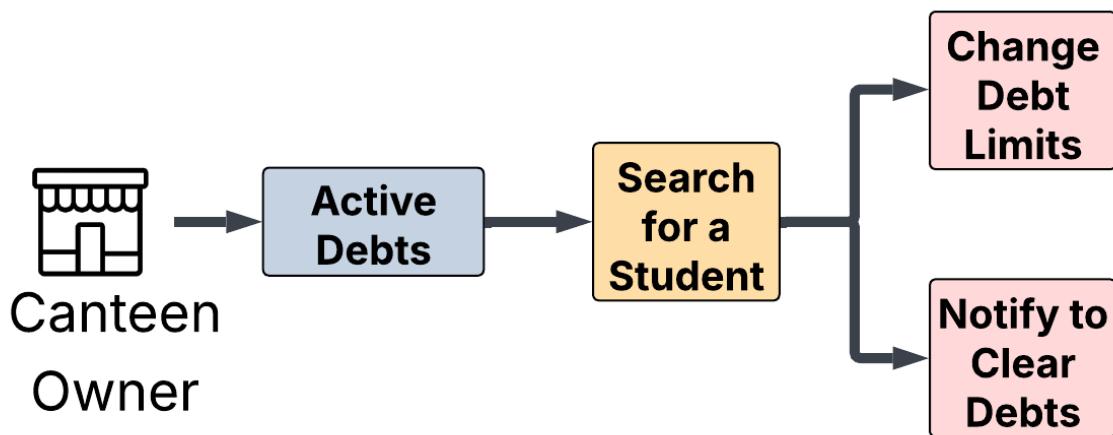
3.1.7 Use Case #7 : Canteen owner navigations



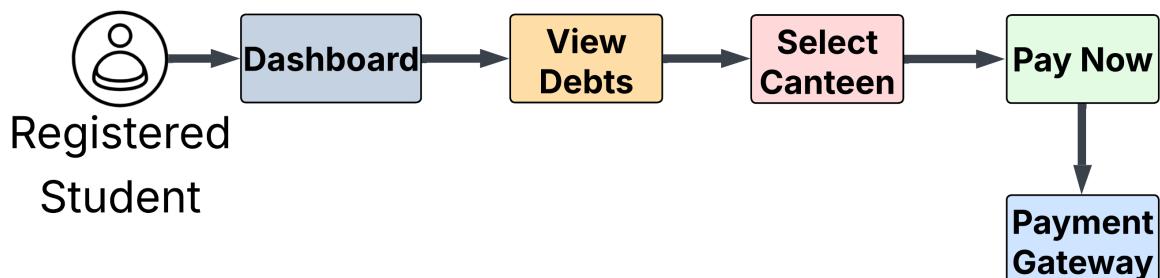
3.1.8 Use Case #8 : History of transactions



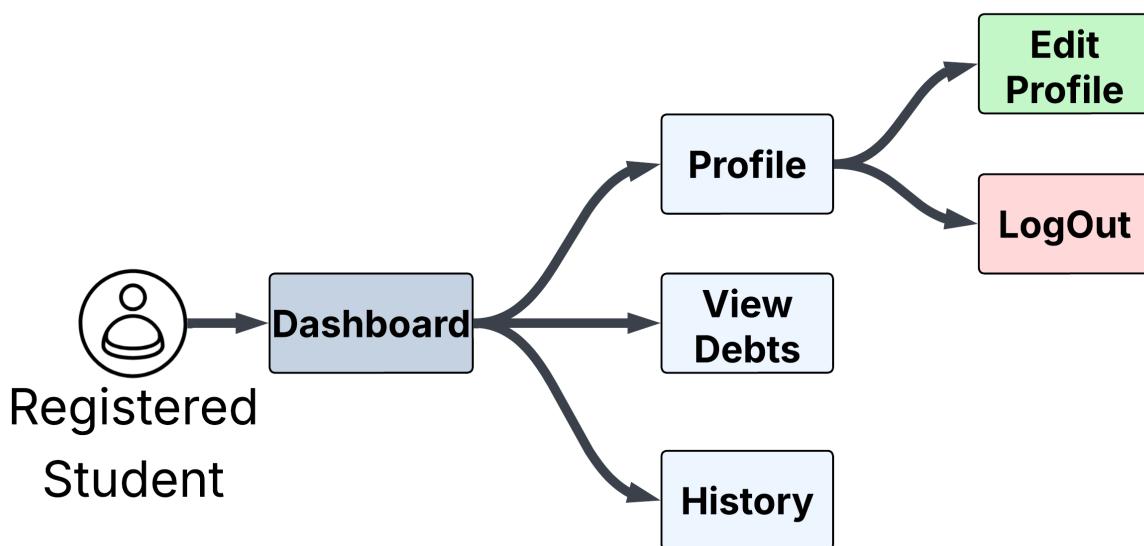
3.1.9 Use Case #9 : Activity flow for managing Active student debts



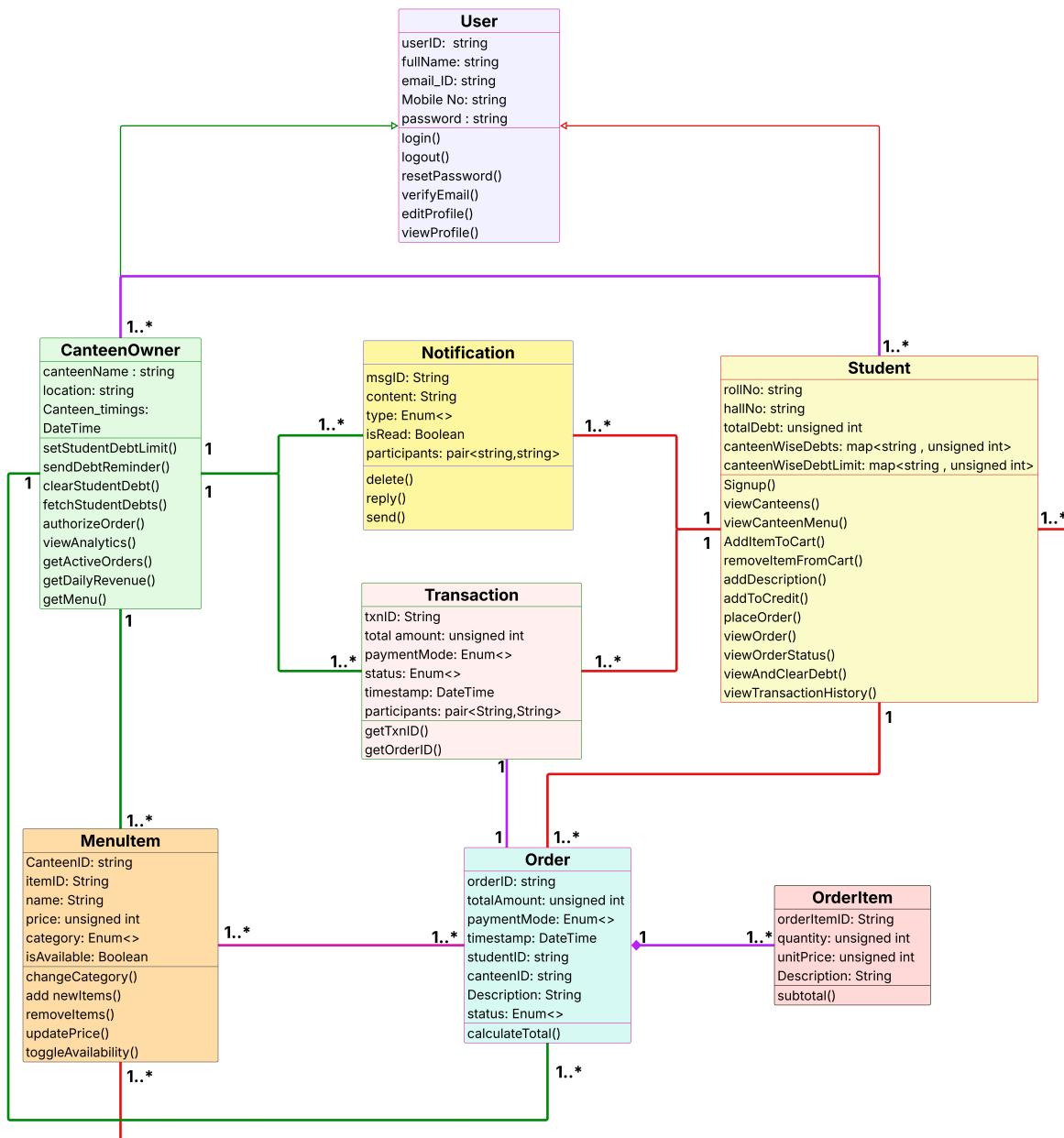
3.1.10 Use Case #10 : Student activity flow for clearing debts



3.1.11 Use Case #11 : Student navigations

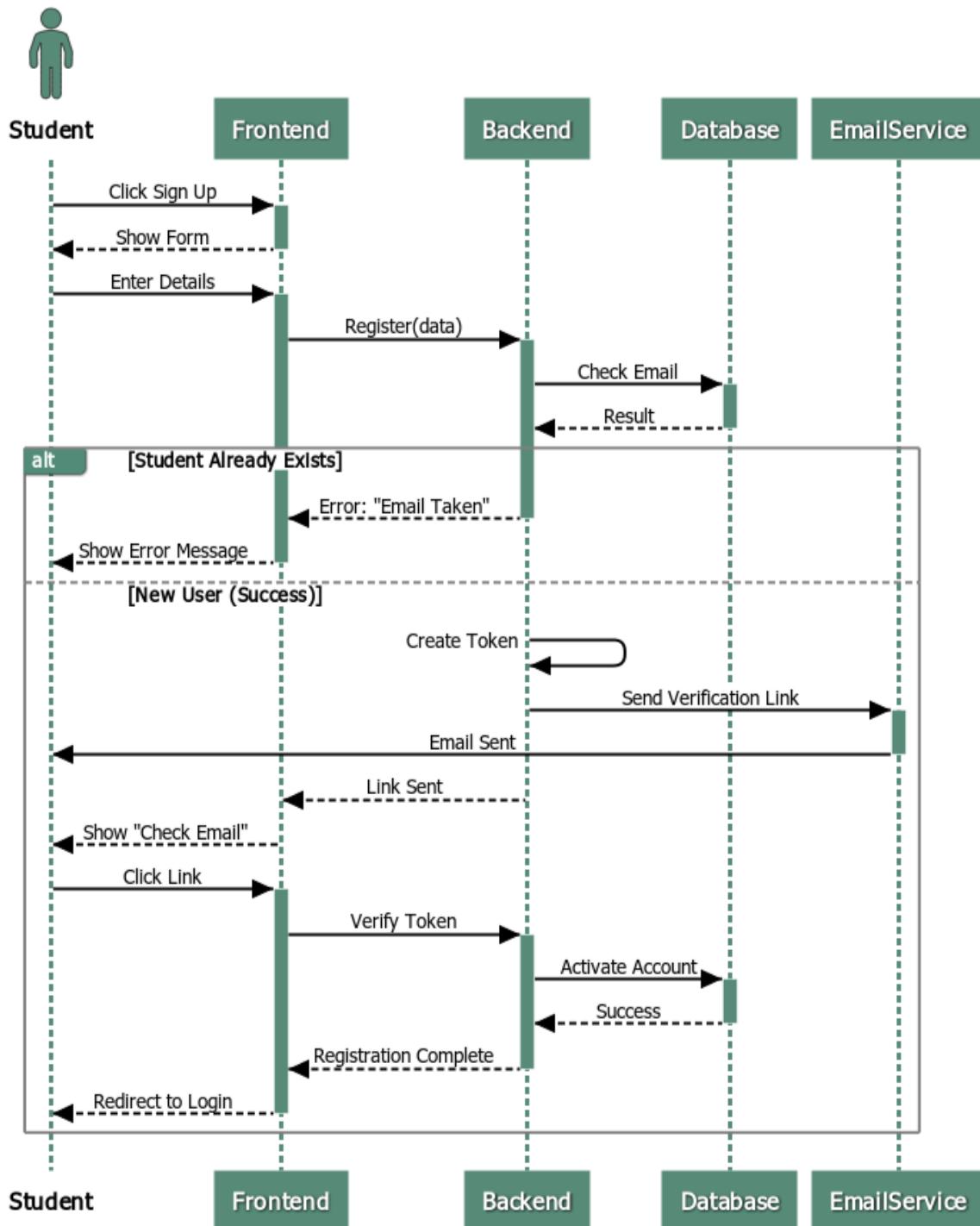


3.2 Class Diagrams

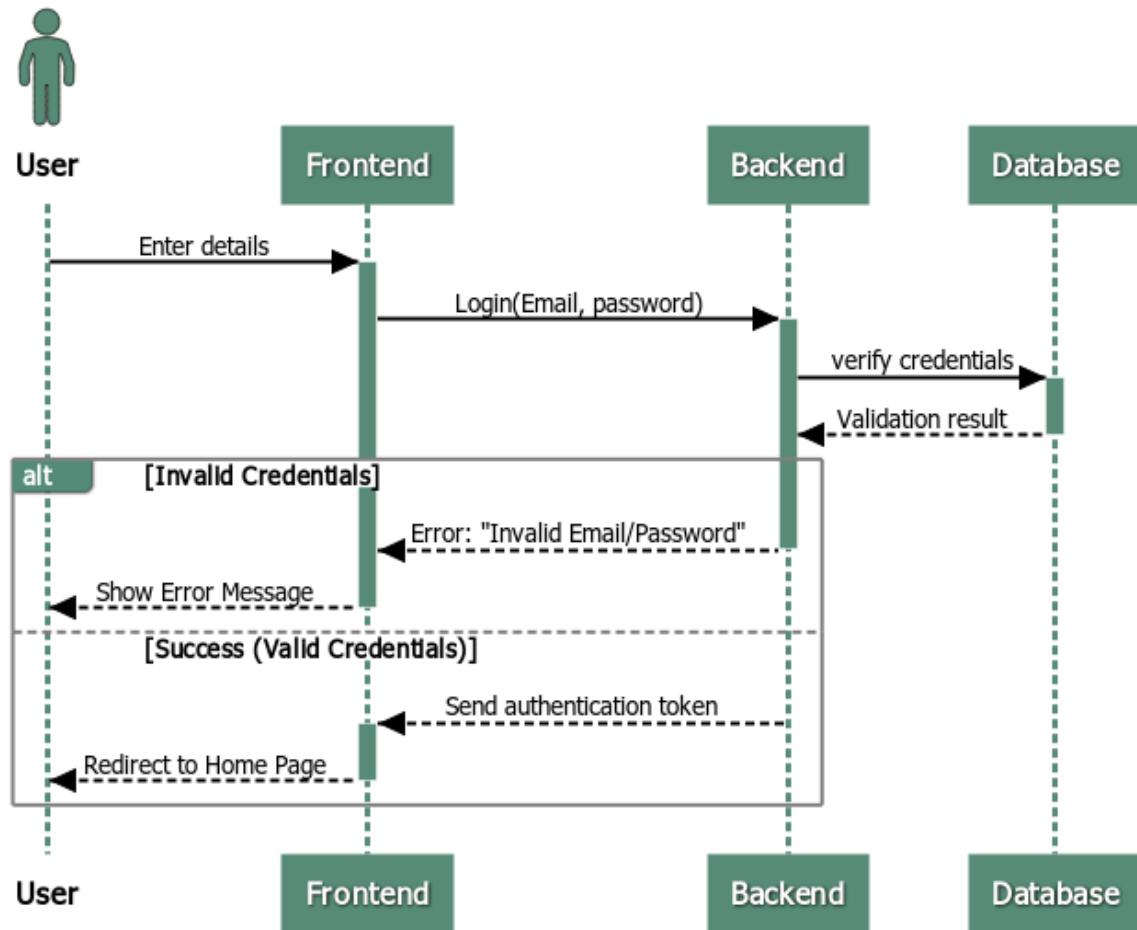


3.3 Sequence Diagrams

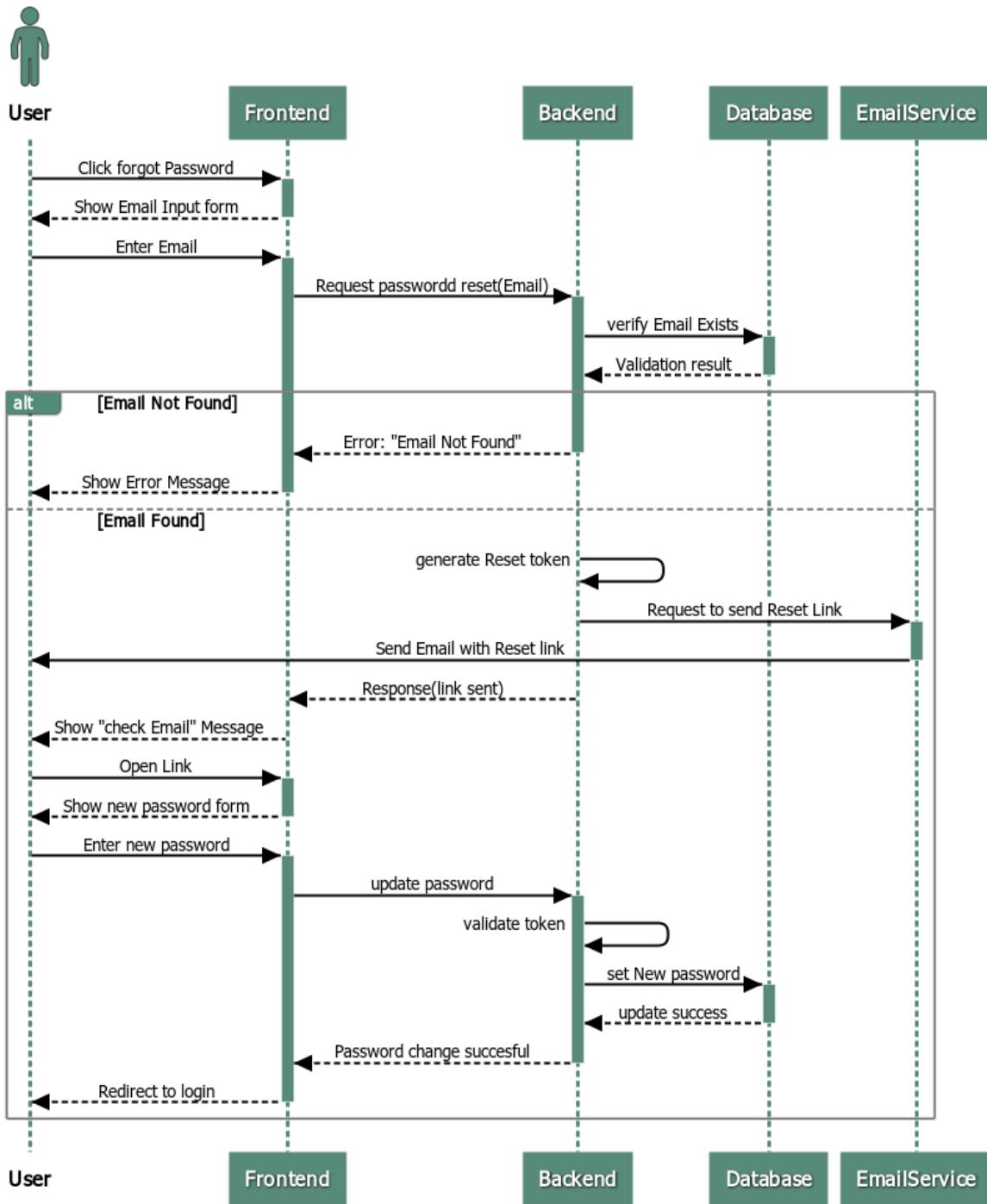
3.3.1 Registration of student :



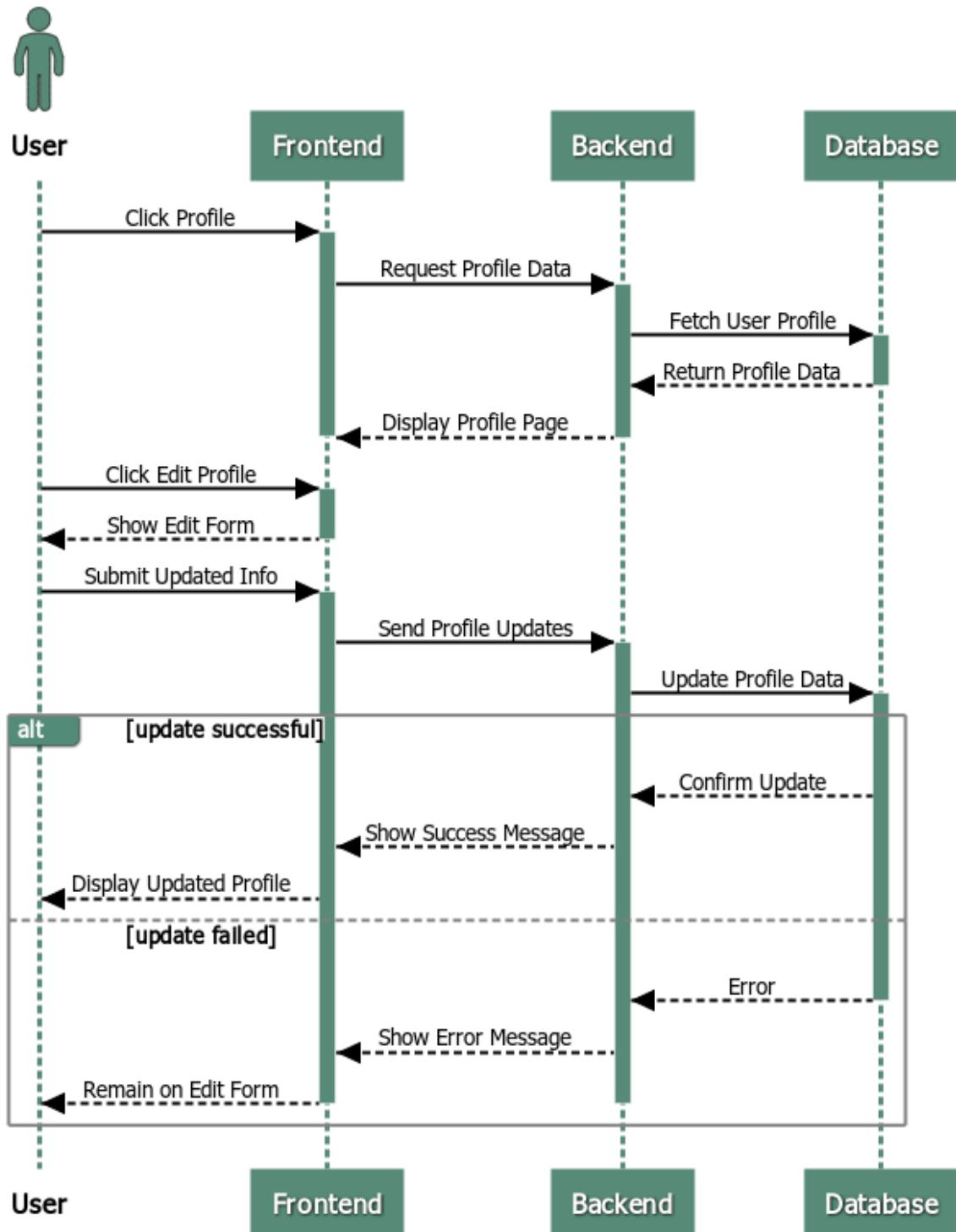
3.3.2 Login Page :



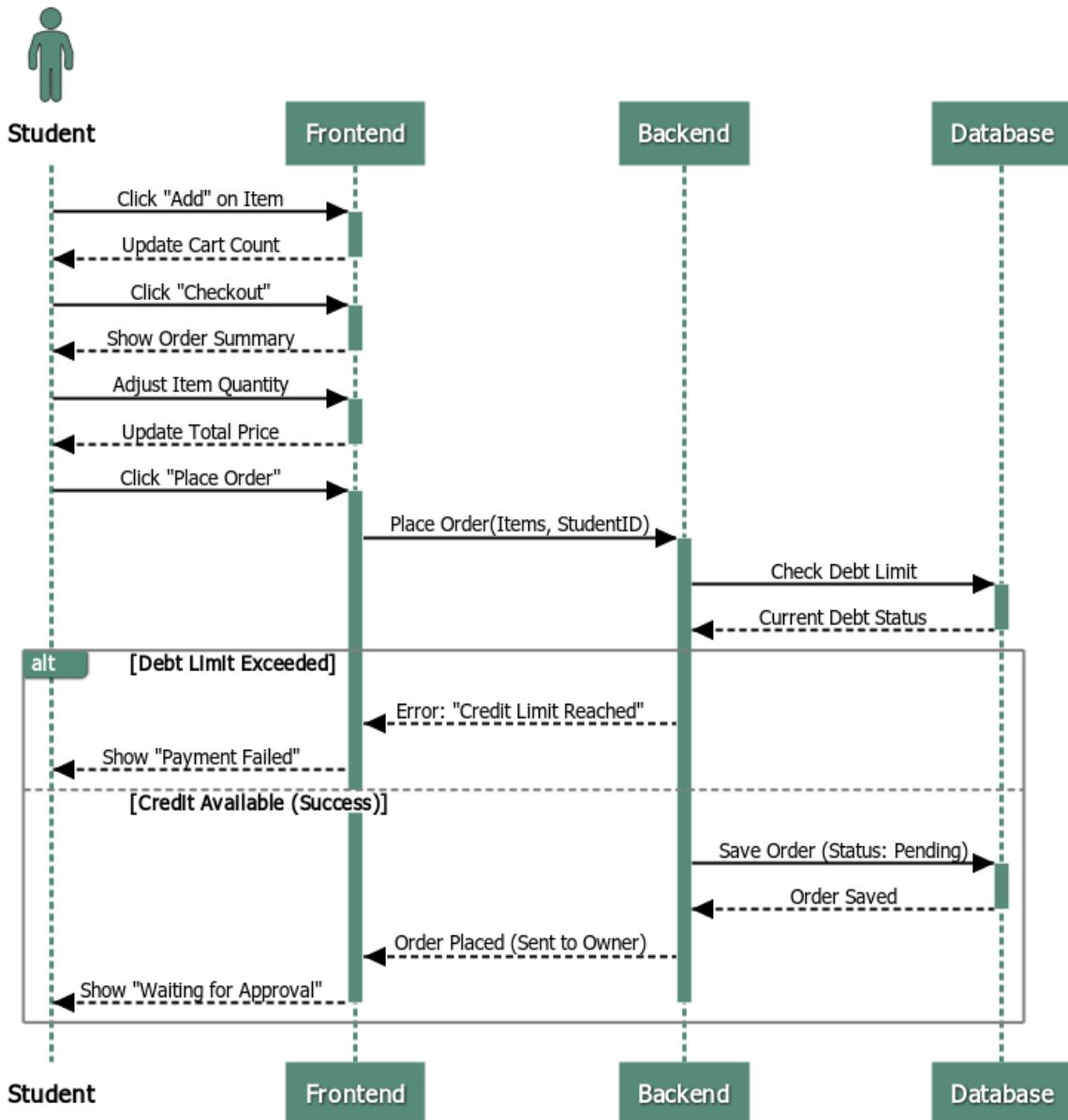
3.3.3 Forgot password :



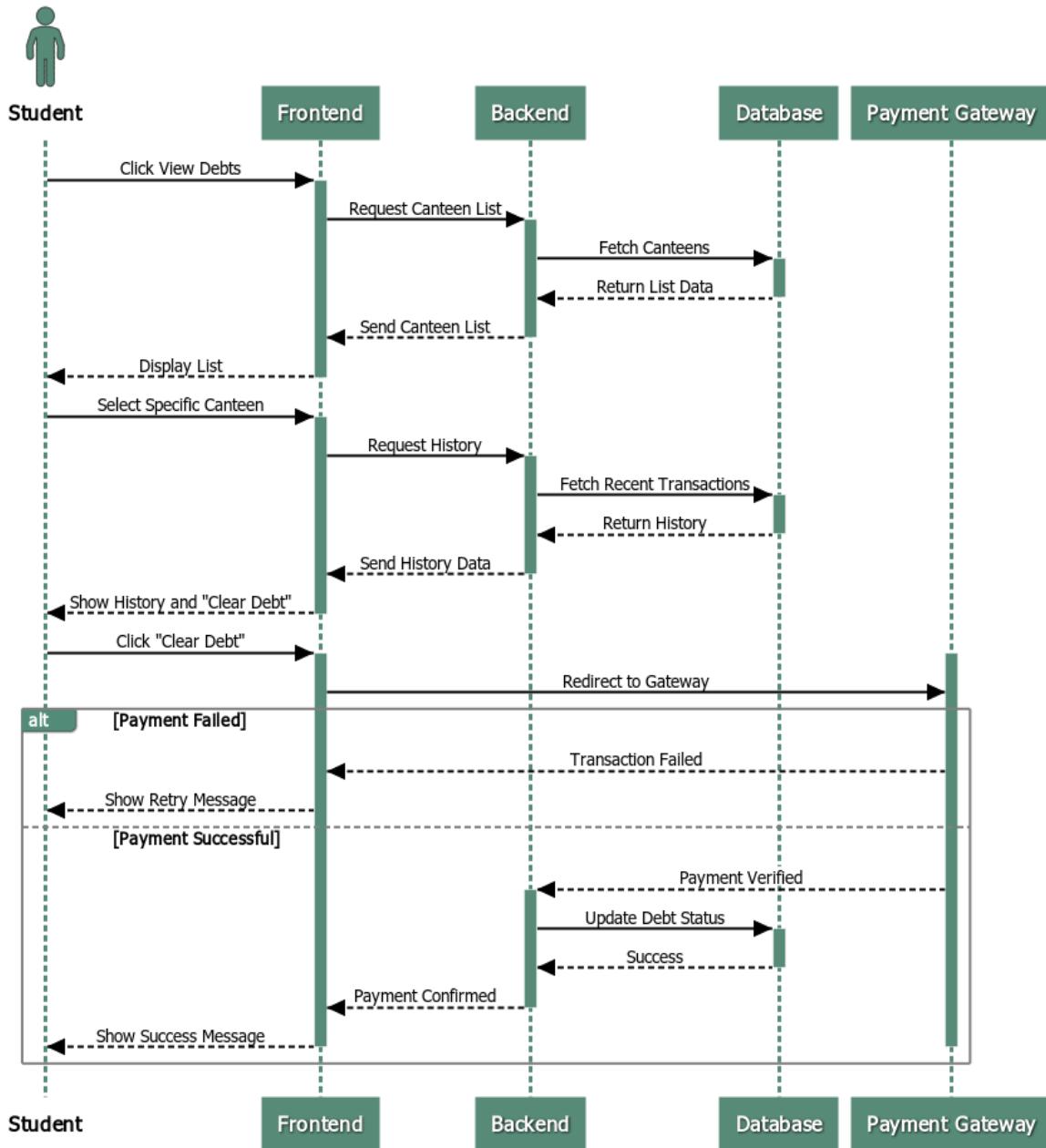
3.3.4 Edit Profile :



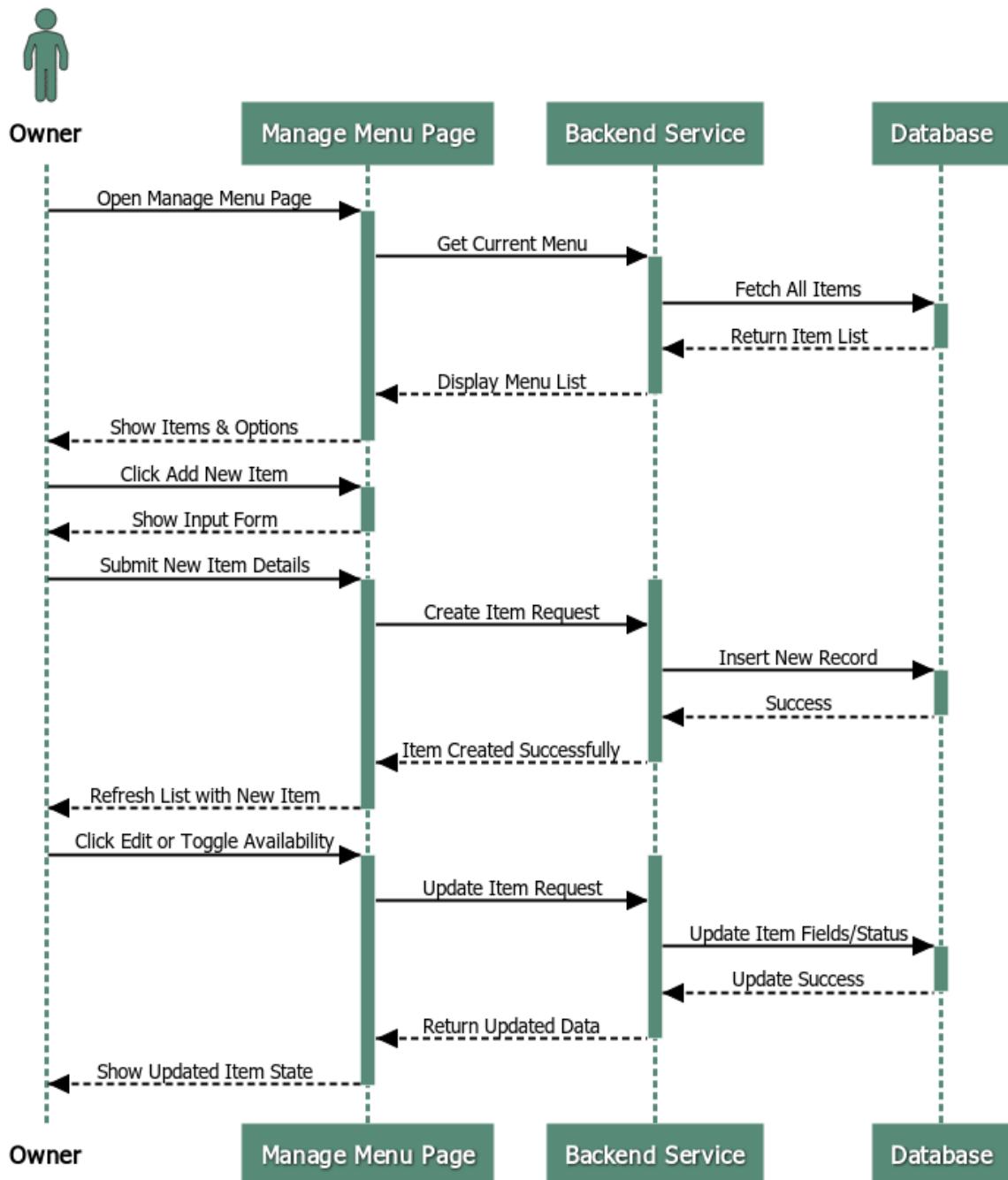
3.3.5 Order Food:



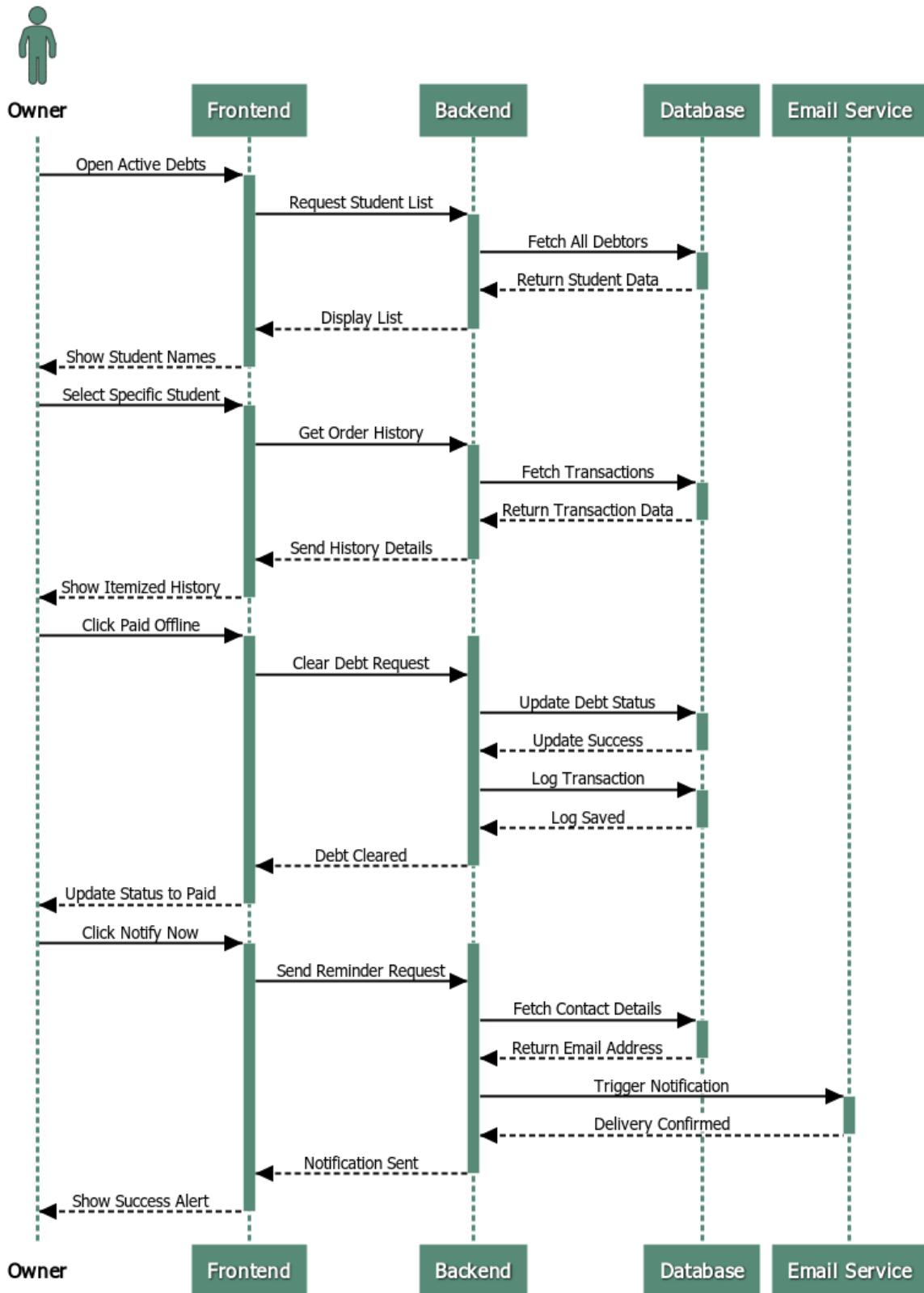
3.3.6 Clear debts:



3.3.7 Edit Menu (Canteen Owner) :

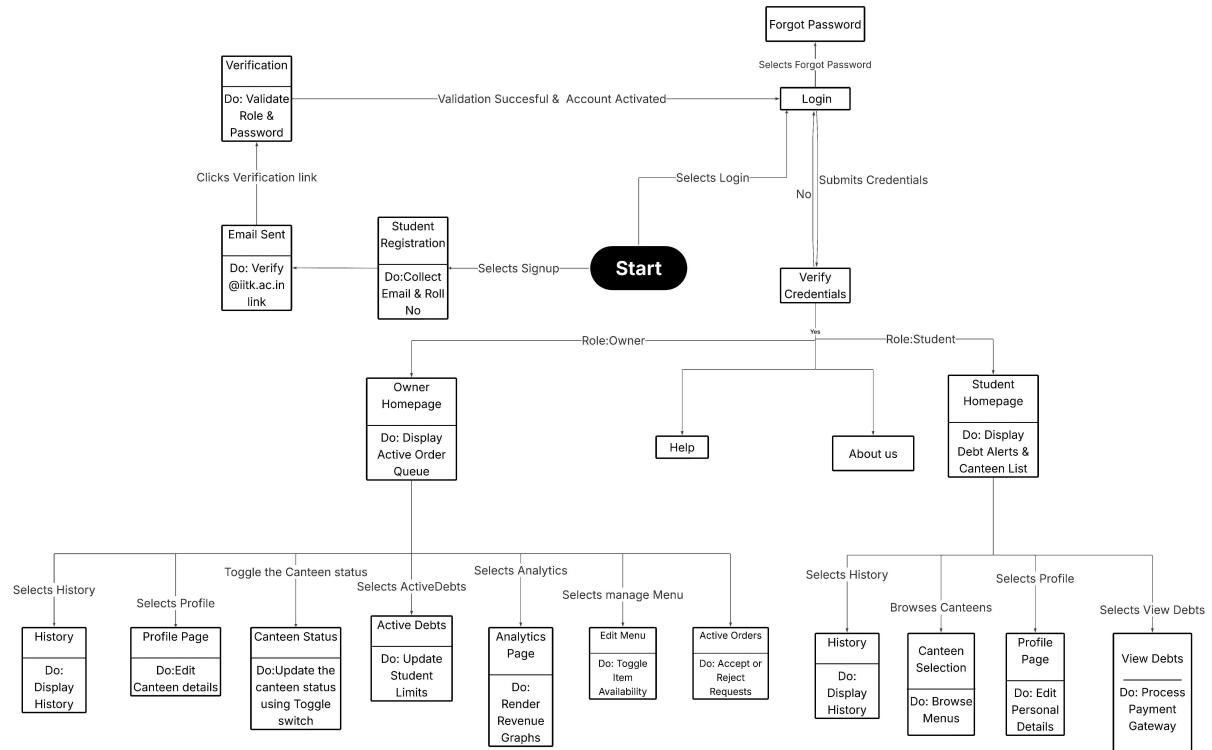


3.3.8 Manage Active Debts :

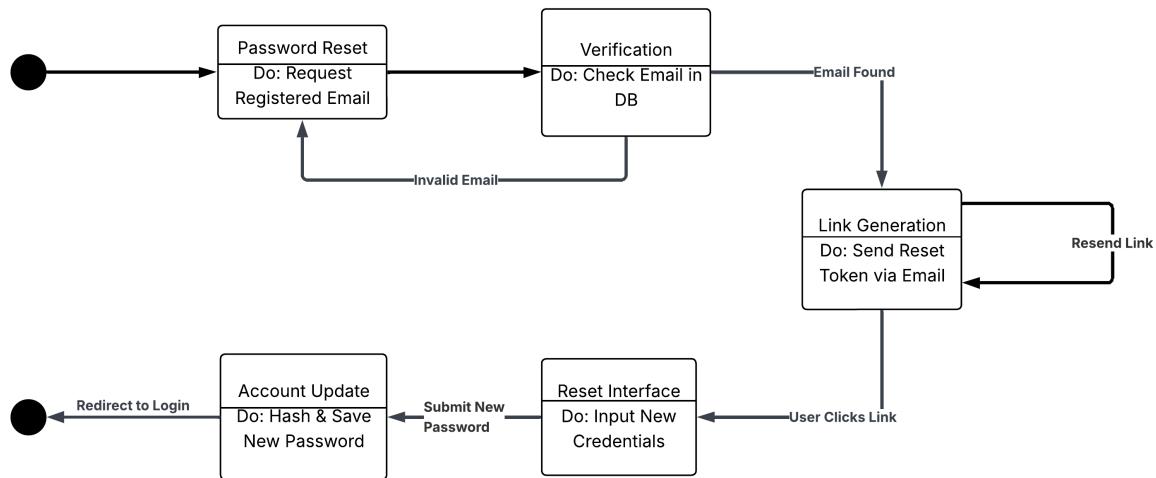


3.4 State Diagrams

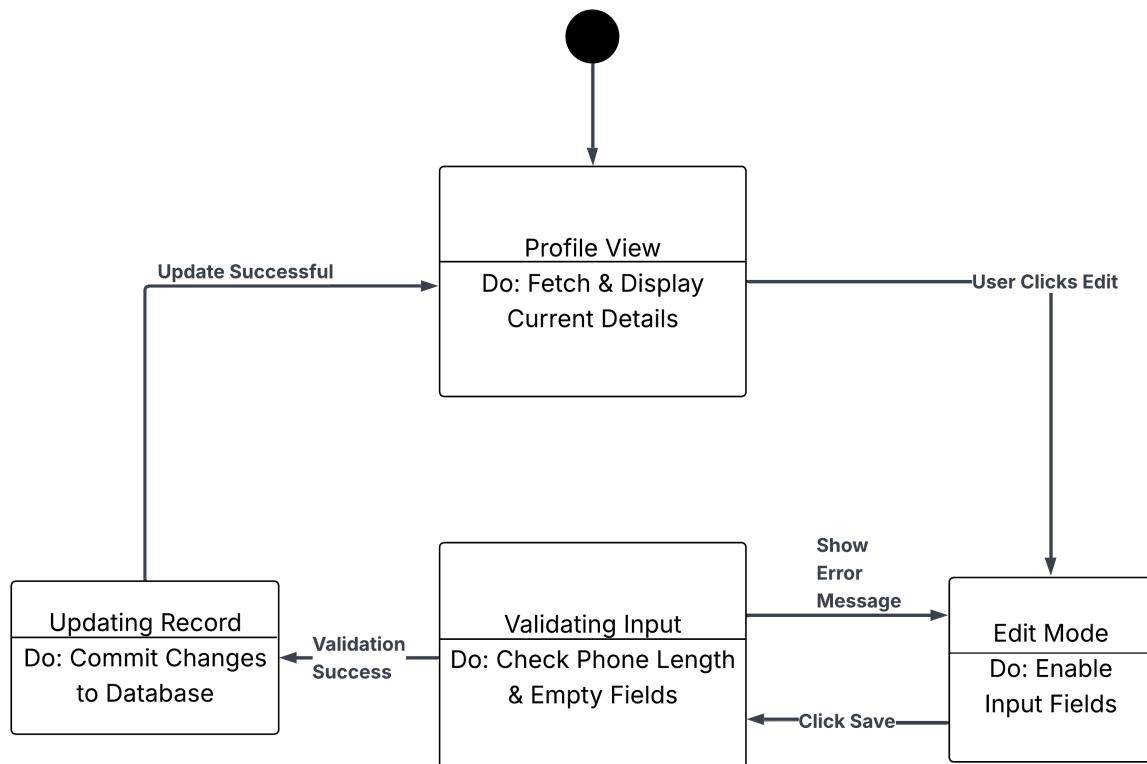
3.4.1 LOGIN, SIGNUP AND HOMEPAGE :



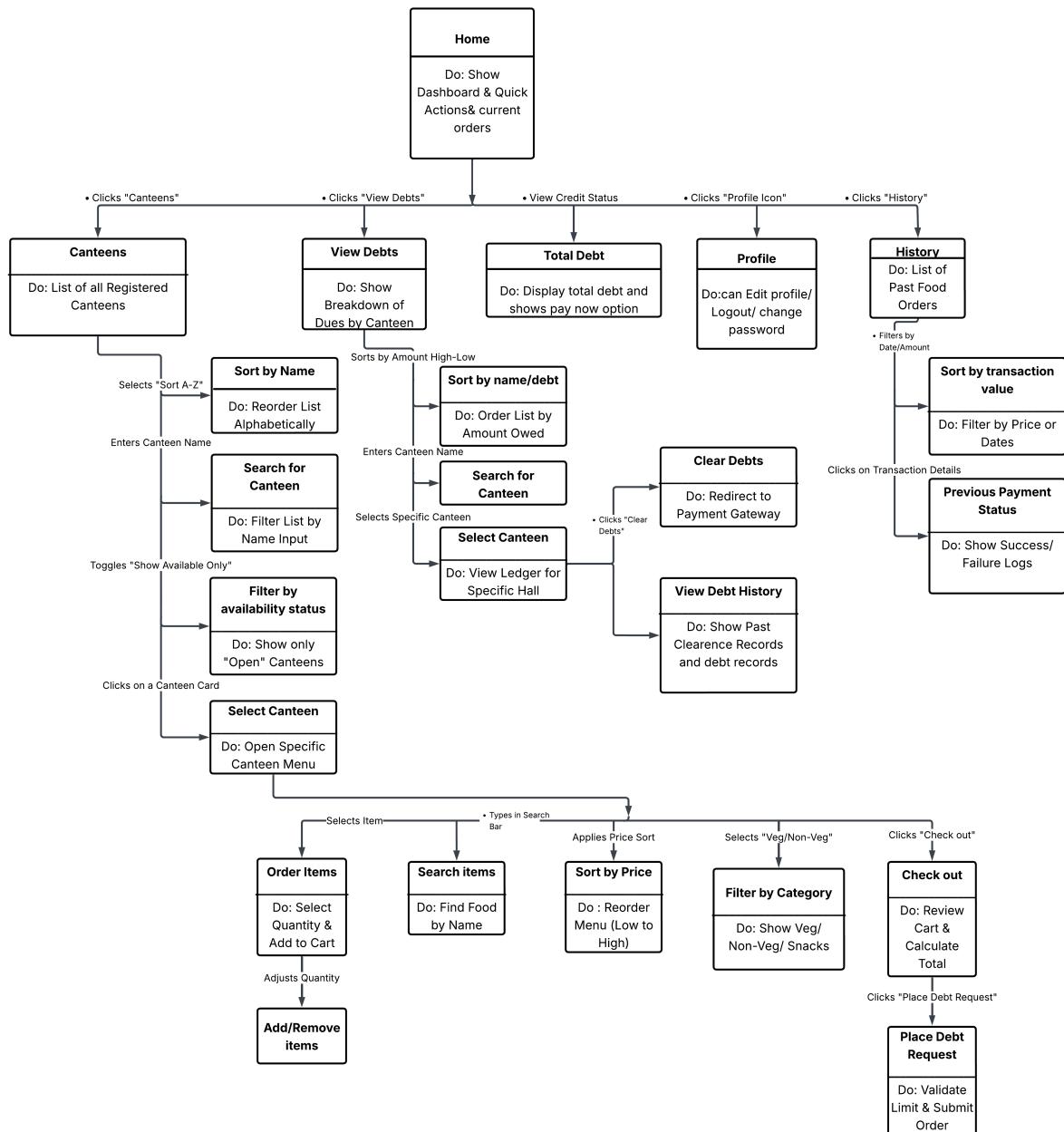
3.4.2 FORGOT PASSWORD :



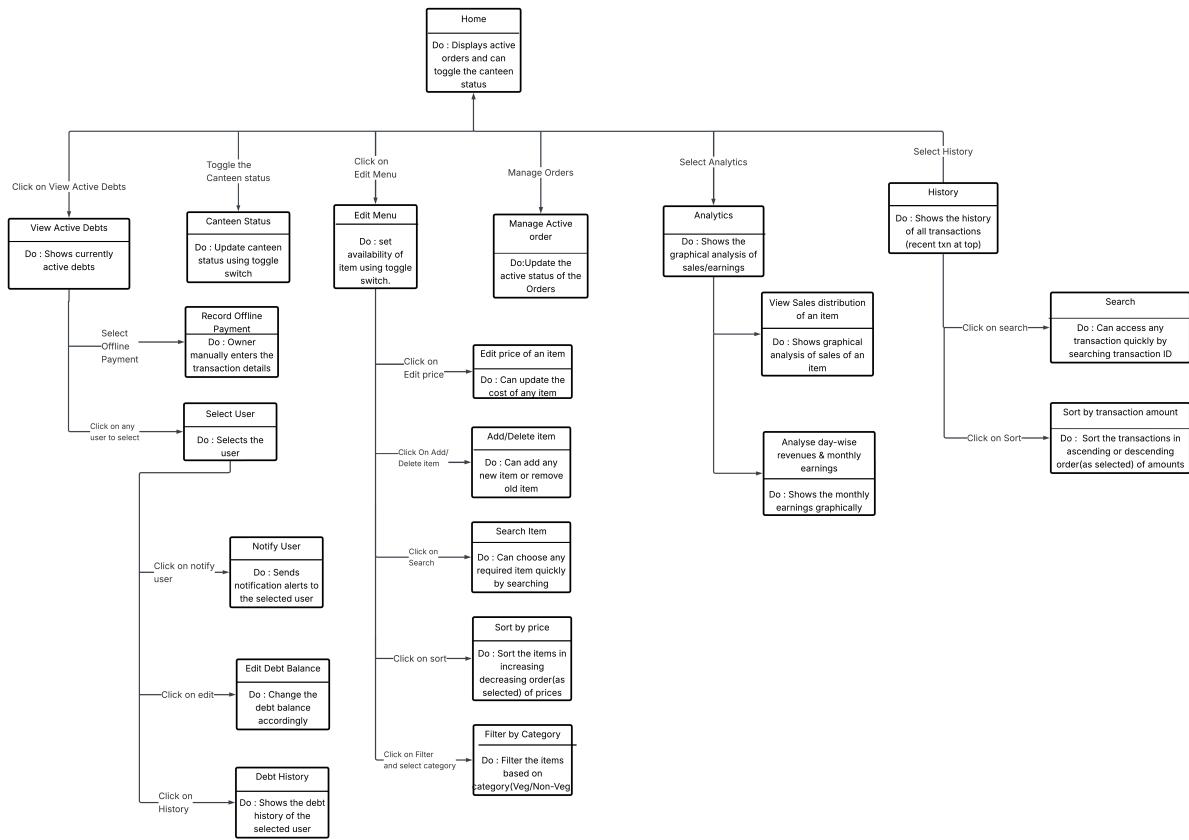
3.4.3 EDIT PROFILE :



3.4.4 STUDENT HOMEPAGE:

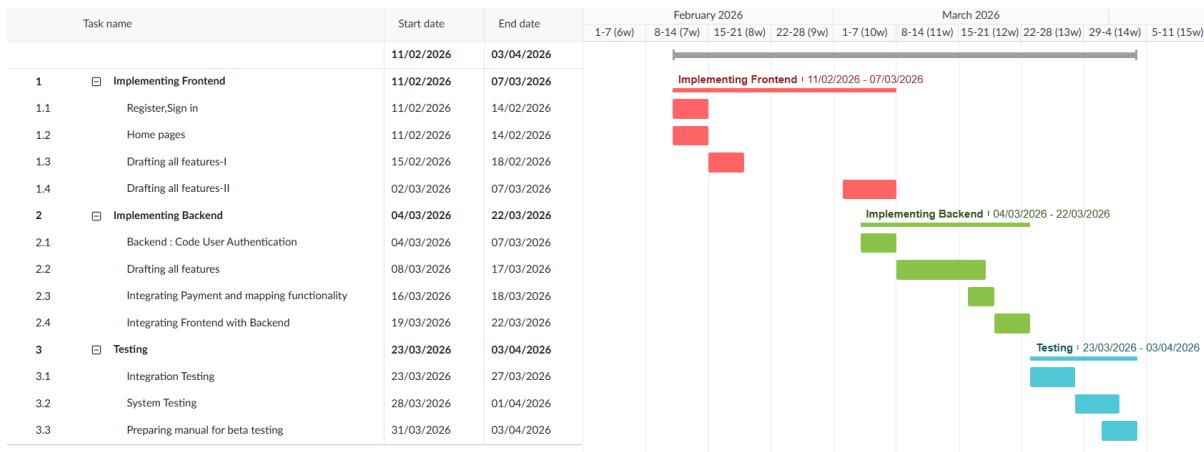


3.4.5 CANTEEN OWNER HOMEPAGE:



4 Project Plan

The following Gantt chart covers our project plan for implementation and testing of the software



The following is the current work division that the team intends to follow:

START DATE	END DATE	TASK DESCRIPTION	TEAM MEMBERS
11-02-25	14-02-25	Frontend:Sign in, Register	Yash raj,Ram charan
11-02-25	14-02-25	Frontend: Home Pages	Tejas,Chaitanya,Ashwin
15-02-25	18-02-25	Frontend: Drafting all features-I	Shreyas,Harshaa,Haneesh,Sathish,Jayavardhan,Yash raj
02-03-25	07-03-25	Frontend: Drafting all features-II	Ramcharan,Tejas,Chaitanya,Ashwin,Sathish
04-03-25	07-03-25	Backend: Code User Authentication	Yash raj, Harshaa, Haneesh
08-03-25	17-03-25	Backend: Drafting features	Full Team
16-03-25	18-03-25	Backend :Integrating payments and mapping functionalities	Tejas,Chaitanya,Shreyas, Ram charan,Jayavardhan
19-03-25	22-03-25	Integrating Frontend with Backend	Full Team
23-03-25	27-03-25	Integration Testing	Full Team
28-03-26	01-04-26	System Testing	Full Team
31-03-26	03-04-26	Preparing manual for beta testing	Full Team

Appendix A - Group Log

The following is the group log for the development of this SDD (Software Design Document).

Sl. No.	Date	Timings	Venue	Description
1	26/01/2026	20:00-22:00	KD-Library	Discussed design document plan and split into groups to handle different aspects of the project
2	30/01/2026	18:00-20:00	RM building	Discussed and finalized Architecture design and some UI diagrams
3	01/02/2026	14:00-16:00	RM building	Had discussions regarding class and state diagrams and finalised them.
4	03/02/2026	21:30-23:00	RM building	Discussed about various functionalities of the application in detail and prepared sequence diagrams
5	04/02/2026	18:30-20:00	RM building	Progressed with other design documentation work.
6	05/02/2026	21:00-22:30	RM building	Had a discussion regarding the future project plan and prepared a Gantt chart.
7	06/02/2026	14:00-17:00	RM Building	We had a final verification and submitted the document.