

Contents

| | | |
|----------|---|-----------|
| 1 | Introduction | 4 |
| 1.1 | Goals and Objectives of the Project | 4 |
| 1.2 | Scope of the Work | 5 |
| 1.2.1 | Current Situation and Context | 5 |
| 1.2.2 | Competing Products (Available in Market) | 5 |
| 1.3 | System Overview | 5 |
| 1.3.1 | Homeowner Panel | 5 |
| 1.3.2 | Student Panel | 6 |
| 1.3.3 | Service Provider Panel | 6 |
| 1.4 | Structure of the Document | 6 |
| 2 | Project Management Plan | 7 |
| 2.1 | Project Organization | 7 |
| 2.1.1 | Individual Contribution to the Project | 7 |
| 2.2 | Process Model Used | 7 |
| 2.2.1 | Rationale for Choosing Life Cycle Model | 8 |
| 2.3 | Constraints to Project Implementation | 9 |
| 2.4 | Hardware and Software Resource Requirements | 9 |
| 2.4.1 | Tools/Language Requirements | 9 |
| 2.5 | Project Timeline and Schedule | 10 |
| 2.6 | Social/Cultural/Environmental Impact of the Project | 10 |
| 3 | Requirement Specifications | 12 |
| 3.1 | Stakeholders for the System | 12 |
| 3.2 | Use Case Diagram with Graphical and Textual Description . . | 13 |
| 3.2.1 | Use Case diagram for User Profile Management | 13 |
| 3.2.2 | Use Case diagram for Post Property | 15 |
| 3.2.3 | Use Case diagram for Book Property | 17 |
| 3.3 | Development of SRS (Software Requirement Specification) . . | 19 |
| 3.4 | Development of Use Case Template | 19 |
| 3.5 | Activity Diagram | 19 |

| | | |
|----------|---|-----------|
| 3.5.1 | Activity diagram for User Profile Management | 20 |
| 3.5.2 | Activity diagram for Post Property | 21 |
| 3.5.3 | Activity diagram for Book Property | 22 |
| 3.6 | Static Model – Class Diagram | 23 |
| 3.7 | Dynamic Model – Sequence Diagram | 24 |
| 3.7.1 | Sequence diagram for User Profile Management | 24 |
| 3.7.2 | Sequence diagram for Post Property | 25 |
| 3.7.3 | Sequence diagram for Book Property | 26 |
| 3.8 | Swimlane diagram | 27 |
| 3.8.1 | Swimlane diagram for User Profile Management | 27 |
| 3.8.2 | Swimlane diagram for Post Property | 28 |
| 3.8.3 | Swimlane diagram for Book Property | 29 |
| 3.9 | Safety and Security Requirements | 30 |
| 3.9.1 | Access Requirements | 30 |
| 3.9.2 | Integrity Requirements | 30 |
| 3.9.3 | Privacy Requirements | 30 |
| 4 | Architecture | 31 |
| 4.1 | Data Centered Architecture for Overall System | 31 |
| 4.1.1 | Rationale for Choosing Data Centered Style | 32 |
| 4.2 | MVC Architecture for the System | 32 |
| 4.2.1 | MVC for User Management System | 33 |
| 4.2.2 | MVC for Posting System | 33 |
| 4.2.3 | MVC for Booking System | 33 |
| 4.2.4 | Rationale for Choosing MVC Architecture | 33 |
| 4.3 | Technology, software, and hardware used | 33 |
| 5 | Design | 35 |
| 5.1 | Component level design pattern | 35 |
| 5.1.1 | Home Reservation | 36 |
| 5.1.2 | Property Management | 36 |
| 5.1.3 | User Profile Management | 37 |
| 5.2 | Dataflow Diagram (DFD) | 38 |
| 5.2.1 | First Level DFD for the System | 38 |
| 5.2.2 | DFD for Booking Property | 39 |
| 5.2.3 | DFD for Booking Property | 40 |
| 5.2.4 | DFD for Post Property Management | 40 |
| 5.2.5 | DFD for User Profile Management | 41 |
| 5.3 | Entity Relationship Diagram (ERD) | 42 |
| 5.4 | UX Design | 44 |
| 5.4.1 | Landing Page | 44 |

| | | |
|----------|---|-----------|
| 5.4.2 | Authentication | 45 |
| 5.4.3 | User Dashboard | 47 |
| 6 | Testing and Sustainability Plan | 49 |
| 6.1 | Requirements/Specifications-based System Level Test Cases . | 49 |
| 6.2 | Techniques used for test generation | 53 |
| 6.3 | Assessment of the goodness of your test suite | 53 |
| 7 | Acknowledgement | 54 |

Chapter 1

Introduction

The introduction provides a comprehensive overview of the Rental Home Management System, highlighting its purpose and key features. It introduces the system as a solution designed to streamline the home rental process for students and homeowners within the university community.

The introduction emphasizes the system's core modules, including User Management, Property Listing, Booking Management, and Task Control. These modules work together to facilitate efficient property searching, booking, and management.

The system's user-friendly interface and accessibility through web browsers are highlighted, emphasizing its convenience for both students and homeowners. The introduction concludes by emphasizing the system's role in fostering a harmonious and supportive living environment within the university community.

1.1 Goals and Objectives of the Project

The primary goal of the Rental Home Management System is to optimize and elevate the management of home rental services, establishing a user-friendly platform for students to effortlessly submit property requests, monitor their progress, and obtain prompt resolutions. Our system strives to enhance communication channels and streamline task management between students, homeowners, and service providers, ultimately fostering a well organized and supportive living environment for the university community.

1.2 Scope of the Work

1.2.1 Current Situation and Context

The Rental House Management System near CUET addresses the challenges of managing rental properties efficiently. It simplifies interactions between homeowners, students, and administrators through features like user management, property listings, and booking management. Additional services such as property maintenance, cleaning, and utilities are seamlessly integrated. The system enhances issue resolution with complaint category management and task tracking, ensuring a responsive and user-friendly experience tailored to the rental community's needs.

1.2.2 Competing Products (Available in Market)

The Rental House Management System near CUET stands out in a market crowded with generic rental platforms by catering specifically to the needs of the CUET community. Unlike competing products, it offers tailored features like student-focused property listings, streamlined booking management, and integrated services such as maintenance and utilities. Some well-known platforms include:

- Focused on meeting the specific needs of the CUET community.
- Provides student-centric property listings and tailored rental solutions.
- Integrated services like property maintenance, cleaning, and utilities.
- Offers a localized, user-friendly, and comprehensive platform.

1.3 System Overview

Rental Home Management System is a robust platform tailored for streamlined home rental service management. It comprises three primary panels. Homeowner, Student, and Administrator ensuring efficient coordination and swift issue resolution.

1.3.1 Homeowner Panel

Central Control Hub: Overseeing all activities within the platform.

Property Management: Homeowners can create and manage property listings.

Booking Approval: Review and approve student booking requests and manage property availability.

Task Assignment: Assign tasks to service providers based on availability, skills, and workload.

1.3.2 Student Panel

Booking Request Submission: Students can submit property booking requests with details and urgency levels.

Notification System: Receive alerts for booking approval and property availability updates.

1.3.3 Service Provider Panel

Task List: View assigned tasks with details and priority.

Task Completion: Confirm task completion with notes or comments.

Record Keeping: Administrators maintain task records and generate reports for efficient tracking.

This system overview ensures a cohesive and user-friendly experience for all stakeholders involved in the home rental process, promoting effective communication and task management.

1.4 Structure of the Document

This document is structured as follows:

- Chapter 1 introduces the project and defines the goals, objectives, scope, and system overview.
- Chapter 2 details the project management plan, including project organization, individual contribution to the project, Process model used, Rational for choosing lifecycle model, Risk Analysis, Constraints, Resources, Timeline and schedule,
- Chapter 3 outlines the system requirements specifications, including use case diagrams, activity diagrams, class diagram, sequence diagrams, swimlane diagrams and security requirements.

Chapter 2

Project Management Plan

2.1 Project Organization

Efficient project organization plays a vital role in software development, involving the distribution of specialized tasks among team members according to their strengths and capabilities. In a three-person team, the project is segmented among members, taking into account the time needed for each stage and leveraging the unique strengths of each individual. The objective is to establish a pragmatic organization that optimizes the likelihood of project success.

2.1.1 Individual Contribution to the Project

The project will be developed by a team of two members, each contributing to different aspects of the project:

- **Member 1:** Responsible for front-end development and UI/UX design.
- **Member 2:** Handles back-end development, database design, and server-side logic.

2.2 Process Model Used

In the development of our home rental system, we employ the Prototype Process Model to ensure agility, rapid iteration, and refinement of the application.

This paradigm enables quick planning, iterative development, and constant feedback from stakeholders, ensuring a user-centric prototype.

- Communication
- Quick Plan
- Quick Design
- Construction of Prototype
- Deployment and Feedback

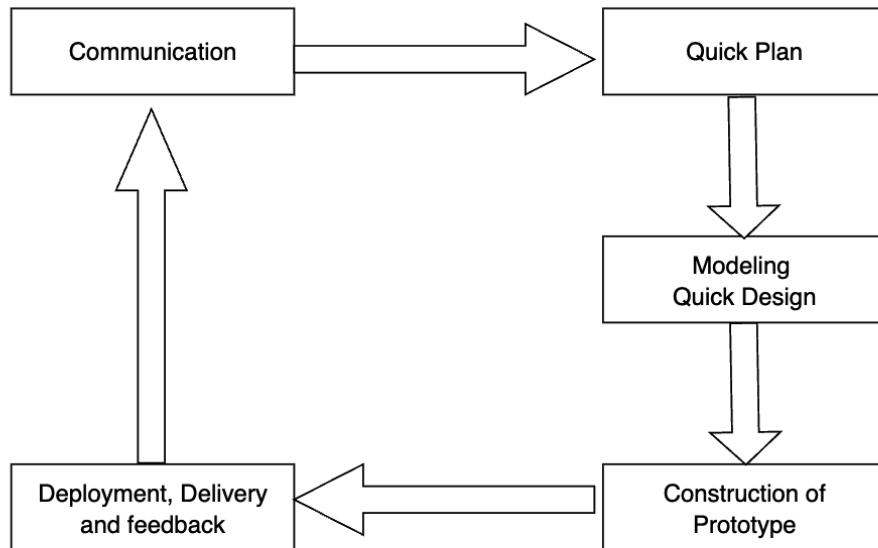


Figure 2.1: Prototyping Process Model.

2.2.1 Rationale for Choosing Life Cycle Model

The selection of the Prototype Process Model for our home rental system development is grounded in several key considerations:

- **Iterative Development:** Our home rental system embraces prototyping, allowing flexibility to adapt and refine based on evolving requirements.
- **User-Centric Design:** Prioritizing an optimal user experience by engaging potential users early, ensuring an intuitive and user-friendly interface.
- **Feedback and Involvement:** Regular stakeholder feedback, including students, homeowners, and administrators, ensures active involvement and aligns the system with diverse user needs.
- **Flexibility to Changes:** Adaptable to dynamic market trends and user feedback, the prototype process model accommodates changes effectively.
- **Continuous Improvement:** iterative nature promotes continuous enhancement, allowing seamless incorporation of new features and optimizations based on lessons learned.
- **Risk Mitigation:** Early user feedback in the prototyping phase helps effectively mitigate risks related to misunderstandings, unclear requirements, or changes in stakeholder expectations.

2.3 Constraints to Project Implementation

The major constraints include:

- **Schedule Constraints:** Limited time for development and testing.
- **Budget Constraints:** Limited resources for purchasing third-party services.
- **Technical Constraints:** Need for integration with existing university systems.

2.4 Hardware and Software Resource Requirements

2.4.1 Tools/Language Requirements

- **Front-End:** HTML, CSS, JavaScript, ReactJS.

- **Back-End:** Node.js, Express.
- **Database:** MongoDB
- **Tools:** Git, GitHub for version control, and VS Code for development.

2.5 Project Timeline and Schedule

The project is expected to be completed within a 6-month period. The timeline includes:

- Month 1: Requirement gathering, initial design, and planning. Front-end and back-end development.
- Month 2: Database integration and system testing.
- Month 3: Deployment, documentation, and user testing.
- Month 4: Final report and presentation.

2.6 Social/Cultural/Environmental Impact of the Project

The implementation of this system will have several positive impacts:

1. Social Impact:

- Convenience and Accessibility: Rental House Management System provides easy access to a variety of rental properties, catering to diverse housing preferences, enhancing convenience for users.
- Opportunities: Creates economic opportunities for property owners, landlords, and real estate professionals, contributing to income sources within the real estate industry.
- Community Well-being: Fosters community well-being by connecting property seekers with suitable homes, contributing to a sense of belonging and stability within the CUET community.

2. Cultural Impact:

- Housing Diversity: Introduces users to a variety of housing options, promoting diversity in residential living experiences.

- Preservation of Local Housing Styles: Plays a role in preserving and promoting regional architectural traditions by featuring local housing styles and preferences

3. Environmental Impact:

- Reduced Commuting and Emissions: Contributes to reducing the need for extensive property visits, leading to decreased commuting and lower carbon emissions associated with property hunting. Practices: Encourages property owners to adopt sustainable practices, minimizing the environmental impact of residential properties listed on the platform. Waste Reduction: Minimizes paper-based processes, promotes digital transactions, and encourages eco-friendly property management practices, contribution.

Chapter 3

Requirement Specifications

3.1 Stakeholders for the System

The primary stakeholders of the system include:

Student

- Primary users of the Rental Home Management near CUET platform, seeking suitable accommodation options near the CUET campus.
- Require easy access to property listings, reservation functionality, and communication channels to connect with homeowners.

Homeowner

- Property owners offering accommodation options to students near the CUET campus.
- Utilize the platform to list their properties, manage bookings, and communicate with potential renters.

Developers

- Responsible for the development, maintenance, and enhancement of the Rental Home Management near CUET platform.
- Ensure the platform's functionality, security, and scalability to meet the needs of stakeholders.

3.2 Use Case Diagram with Graphical and Textual Description

3.2.1 Use Case diagram for User Profile Management

Textual Description of Use Case

Actors: Student,Landlord are primary actors and Database are secondary actor. **Use Cases :**

1. View Booked Lists:
 - Landowners and students can access this use case.
 - It allows users to view lists that have been booked or reserved within the system.
2. View Total Posts:
 - Both landowners and students can use this functionality.
 - It enables users to see the total number of posts or listings within the system.
3. Manage Profile:
 - This use case is accessible to both landowners and students.
 - It provides users with the ability to manage their own profiles, possibly including updating personal information or preferences.
4. Edit Profile:
 - Again, accessible to both landowners and students.
 - This allows users to modify specific details within their profiles.
5. Upload Profile Picture:
 - Both landowners and students can utilize this feature.
 - It enables users to upload a profile picture to their account.

Graphical Description of Use Case

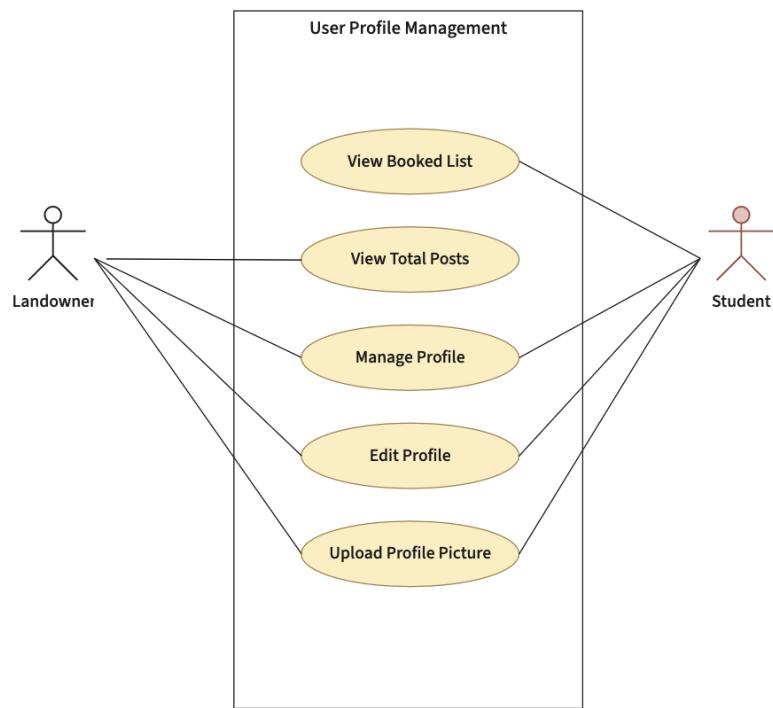


Figure 3.1: Use case diagram for user profile management.

3.2.2 Use Case diagram for Post Property

Textual Description of Use Case

Actors: Landlord is primary actor and Admin are secondary actor.

Use Cases :

1. SignUp:
 - Landlords can access this use case.
 - It allows new users to create an account within the system.
2. Login:
 - This use case is accessible to both existing landlords and new users who have signed up.
 - It enables users to log into their accounts.
3. Post Property:
 - This is the core use case for landlords.
 - It allows them to create listings for their properties.
4. Upload Photos:
 - This is included within the "Post Property" use case.
 - Landlords can upload photos of their property to enhance the listing.
5. Property Description:
 - This is also included within the "Post Property" use case.
 - Landlords can provide detailed descriptions of their properties.
6. Edit Post:
 - This use case enables landlords to modify existing property listings.

Graphical Description of Use Case

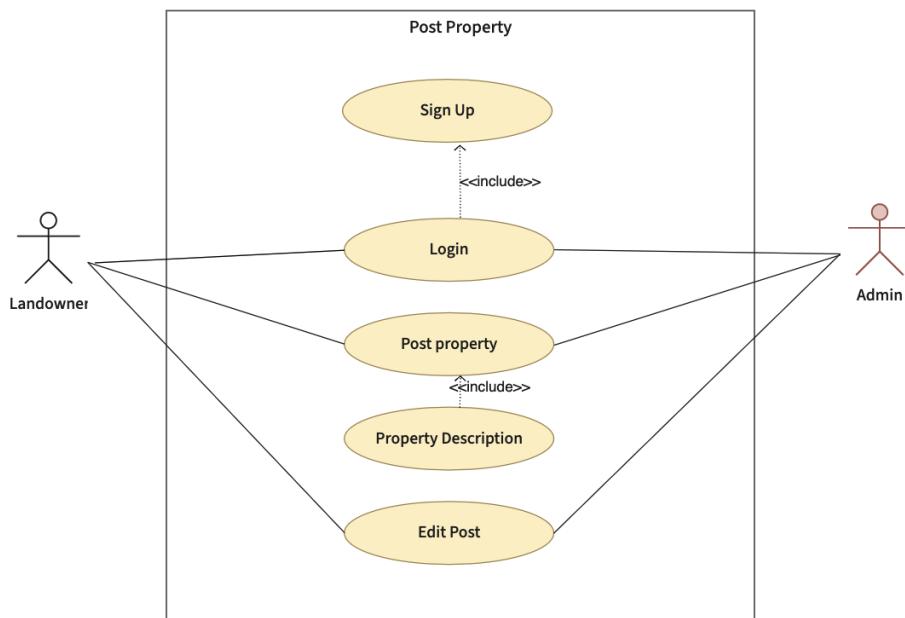


Figure 3.2: Use case diagram for Post Property.

3.2.3 Use Case diagram for Book Property

Textual Description of Use Case

Actors: Landowner and customer are primary actors and Admin is secondary actor.

Use Cases :

1. SignUp:
 - Customers and Landowners can access this use case.
 - It allows new users to create an account within the system.
2. Login:
 - This use case is accessible to both existing customers, landowners, and new users who have signed up.
 - It enables users to log into their accounts.
3. View Property:
 - Customers can access this use case.
 - It allows customers to view available properties for booking.
4. Book Property:
 - This is the core use case for customers.
 - It allows them to book a property.
5. Update Status:
 - This use case is accessible to landowners.
 - It enables landowners to update the status of a property (e.g., booked, available).
6. Show Notification:
 - This use case is included within the "Update Status" use case.
 - Landowners can send notifications to customers regarding property status updates.

Graphical Description of Use Case

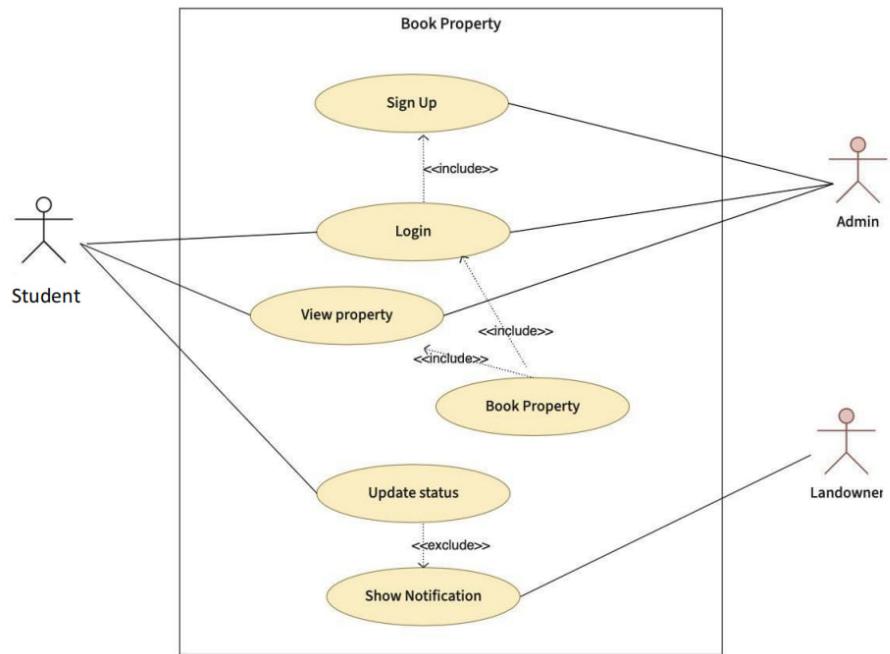


Figure 3.3: Use case diagram for Book Property.

3.3 Development of SRS (Software Requirement Specification)

The Software Requirement Specification document will cover:

- Functional requirements, including user interfaces, and event management features.
- Non-functional requirements like performance, scalability, and security.
- System interfaces and data flow.

3.4 Development of Use Case Template

The use case template will include the following elements:

- Use case name
- Description of the functionality
- Actors involved
- Preconditions and postconditions
- Normal and alternate flows

3.5 Activity Diagram

An activity diagram illustrating the flow of activities for event creation and registration will be created. This diagram will represent the steps taken by users and system interactions.

3.5.1 Activity diagram for User Profile Management

The activity diagram for user profile management delineates steps from user authentication to role-based functionalities. Landowners manage profiles, update listings, and finalize deals, while students maintain profiles, search for properties, negotiate, and book.

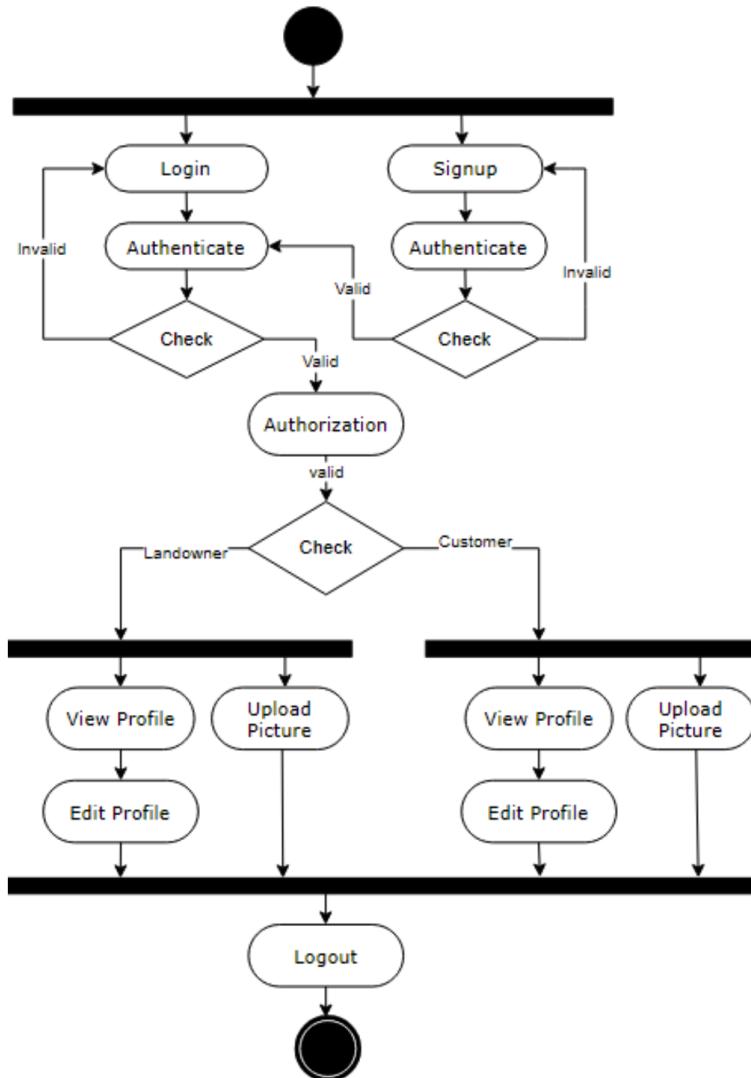


Figure 3.4: Activity diagram for user profile operation

3.5.2 Activity diagram for Post Property

The activity diagram for posting a property outlines the sequential steps involved in the process, from authentication to the completion of property listing. This diagram represents the activities performed by both landowners and students, capturing their distinct roles in the property posting process.

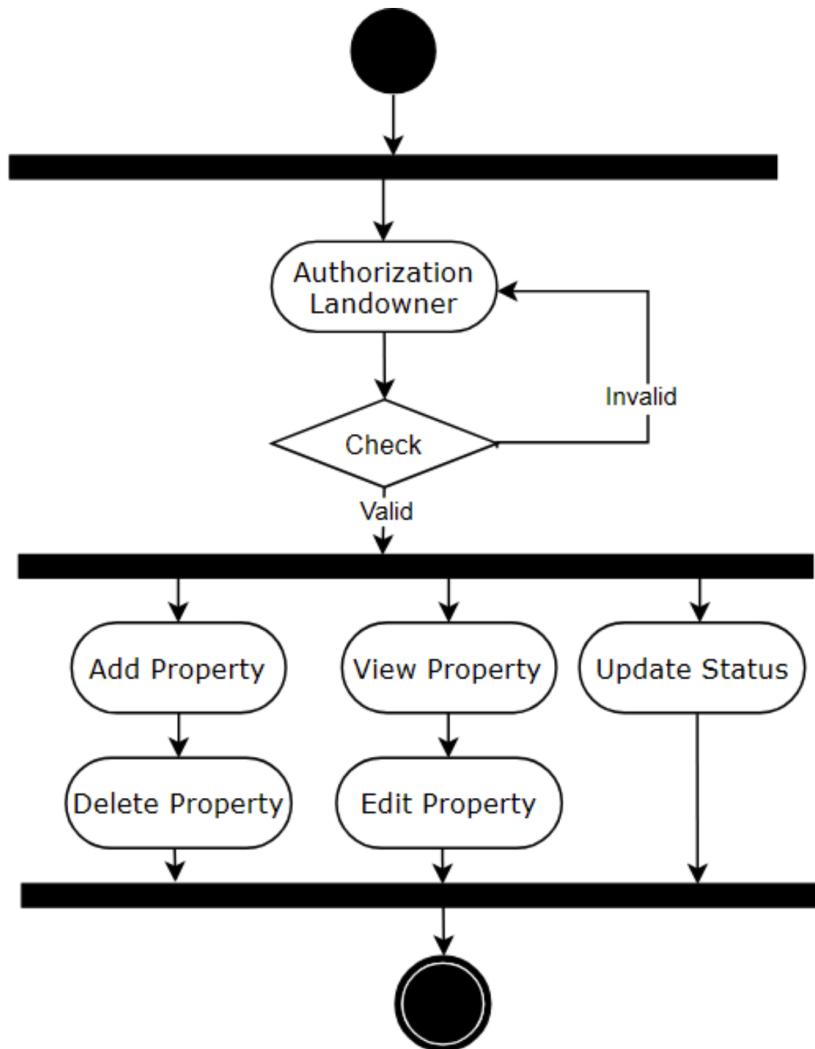


Figure 3.5: Activity diagram for Post Property operation

3.5.3 Activity diagram for Book Property

An activity diagram for "Book Property" visually outlines the workflow involved in securing a rental. It demonstrates the user's activities: initiating a property search, selecting a listing, entering booking details, and proceeding to payment. Simultaneously, it illustrates system actions: verifying availability, processing payments, confirming reservations, and sending notifications. The diagram clarifies the sequential steps and decision points in the booking process.

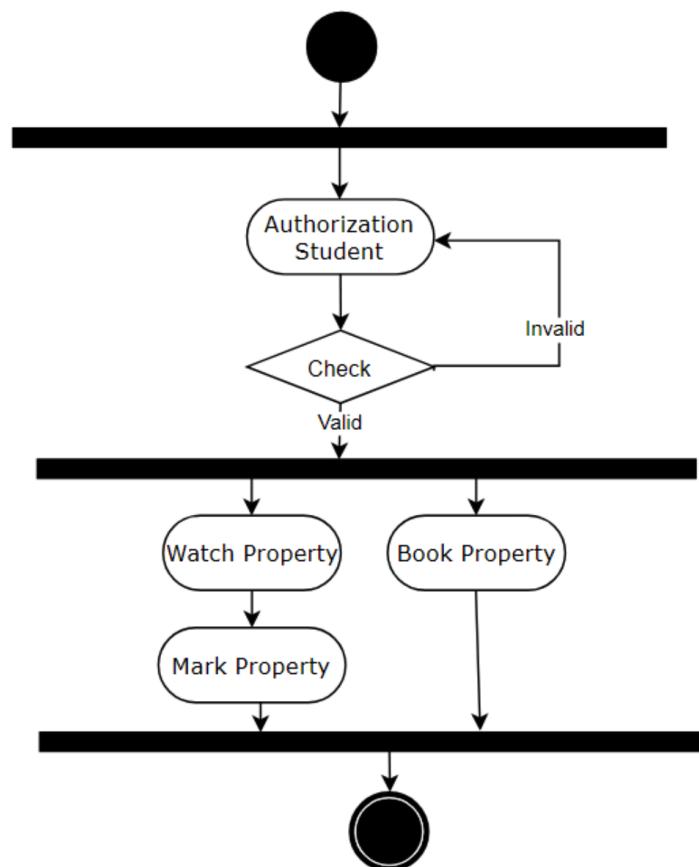


Figure 3.6: Activity diagram for book Property operation

3.6 Static Model – Class Diagram

Class diagram visually represents the static structure of a system, depicting relationships and attributes of classes. The system diagram of Rental Home Management Near CUET is given below drawn according to the reference of book chapter-8 Requirements Modeling.

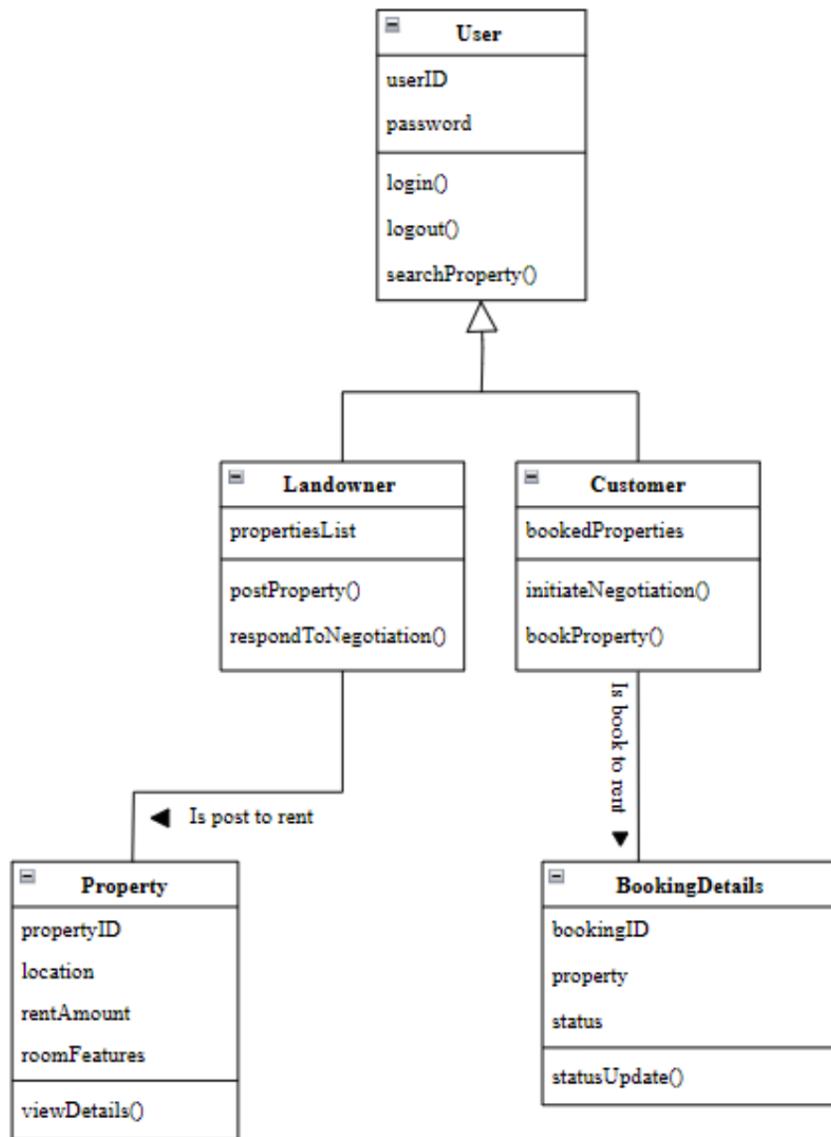


Figure 3.7: Class diagram for Rental Home Management System near CUET

3.7 Dynamic Model – Sequence Diagram

3.7.1 Sequence diagram for User Profile Management

In the sequence diagram for user profile management, the process initiates with a user attempting authentication through login or sign-up. Upon successful authentication, the system verifies permissions and grants access to role-specific functionalities. Landowners engage in managing profiles, updating listings, and negotiating deals, while students access profiles, search properties, negotiate terms, and book. Throughout, the system facilitates secure communication and offers landowners tools for property management, ensuring a seamless user experience.

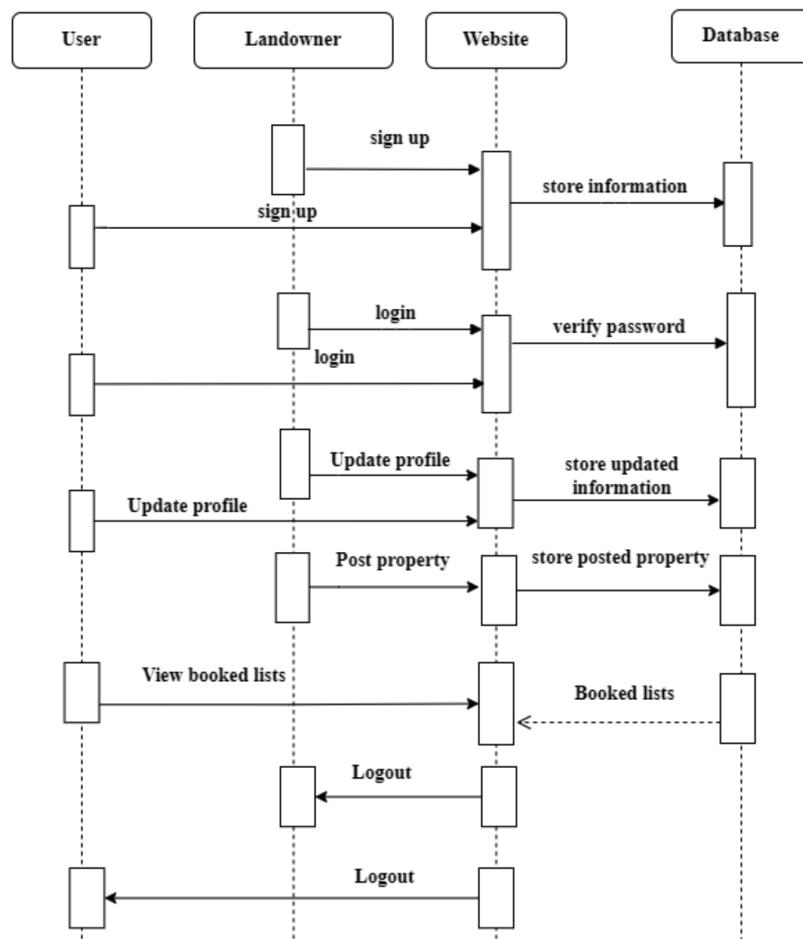


Figure 3.8: Sequence diagram for user profile management.

3.7.2 Sequence diagram for Post Property

The sequence diagram for posting a property illustrates the dynamic interactions between users, the authentication process, and the system functionalities involved in creating a property listing. The sequence begins with a user attempting authentication through login or sign-up and proceeds through role-specific actions for both landowners and students.

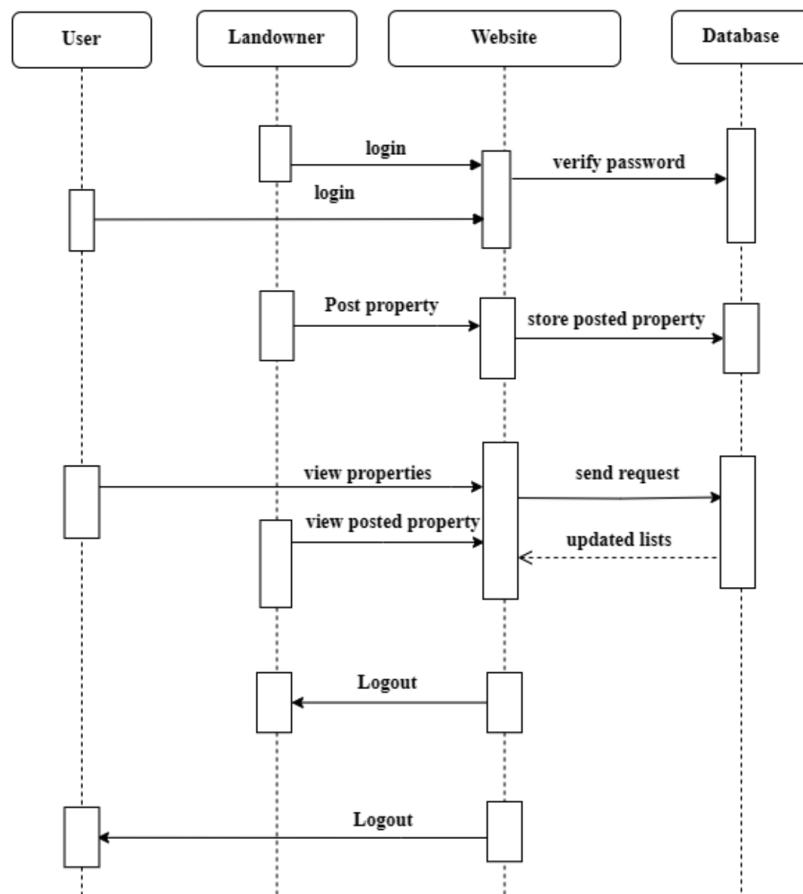


Figure 3.9: Sequence diagram for Post Property.

3.7.3 Sequence diagram for Book Property

A sequence diagram for "Book Property" showcases the chronological interaction between user actions and system components. It delineates the step-by-step process: The user searches for a property, selects it, submits booking details, the system checks availability, confirms the reservation, and notifies both user and the property owner. The diagram captures the dynamic exchange of messages and actions among these entities.

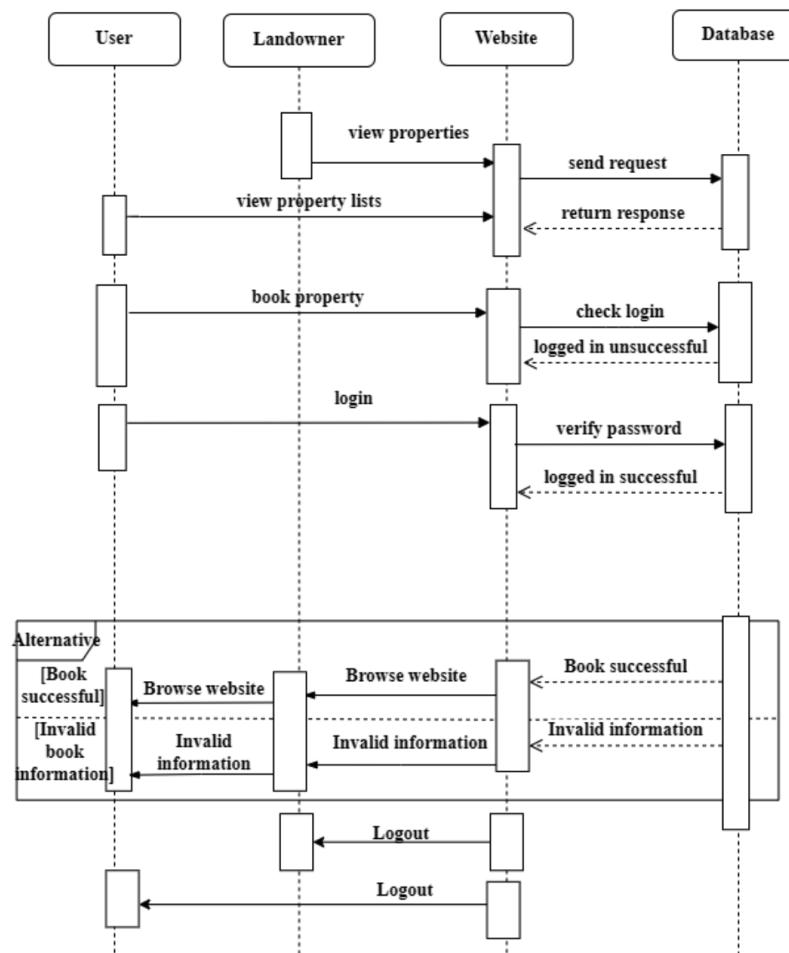


Figure 3.10: Sequence diagram for book Property.

3.8 Swimlane diagram

3.8.1 Swimlane diagram for User Profile Management

In the swimlane diagram for user profile management, distinct lanes represent users' roles: students and landowners. Landowners navigate activities such as profile management, listing updates, negotiation responses, and deal finalization. Meanwhile, students engage in profile maintenance, property searches, negotiation initiation, and booking.

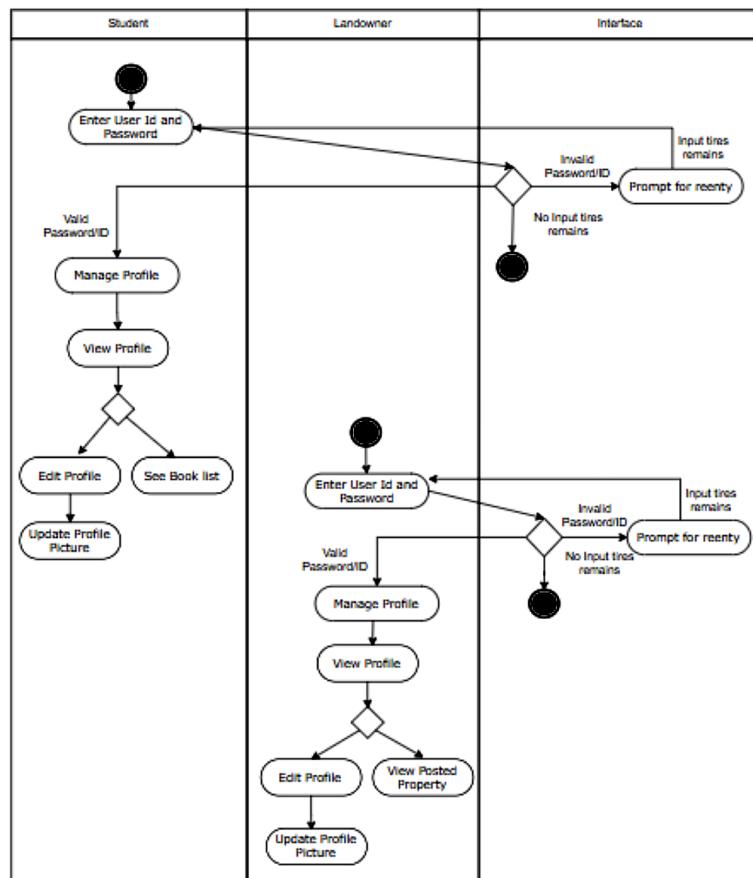


Figure 3.11: Swimlane diagram for user profile management.

3.8.2 Swimlane diagram for Post Property

The swimlane diagram depicts property posting, showcasing landowners' actions like property detail submission, negotiation responses, and deal finalization, while students engage in property search, negotiation initiation, and booking. It highlights the distinct roles and collaborative efforts of both parties in the property posting process.

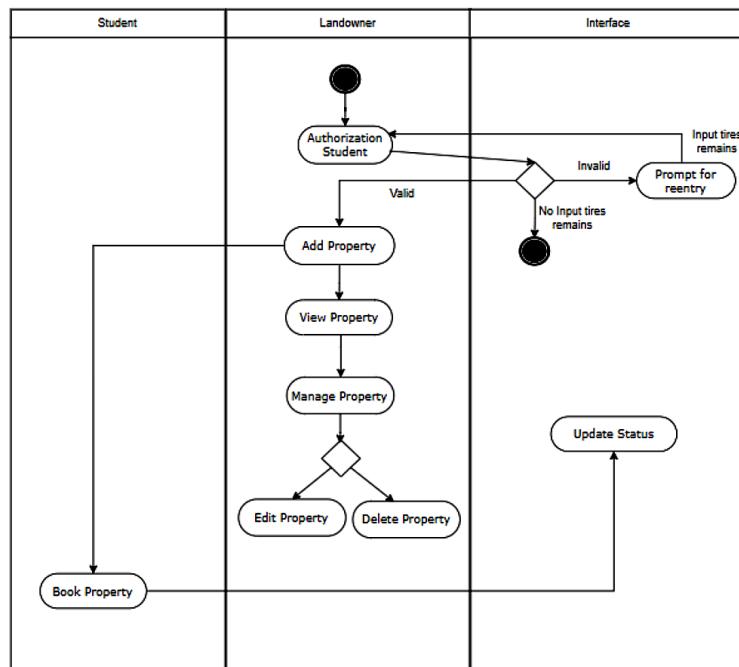


Figure 3.12: Swimlane diagram for Post Property.

3.8.3 Swimlane diagram for Book Property

In the swimlane diagram for book property, distinct lanes represent users' roles: interface, students, and landowners. The process typically begins with the user initiating a search for available properties. The website's interface interacts with the user, displaying property options based on their preferences. Once a property is selected, the user fills out booking details, such as rent duration and the user's permanent address.

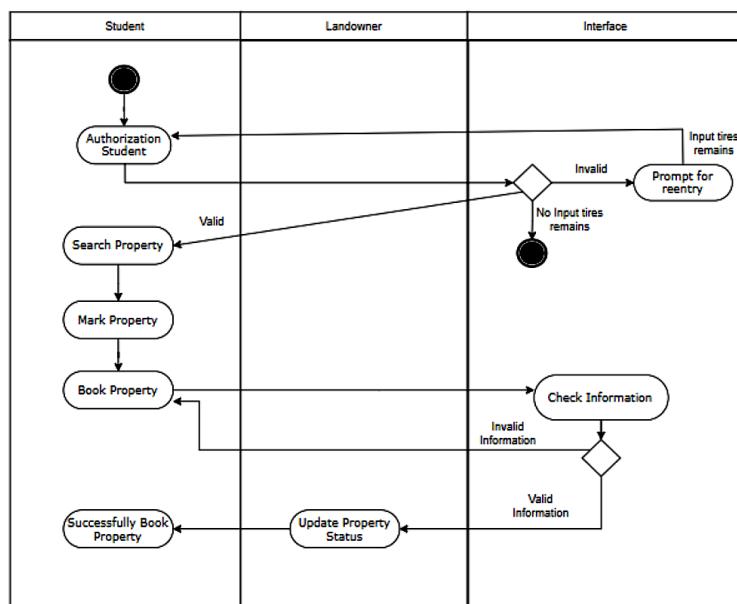


Figure 3.13: Swimlane diagram for user Book Property.

3.9 Safety and Security Requirements

3.9.1 Access Requirements

Users must authenticate to access the system. Different access levels will be assigned, such as Student, Club Organizer, and Administrator.

3.9.2 Integrity Requirements

The system will ensure data consistency through the use of secure transactions and database management practices. Any modification in event or club data will require proper validation.

3.9.3 Privacy Requirements

The system will implement encryption and secure protocols to protect user data. Sensitive information such as personal student data will be accessible only to authorized personnel.

Chapter 4

Architecture

4.1 Data Centered Architecture for Overall System

A data centered architecture for Rental Home Management system efficiently manages property bookings, user profiles, and property postings. It relies on a database system that integrates seamlessly with a user interface, enabling swift property searches, secure bookings, and easy profile management. Through structured data organization, it optimizes property listings, user information, and booking records, ensuring a streamlined and user-friendly experience for both property owners and renters.

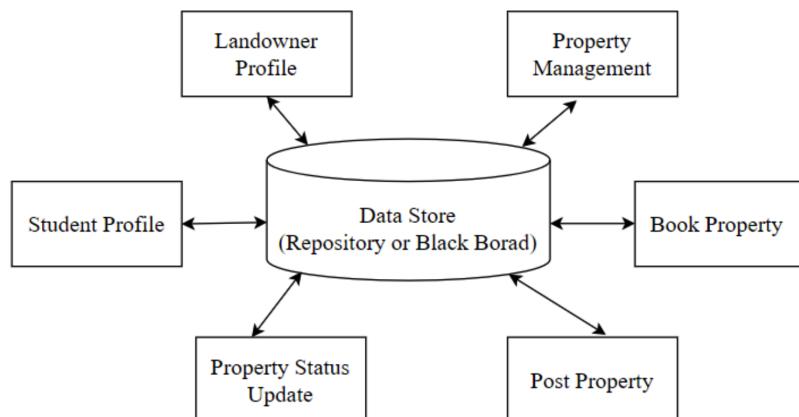


Figure 4.1: Data Centered Architecture for the overall system.

4.1.1 Rationale for Choosing Data Centered Style

- **Scalability:** Data-centered architecture allows for easy scaling as the system grows, accommodating increased property listings and user profiles effortlessly.
- **Efficiency:** Centralizing data improves system efficiency by enabling quick access, retrieval, and manipulation of information, enhancing overall system performance.
- **Consistency:** With a centralized data approach, maintaining data integrity and consistency across different functionalities becomes more manageable, ensuring reliability.
- **Flexibility:** This architecture facilitates easy integration of new features or updates, enabling adaptability to changing user needs and technological advancements.
- **Security:** Centralized data management enhances security measures, enabling better control and safeguarding of user information and transactions.

4.2 MVC Architecture for the System

In an MVC (Model-View-Controller) architecture for Rental Home Management system, the emphasis is on separating concerns between user management, booking, and posting property.

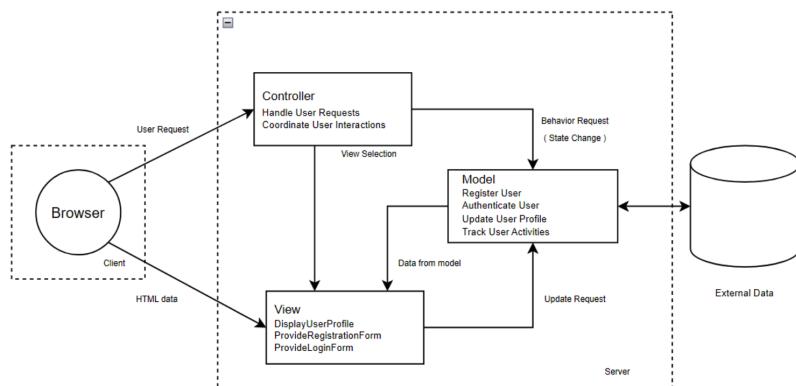


Figure 4.2: MVC Architecture for the System

4.2.1 MVC for User Management System

In the User Management System using MVC architecture, the Model stores user data, the View is the user interface, and the Controller manages interactions between them. This modular structure enhances organization and simplifies user-related operations.

4.2.2 MVC for Posting System

In the Posting System with MVC architecture, the Model stores post data, the View displays posts, and the Controller manages interactions between them for a streamlined posting experience.

4.2.3 MVC for Booking System

In the Booking System employing MVC architecture, the Model stores booking data, the View provides the user interface for booking interactions, and the Controller manages the communication between the Model and View, facilitating seamless booking processes.

4.2.4 Rationale for Choosing MVC Architecture

For our home rental system project, we opt for the Model-View-Controller (MVC) architecture due to its inherent benefits that align with the specific needs of our application. MVC provides a modular structure, allowing us to compartmentalize the different aspects of the system. This modularity enhances maintainability, as updates or modifications can be made to one component without affecting others. Additionally, MVC's clear distinction between the Model (data and business logic), View (user interface), and Controller (interaction and processing) ensures a scalable and adaptable architecture. This separation not only streamlines development but also promotes code reusability and ease of testing. Ultimately, choosing MVC for our project enhances the efficiency, flexibility, and collaborative aspects of our development process, aligning well with the dynamic nature of a home rental system.

4.3 Technology, software, and hardware used

The booking system utilizes a modern technology stack. On the front end, it employs React for dynamic user interfaces, CSS for styling, and Bootstrap

and React Strap for responsive design. The back end is powered by Node.js for efficient server-side processing.

- **Frontend:** React, CSS, Bootstrap
- **Backend:** Node.js
- **Database:** MongoDB

Chapter 5

Design

5.1 Component level design pattern

A software component is an encapsulated, interchangeable unit offering specific functionalities, designed for seamless interaction with other components. It's a modular, portable, and reusable package defining clear interfaces and dependencies. It's a building block that ensures well-defined functionality within a software system. Website features a modular, encapsulated system for property booking, listing, and user profile management. It offers replaceable, reusable components with clear interfaces, ensuring seamless interaction between customers, landowners, and the platform. Each component defines specific functionalities, fostering a cohesive and easily adaptable user experience.

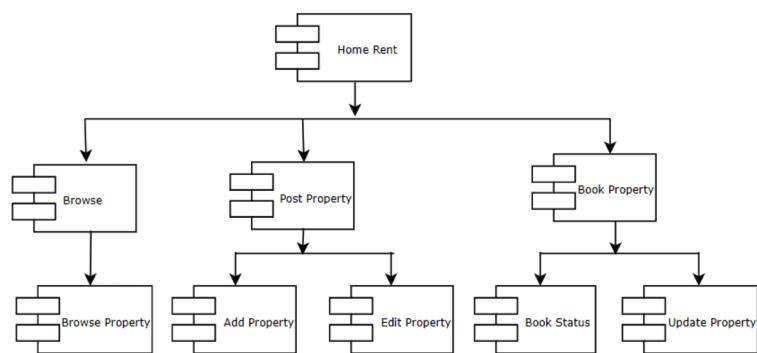


Figure 5.1: Structure Chart for Rental Home Management System

5.1.1 Home Reservation

The component-level design outlines the Home Reservation feature using PlantUML. It comprises two components: PropertyDetails for displaying property information and BookingManager for handling the booking process. The PropertyBooking class interacts with these components to view property details and initiate bookings. This design promotes modularity and encapsulation, enhancing the maintainability of the Home Reservation feature.

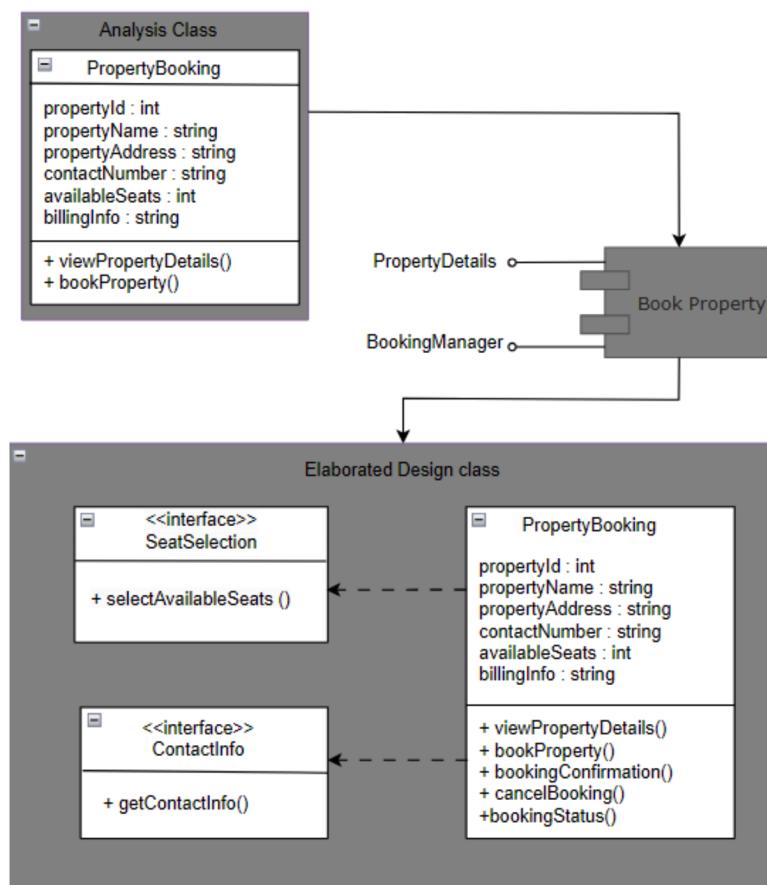


Figure 5.2: Component level design for Home Reservation

5.1.2 Property Management

Our apartment renting platform empowers landlords to post their empty apartments with detailed descriptions and images effortlessly. Through a

streamlined interface, landlords can manage their listings while users browse available apartments, creating a seamless experience for both renters and property owners.

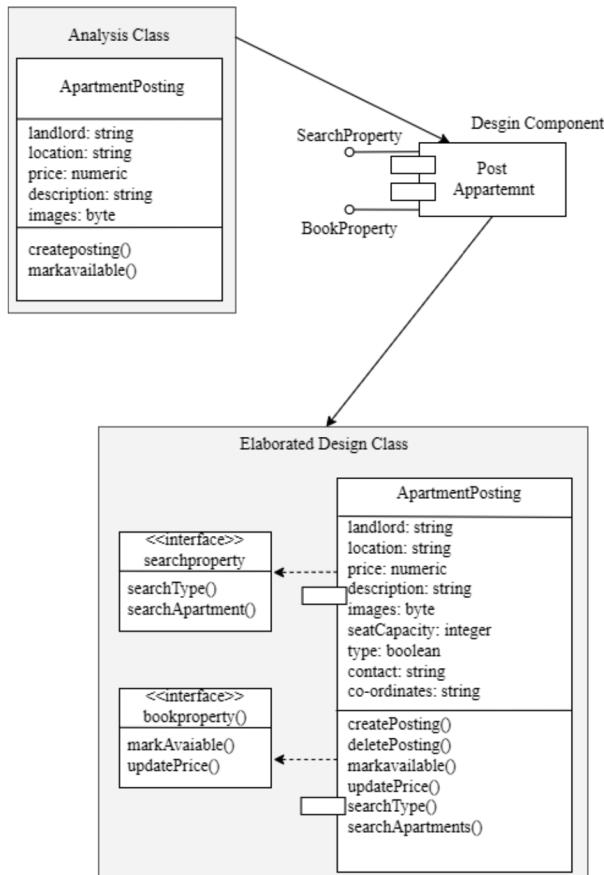


Figure 5.3: Component level design for Property Management.

5.1.3 User Profile Management

Our user profile management system offers a centralized platform for users to effortlessly update personal information, manage privacy settings, and adjust preferences. With a clean and intuitive interface, users can navigate through sections like security settings, notification preferences, and connected accounts seamlessly. Leveraging robust backend integration and stringent testing, our system ensures data security, reliable functionality, and a user-centric experience.

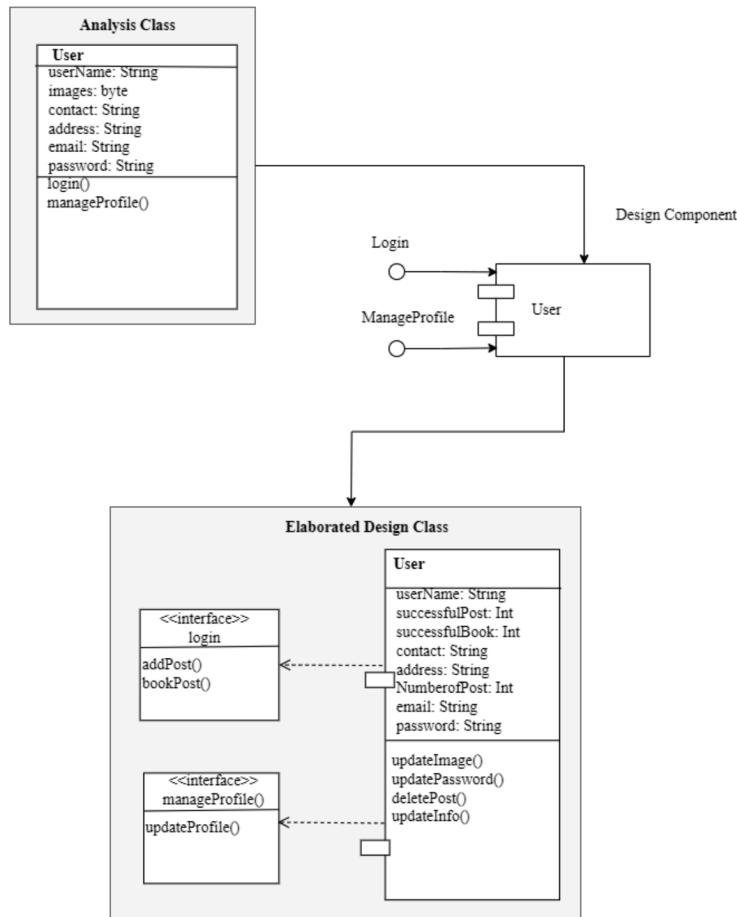


Figure 5.4: Component level design for user profile management.

5.2 Dataflow Diagram (DFD)

5.2.1 First Level DFD for the System

The Level 1 Data Flow Diagram (DFD) for the system includes entities like "User", "Property", and "Booking". It demonstrates how users interact with the system by posting properties and making bookings, illustrating data flow among these components for property management and booking processes.

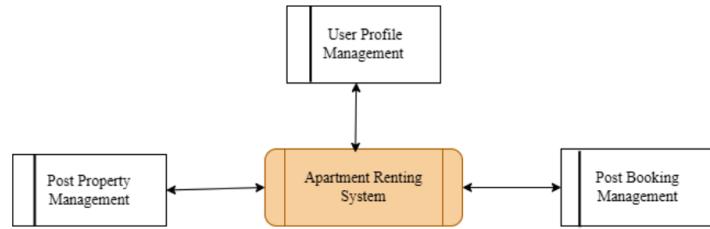


Figure 5.5: First Level Dataflow diagram for Home Reservation System.

5.2.2 DFD for Booking Property

The Data Flow Diagram (DFD) illustrates the home reservation feature. Users interact with the system through the user interface, entering booking details. The Booking System processes the booking, updates the User on confirmation, and adjusts the property's availability. The flow emphasizes simplicity, with data moving from user input to system processes and updates.

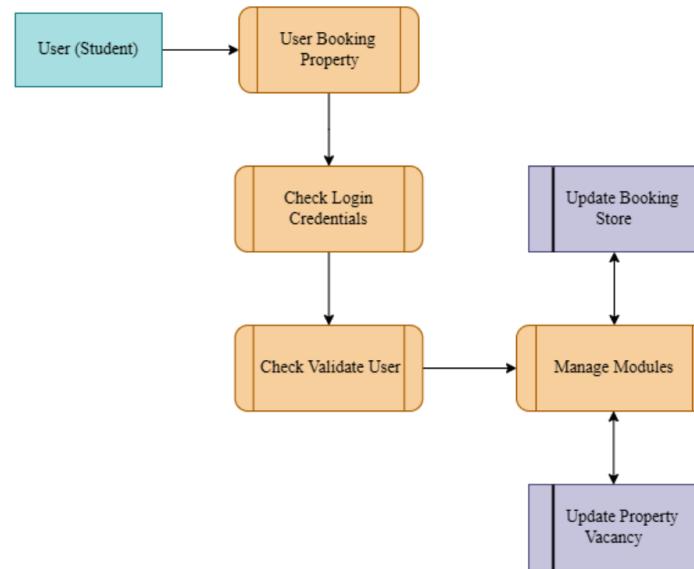


Figure 5.6: First Level Dataflow diagram for Home Reservation.

5.2.3 DFD for Booking Property

The Data Flow Diagram (DFD) illustrates the home reservation feature. Users interact with the system through the user interface, entering booking details. The Booking System processes the booking, updates the User on confirmation, and adjusts the property's availability. The flow emphasizes simplicity, with data moving from user input to system processes and updates.

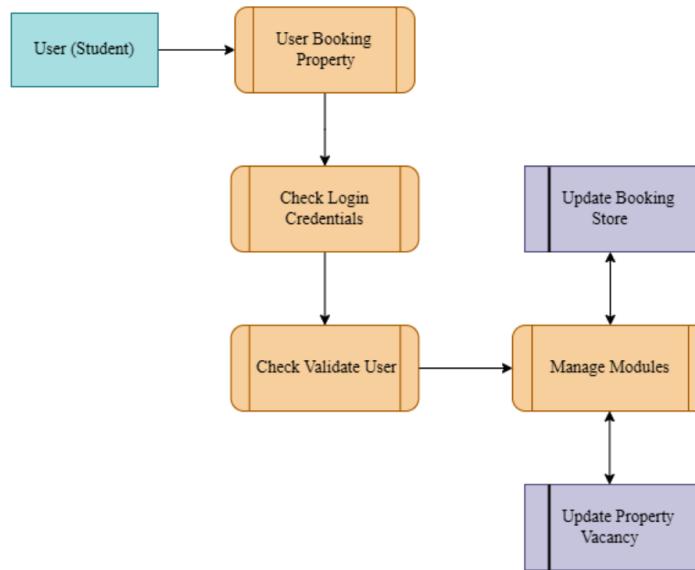


Figure 5.7: First Level Dataflow diagram for Booking Management.

5.2.4 DFD for Post Property Management

Property Management dataflow diagram shows how the landowner inputs apartment details, funneling information into the Property Listings data store. Published via the listing platform, potential tenants search listings. Inquiries submitted by tenants are handled, and responses are directed back to potential tenants.

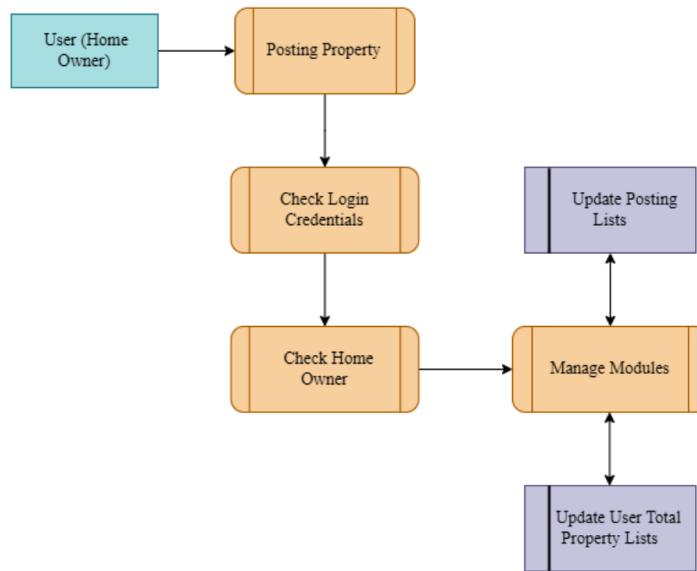


Figure 5.8: First Level Dataflow diagram for Property Management.

5.2.5 DFD for User Profile Management

The Level 0 DFD for user profile management showcases interactions between users and the system. Users input data for profile updates, which undergo authentication before updating the User Profile Database. At Level 1, the process delves deeper: user inputs are validated, profiles updated, and identity verified before retrieving or storing data in the central User Profile Database, ensuring a secure and accurate user profile management system.

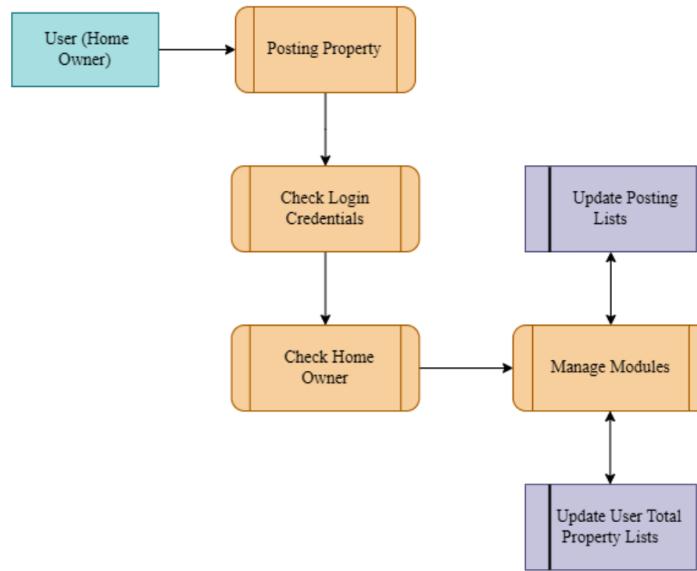


Figure 5.9: First Level Dataflow diagram for user profile management.

5.3 Entity Relationship Diagram (ERD)

The ER (Entity-Relationship) diagram for a website involving booking and property posting includes entities like User, Property, and Booking and their relationships. Users can post properties, which can have multiple bookings, establishing associations among these entities to facilitate property management and booking functionalities. Bookings include details such as booking date, duration, and total cost. The relationships indicate that a user can make multiple bookings, and each property can be booked multiple times.

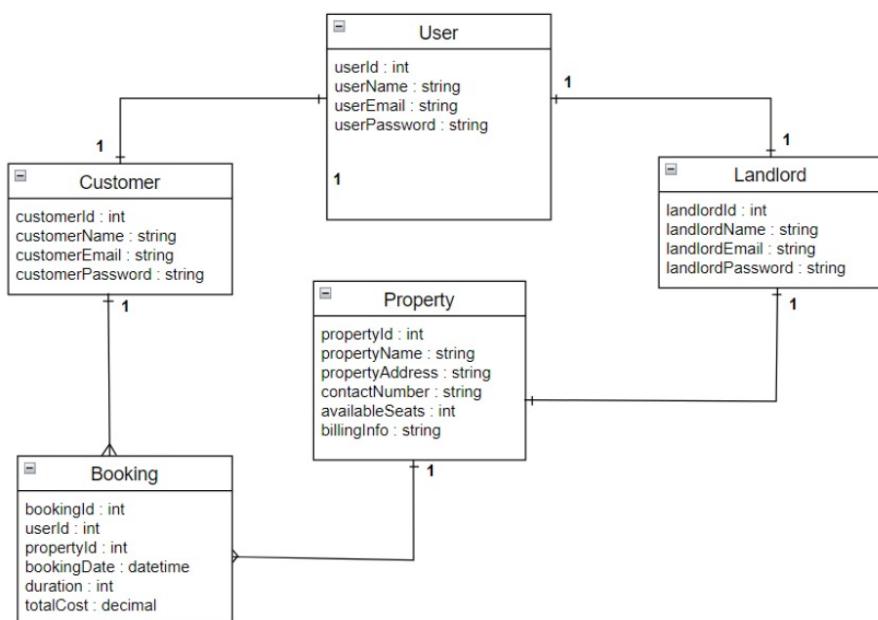
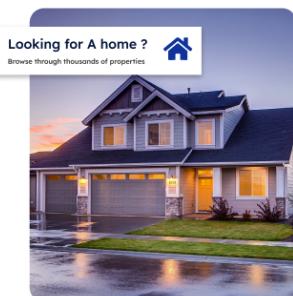
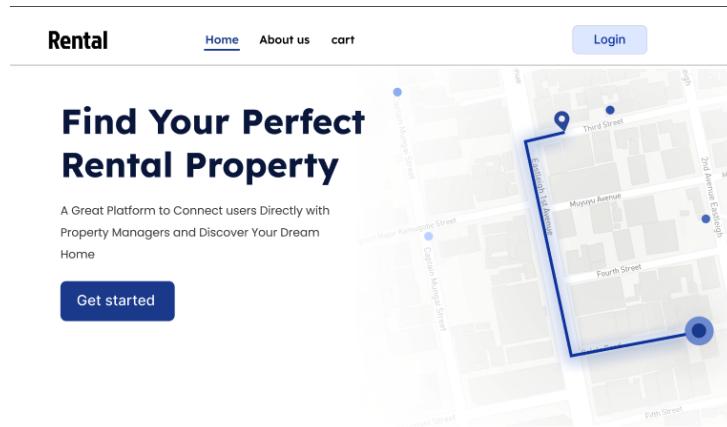


Figure 5.10: ER Diagram for Rental Home Management System

5.4 UX Design

5.4.1 Landing Page



What We Do

We Simplify Your Rental Home Search

Experience hassle-free rental home searching with our platform. We simplify the process by linking you directly to legit property managers by cutting out the middleman

Figure 5.11: Landing Page



Figure 5.12: Landing Page Footer

5.4.2 Authentication

Login

Don't have and account ! Create One

or

Login

[Forgot Password](#)

Figure 5.13: Login page for Our Website

Sign-up

[Already have an account ! Login](#)

Name

Email Address

username

Password

Sign Up

Figure 5.14: signup page for Our Website

5.4.3 User Dashboard

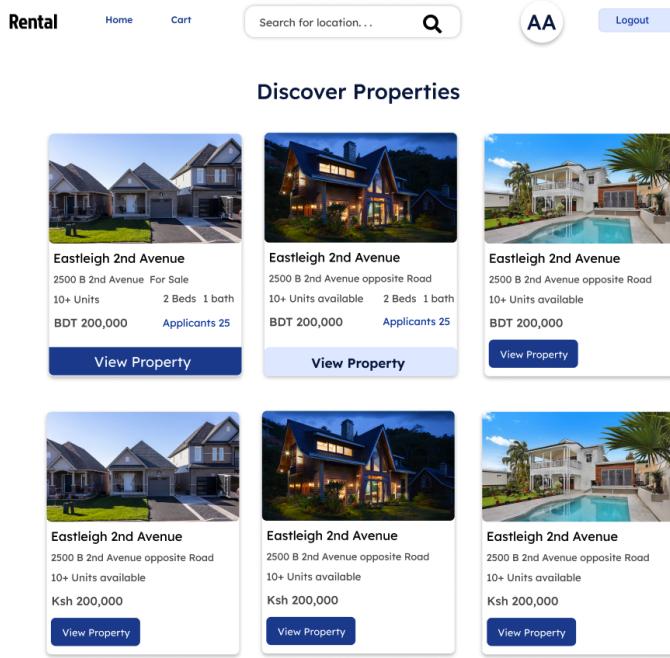


Figure 5.15: Property dashboard for Our Website

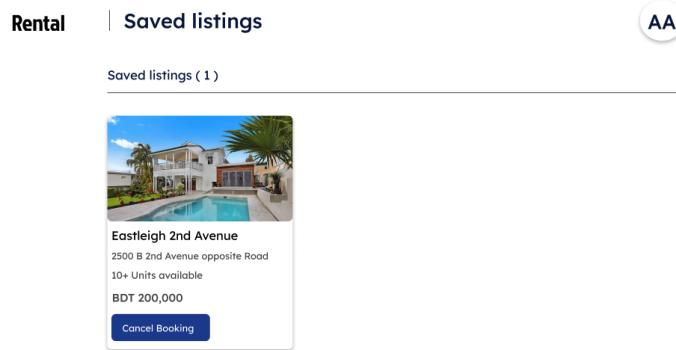


Figure 5.16: Booking info for Our Website

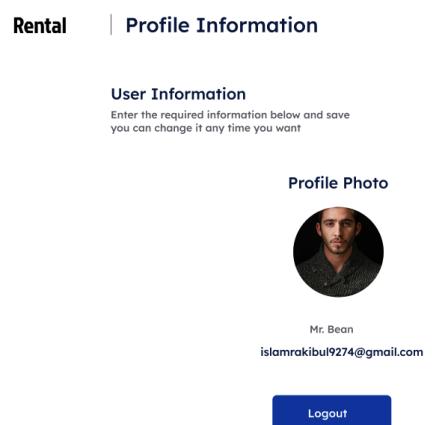


Figure 5.17: User profile page for Our Website

Chapter 6

Testing and Sustainability Plan

6.1 Requirements/Specifications-based System Level Test Cases

In the property renting project, system testing relies on a comprehensive specification document. It involves isolating each component for thorough examination, ensuring alignment with end-user expectations. A concise tabulated representation of system-level test cases, organized by requirements, ensures systematic validation, affirming the system's reliability and adherence to specified criteria. Table for Requirements/Specifications-based system-level test cases is given below:

| Requirement ID | Requirement Statement | Must/Want | Comment |
|-----------------------|---|------------------|----------------|
| R-Upload-01 | Users can't post a property listing without providing the address | Must | N/A |
| R-Upload-02 | Display an error message if the user attempts to post a property without specifying the available seats | Want | N/A |
| R-Search-01 | Users must be able to search for properties based on location and other relevant criteria | Must | N/A |
| R-Search-02 | Display a notification if there are no matching properties for the user's search criteria | Want | N/A |
| R-Booking-01 | Users cannot book a property without providing necessary details such as duration and contact information | Must | N/A |

Figure 6.1: Requirements/Specifications-based System Level Test Cases (Rental Home Management System System Near Cuet)

| Project Name | Rental Home Management System Near Cuet | | | | | |
|-------------------------|--|---|--|---|---|--------|
| Module Name | Upload Post | | | | | |
| Created By | Md.Minhazul Islam | | | | | |
| Reviewed By | Rakibul Islam Rumel | | | | | |
| Date of Creation | 11-01-2025 | | | | | |
| Date of Review | 14-01-2025 | | | | | |
| Test Case ID | Scenario | Steps | Test Data | Expected Result | Actual Result | Status |
| R-Upload-01 | Check upload image option with invalid file extension. | 1. Go to the user interface 2. Click 'choose file' option 3. Upload file from PC's directory | File: new.pdf | Redirect to the initial state of user interface | Redirect to the initial state of user interface | Passed |
| R-Upload-02 | Attempt to post a property listing without specifying the available seats. | 1. Navigate to property upload section. 2. Skip entering the available seats field. 3. Attempt to submit the listing. | Property details without specifying available seats. | Display an error message indicating that specifying available seats is mandatory. | Error message appears, indicating that specifying available seats is mandatory. | Passed |

Figure 6.2: Requirements/specifications-based system level test cases (01)

| Project Name | Rental Home Management System Near Cuet | | | | | |
|-------------------------|--|---|--|--|---|--------|
| Module Name | Search Posts | | | | | |
| Created By | Md.Minhazul Islam | | | | | |
| Reviewed By | Rakibul Islam Rumel | | | | | |
| Date of Creation | 11-01-2025 | | | | | |
| Date of Review | 14-01-2025 | | | | | |
| Test Case ID | Scenario | Steps | Test Data | Expected Result | Actual Result | Status |
| R-Search-01 | Search for properties based on location and relevant criteria. | 1. Navigate to the search section. 2. Enter location and other relevant criteria. 3. Initiate the search. | Location: "City A", Criteria: "Pet-friendly" | Display a list of properties matching the specified location and criteria. | List of properties appears, matching the specified location and criteria. | Passed |
| R-Search-02 | Search for properties with criteria that have no matching results. | 1. Navigate to the search section. 2. Enter criteria with no matching results. 3. Initiate the search. | Criteria: "Swimming Pool, Sauna" | Display a notification indicating that there are no matching properties. | Notification appears, indicating that there are no matching properties. | Passed |

Figure 6.3: Requirements/specifications-based system level test cases (02)

6.2 Techniques used for test generation

At Rental Home Management System Near CUET, we utilize Black Box Testing methods, including Equivalence Partitioning. Equivalence Partitioning for our platform entails segmenting the input range into separate data categories and using these categories to generate test cases. This technique involves examining equivalence classes associated with a specific input condition to distinguish between valid and invalid states. The input condition is Boolean, delineating one valid category and one invalid category.

6.3 Assessment of the goodness of your test suite

In evaluating the reliability and accuracy of our Rental Home Management system Near Cuet platform during the testing phase, we adopt a meticulous and systematic approach. We utilize black box testing methodology, primarily focusing on functional requirements. Specifically, we employ equivalence partitioning to evaluate different input conditions thoroughly. This technique categorizes input conditions into valid and invalid equivalence classes, facilitating a comprehensive assessment of the platform's functionality. Through these black box testing methods, we ensure the robustness and dependability of the Rental Home Management System Near Cuet.

Chapter 7

Acknowledgement

We want to express our heartfelt gratitude for the tireless dedication and precious time invested by our course teachers.

Mir. Md. Saki Kowsar
Assistant Professor
Dept. of CSE, CUET

Md.Al-Mamun Provath
Lecturer
Dept. of CSE, CUET

Their valuable guidance and feedback have helped us in completing this project.

Bibliography

- [1] Roger S. Pressman, *Software Engineering: A Practitioner's Approach*
- [2] GeeksforGeeks, <https://www.geeksforgeeks.org/software-engineering-architecture>