
Design Document

for

HallBuddy

Version 1.1

Prepared by

Group #: 12

Ritesh Hans
Mridul Gupta
Rohan Batra
Krutuparna Paranjape
Mrdul Agarwaal
Apoorv Tandon
Tanishq Maheshwari
Taneshq Zendey
Ayush
Samarpan Verma

220893
220672
210868
210536
210632
220192
221128
221123
220259
220943

Group Name: CodeMonk

riteshhans22@iitk.ac.in
mridulg22@iitk.ac.in
rohanb21@iitk.ac.in
krutuparna21@iitk.ac.in
mrdula21@iitk.ac.in
apoorvt22@iitk.ac.in
tanishqm22@iitk.ac.in
taneshq22@iitk.ac.in
ayushs@iit.ac.in
samarpanv22@iitk.ac.in

Course: CS253

Mentor TA: Vaibhav Tanwar

Date: 26.01.2024



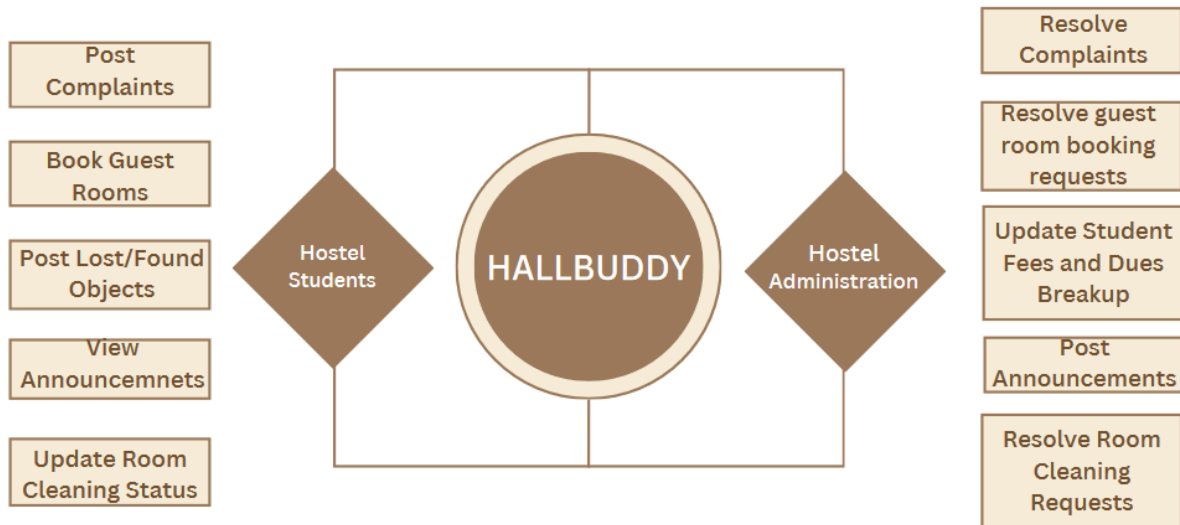
CONTENTS.....	II
REVISIONS.....	II
1 CONTEXT DESIGN.....	1
1.1 CONTEXT MODEL.....	1
1.2 HUMAN INTERFACE DESIGN.....	2
2 ARCHITECTURE DESIGN.....	9
3 OBJECT-ORIENTED DESIGN.....	11
3.1 USE CASE DIAGRAM.....	11
3.2 CLASS DIAGRAM.....	15
3.3 SEQUENCE DIAGRAM	18
3.4 STATE DIAGRAM	22
4 PROJECT PLAN.....	25
APPENDIX A - GROUP LOG.....	27

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
1.1	Apoorv Tandon Ayush Mridul Gupta Mrdul Agarwaal Krutuparna Paranjape Rohan Batra Samarpan Verma Ritesh Hans Tanishq Maheshwari Taneshq Zendey	First Draft : This version contains details of the user interface diagrams, use case diagrams, class diagrams, sequence diagrams, and state diagrams for our software.	09/02/24

1. Context Design

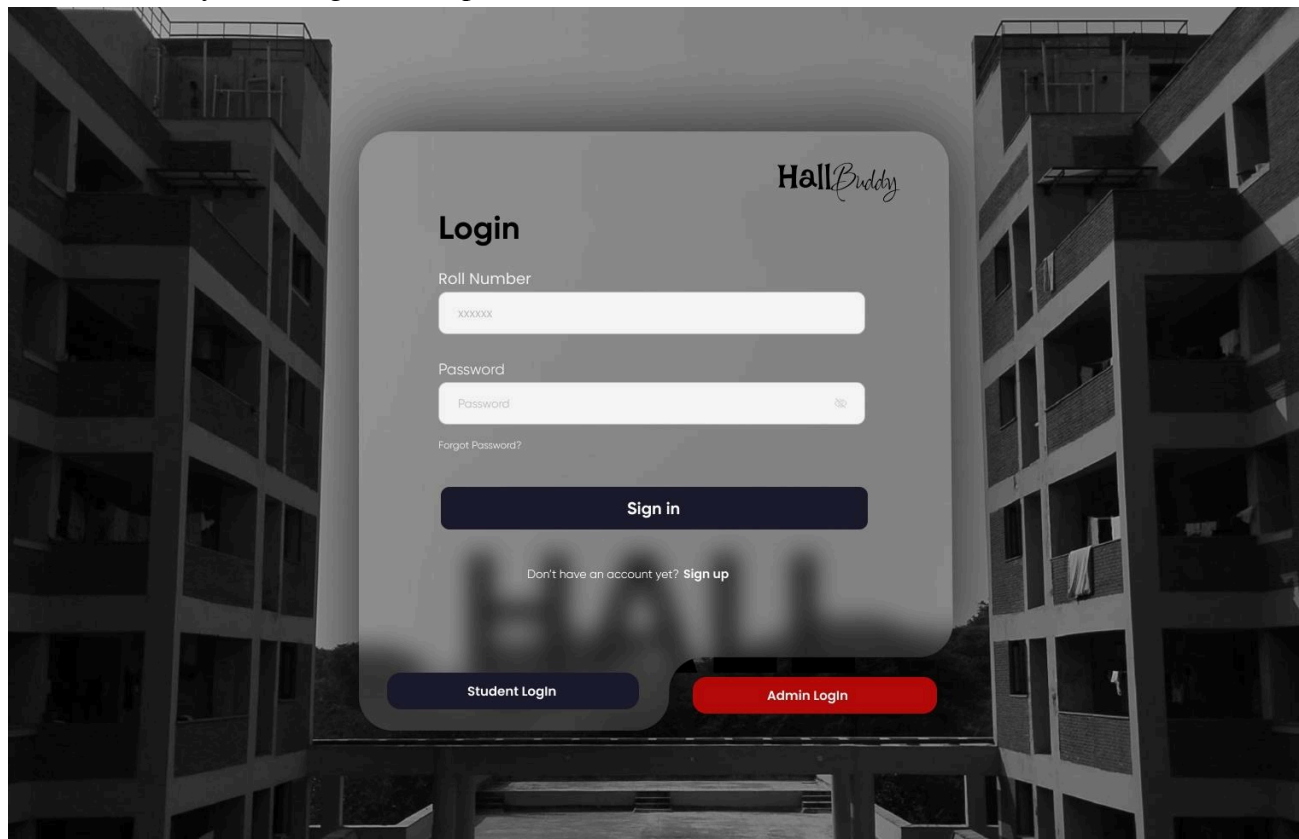
1.1 Context Model



1.2 Human Interface Design

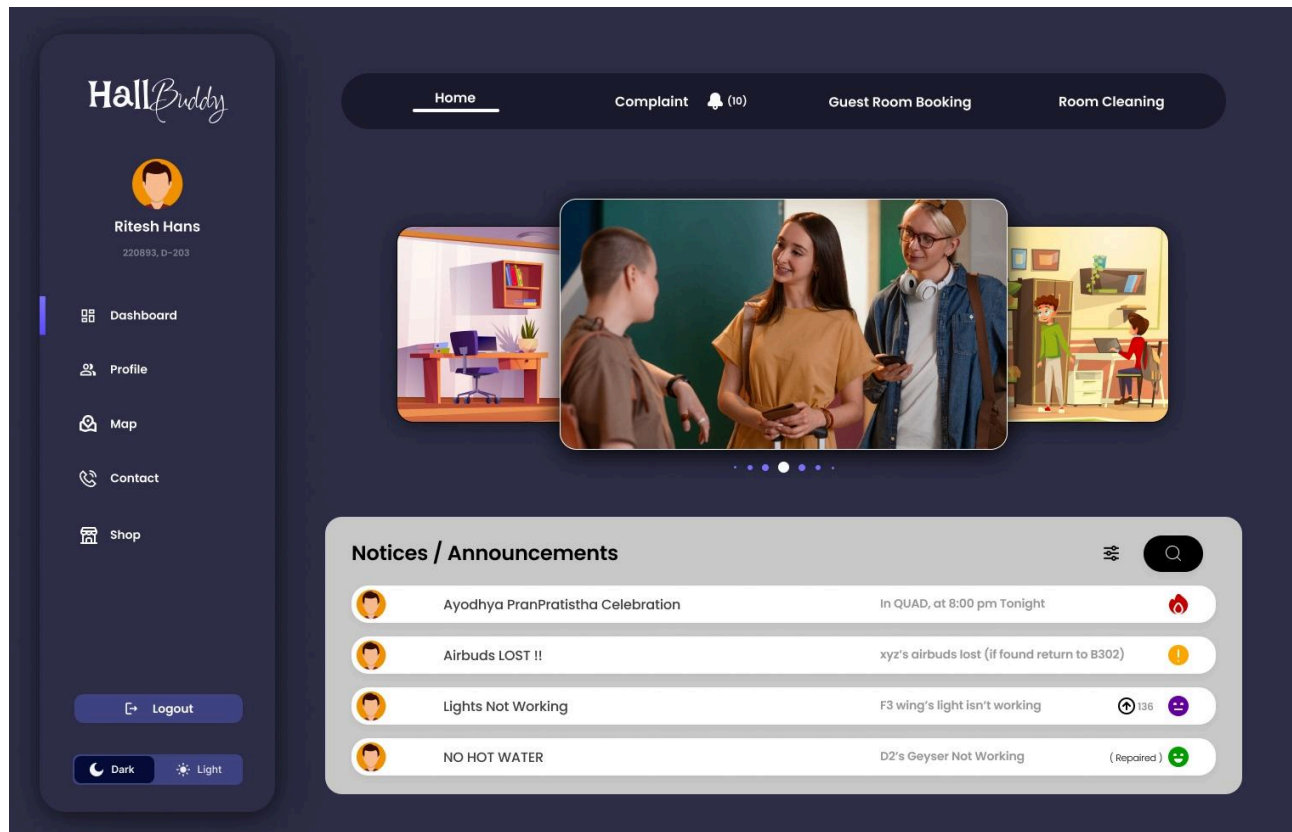
1. Log-In Page

This page allows both students and admin to login and access their account. It also contains fields to signup if the user is not registered as well as a forgot password option if a registered user may have forgotten his password.



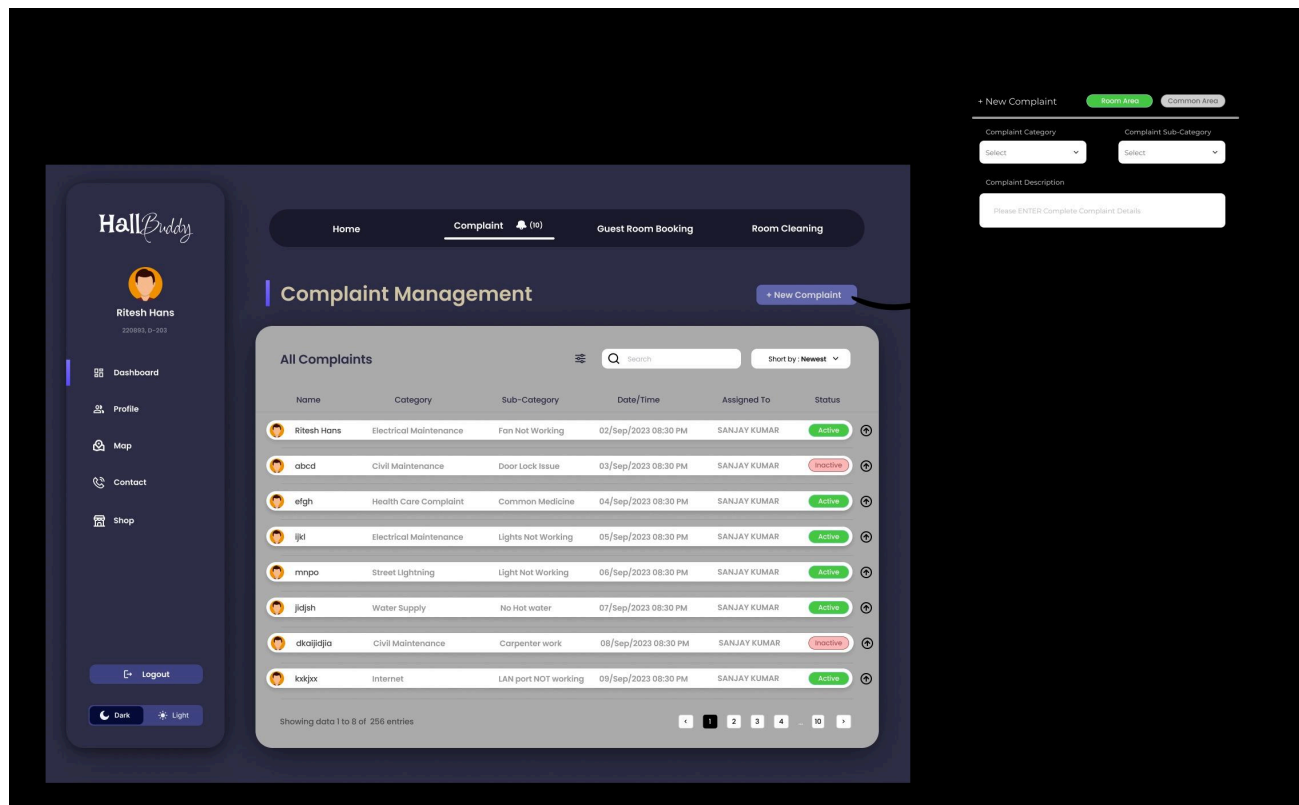
2. Dashboard

The dashboard contains options to access one's profile, hall map, lodge a complaint, book the guest room, request/update room cleaning, check the hall shop catalog or contact the warden/HEC.



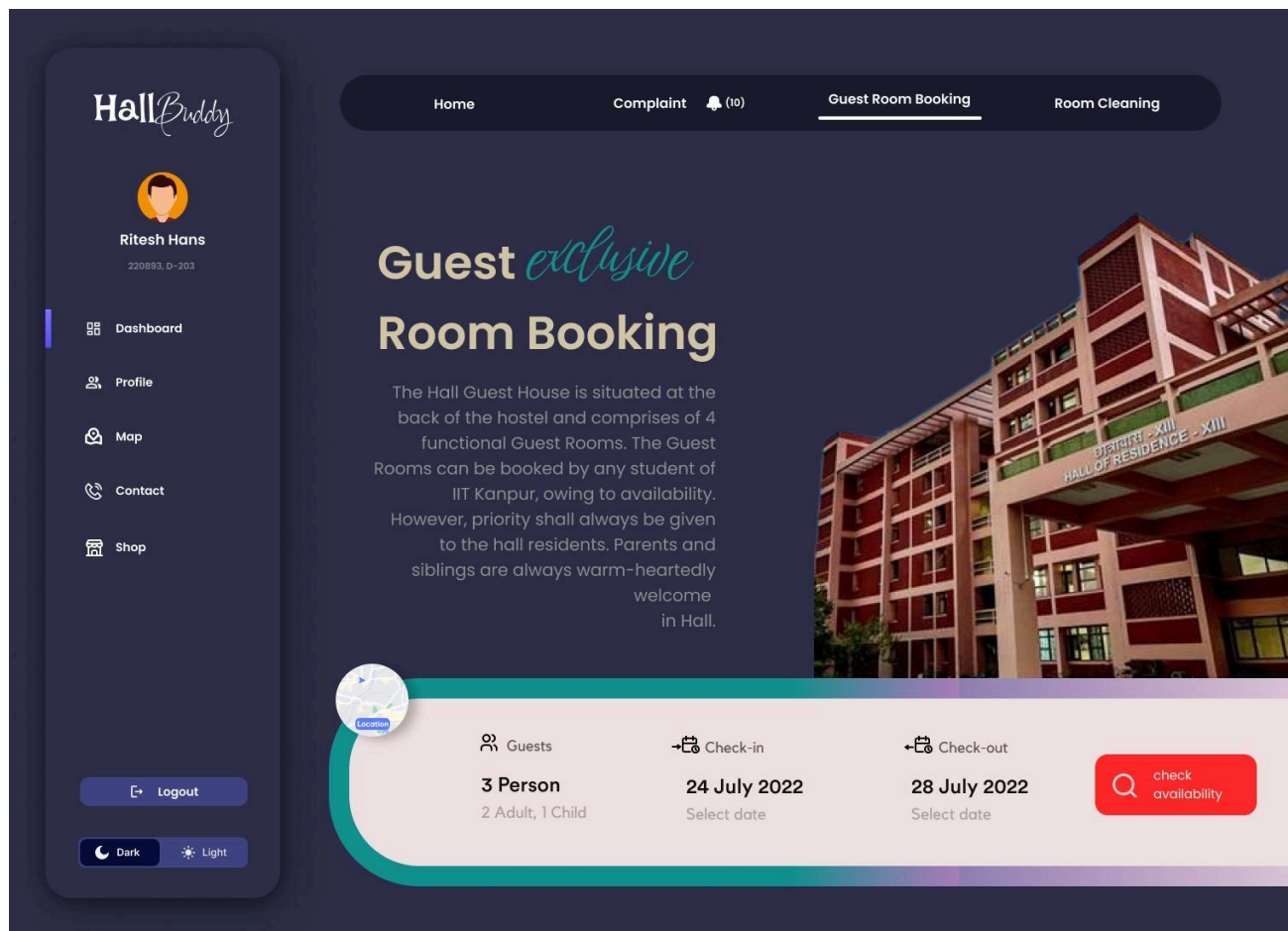
3. Complaint Management Page

This is the complaints page wherein one can lodge a new complaint by clicking on the new complaint option or view the existing complaints.



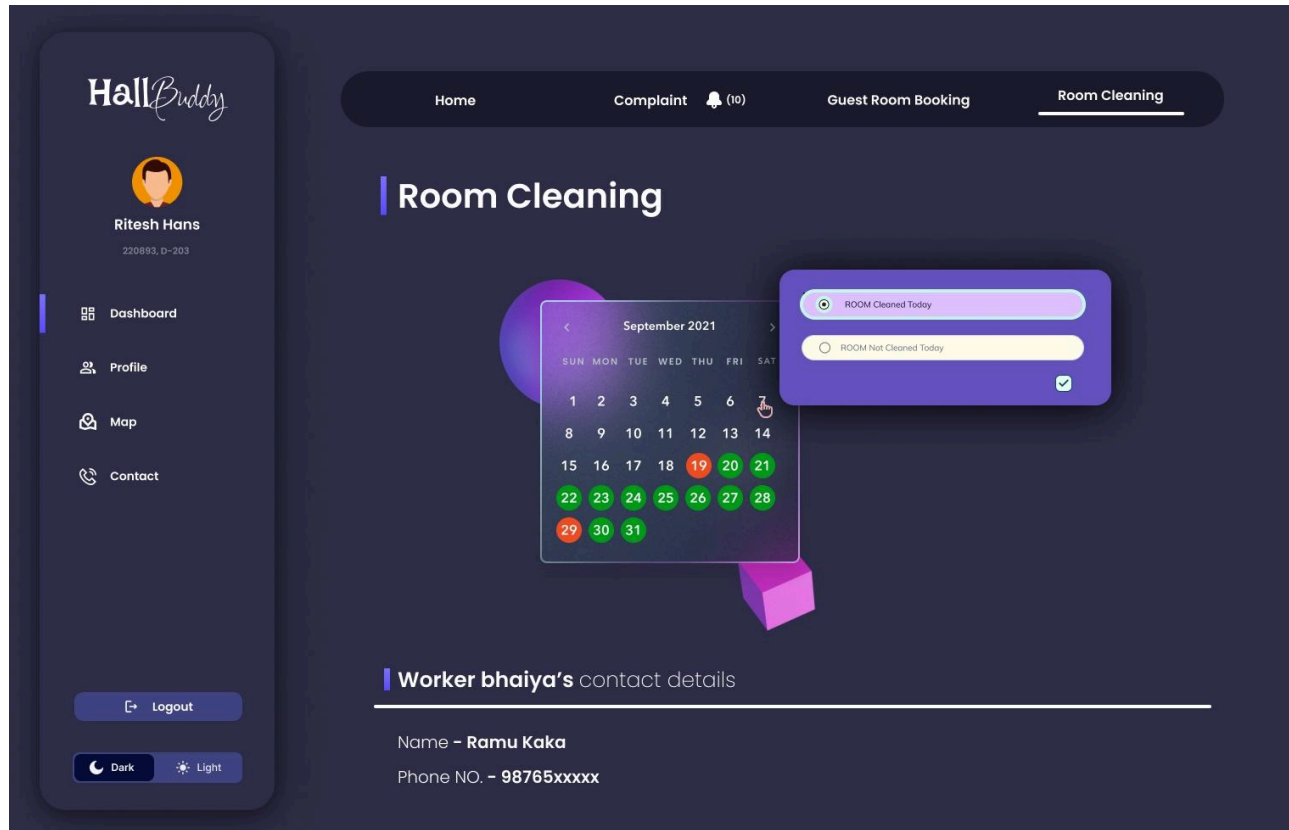
4. Guest Room Booking

The Guest Room Booking Page allows the user to check the price and the availability of the guest room after entering the Check-in Dates, Check-Out Date and the number of guests.



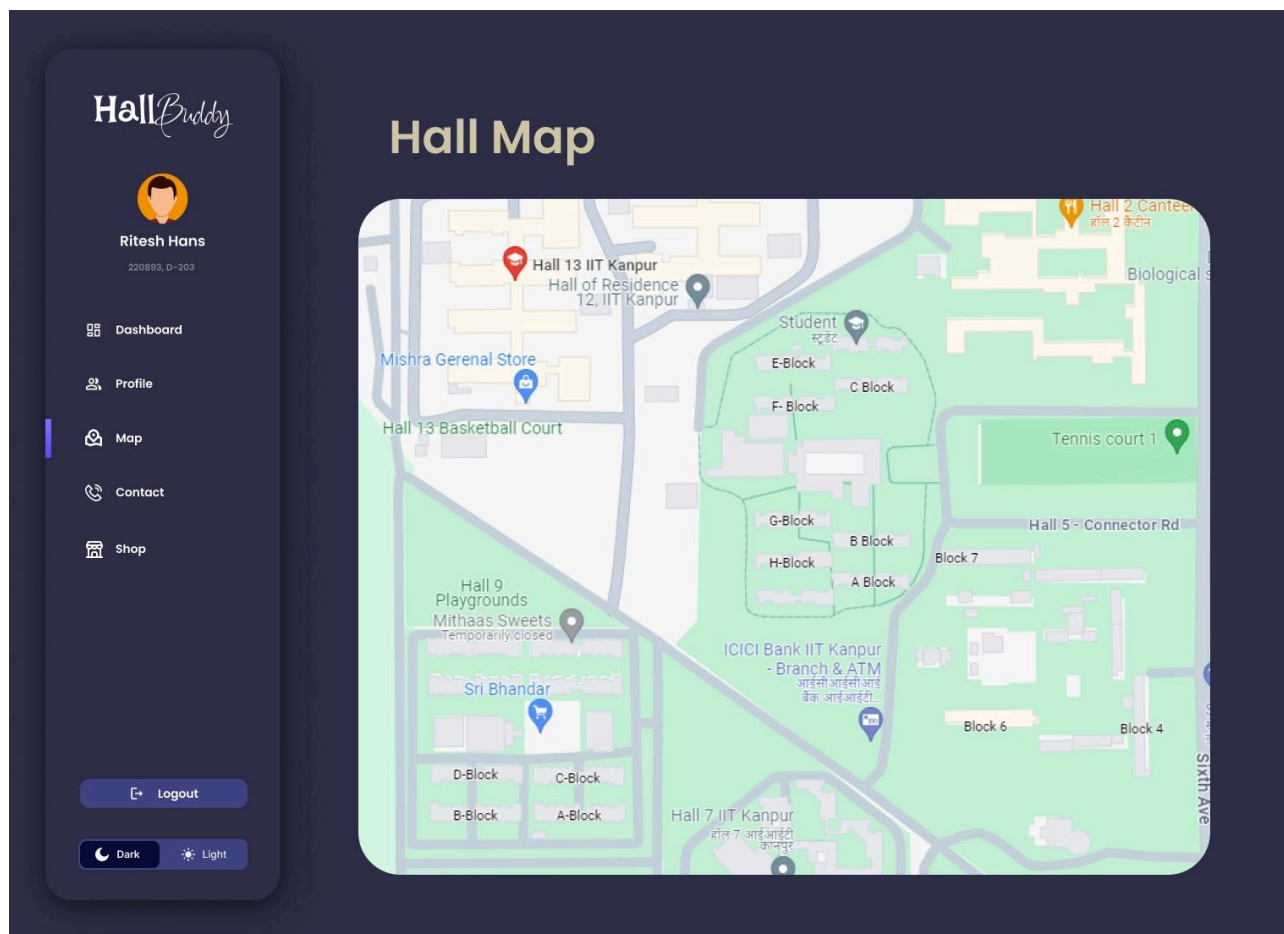
5. Room Cleaning

The room cleaning portal would allow the user to update whether the room has been cleaned on a particular day. The student can also contact the respective worker responsible for his wing.



6. Hall Map

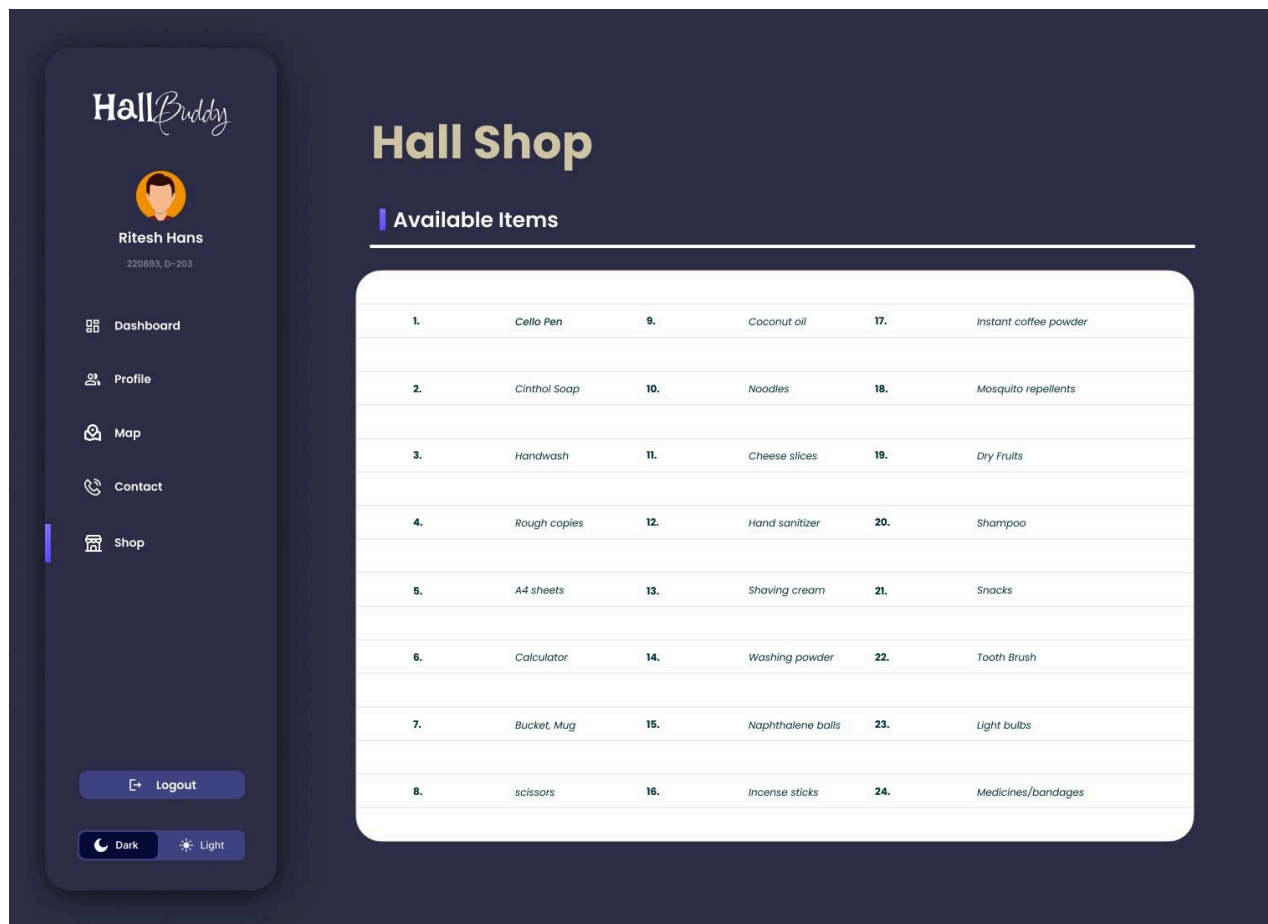
The Hall map contains the map of the hall for easier reachability and access to different hostel facilities.



7. Hall Shop

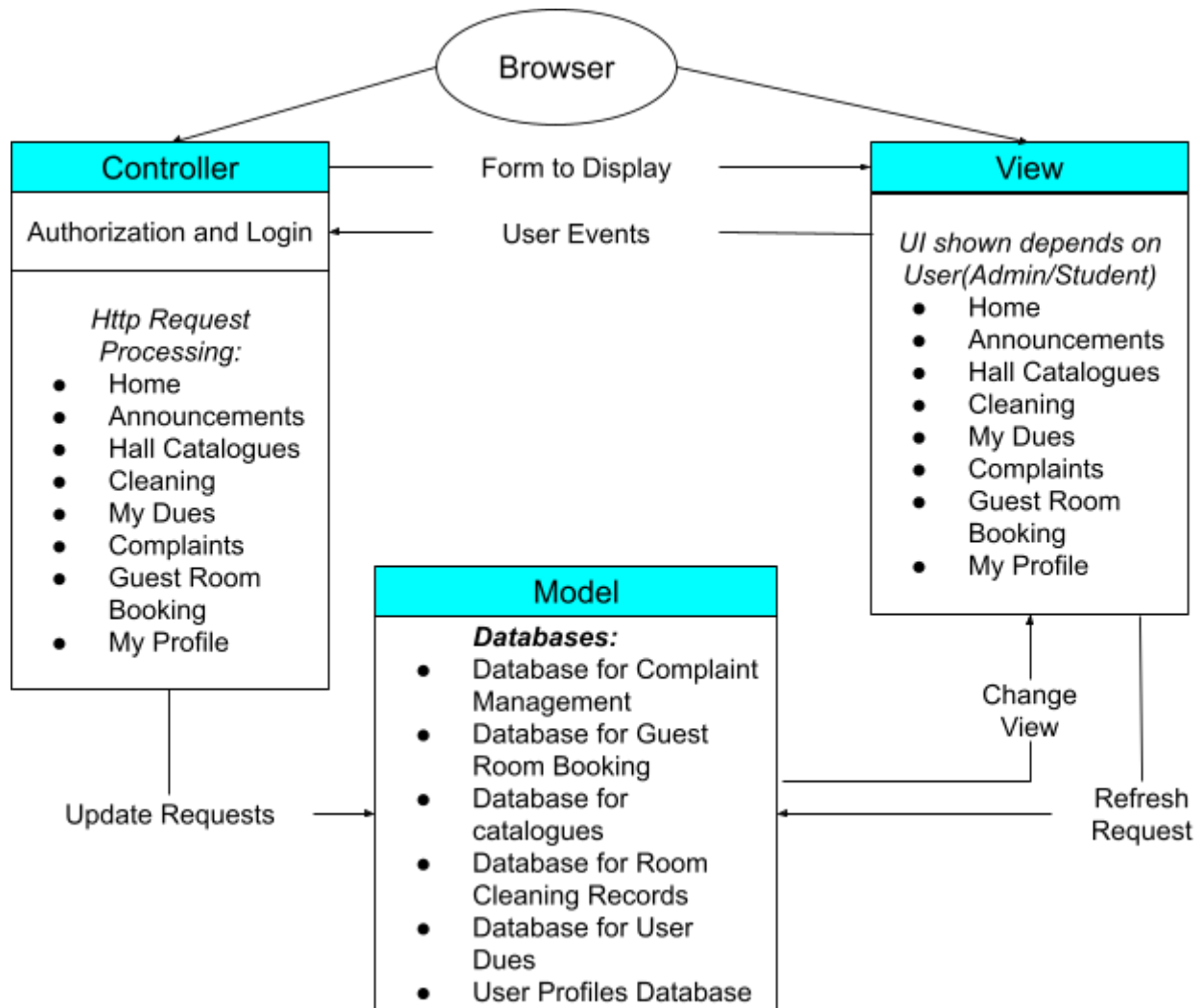
The hall shop contains the hall shop catalog wherein the student can check if the requisite item is available or not. They can also contact the Hall Owner to request certain items

The hall shop owner would have a different interface wherein he can update the time available with him .



2. Architecture Design

Model-View-Control Pattern



The Model-View-Controller architecture is very common amongst Web Applications since it allows maintenance and editing of a uniform database while allowing for different presentations of that data based on request/user.

- This model does not interfere with implementation of security, latency and responsiveness constraints.
- The model ensures reliability of data due to the streamlined mechanisation of data queries, presentation, and editing, as depicted in the model above.

Model:

1. Multiple Databases will be maintained in the software:
 - a. Database for Complaint Management
 - b. Database for Guest Room Booking
 - c. Database for catalogues
 - d. Database for Room Cleaning Records
 - e. Database for User Dues
 - f. User Profiles Database

Controller:

The main parts of the controller will be:

- User Authorization
- HTTP Requests

Authorization: Will include Sign-Up, Sign-In, Forgot Password functionality

HTTP Requests: Will be passed and evaluated based on the current state of the software

- Home
- Announcements
- Catalogues
- Cleaning
- My Dues
- Complaints
- Guest Room Booking
- My Profile

View:

According to their identity, users are shown different UIs amongst:

- Home
- Announcements
- Catalogues
- Cleaning
- My Dues
- Complaints
- Guest Room Booking
- My Profile

The links between Model, Controller and View are as follows:

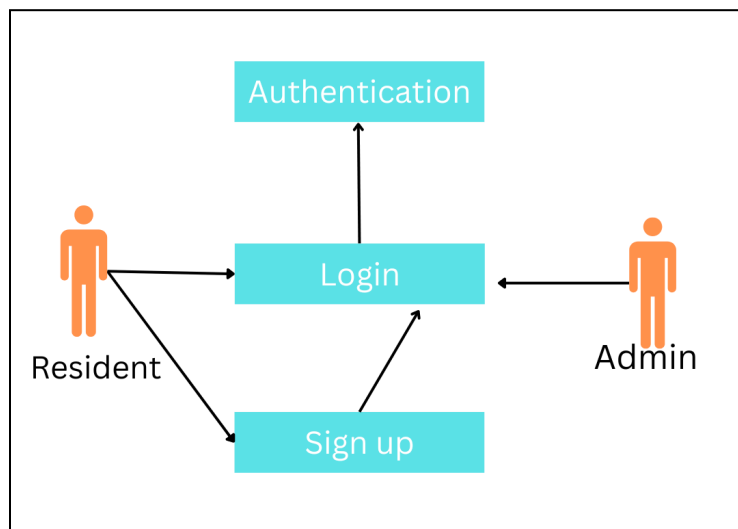
- The Controller authorizes based on User data stores in the model, and implements any updates in the model that are initiated by Admin
- Views communicate with Controllers and Models to cater to requests sent by the user.

3. Object Oriented Design

3.1 Use Case Diagrams

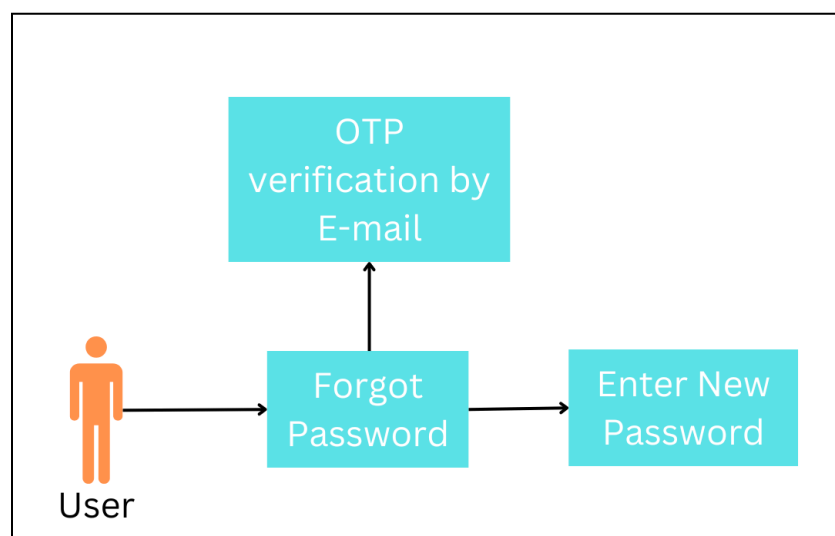
1. Sign Up and Login

Purpose: This use case is used for the Login/ Signup of the Resident and the Admin. Login is used for the registered users and the Signup is used for the first-time users.



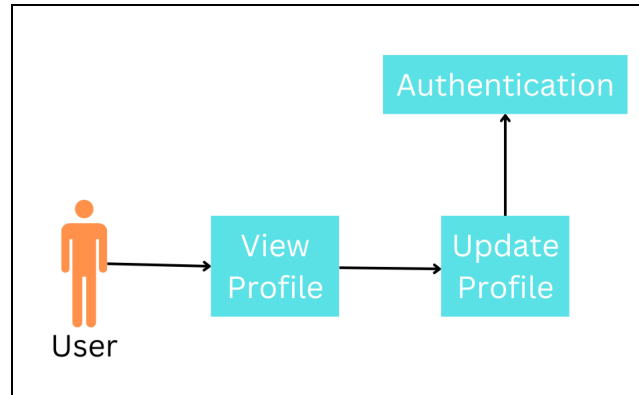
2. Forgot Password

Purpose: This use case is used for new password creation when the user forgets their password.



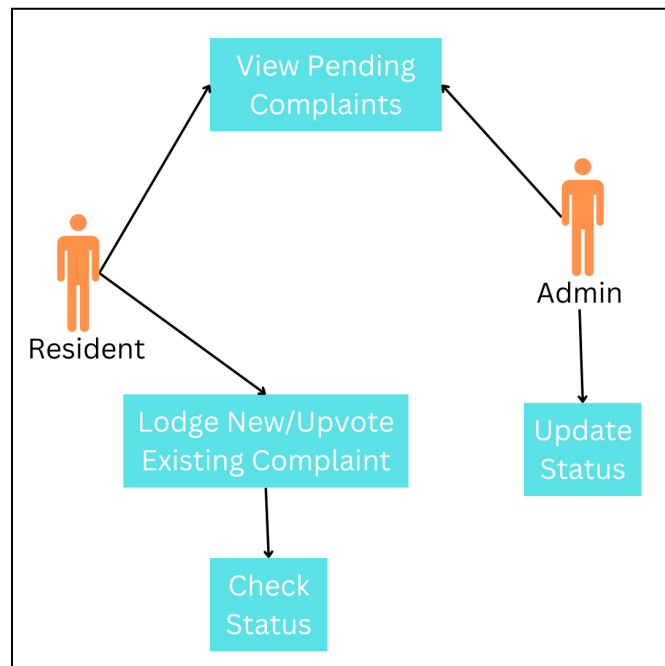
3. View and Update Profile

Purpose: This use case allows the Resident to view and update their Profile Information including Room Number and Contact Details.



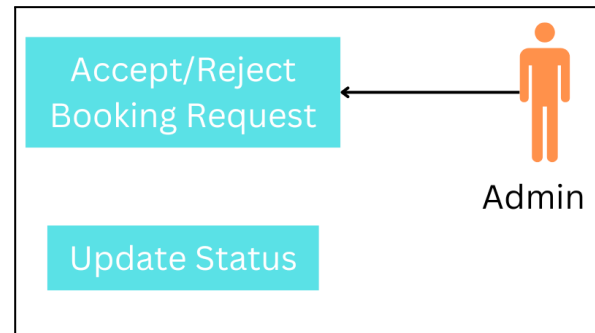
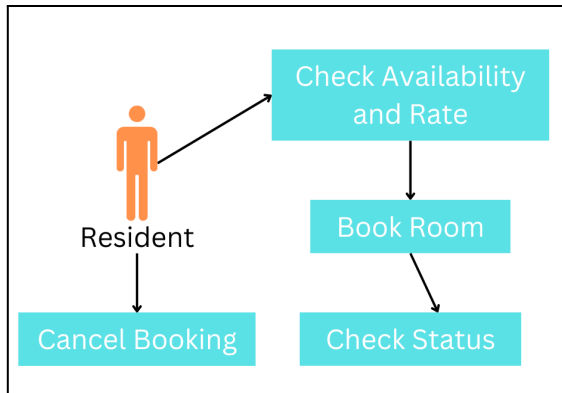
4. Complaint management

Purpose: This use case allows the users to view the existing complaints and check their response status. The Resident can upvote existing complaints if they concern them. The admin can manage the response status of the complaints as – “Resolved”, “In Progress”. It also allows the Resident to create new complaints. The resident will be able to enter text description, attach images and select suitable filters based on the category of complaint (e.g. Electrical, Civil maintenance etc.).



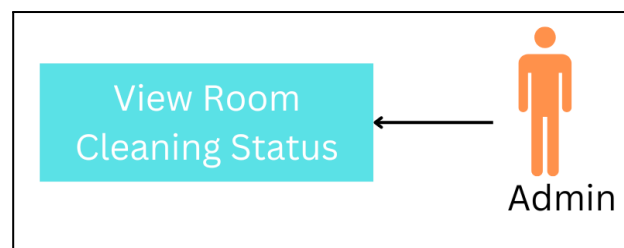
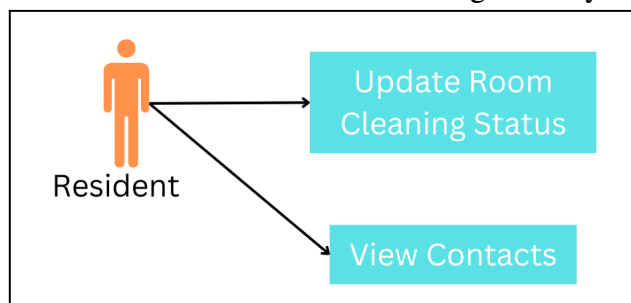
5. Guest Room Booking

Purpose: This use case is for handling guest room bookings. The user can view prices and availability and book the guest room. The admin must approve the request before the booking is confirmed. This also allows the user to cancel his/her booking



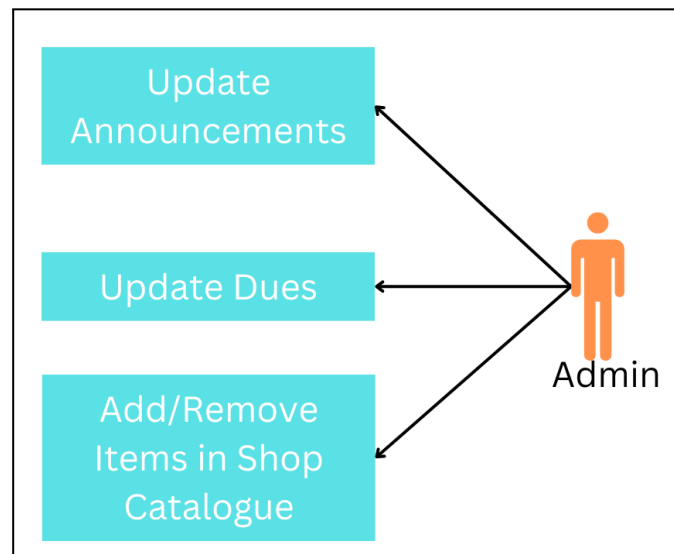
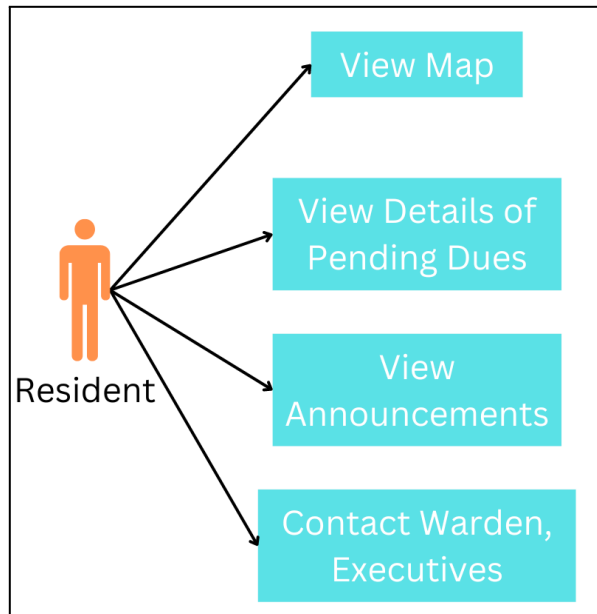
6 .Room Cleaning Status

Purpose: The purpose of this user case is to provide users with the ability to update the cleaning status of rooms within the hall management system.

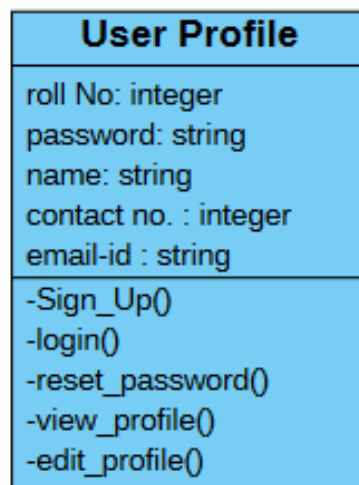
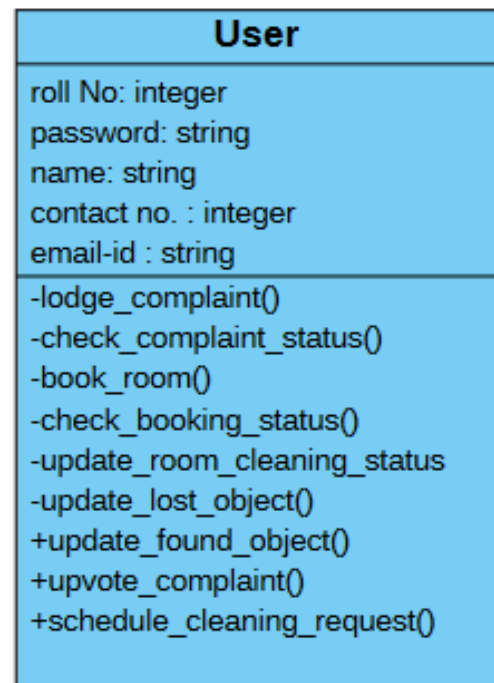
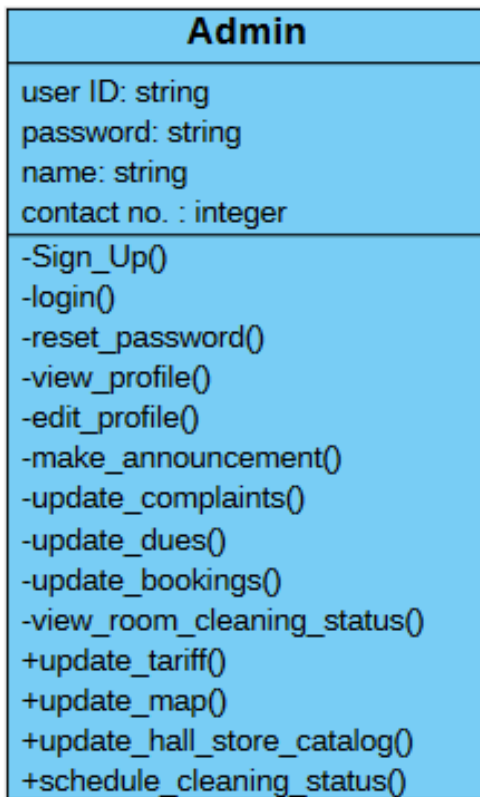


7. Miscellaneous Use Cases

Purpose: This use case covers all miscellaneous actions of the user/admin such as viewing map of the hostel, contacts of the hall executives, view pending dues.



3.2 Class Diagrams



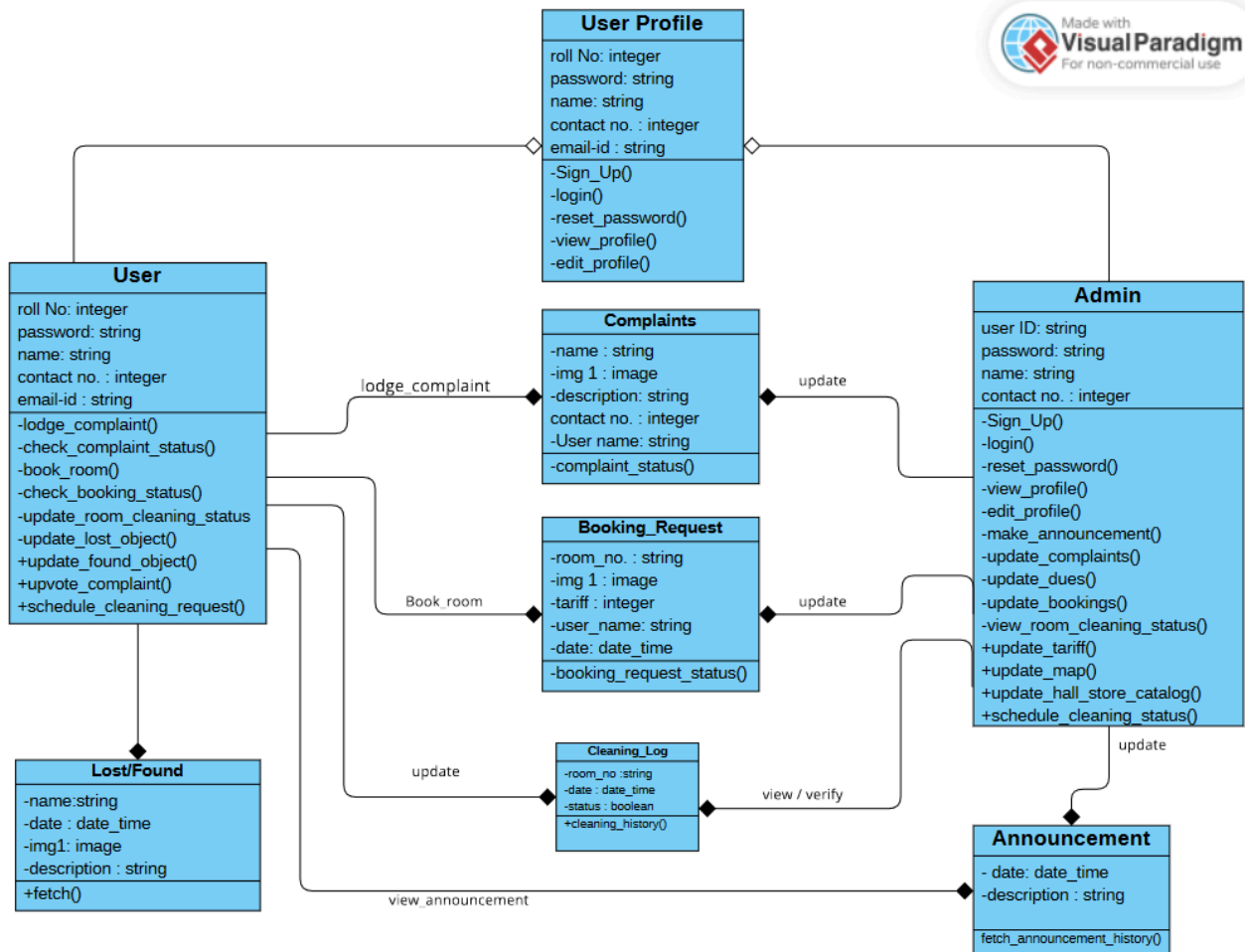
Complaints
-name : string -img 1 : image -description: string contact no. : integer -User name: string -complaint_status()

Booking_Request
-room_no. : string -img 1 : image -tariff : integer -user_name: string -date: date_time -booking_request_status()

Cleaning_Log
-room_no :string -date : date_time -status : boolean +history()

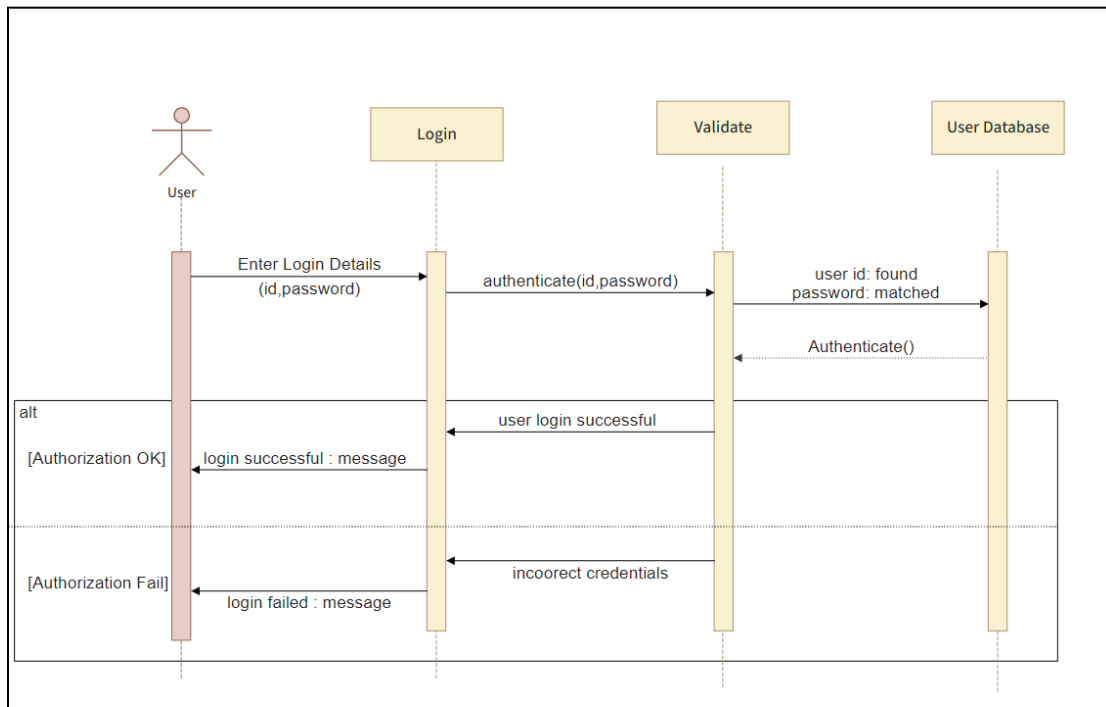
Lost/Found
-name:string -date : date_time -img1: image -description : string +update lost/found()

Announcement
- date: date_time -description : string fetch_announcement_history()

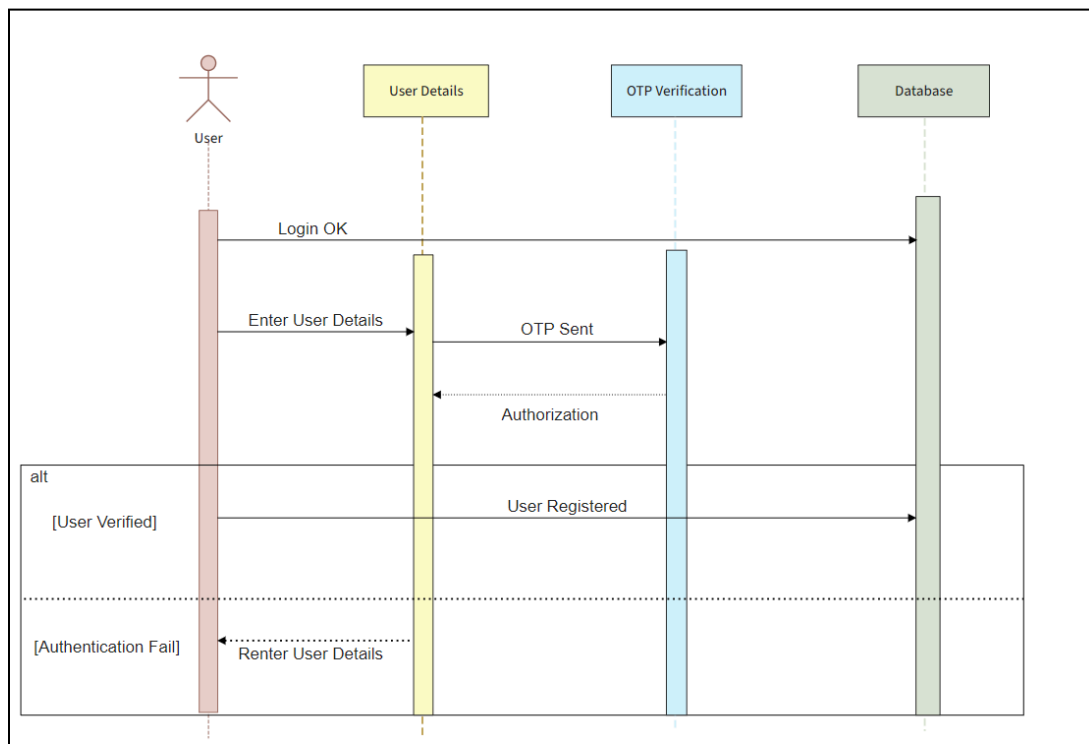


3.3 Sequence Diagrams

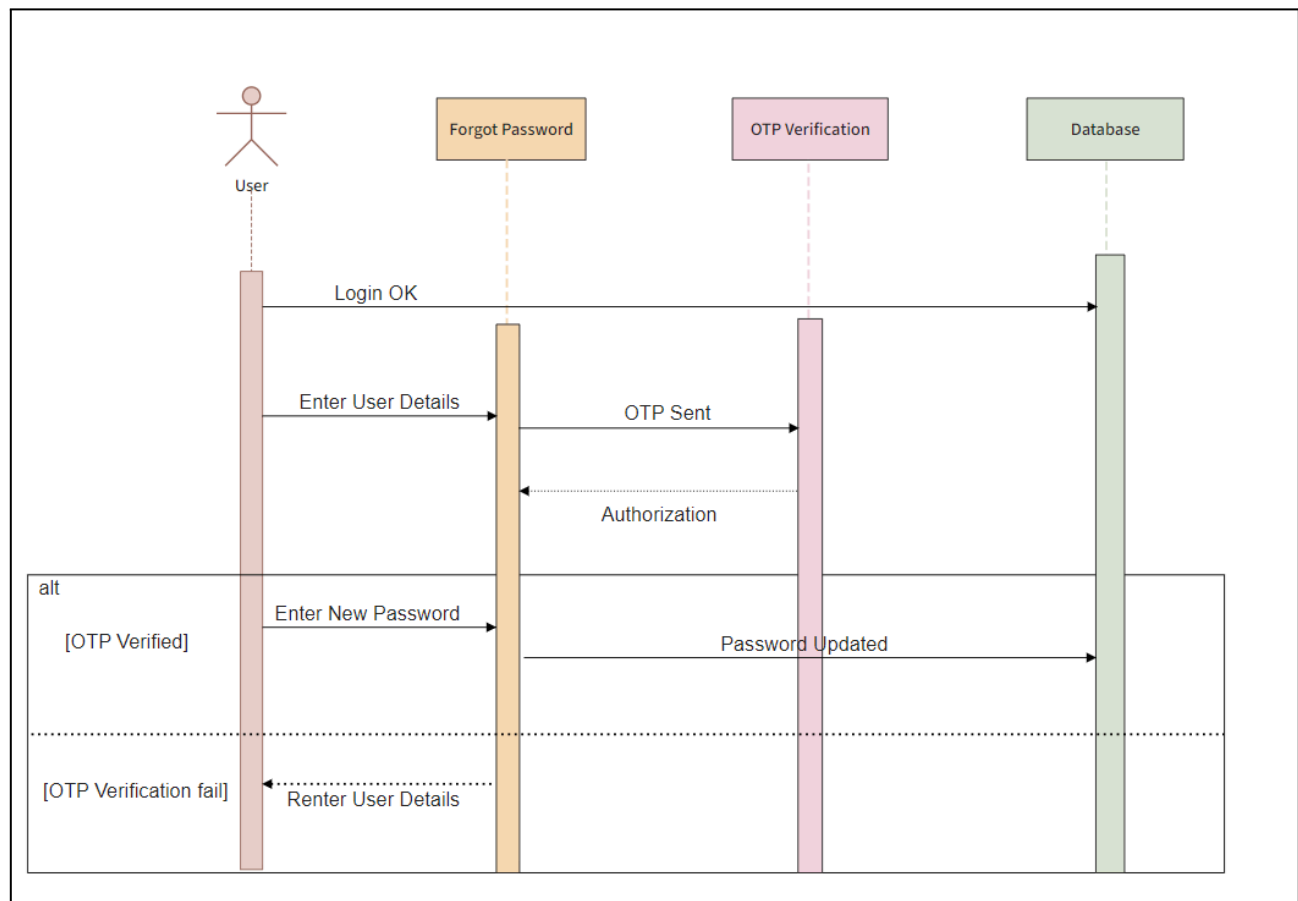
Login:



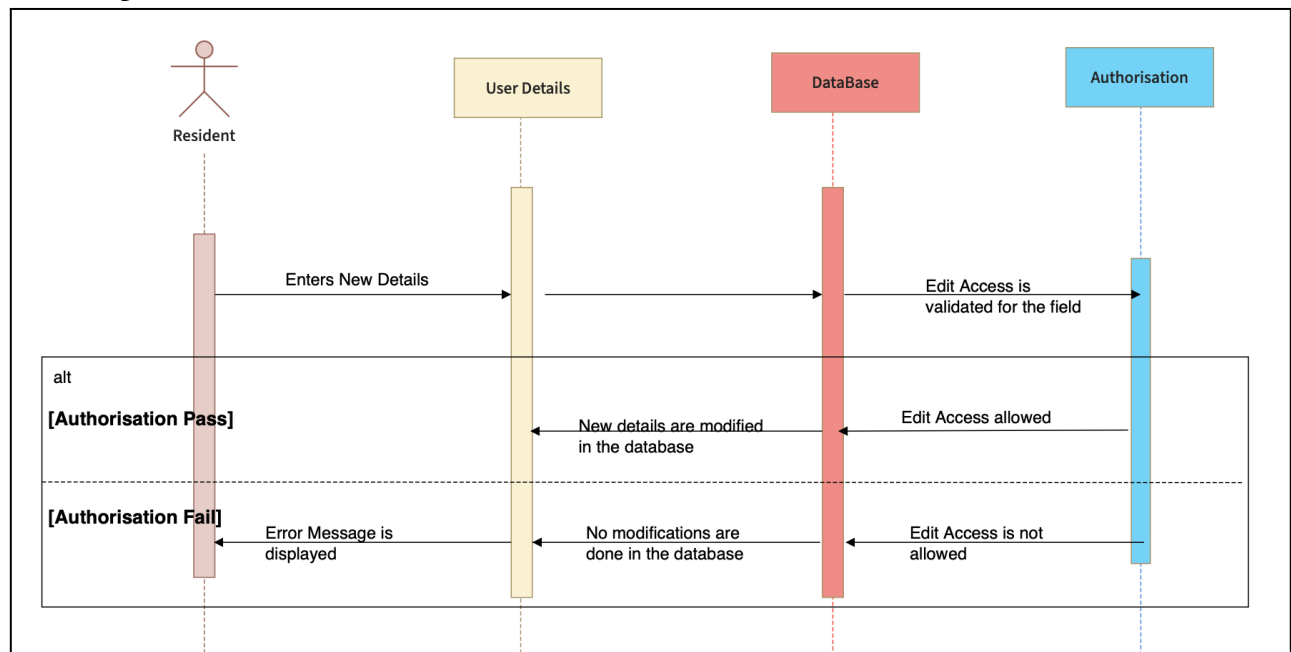
Sign Up:



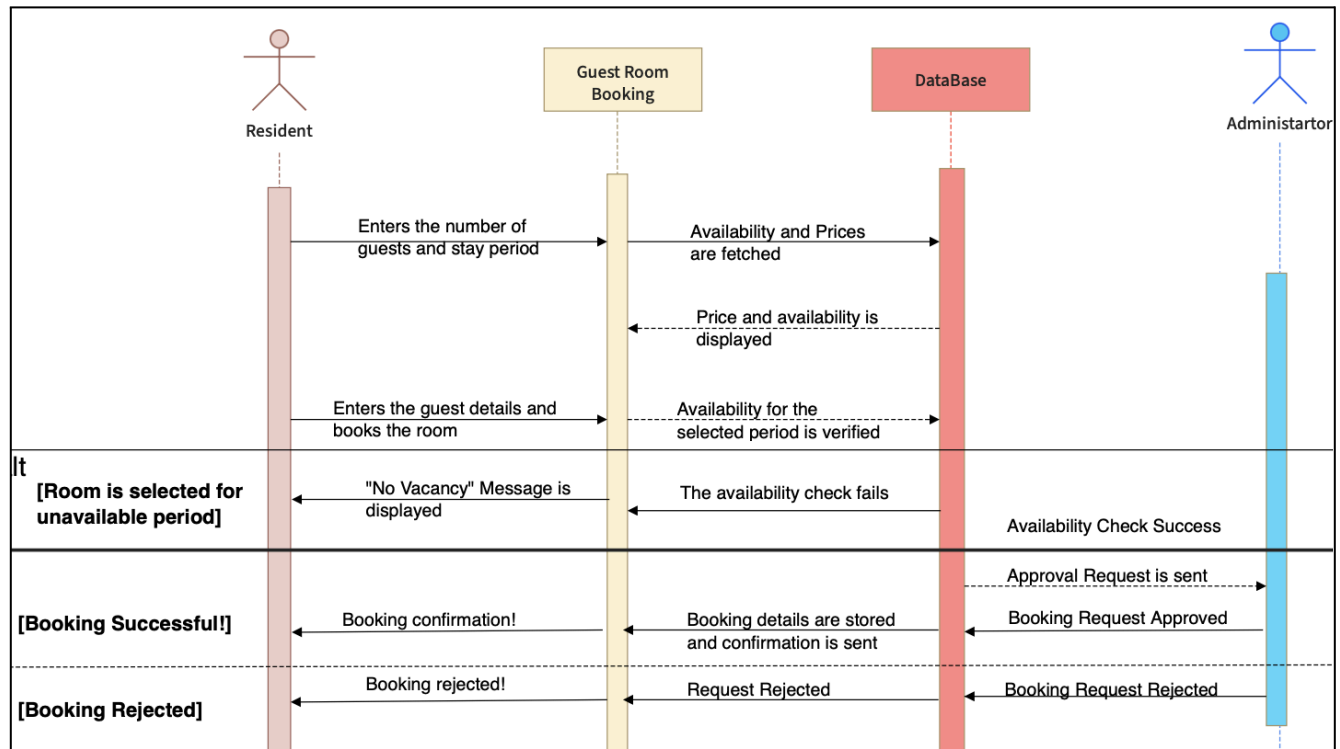
Forgot Password:



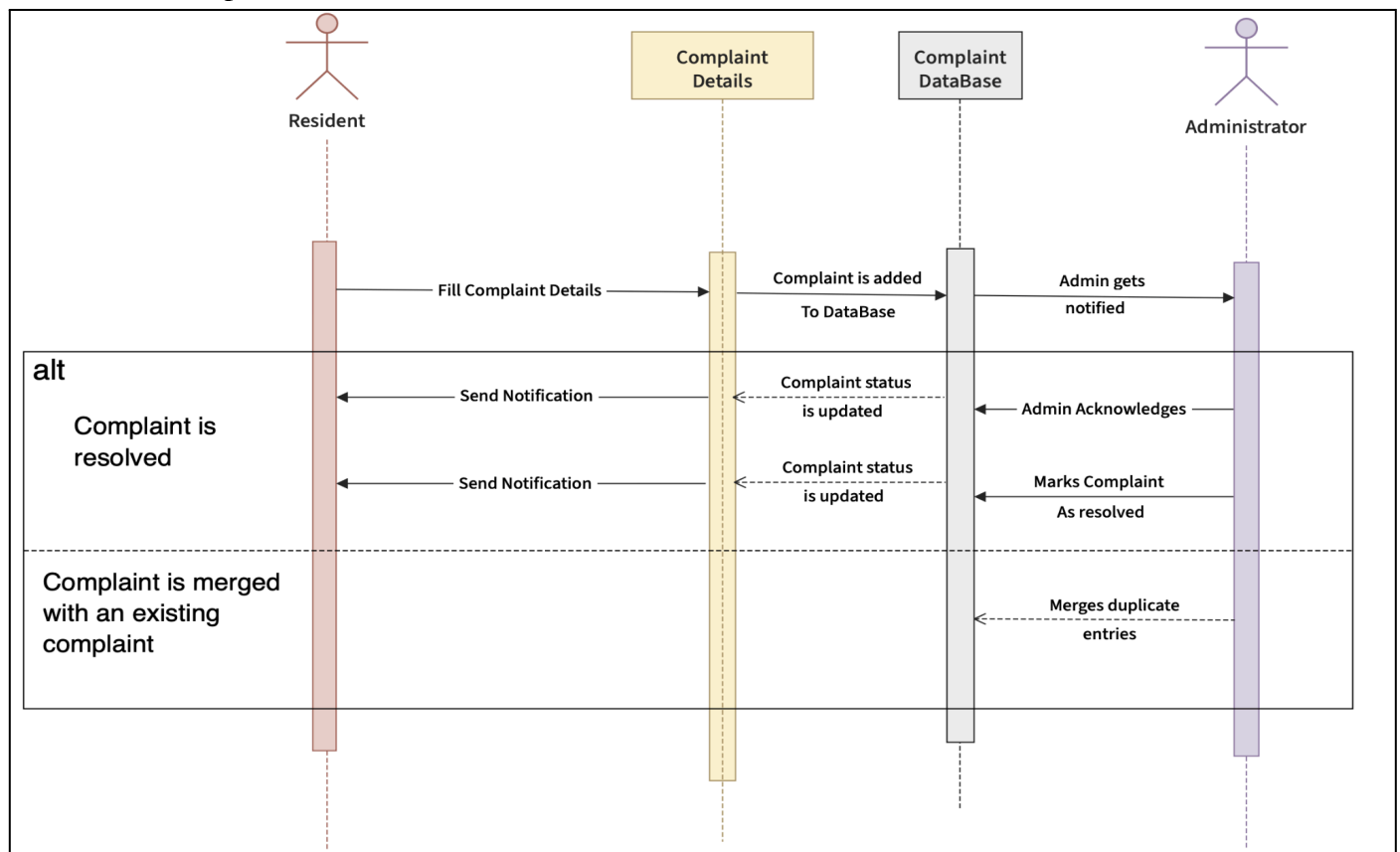
Profile Updation:



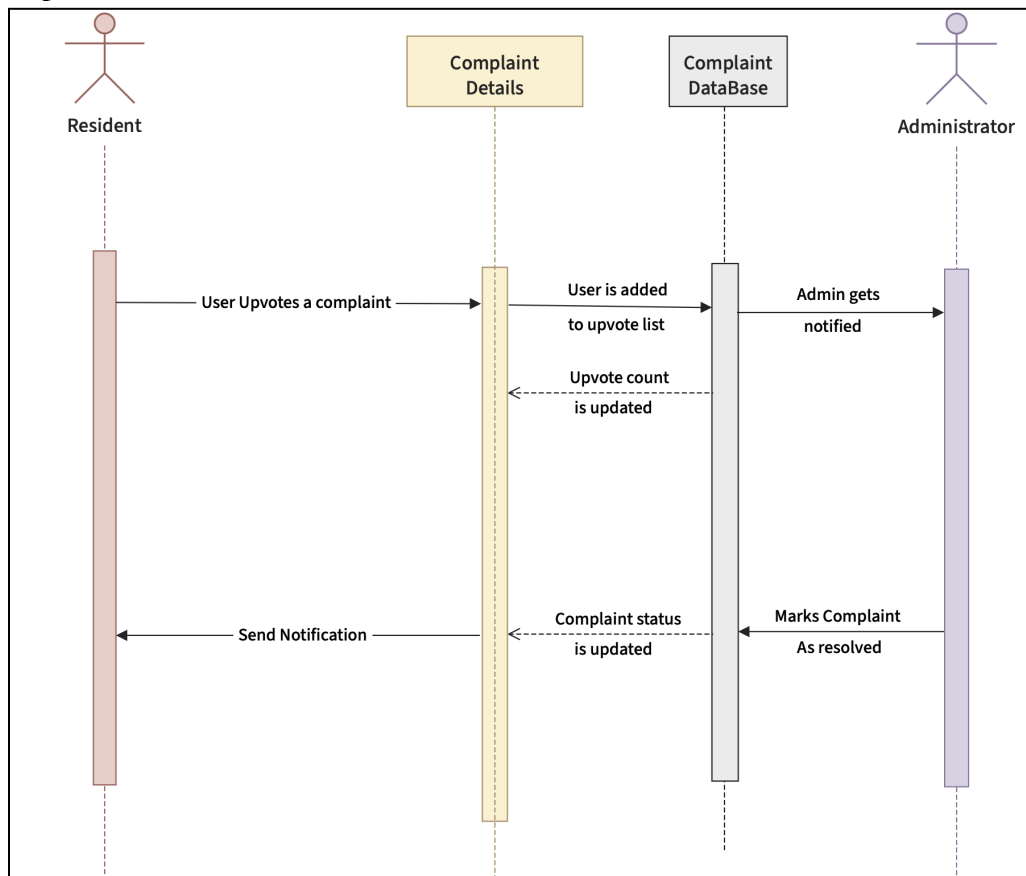
Guest Room Booking:



Create New Complaint:

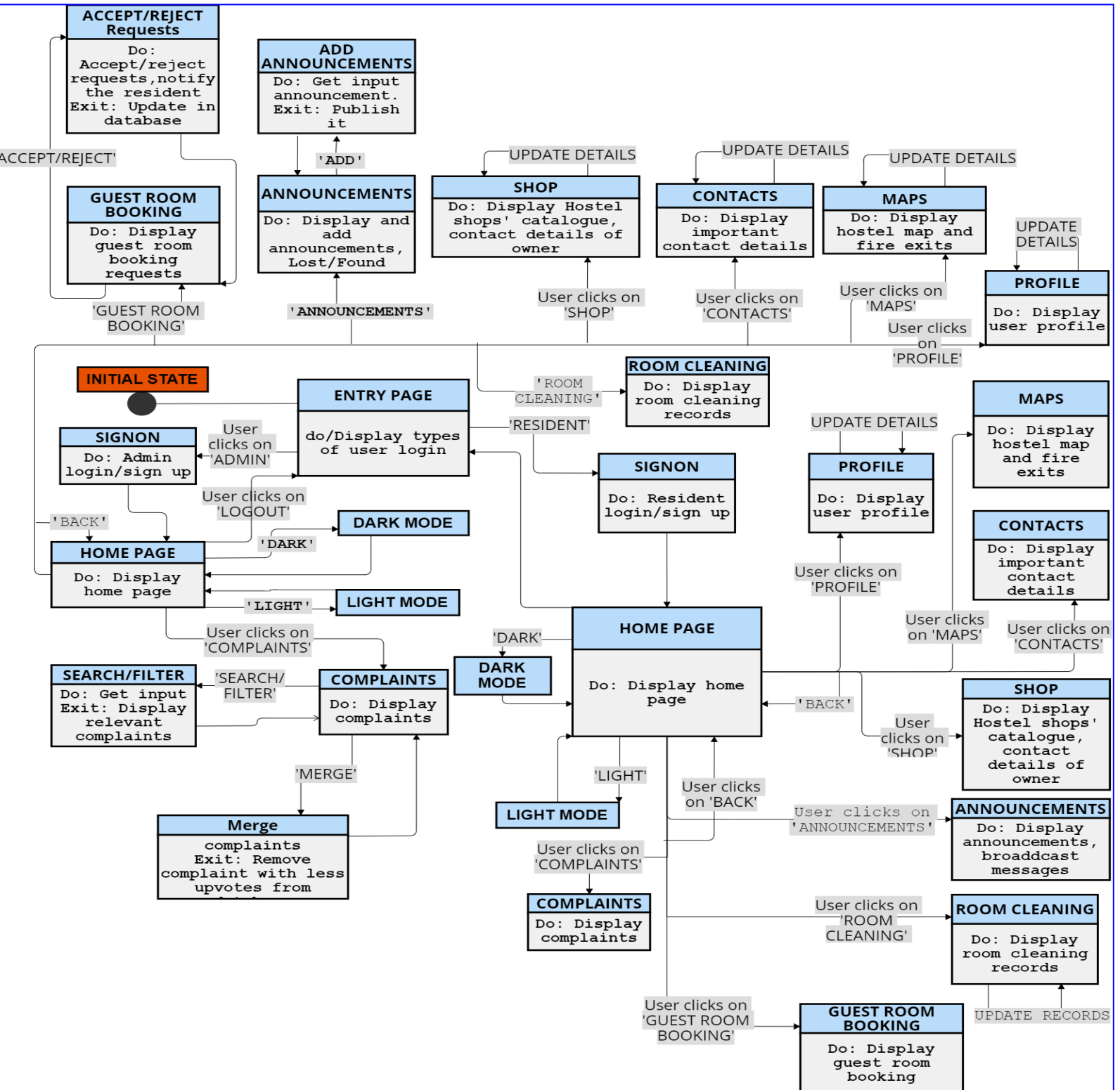


Upvote Complaint:

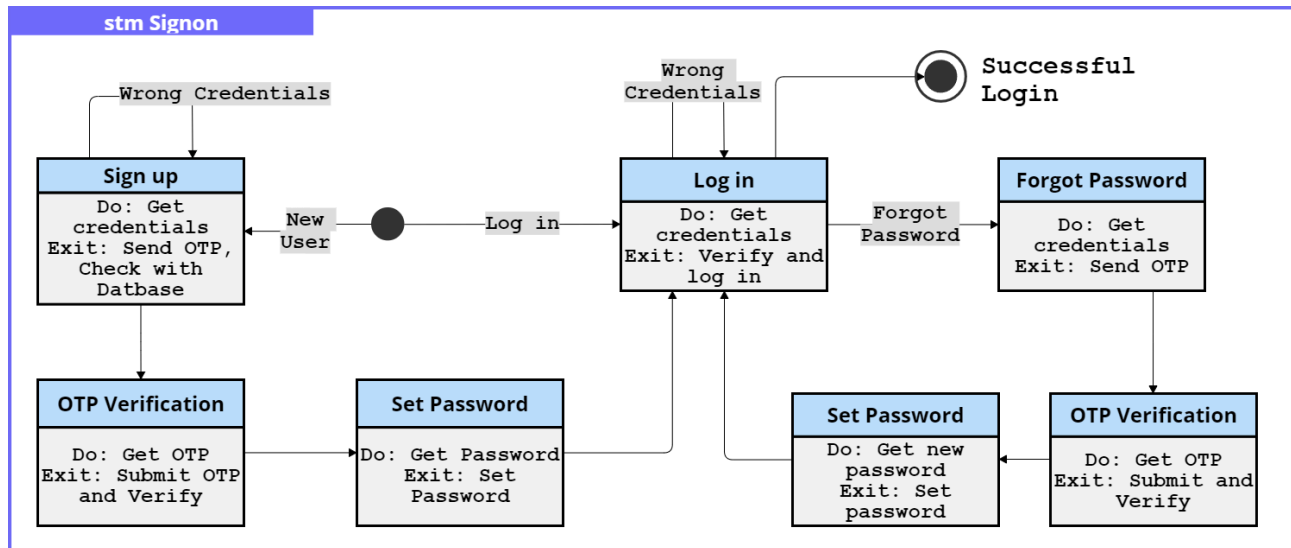


3.4 State Diagrams

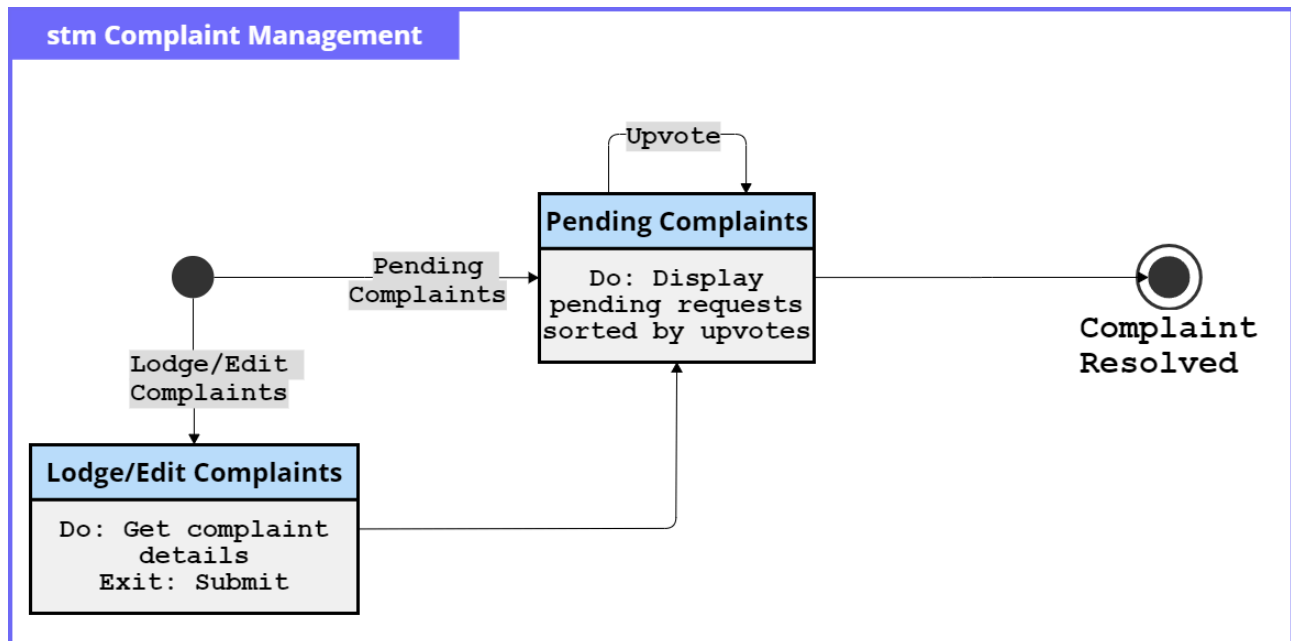
Overall state diagram:



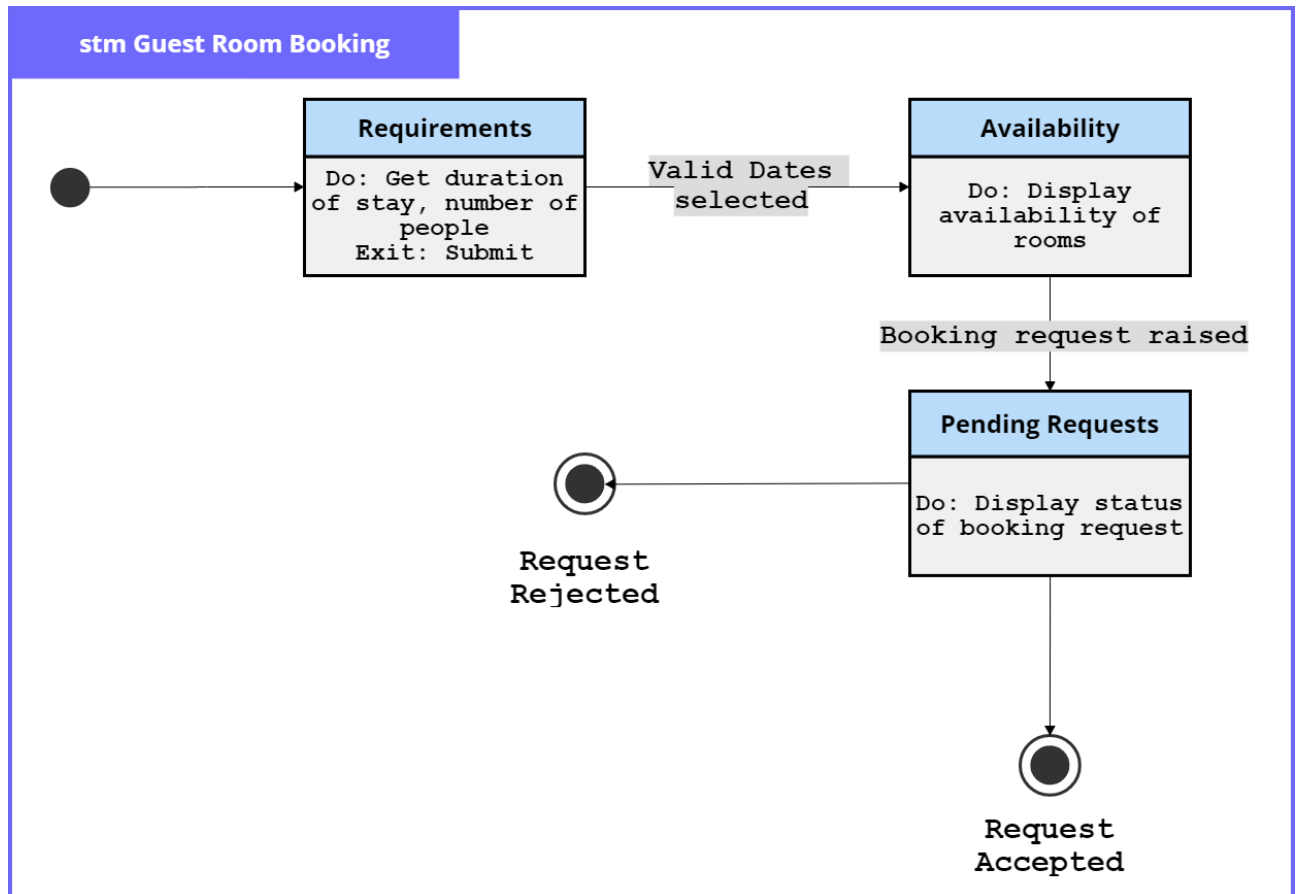
1. Signon



2. Complaint Management



3. Guest Room Booking



4. Project Plan

Timeline of development :

1. 12th January 2024 - 26th January 2024: Requirement generation, Analysis and Documentation of Software Requirements Specification (SRS).
2. 27th January 2024 - 9th February 2024: Documentation of Software Design Document
3. 9th February 2024 - 15th March 2024: Implementation, unit testing, integration testing
4. 15th March 2024 - 29th March 2024: System testing, manual preparation for beta testing
5. 29th March 2024 - 12th April 2024: : Code Improvement and Beta Testing:- With the results of the previous testing sessions, improvements on the software would be done to increase speed, security and its overall quality
6. 12th April 2024 - 19th April 2024: Addressing Beta Testing Feedback and Delivering the Final Project Report.

Broad classification of the tasks along with the team members working on it are :-

1. **Front End Design:** A front-end designer not only designs the look and layout of a product but also translates these designs into code. Any interactive element on a website that users can engage with falls under the category of front-end development.

The members responsible for the front-end design are:

- Ayush
- Ritesh Hans
- Mrdul Agarwaal
- Rohan Batra

The following software might be useful for Front End Development:-

a. **React JS:** A free and open-source front-end JavaScript library for building user interfaces based on UI components.

b. **JavaScript:** JavaScript, often abbreviated JS, is a programming language that is one of the core technologies of the World Wide Web, alongside HTML and CSS.

c. **HTML:** A standard markup language for documents designed to be displayed in a web browser.

d. **CSS:** Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language such as HTML.

1. **Back End Design:** Anything that is not directly visible to the user and contains and processes the data behind the scene comes under the back-end. It focuses mainly on APIs, backend logic, databases etc.

The members responsible for the back-end development are:

- Apoorv Tandon
- Tanishq Maheshwari

- Taneshq Zendey
- Samarpan Verma
- Krutuparna Paranjape
- Mridul Gupta

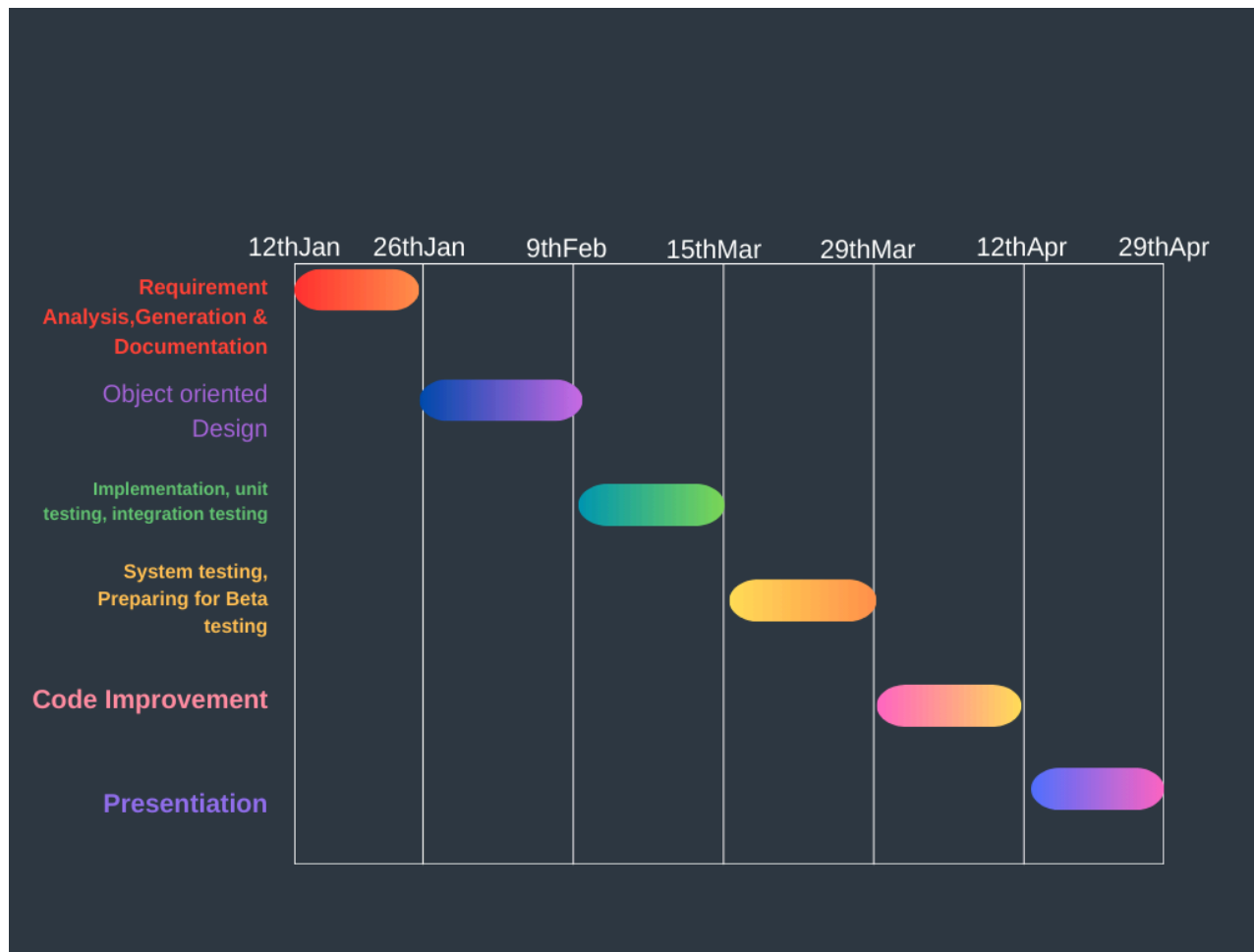
The following software might be useful for Back End Development:-

a. **JavaScript:** Can be used for both Front and Back End Development.

b. **MySQL:** MySQL is an open-source relational database management system.

c. **Node.JS:** Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside a web browser.

d. **MongoDB:** MongoDB is a free and open-source document-oriented database that is very much popular among web developers.



Appendix A - Group Log

<Please include here all the minutes from your group meetings, your group activities, and any other relevant information that will assist in determining the effort put forth to produce this document>

All the group members were in constant touch with each other and the TA through the WhatsApp group. Suggestions were given by the TA regarding some corrections and elaboration in the architecture diagram, class diagrams, and state diagram. All those suggestions have been seriously incorporated into the document. Work was divided and team-meetings were held regularly.

Date	Members Present	Topic Of Discussion
29/01/2024	All group members	Collaboratively discussed the appearance of user interfaces and the required features for each webpage.
02/02/2024	All group members	Distributed work on context, use case, class, sequence, and state diagrams.
04/02/2024	TA & All group members	Addressed and resolved TA's questions and suggestions regarding our software architecture, determining suitable classes for inclusion in the class diagram, and clarifying uncertainties related to the state diagram.
08/02/2024	All group members	Examined diagrams and interfaces to ensure consistency of features, functions, and attributes across the design document.