
Design Document

for

KnowYourCibo

Version 1.0

Prepared by

Group #: 6

| | |
|----------------------|--------|
| K S U Rithwin | 220537 |
| R V S Havish | 220879 |
| Mahanthi Vijay Kumar | 220602 |
| G N L Mahathi | 220395 |
| Rohan Aditya | 220741 |
| Rushikesh Chary | 220336 |
| Raj Vinayak Meena | 220861 |
| Laveti Bhanu Prakash | 220583 |
| Sai Dishanth | 220282 |
| Ravula Harshith Sai | 220878 |
| Gowtham Chand | 220313 |

Group Name: Code Crafters

| |
|--|
| ksupendra22@iitk.ac.in |
| rviswas22@iitk.ac.in |
| mvijay22@iitk.ac.in |
| gnagal22@iitk.ac.in |
| praditya22@iitk.ac.in |
| drushikesh22@iitk.ac.in |
| rajvm22@iitk.ac.in |
| ibhanu22@iitk.ac.in |
| banoths22@iitk.ac.in |
| ravula22@iitk.ac.in |
| cgchand22@iitk.ac.in |

Contents

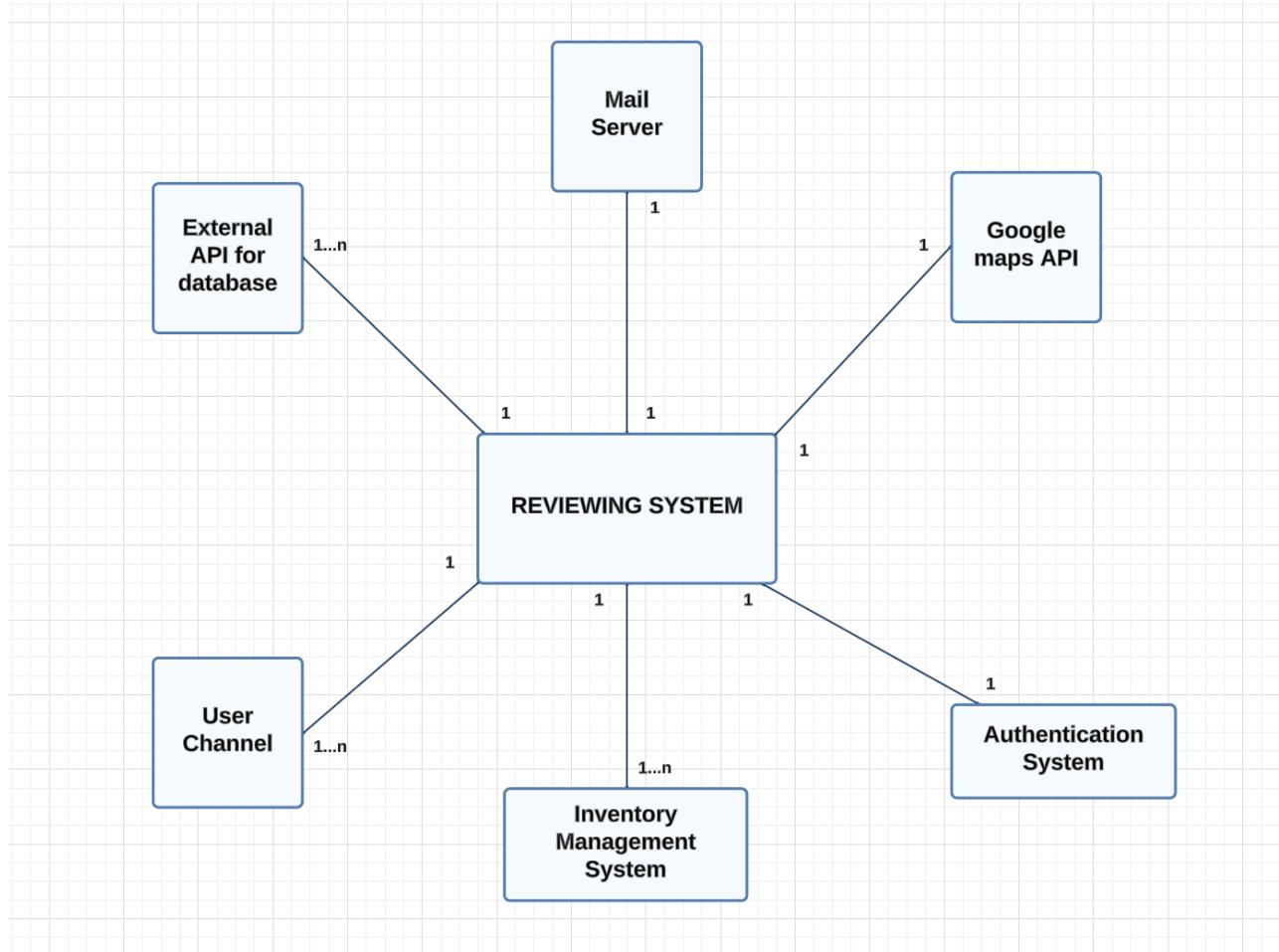
| | |
|--------------------------------------|-----------|
| CONTENTS..... | II |
| REVISIONS..... | II |
| 1 CONTEXT DESIGN..... | 4 |
| 1.1 CONTEXT MODEL..... | 4 |
| 1.2 HUMAN INTERFACE DESIGN..... | 5 |
| 2 ARCHITECTURE DESIGN..... | 12 |
| 3 OBJECT-ORIENTED DESIGN..... | 14 |
| 3.1 USE CASE DIAGRAM..... | 14 |
| 3.2 CLASS DIAGRAM..... | 18 |
| 3.3 SEQUENCE DIAGRAM | 20 |
| 3.4 STATE DIAGRAM | 28 |
| 4 PROJECT PLAN..... | 31 |
| 5 OTHER DETAILS..... | 34 |
| APPENDIX A - GROUP LOG..... | 35 |

Revisions

| Version | Primary Author(s) | Description of Version | Date Completed |
|---------|-----------------------|------------------------------|----------------|
| v1.0 | Group - Code Crafters | Initial commit (First Draft) | 09/02/24 |

1 Context Design

1.1 Context Model



KNOW YOUR CIBO SYSTEM: This is the central system being described. It's likely a software application or platform focused on providing services or information related to "CIBO," which might stand for a specific concept or company name.

External API for database: This is an external database API that the system interacts with. It suggests that the system retrieves or sends data to an external database that is not maintained within the system itself.

Mail Server: The system has a connection to a mail server, indicating that it can send or receive emails, possibly for notifications, alerts, or communications with users.

Google Maps API: The system uses the Google Maps API, which could imply that it has location-based services or functionalities, like mapping, location tracking, or providing directions.

Authentication System: This external system handles the authentication of users. It ensures that only authorized users can access the system, likely through login credentials or tokens.

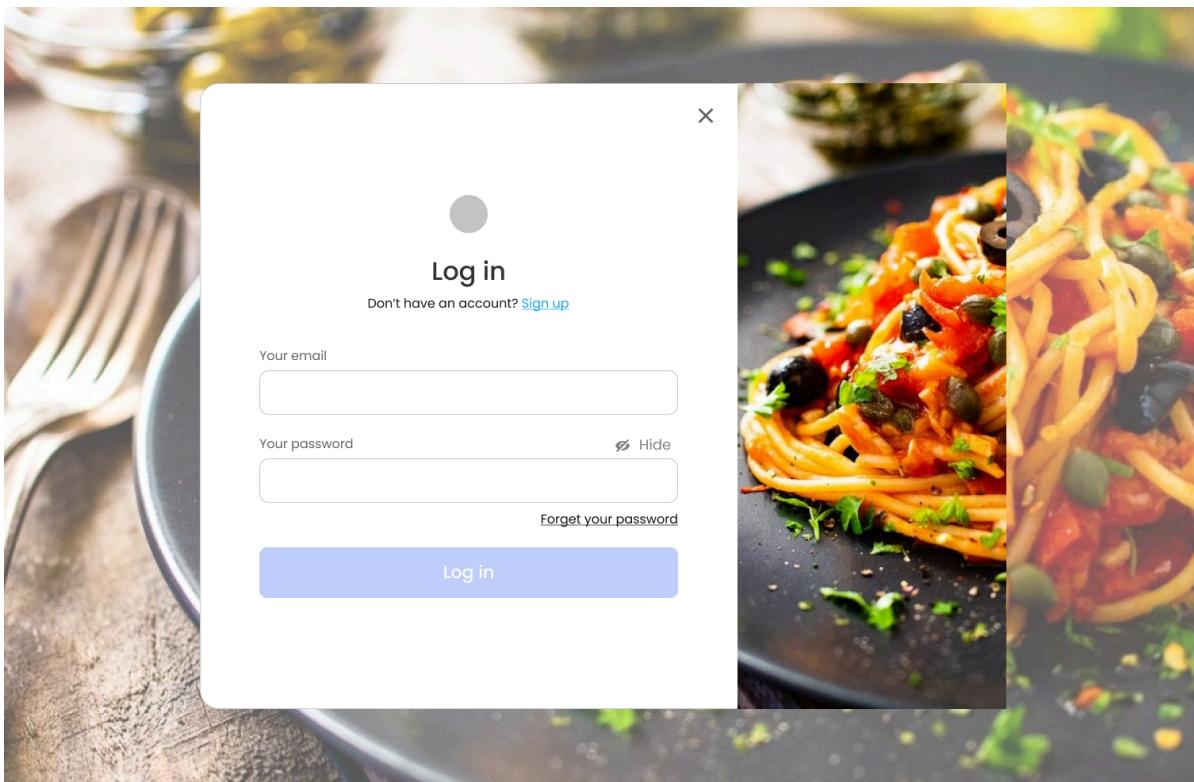
Inventory Management System: An inventory management system with a review feature focuses solely on organizing and managing the reviews database for each food item. This system allows businesses to efficiently track and analyze customer feedback, facilitating informed decisions about product quality, marketing strategies, and potential improvements.

User Channel: This represents the different ways users can interact with the system. It may include web interfaces, mobile apps, or other forms of user interaction.

The **system** is at the center, which communicates bidirectionally with the User Channel, the Inventory Management System, the Authentication System, and sends or retrieves information from the external database API, the Mail Server, and the Google Maps API. This context diagram provides a high-level overview of the system's interactions with its environment without detailing the internal workings of the system itself.

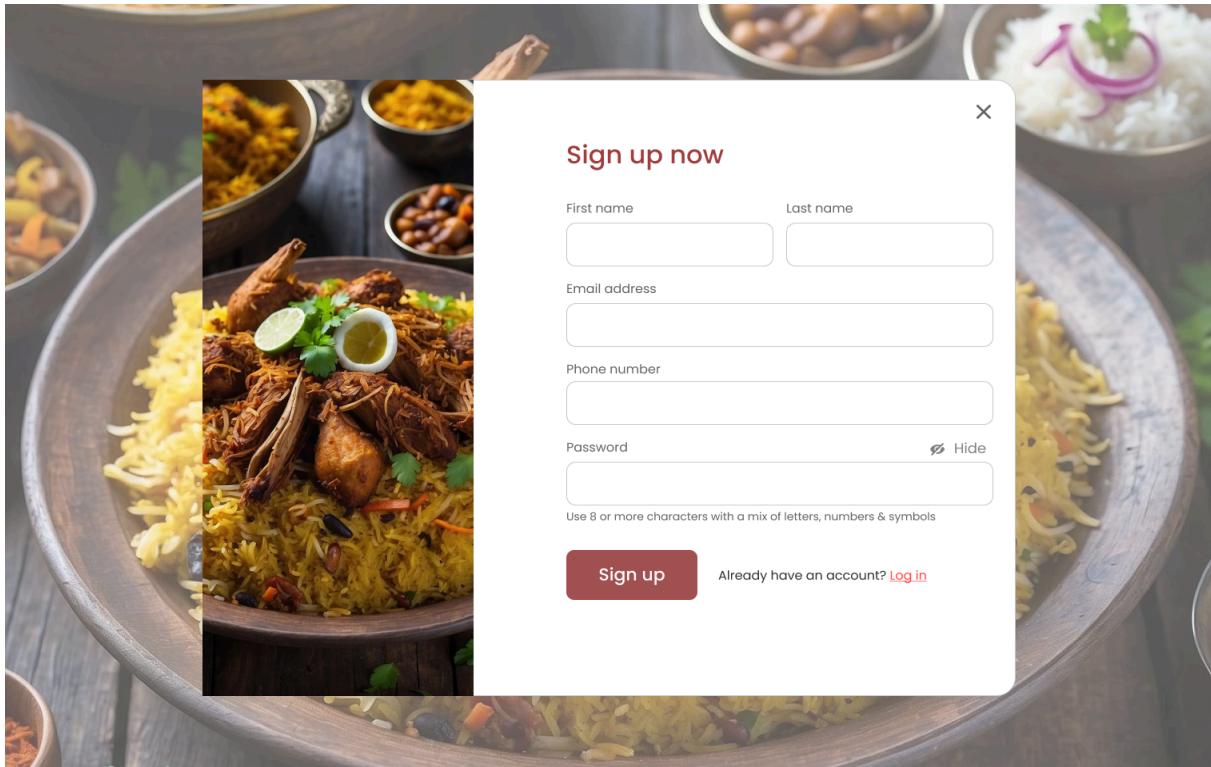
1.2 Human Interface Design

1. Login Page:



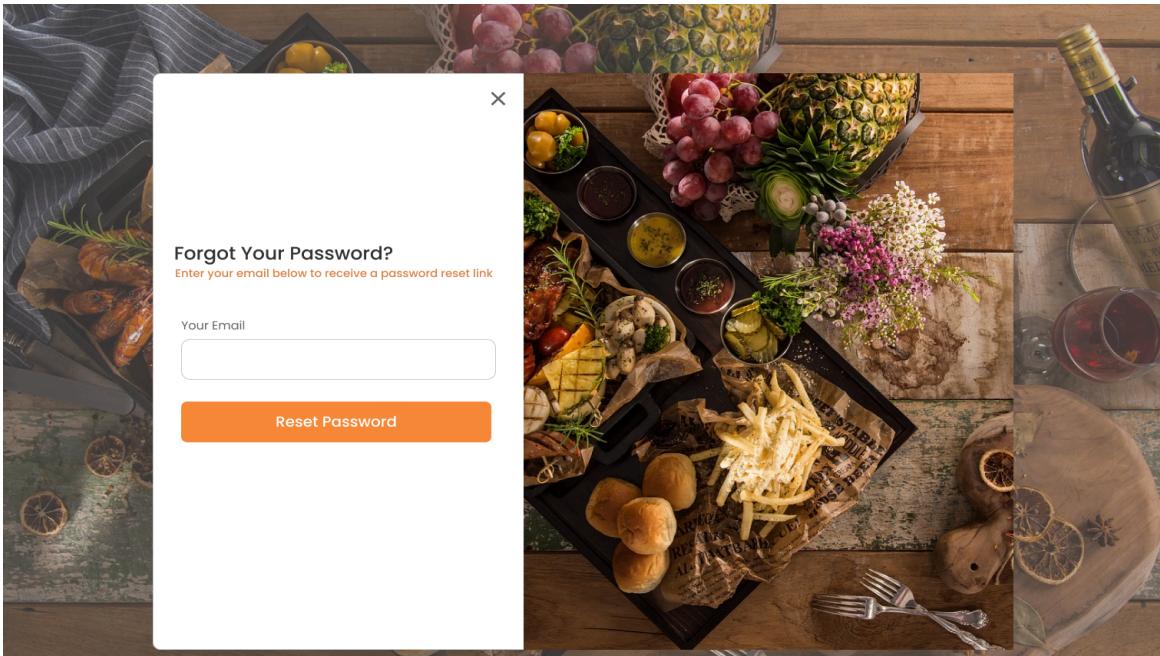
The "Log in" page is a critical component of "Know Your Cibo," providing a secure entry point for returning users. It prompts for email and password credentials, includes a password visibility toggle for user convenience, and offers a link for users who need to recover forgotten passwords. Additionally, it provides an option for new users to navigate to the "Sign Up" page, ensuring seamless navigation between user authentication states.

2. SignUp Page:



The "Sign Up" page of "Know Your Cibo" serves as the entry point for new users to create an account, essential for a personalized experience on the platform. It prompts the user to enter personal information, such as first and last name, email address, phone number, and a secure password. The form enforces strong password creation for security and provides a direct link to the login page for existing users, streamlining the process of joining and navigating the food reviewing community.

3. Forgot Password:



The "Forgot Your Password?" page on "Know Your Cibo" provides users with a secure method to reset their password. Upon entering their registered email address, users will receive a password reset link. This feature ensures that users can regain access to their accounts with minimal disruption, maintaining the security and integrity of user access within the system.

4. View and Edit Profile:

KnowYourCibo

Home Restaurants **Profile** Feedback Contact us

Mahathi
gnagal22@iitk.ac.in

22 Favorites 67 Ratings 5 Reviews

Edit profile

FAVORITE PICKS

| Pizza Name | Rating | Reviews |
|-----------------|--------|---------|
| Farmhouse | 3.8 | 459 |
| Peppy Pancer | 3.8 | 459 |
| Veggie Paradise | 3.8 | 539 |

KnowYourCibo

Home Restaurants **Profile** Feedback Contact us

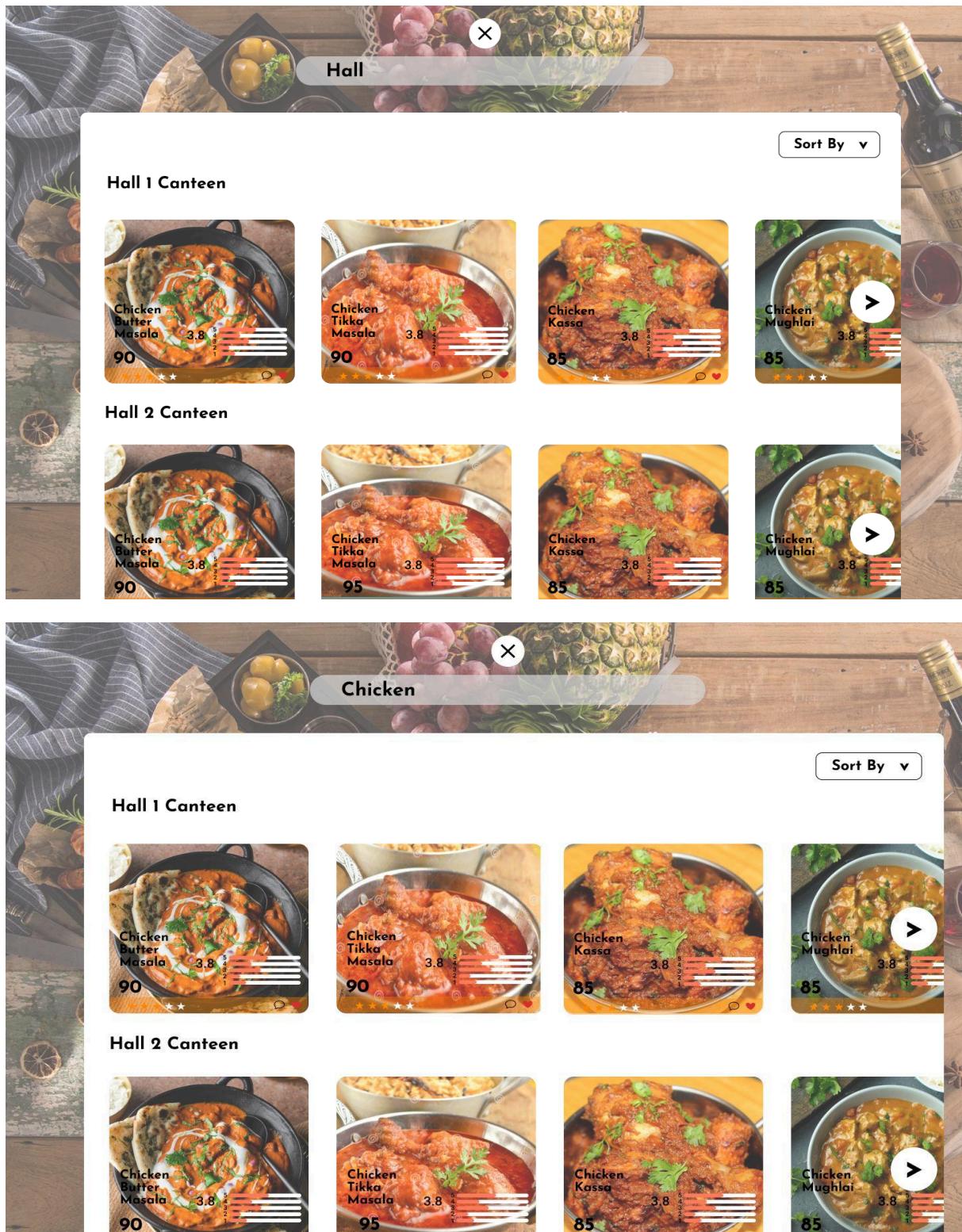
Edit Profile

First name Last name
 Email address
 Phone number
 Password Hide
Use 8 or more characters with a mix of letters, numbers & symbols

Update

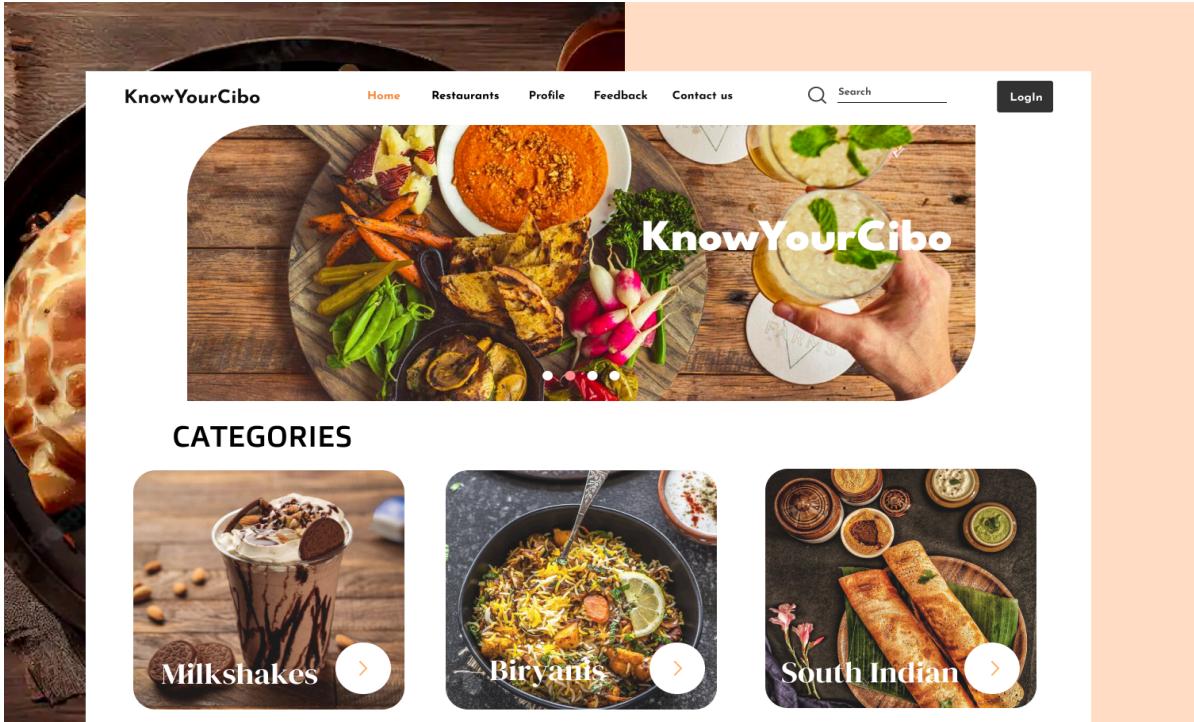
The 'Edit Profile' and 'User Profile' pages are interconnected facets of the "Know Your Cibo" user interface. The 'Edit Profile' page allows users to update their personal details such as name, email, phone number, and password to maintain their account's accuracy and security. The 'User Profile' page displays the user's chosen favorites, ratings, and reviews, enhancing the personalized experience by allowing quick access to preferred items and a summary of past interactions on the platform.

5. Functionality of search:



The pages feature a categorized display of dishes from different restaurants, providing users with a visual menu to browse. Each dish is accompanied by an image, name, price, and a rating system, allowing for an intuitive and engaging user experience. The pages are designed to facilitate easy comparison and selection of dishes based on user preference, with functionality to sort the items and navigate for more details or to place orders directly from the platform.

6. Home Page:



This page serves as the homepage for "**Know Your Cibo**," a food discovery platform. It features a navigation bar for easy access to various sections like Home, Restaurants, Profile, Feedback, and Contact Us. A search bar is prominently displayed for quick searches. The main focus of the page is the "Categories" section, showcasing different food categories like Milkshakes, Biryani, and South Indian cuisine, each with a visually appealing image and a call-to-action button, inviting users to explore the offerings within each category. The design aims to engage users with an attractive and intuitive layout, encouraging exploration and interaction with the site's content.

7. Restaurant Page, Ratings and Reviews:

Pizza, Italian, Pasta, Fast Food, Desserts
OAT, IIT Kanpur
2pm - 12am

VEG PIZZA

- Margherita
- Farmhouse
- Peppy Paneer
- Veggie Paradise

Pizza, Italian, Pasta, Fast Food, Desserts
OAT, IIT Kanpur
2pm - 12am

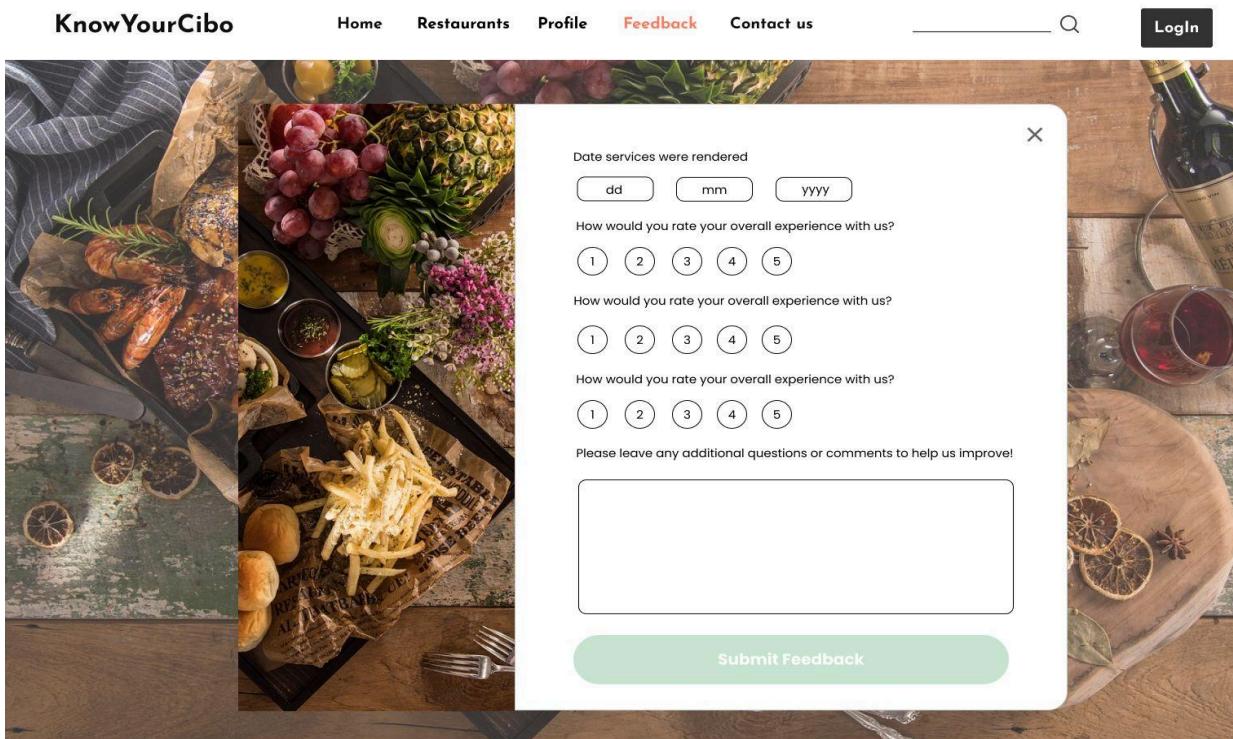
VEG PIZZA

- Margherita
- Farmhouse
- Peppy Paneer

Write a Review

The pages from the "KnowYourCibo" website offer a user-centric interface for engaging with the food community at IIT Kanpur. The restaurant page provides a visual and interactive menu of dishes available at specific restaurants, complete with images, prices, and user ratings, allowing for informed dining decisions.

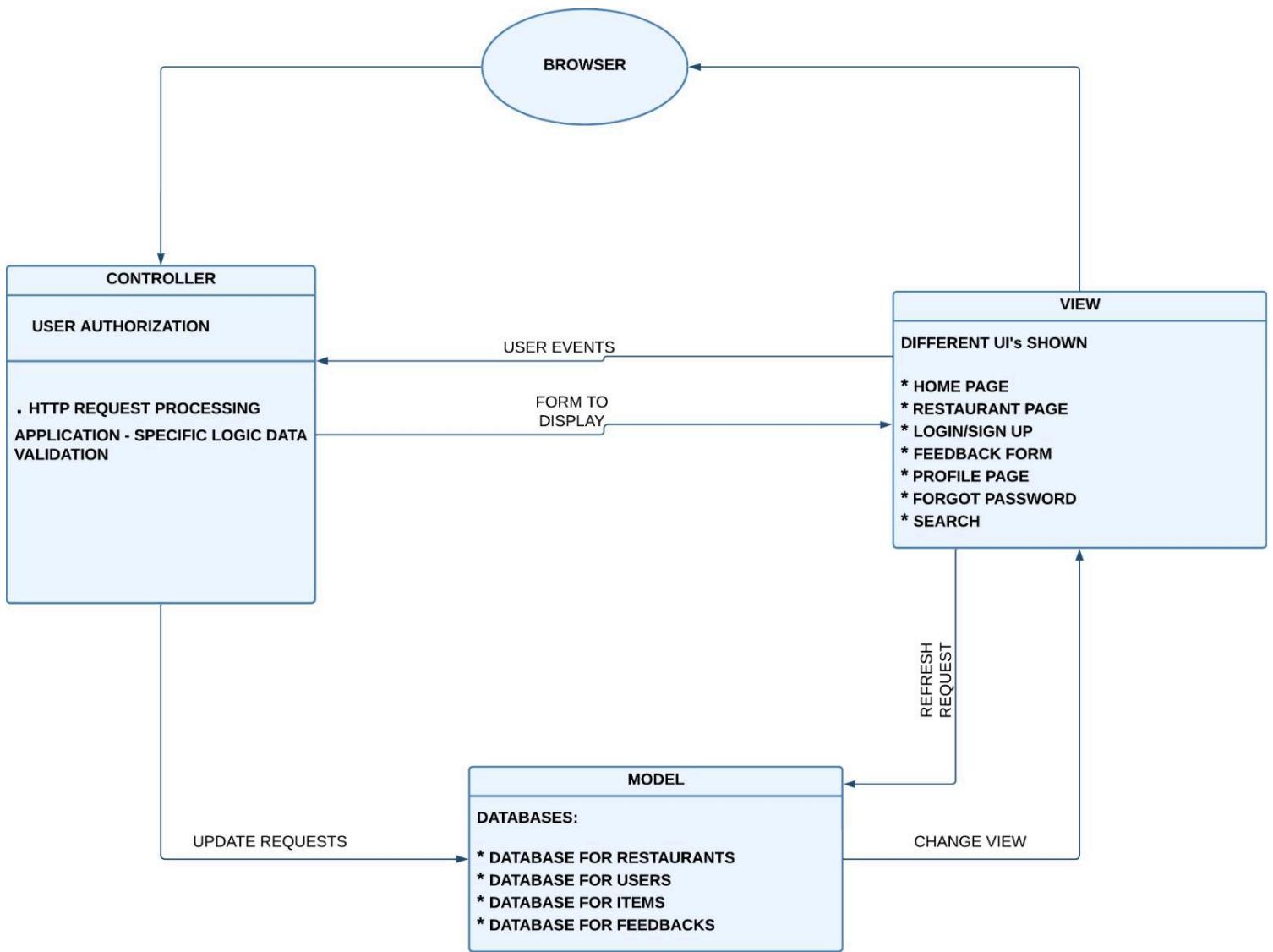
8. Feedback form:



The "Feedback" page on "Know Your Cibo" is designed for users to rate their dining experiences and provide detailed feedback. It features a date selection for service rendered, a rating scale for multiple aspects of the service, and a text box for additional comments or suggestions. This functionality allows the platform to gather actionable insights to enhance service quality and user satisfaction.

2 Architecture Design

Model-View-Controller Pattern



Many web applications use the mentioned architecture pattern. The architecture pattern of our software also corresponds to the Model-View-Controller Pattern. This is a common design pattern used in software development to separate the internal representations of information from the ways that information is presented to and accepted from the user.

Browser: This represents the user's point of interaction with the system, likely a web browser.

View: This component is responsible for displaying the user interface (UI) to the user. It includes different UI elements, such as:

- Home Page
- Restaurant Page
- Login/Sign Up
- Feedback Form
- Profile Page
- Forgot Password
- Search

The View interacts with the Controller to refresh data and change the UI as needed based on user interactions.

Controller: This is the component that processes HTTP requests, handles user authorization, and contains the application-specific logic for data validation. It also reacts to user events, updates requests, and interacts with the Model to reflect changes.

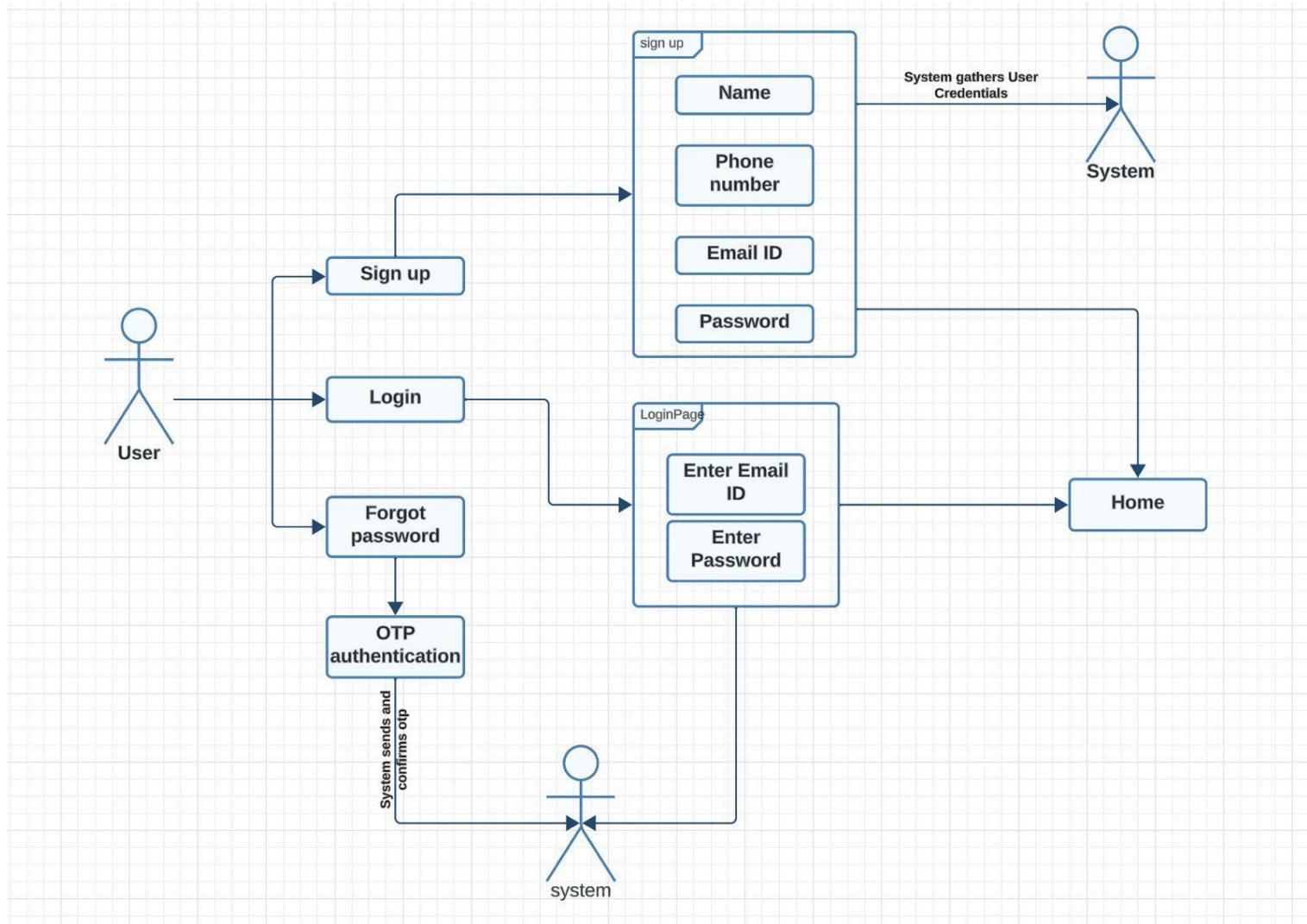
Model: The Model is responsible for managing the data, logic, and rules of the application. In the diagram, it includes databases for restaurants, users, items, and feedback. The Model receives updates from the Controller and informs the View when data changes so that the UI can be updated.

The diagram suggests a web application where the user interface changes dynamically. The Controller processes and responds to user inputs, and the Model handles data persistence and business logic.

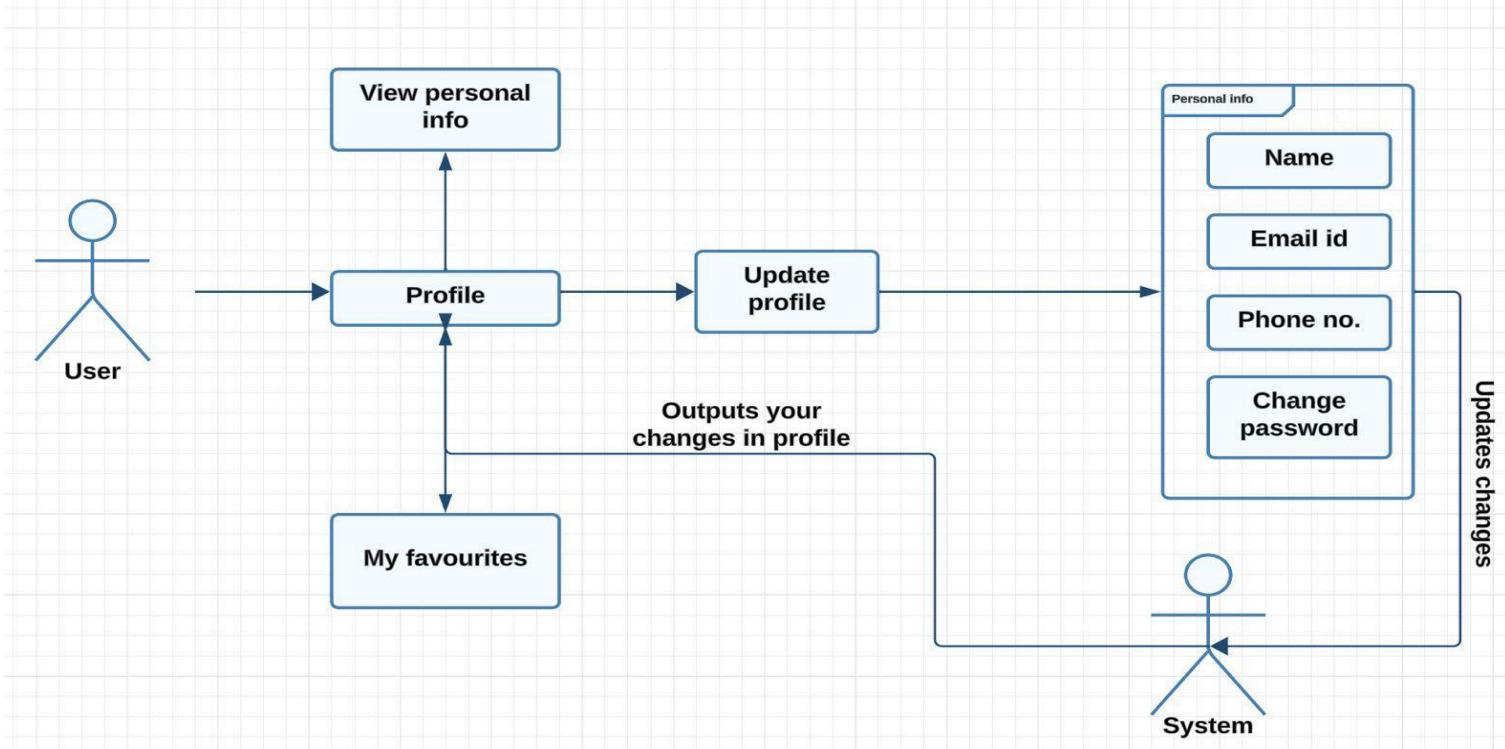
3 Object Oriented Design

3.1 Use Case Diagrams

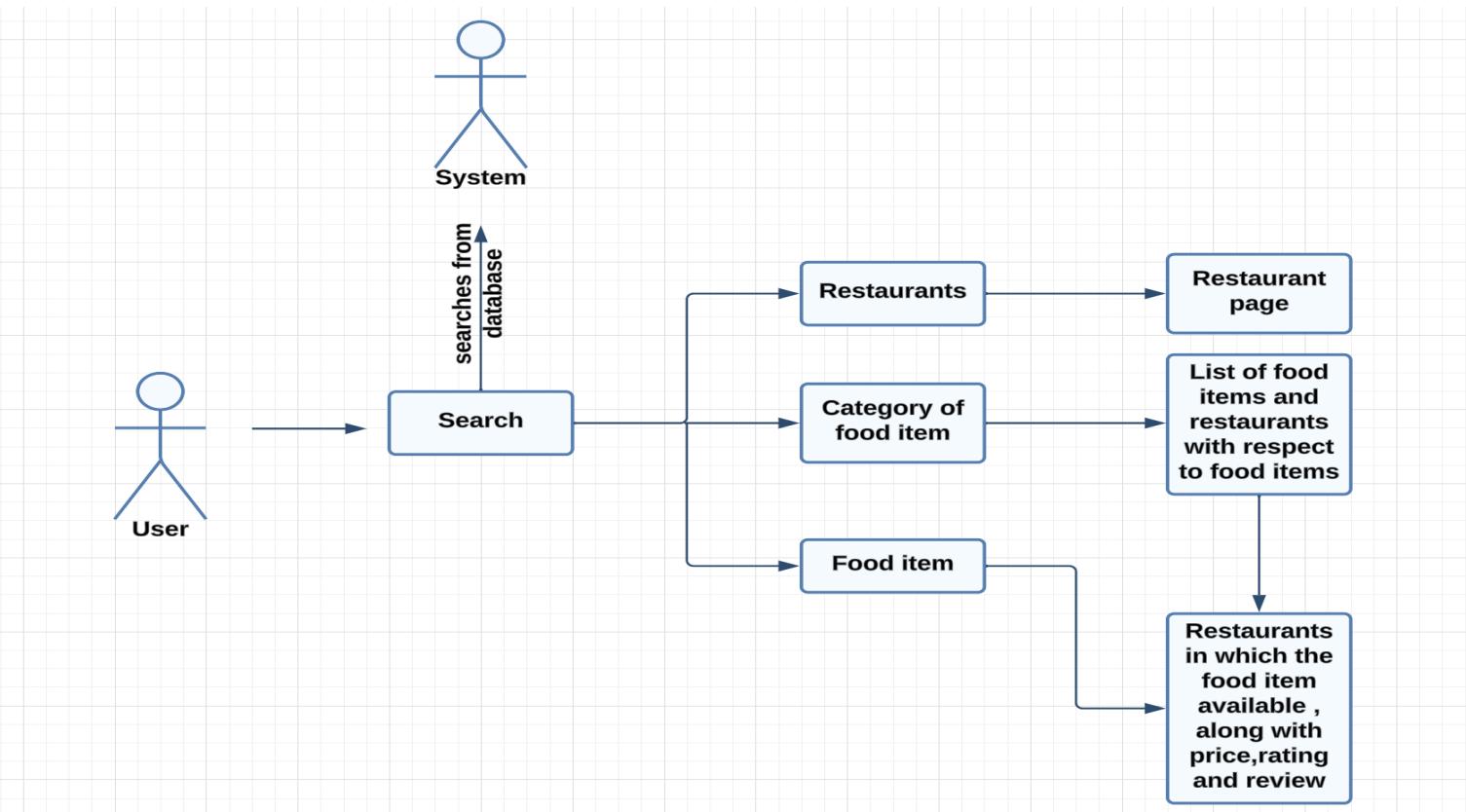
3.1.1 Use Case #1: Authentication



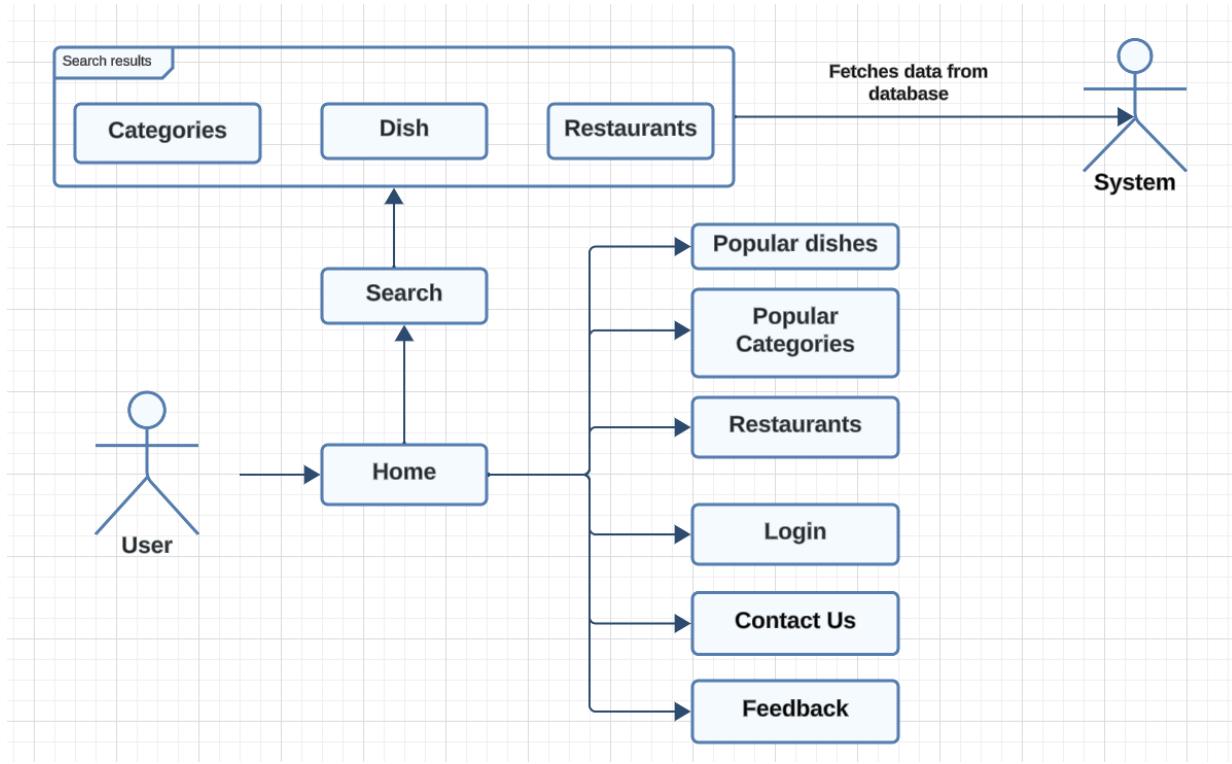
3.1.2 Use Case #2: Updating User Profile Page



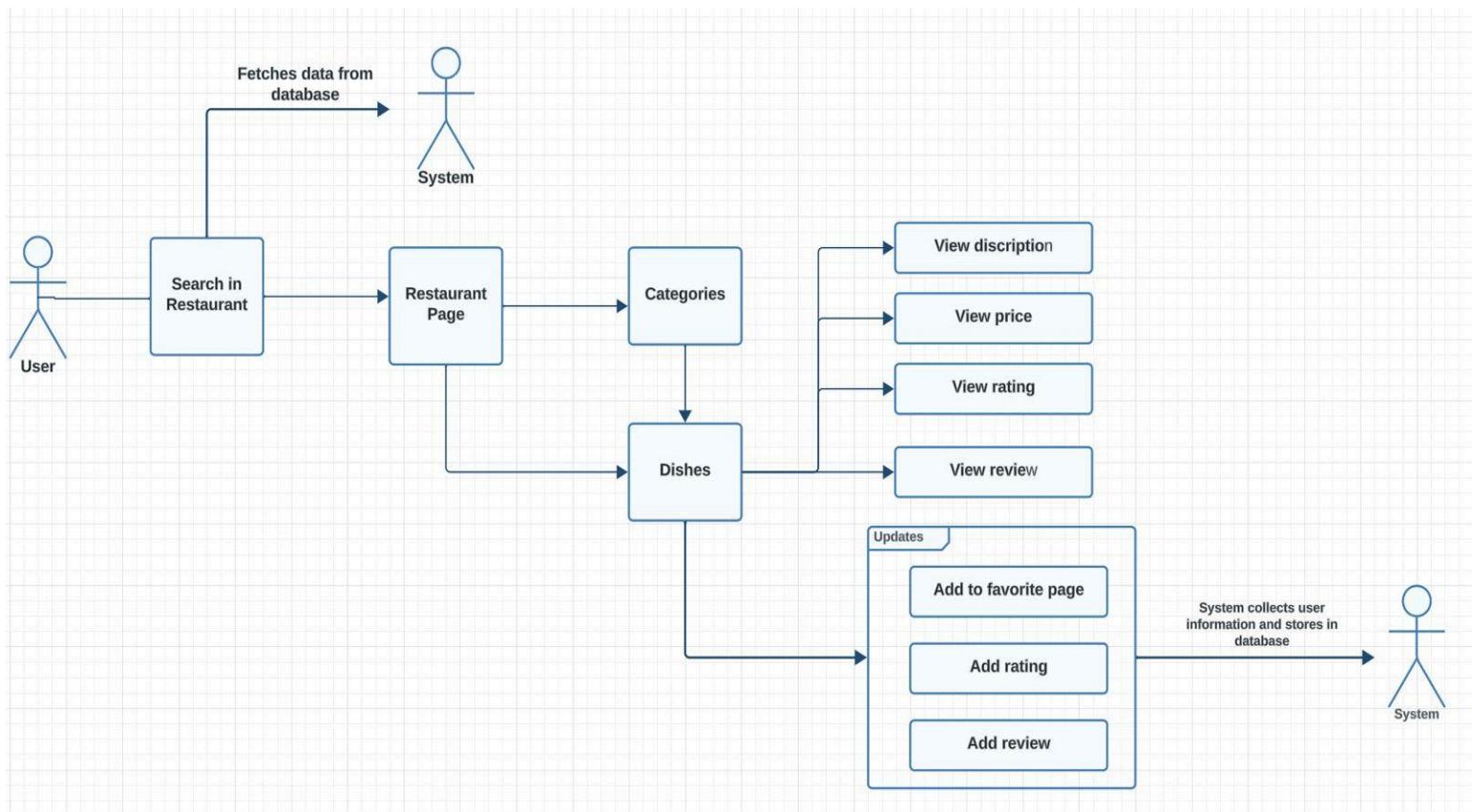
3.1.3 Use Case #3: Functionality of Search



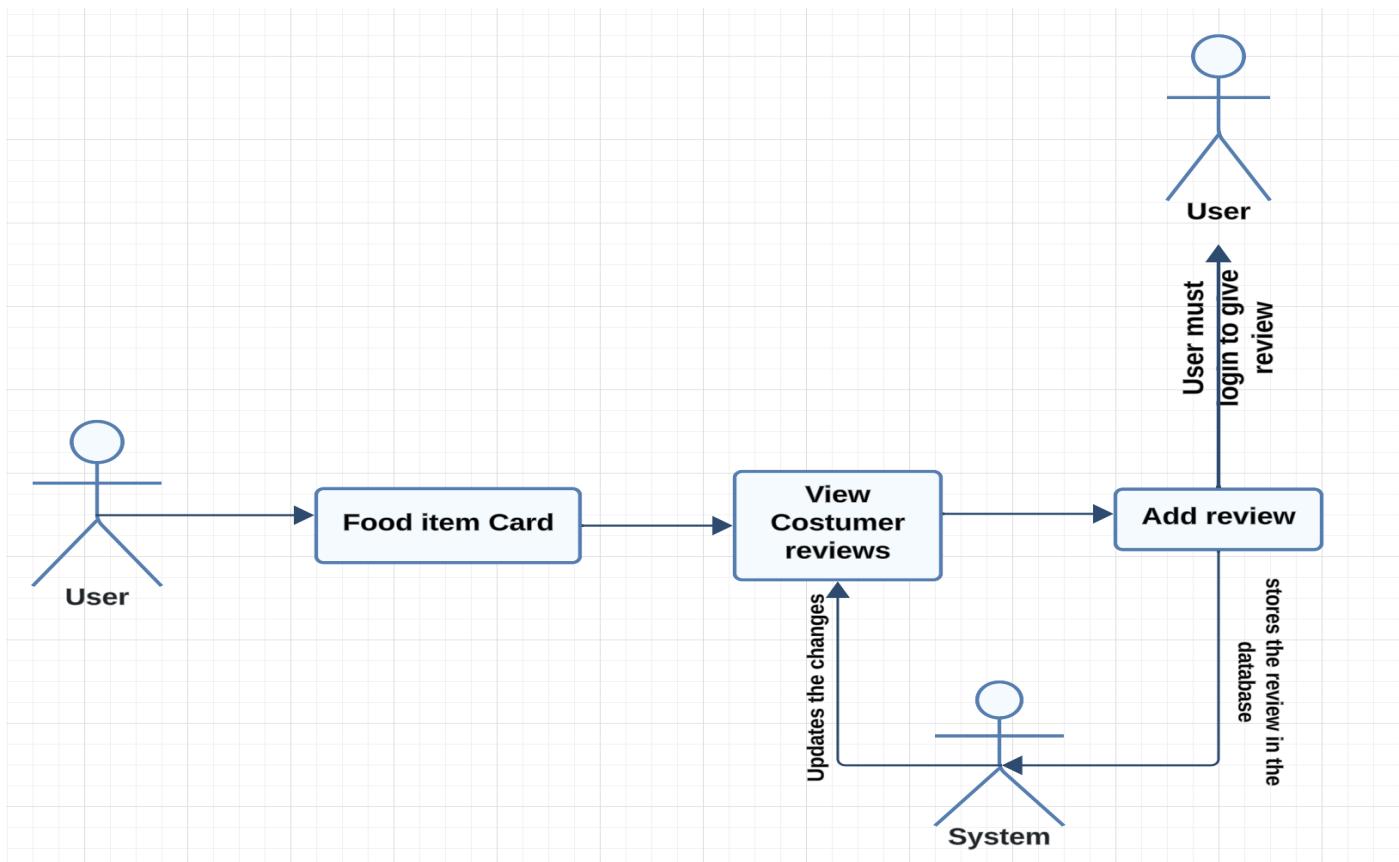
3.1.4 Use Case #4: Home Page Navigation



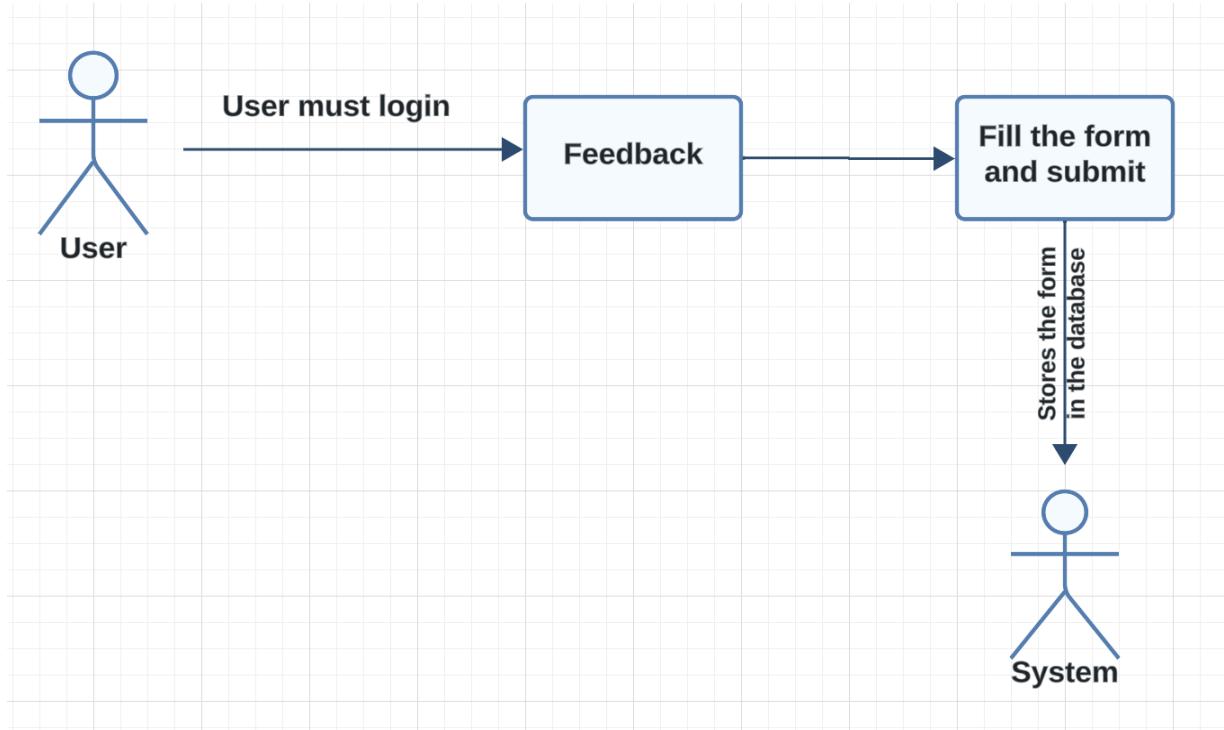
3.1.5 Use Case #5: Restaurant Page Navigation



3.1.6 Use Case #6: Giving Reviews

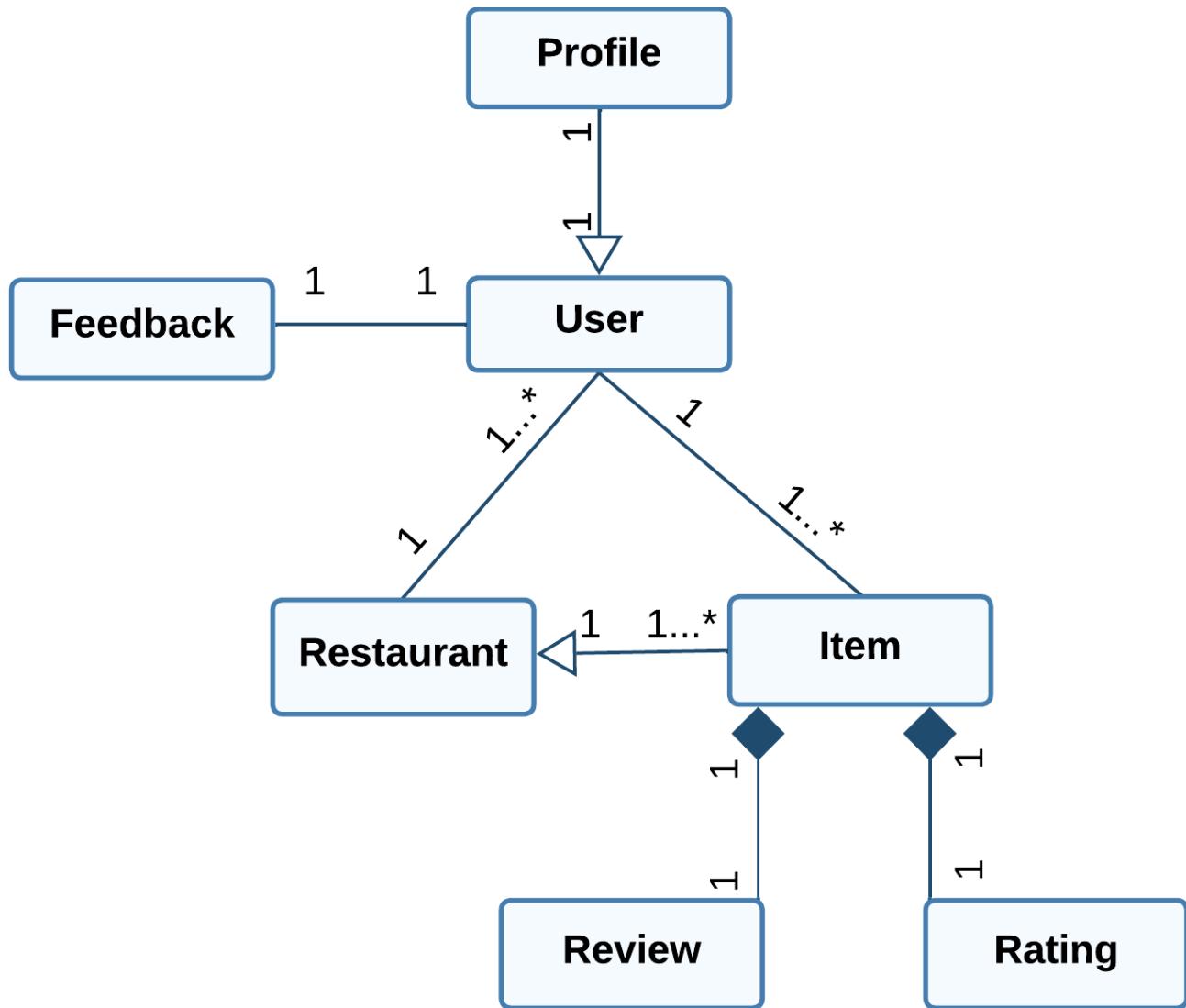


3.1.7 Use Case #7: Giving Feedback

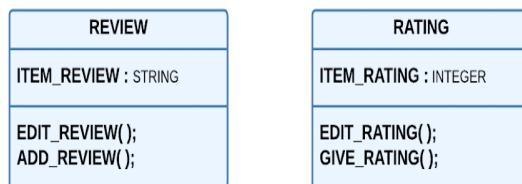
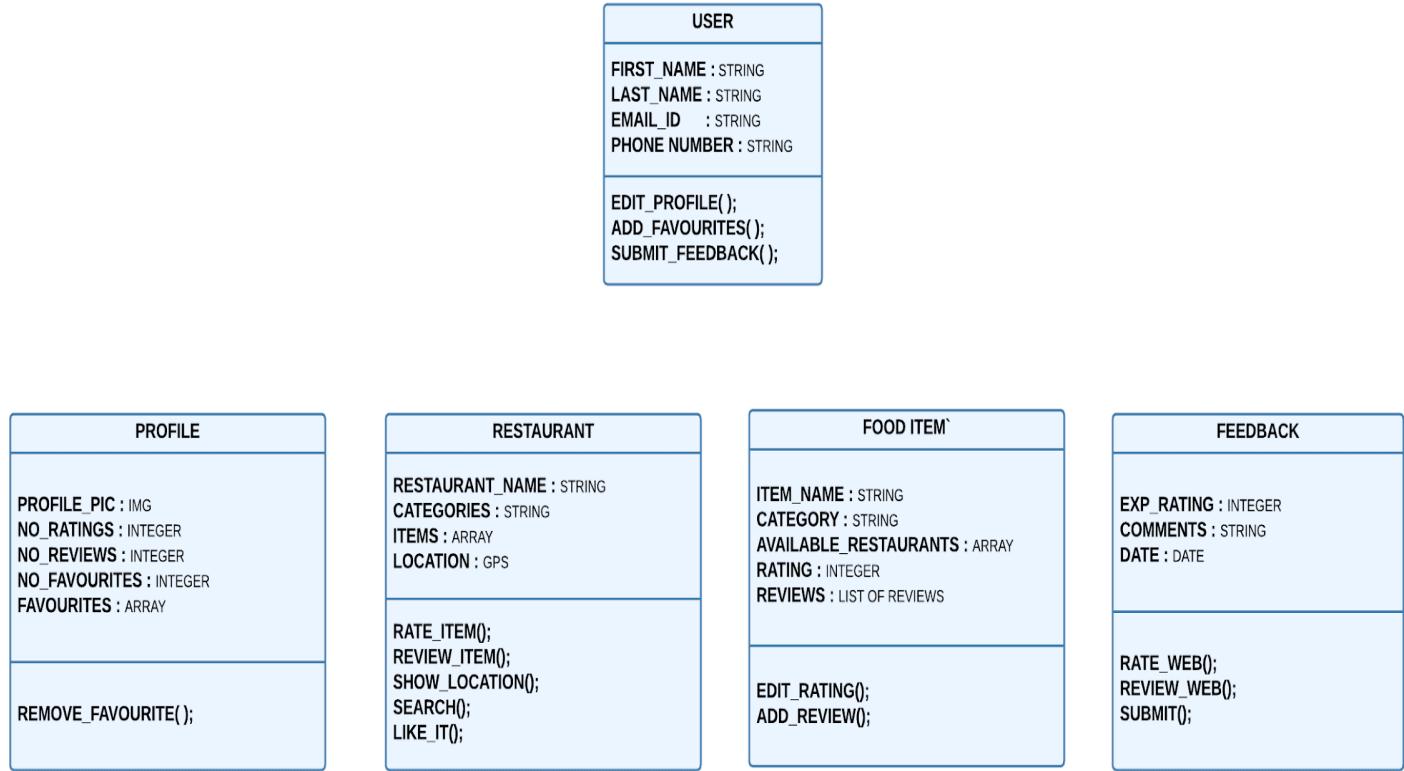


3.2 Class Diagrams

The classes to be used in the software system along with their various association is as follows:

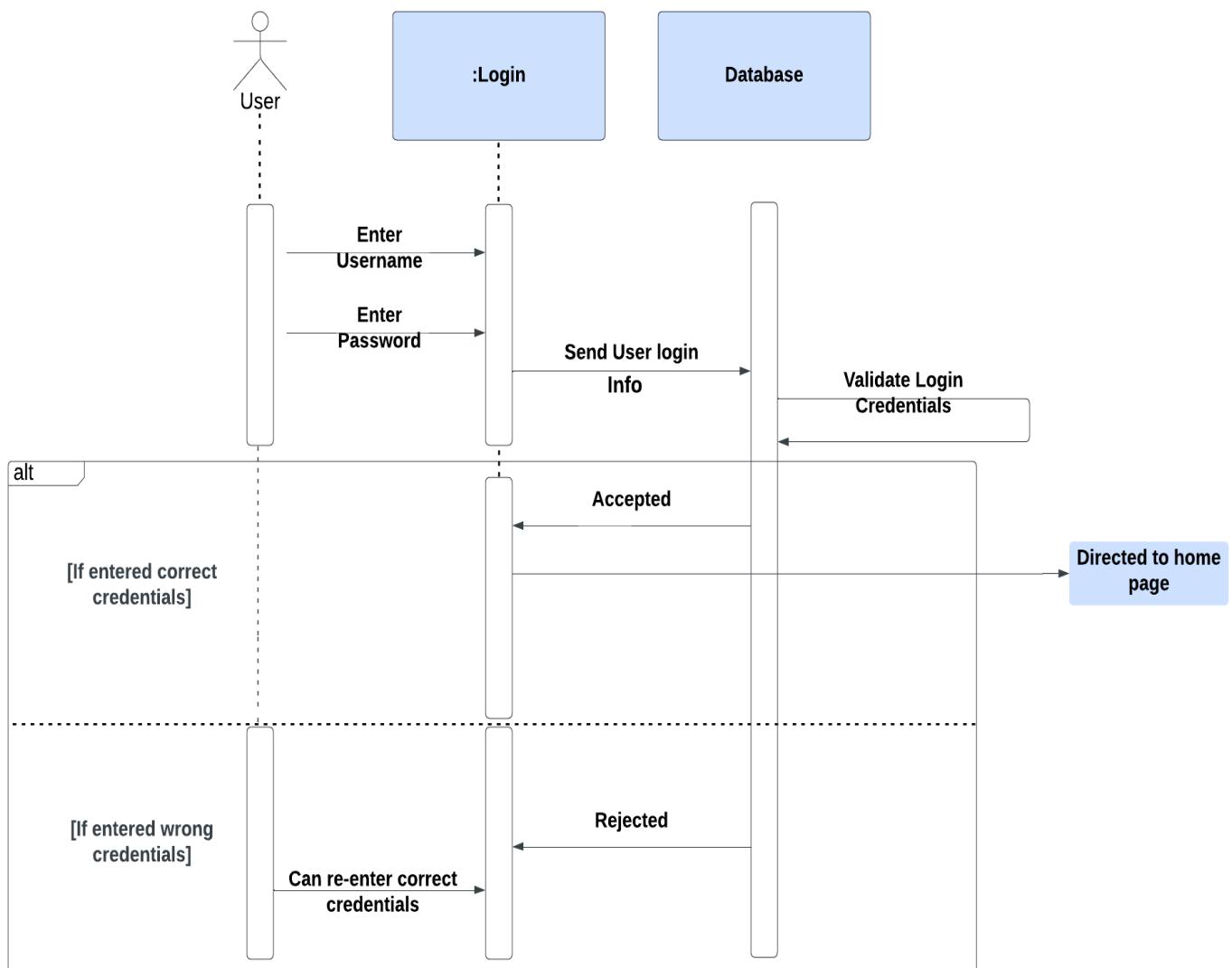


Detailed description of each class along with their attributes and functions is as follows:

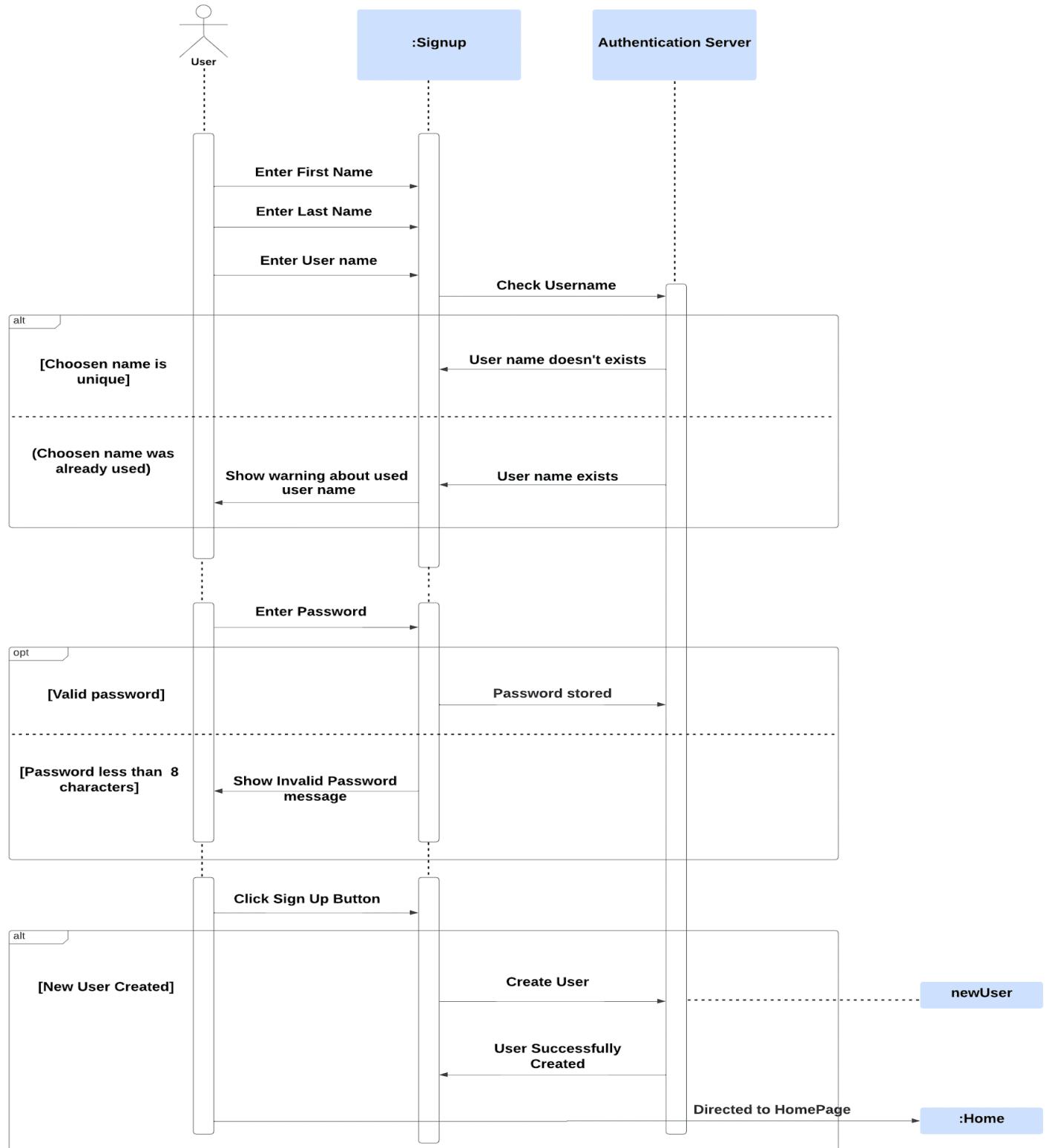


3.3 Sequence Diagrams

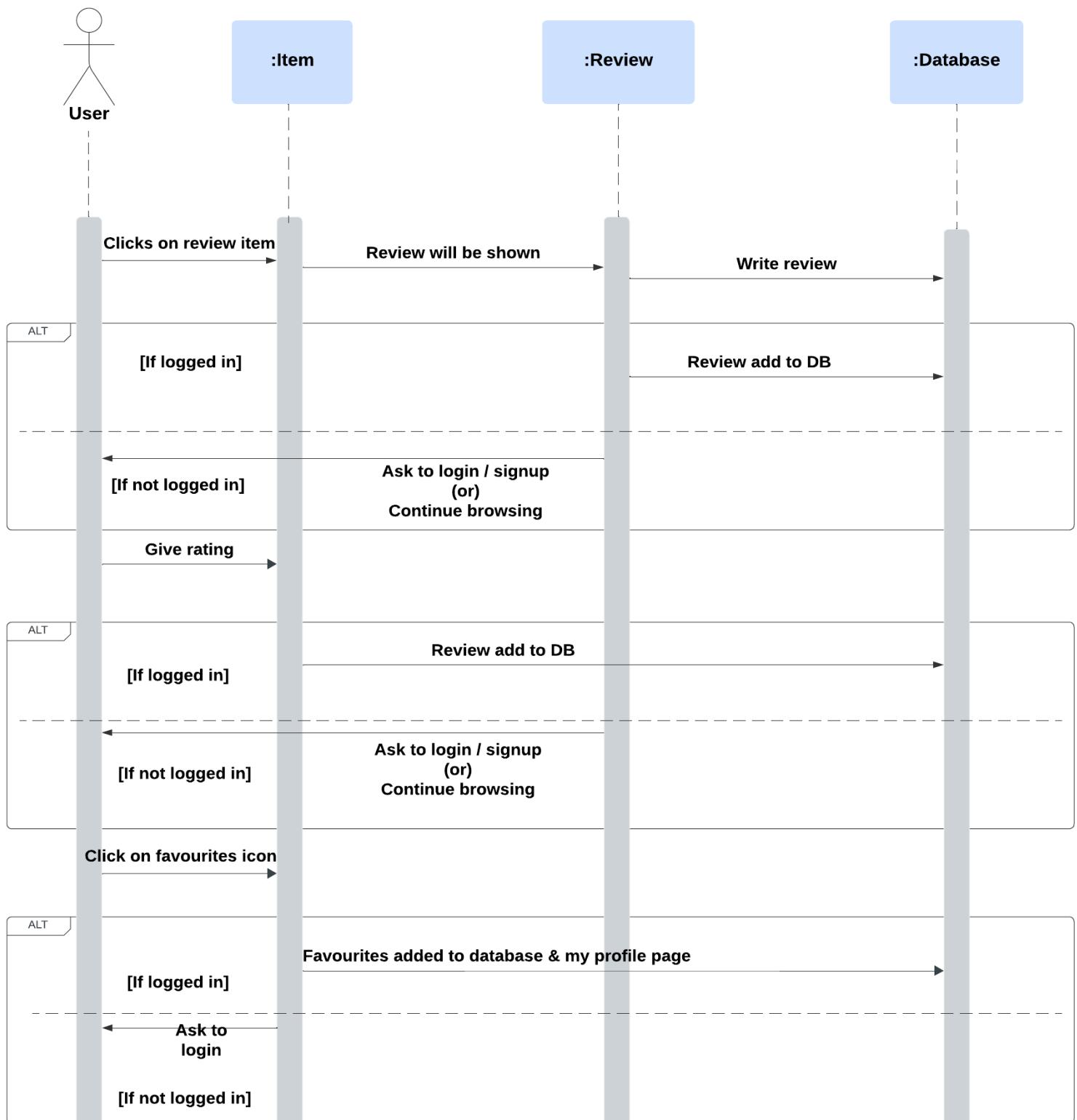
1. Login:



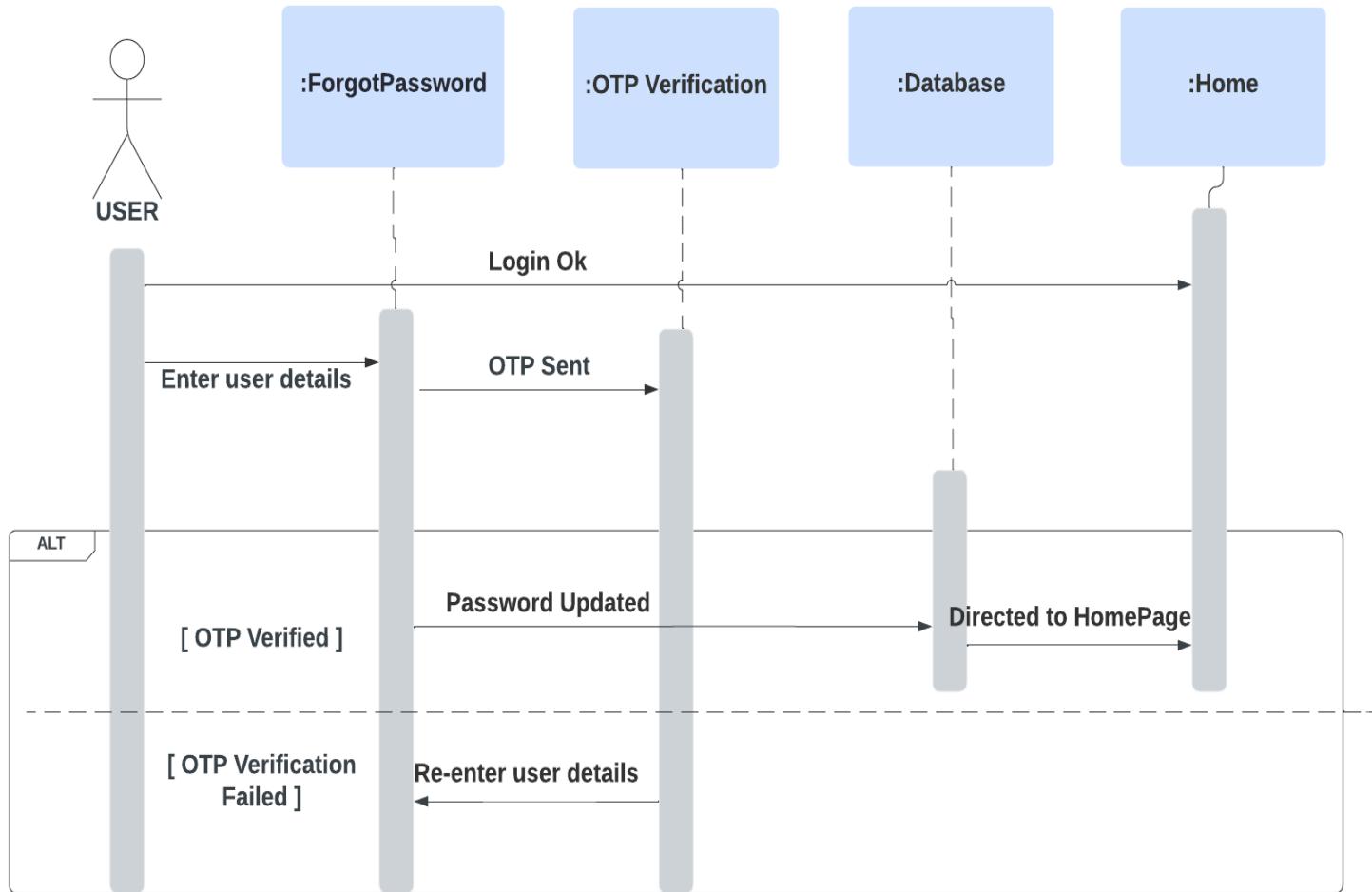
2. SignUp:



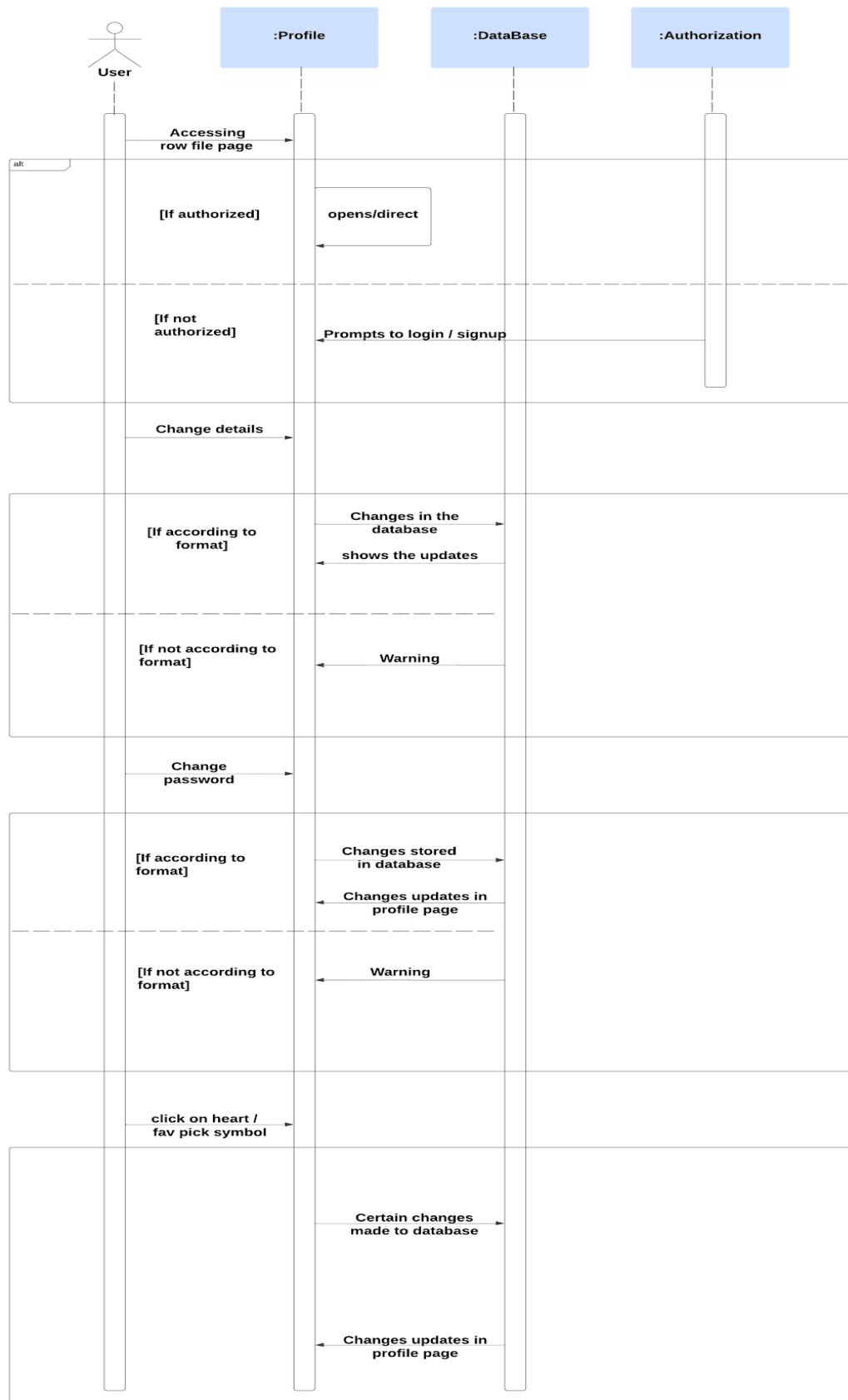
3. Rating & Review:



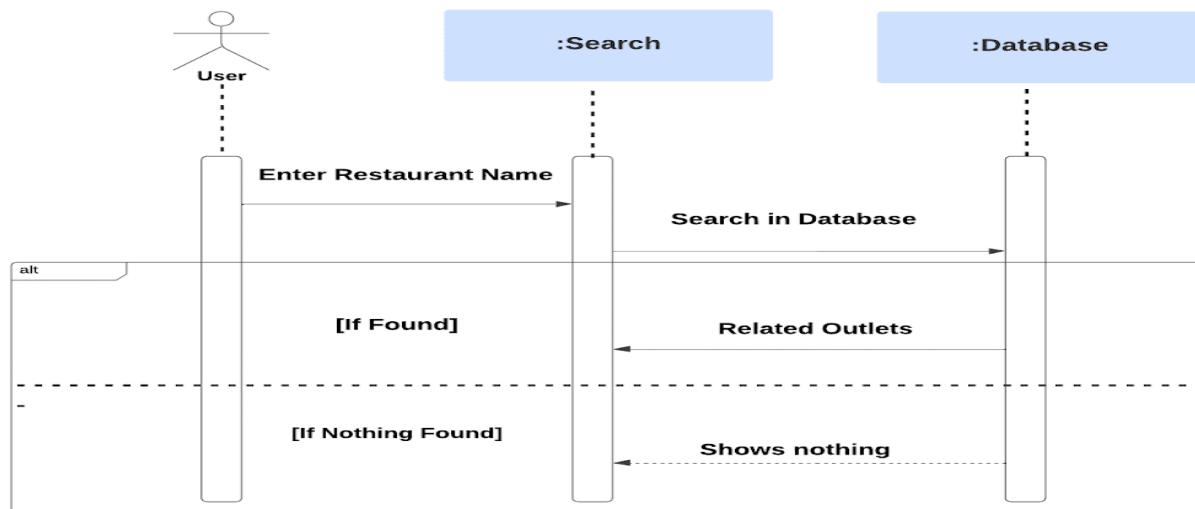
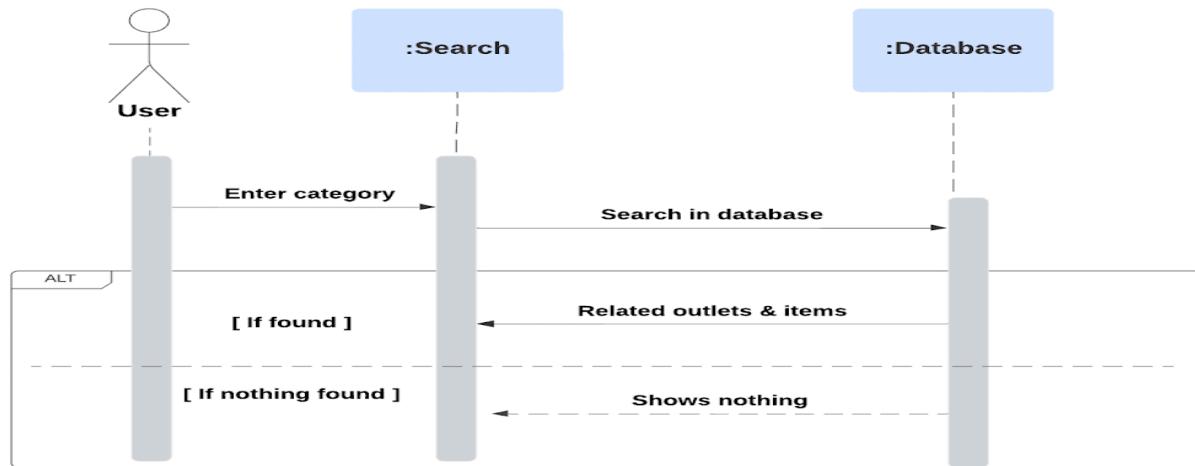
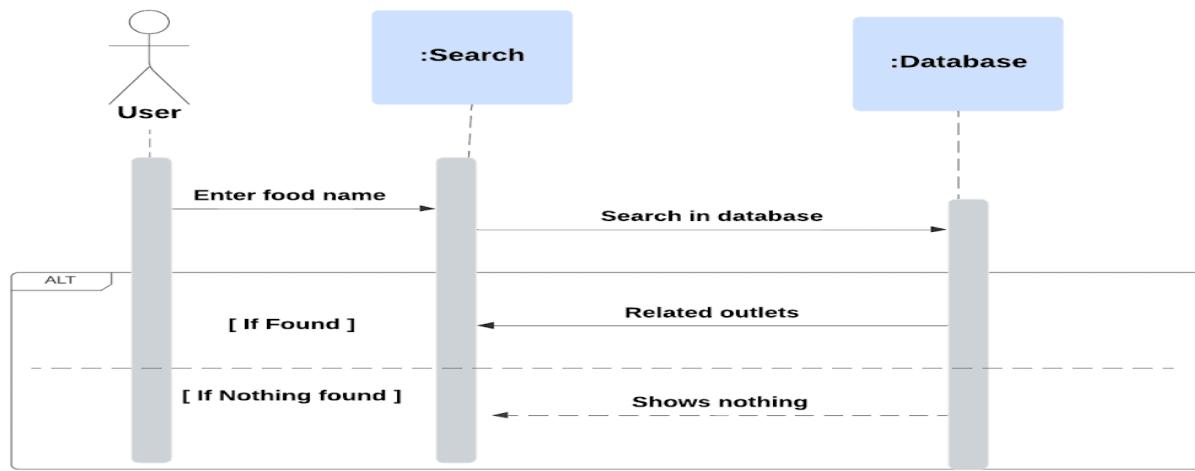
4. Forgot Password:



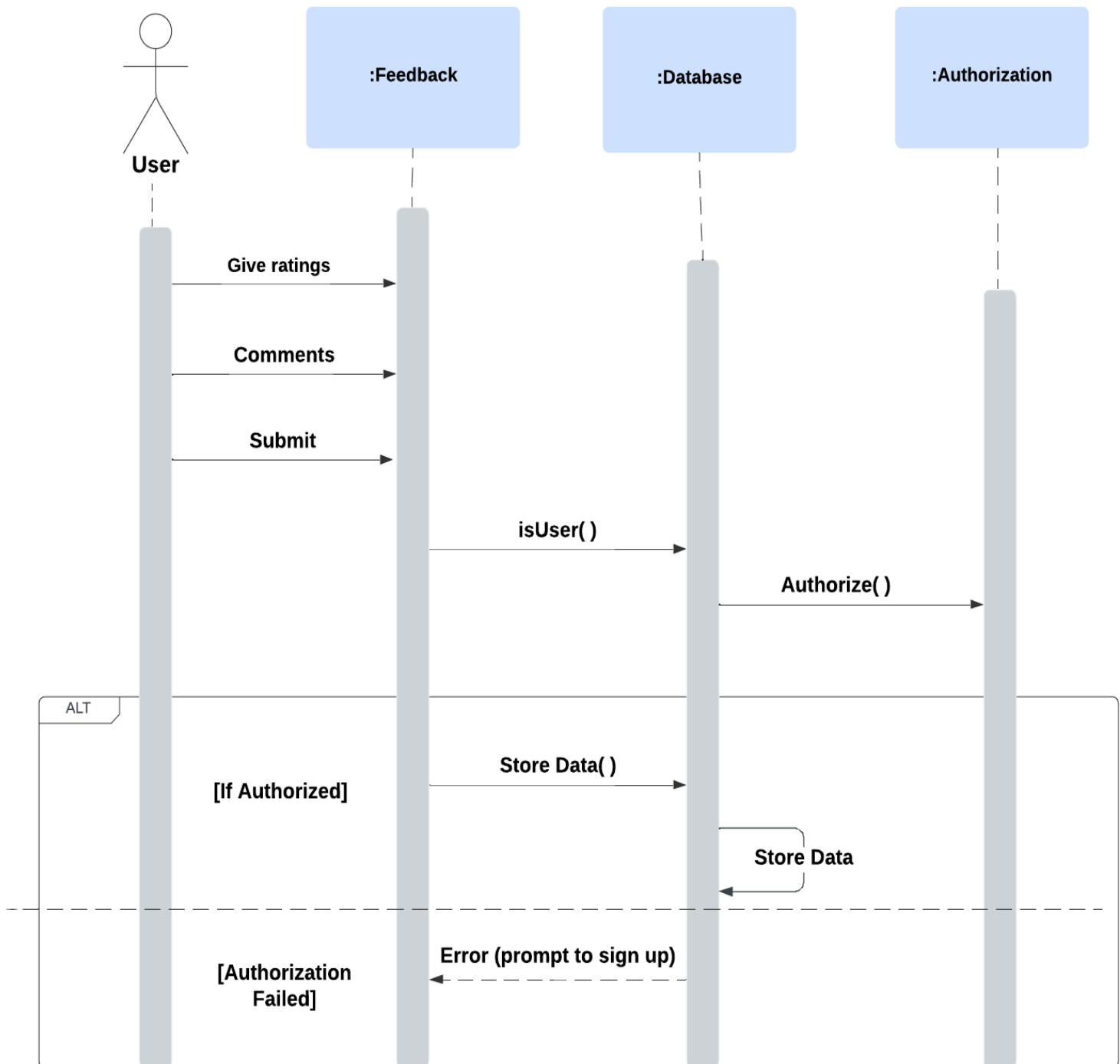
5. Profile Page:



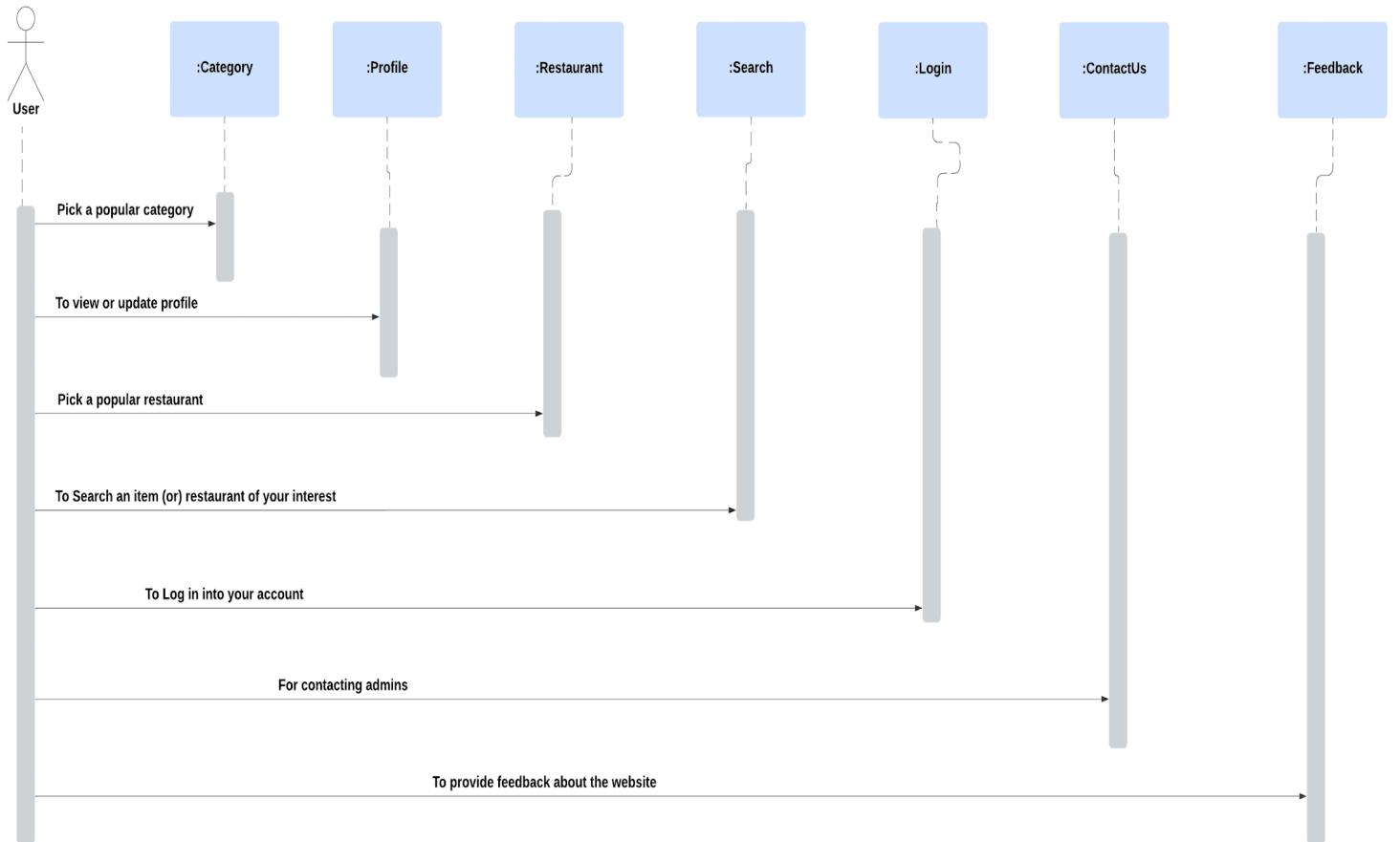
6. Search :



7. Feedback:

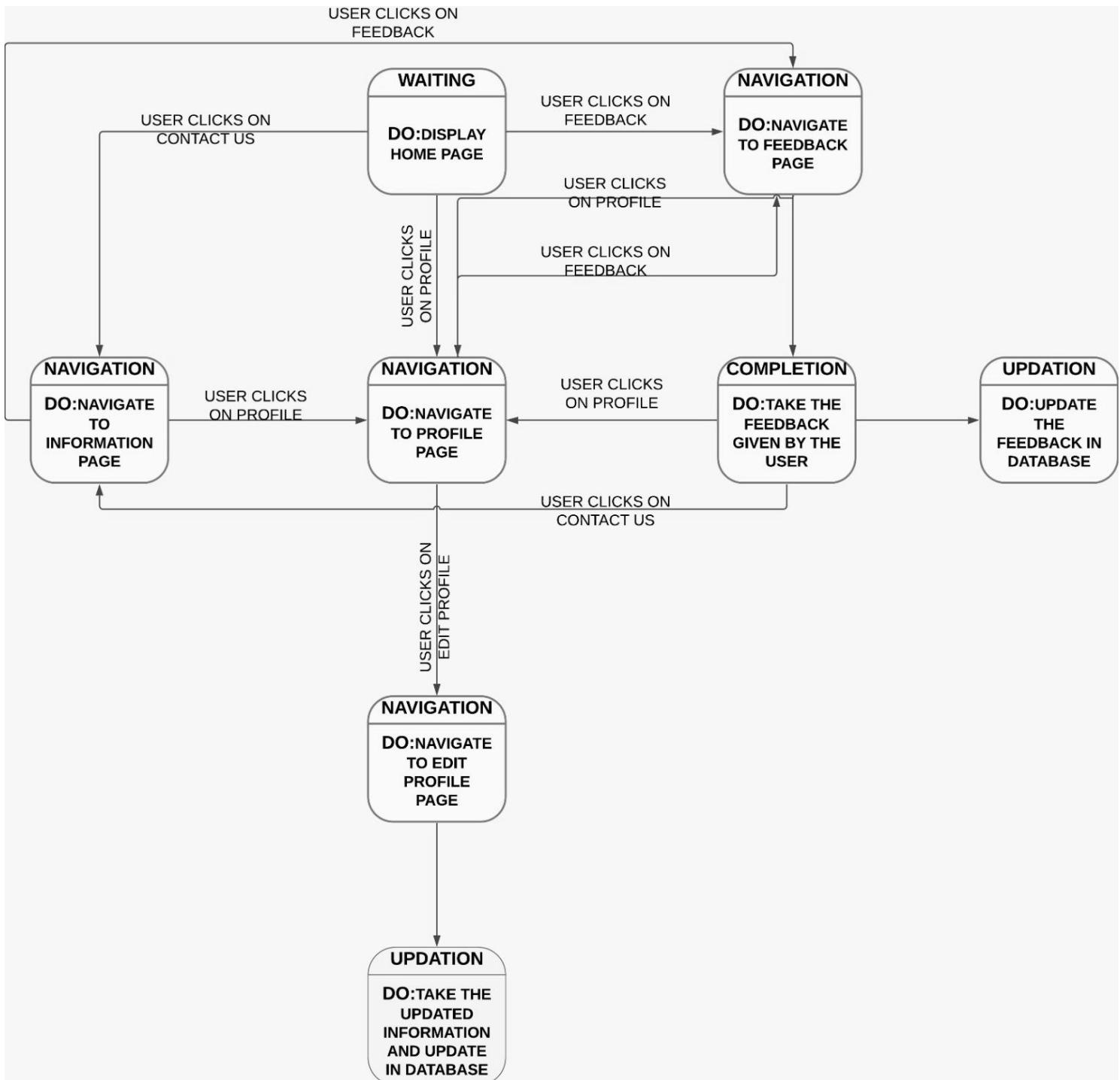


8. Home Page Navigation:

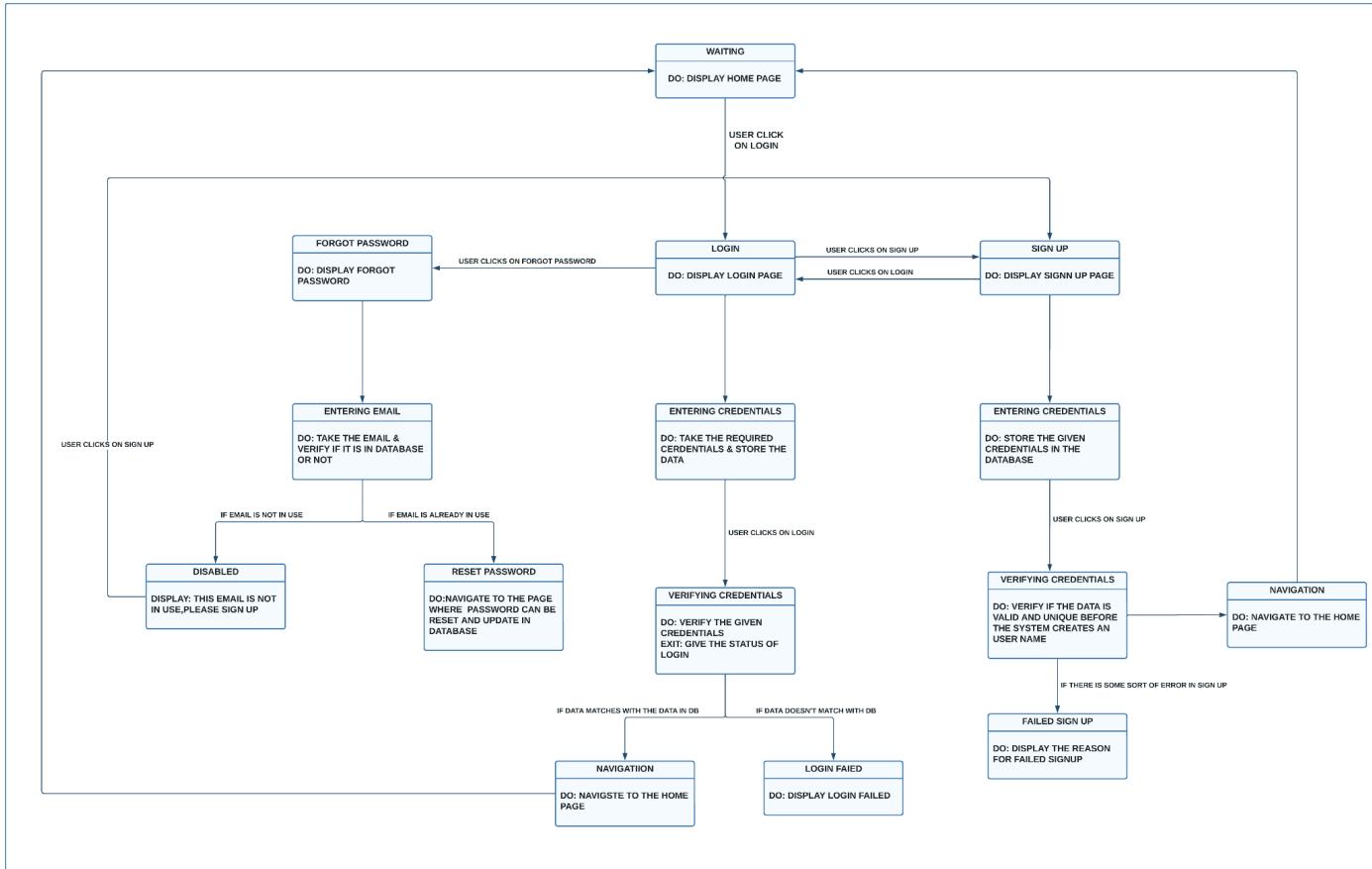


3.4 State Diagrams

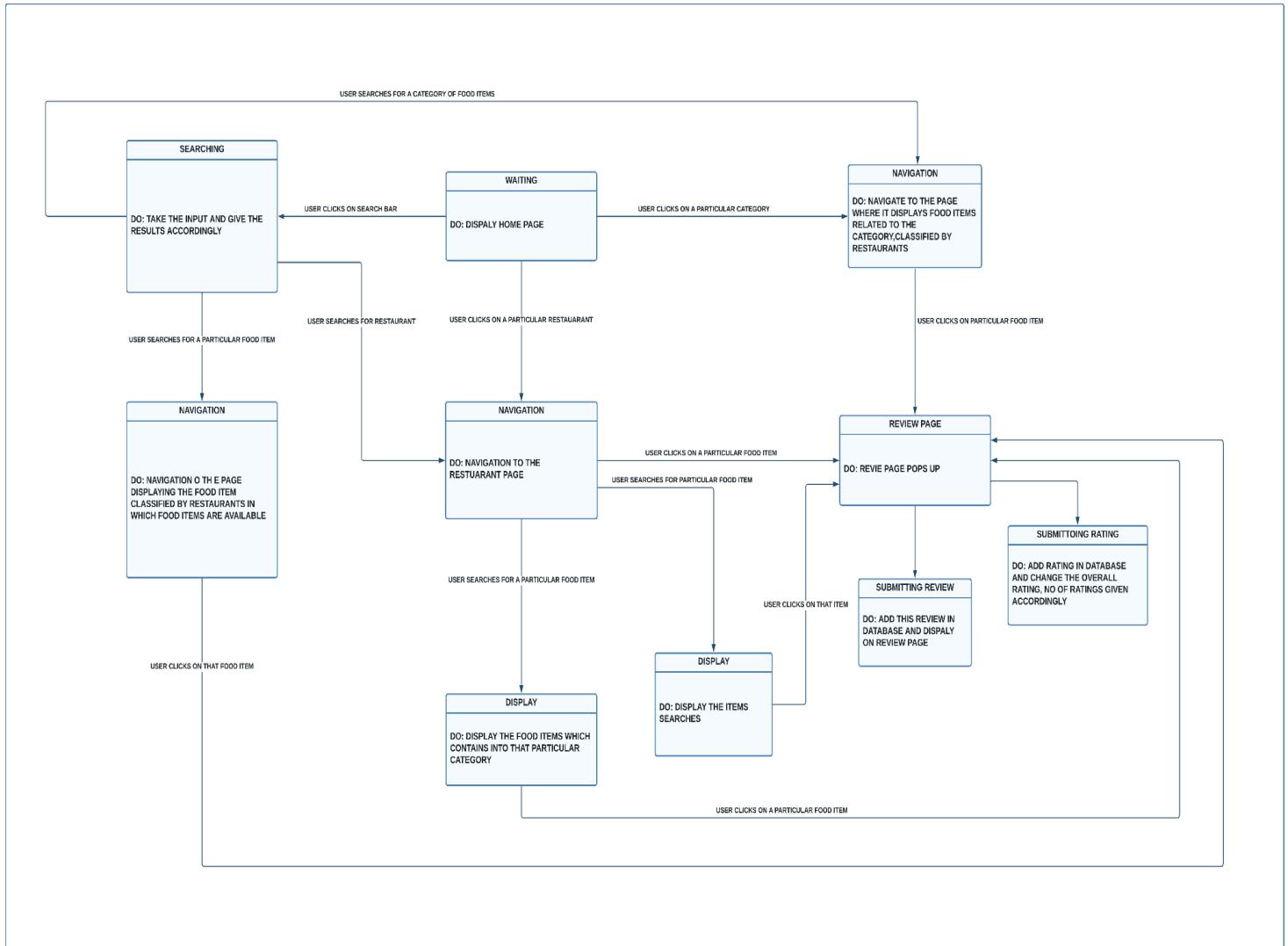
3.4.1 Edit Profile, Contact Us and Feedback



3.4.2 Authentication



3.4.3 Rating and Reviews



4 Project Plan

UI/UX Design Phase: (14'Jan - 26'Jan)

- **Conducting user research and gathering requirements:**
 - This phase involves understanding user needs and preferences, which informs both frontend and backend development.
 - Timeline:- (3-5 Days)
 - Team member :- Harshith, Sai Dishanth
- **Creating wireframes and prototypes:**
 - These are typically part of the frontend design process, focusing on the layout and structure of the user interface.
 - Timeline:- (3-5 Days)
 - Team member :- Rohan, Gowtham
- **Finalizing UI/UX designs:**
 - Ensuring the user interface meets design and usability standards.
 - Timeline:- (3-5 Days)
 - Team member :- Harshith, Sai Dishanth

Frontend Development Phase: (10'Feb - 29'Feb)

- **Setting up the development environment:**
 - This is the initial step in preparing for frontend development work.
 - Timeline:- (3-5 Days)
 - Team member :- Bhanu Prakash, Raj Vinayak Meena
- **Developing the frontend architecture:**
 - This involves structuring the frontend codebase and deciding on frameworks and libraries to use.
 - Timeline:- (3-5 Days)
 - Team member :- Vijay, Rohan
- **Implementing UI designs using HTML, CSS, and JavaScript:**
 - This is the core of frontend development, translating design mockups into functional user interfaces.
 - Timeline:- (3-5 Days)
 - Team member :- Rithwin, Rushikesh Chary
- **Implementing responsive design:**
 - Ensuring the application works well on various devices is a key frontend concern.
 - Timeline:- (3-5 Days)
 - Team member :- Havish, Mahathi
- **Implementing user authentication and authorization:**
 - While authentication logic may involve backend processes, the implementation on the frontend involves user interface elements like login forms.
 - Timeline:- (3-5 Days)
 - Team member :- Raj Vinayak Meena, Bhanu Prakash
- **Implementing features such as search functionality, user comments, and ratings display:**
 - These are frontend components that interact with backend APIs to fetch and display data.
 - Timeline:- (3-5 Days)
 - Team member :- Gowtham, Rohan

Backend Development Phase: (10'Feb - 29'Feb)

- **Defining database schemes:**
 - Backend developers design the structure of the database to store and manage application data.
 - Timeline:- (3-5 Days)
 - Team member :- Rithwin, Mahathi
- **Setting up the backend server and database:**
 - This involves configuring the server environment and database systems to support backend functionality.
 - Timeline:- (3-5 Days)
 - Team member :- Rithwin, Mahathi
- **Implementing CRUD operations:**
 - Backend developers write the logic to create, read, update, and delete data from the database.
 - Timeline:- (3-5 Days)
 - Team member :- Vijay, Bhanu Prakash
- **Implementing API endpoints for frontend-backend communication:**
 - This is a critical aspect of backend development, allowing frontend and backend systems to exchange data.
 - Timeline:- (3-5 Days)
 - Team member :- Rushikesh Chary, Havish
- **Implementing business logic for features:**
 - Backend developers handle the logic and algorithms that power application features like search bar, user authentication and comment section.
 - Timeline:- (3-5 Days)
 - Team member :- Rithwin, Mahathi
- **Handling security considerations:**
 - Backend developers implement measures like input validation, authentication, and authorization to protect the application and its data.
 - Timeline:- (3-5 Days)
 - Team member :- Rushikesh Chary, Havish
- **Implementing third-party integrations:**
 - Backend developers may integrate external services like email services into the application.
 - Timeline:- (3-5 Days)
 - Team member :- Vijay, Bhanu Prakash

Integration Phase: (27'Feb - 05'Mar) This phase involves combining frontend and backend components to ensure they work together seamlessly. It focuses on communication between the two layers and resolving any integration issues.

Team member :- All

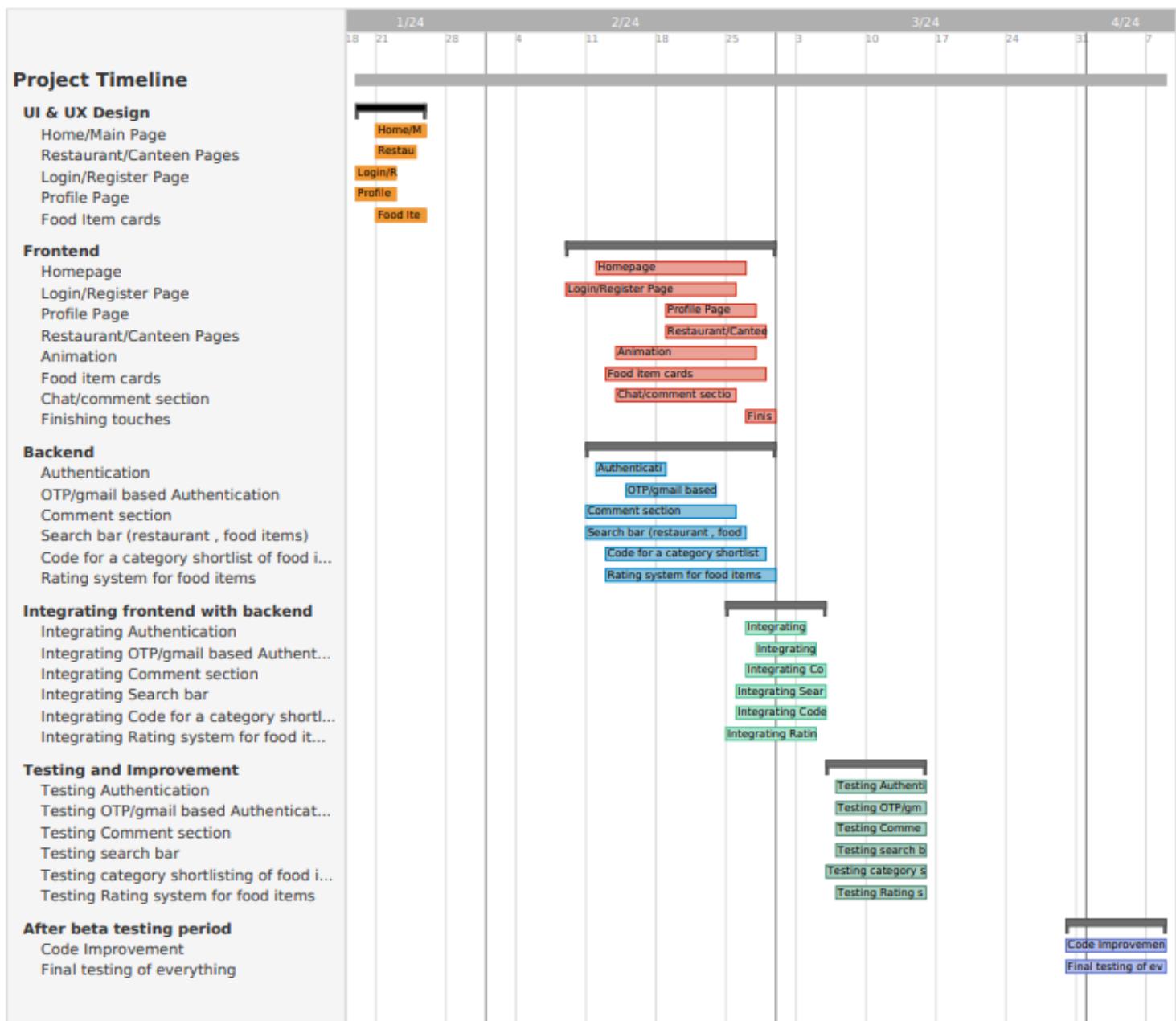
Testing Phase: (05'Mar - 09'Mar) Both frontend and backend components undergo testing to ensure they function correctly and meet requirements. This includes unit testing, integration testing, regression testing, and usability testing.

Team member :- All

Final Testing Phase: (15'Mar - 8'April) This phase focuses on comprehensive testing before deployment, including beta testing with real users, performance testing, and security testing.

Team member :- All

Timeline: (using gantt chart)



5 Other Details

Appendix A - Group Log

| S.No | Date | Venue | Description |
|------|------------|-------------|---|
| 1 | 27/01/2024 | RM building | Studied the given template and discussed on how to approach with the document |
| 2 | 28/01/2024 | RM building | Distributed the work among team members and shared the ideas among members |
| 3 | 03/02/2024 | RM building | Had a brief meeting to review the extent of work done by members and planned to proceed with the remaining document |
| 4 | 07/02/2024 | RM building | Had a brief interaction with TA, clarifying the doubts and proceeded with the document |
| 5 | 08/02/2024 | RM building | Reviewed the whole document and proceeded with any mistakes or work left |
| 6 | 09/02/2024 | RM building | Had the final review and submitted the document |