Software Requirements Specification

for

Roomble

Version 1.0

Prepared by

Group #: 7 **Group Name: Marauders**

Aarsh Jain	**230015**	**aarshjain23@iitk.ac.in**
Aritra Ambudh Dutta	**230191**	**aritraad23@iitk.ac.in**
Aritra Ray	**230193**	**aritrar23@iitk.ac.in**
Bhukya Vaishnavi	**230295**	**bhukyav23@iitk.ac.in**
Bikramjeet Singh	**230298**	**bsingh23@iitk.ac.in**
Hitarth Makawana	**230479**	**hitarthkm23@iitk.ac.in**
Shlok Jain	**230493**	**jainshlok23@iitk.ac.in**
Ronav Puri	**230815**	**ronavg23@iitk.ac.in**
Rathod Ayushi	**230844**	**rathoday23@iitk.ac.in**
Saksham Verma	**230899**	**sakshamv23@iitk.ac.in**
Surepally Pranaysriharsha	**231057**	**surepally23@iitk.ac.in**

Course: **CS253**

Mentor TA: ***Nij Bharatkumar Padariya***

Date: **24/01/2025**

CONTENTS II

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
Version 1.0	Aarsh Jain Aritra Ambudh Dutta Aritra Ray Bhukya Vaishanavi Bikramjeet Singh Hitarth Makawana Shlok Jain Ronav Puri Rathod Ayushi Saksham Verma Surepally Pranaysriharsha	First Version of the Software Requirement Specification (SRS) Document	23/01/2024

1 Introduction

1.1 Product Scope

Whenever an individual moves to a new place, be it for a job, education, or anything else, finding accommodation is often a tough challenge. Not only for them but finding flatmates or tenants (for landlords) may prove to be challenging in most cases owing to the lack of effective communication between the two sides. The purpose of our application is to digitize the process of property-based dealing and looking for flatmates for sharing the property, thereby bridging the communication gap efficiently, which helps eliminate certain restrictions like location-based restrictions and compulsion of physical involvement for deal finalization, and hence helps users to explore a wide range of options with minimal compromisation of their preferences and enjoy a satisfactory user experience.

Deployment of this product aims at helping the concerned people be it tenants, people searching for a flatmate, or landlords to make well-informed and efficient choices depending upon their preferences. Access to a database of available properties and people along with the ease of communication sums up the motive of the platform.

1.2 Intended Audience and Document Overview

The document is mainly intended for the people connected to the project, namely, the course instructor, our TA, our team, and the team that intends to beta-test the application.

The document may also assist the people we intend to use the finished product with, namely landlords, tenants, and people searching for a flatmate.

Section 1 (INTRODUCTION):

This section gives a brief overview that would be useful in reading the SRS document. Although the reader is advised to go through this section, but in case he/she is familiar with the basic terminology they can skip the section.

Section 2 (OVERALL DESCRIPTION):

This section offers a bird’s eye view of the software system and its functionalities, assumptions and dependencies. This will be a useful read for those seeking to familiarize themselves with the system at a quick glance. A reader is encouraged to read this part as it provides a good basis for understanding the next section of the SRS.

Section 3 (SPECIFIC REQUIREMENTS):

This section contains detailed information about the software and explains the functions in detail through the use of multiple diagrams. This is essential for end-users, clients and developers as it will serve as a guide in the development process and also an instruction manual for end-users.

Section 4 (OTHER NON-FUNCTIONAL REQUIREMENTS):

This section contains detailed information about the software and explains its functions in detail through the use of numerous tree diagrams. This proves indispensable for end-users, clients, and developers alike, serving as a roadmap during the development phase and a user manual for end-users.

1.3 Definitions, Acronyms and Abbreviations

- **DM** – Direct Messaging
- **HTTPS** – HyperText Transfer Protocol Secure
- **MTTF** – Mean Time to Failure
- **SRS** – Software Requirements Specification
- **UI** – User Interface
- **UCi** – ith Use Case
- **WS** – Web Sockets
- **DPDPA** – Digital Personal Data Protection Act
- **GDPR** – General Data Protection Regulation
- **GPS** – Global Positioning System

1.4 Document Conventions

We have used the following conventions while writing the document to enhance readability and capture the attention of the reader:

- The headings of all the sections are written in ***bold*** using Arial font size 18.
- The headings of subsections are ***bold*** and written in Arial in font size 14.
- The headings of sub-subsections are ***bold*** and written in Arial with font size 12.
- The content of the section is written in Arial font size 11.
- Any important term or short form is written in ***bold***.
- The alignment of the whole content is ***justified***.
- Text has been indented wherever required to highlight the hierarchy of the content.
- The document follows the IEEE formatting, indenting and numbering conventions. Any deviations from the same will be explicitly specified.

1.5 References and Acknowledgments

1.5.1 References:

Style guide (for this document): IEEE Software Requirements Specifications Guide

1.5.2 Acknowledgement:

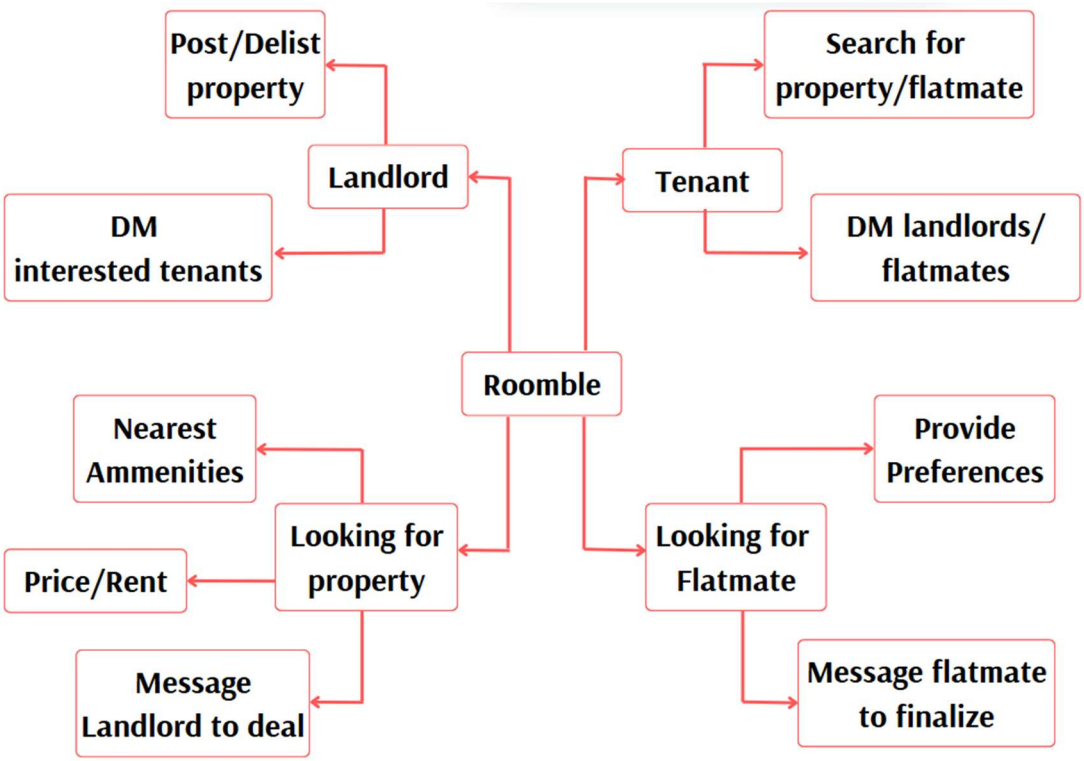
We are highly indebted to Professor Indranil Saha for his excellent lectures and notes, which largely provided the basis for our system design, and for diligently resolving all our queries regarding various functional and organizational aspects of the software. We are also obliged to our Teaching Assistant, Nij Bharatkumar Padariya, for his timely assistance whenever required.

We used Canva for creating the various diagrams of the use cases as well as for the various UI designs that have been used to present the basic ideas of our application effectively.

2 Overall Description

2.1 Product Overview

The app features a user signup/login system and validation. After signing in, users can fill out necessary details according to their preferences, which enables them to either post property as landlords to look for tenants or filter rentable properties based on their preferences, or find potential roommates accordingly. Users can filter based on living habits when searching for a flatmate or based on locality/nearby amenities when looking for a place to rent. Additionally, the app includes direct messaging capabilities for users to address any concerns, making it easier for them to make informed and efficient choices. Moreover, the app also includes a bidirectional rating/review system which allows the user to rate their overall experience about, either the property that they've stayed in or the tenant who has rented the user's property. The rating/review system can be availed by the user only after a validation process that is mutual with both landlord/tenant parties to maintain authenticity. Overall, the app bridges the gap between landlords and tenants, streamlining the renting and lending process.



2.2 Product Functionality

- **Creating Account**: Users will be allowed to register as a landlord or as a tenant.

- **Landlord**: As a Landlord, you will have the ability to list your properties for rent, making them visible and accessible to potential tenants searching for rental options.
- **Tenant**: As a Tenant, you will have the ability to browse and explore a variety of rental properties, with the option to rent independently or find a property to share with a flatmate.
- **Renting Out Property:** Landlords can list properties with photos, location, price, and other relevant details to attract suitable tenants.
- **Searching For Property:** Tenants can search for properties using filters such as location, price, property type, and preferences to find their ideal match.
- **Searching for Flatmates:** Users seeking flatmates for shared apartments can connect with those who share similar preferences.
- **Direct Message Feature:** Tenants can use the direct message feature to contact the landlords of the interested property, or to talk with potential flatmates.
- **De-Listing Properties:** Landlords would have the option to de-list their properties, in case their property is unavailable for rent.
- **Review and Rate**: Tenants can rate the properties they have stayed in, and also write detailed reviews about their experience. Similarly, landlords would also be able to provide ratings and reviews on their tenants.

2.3 Design and Implementation Constraints

- The landlord and tenant databases are an integral part of the relevance of the software, and its memory needs to be sufficient.
- The properties database is an integral part of the relevance of the software, and its memory needs to be sufficient for the timely addition (and updating) of properties.
- The server should have sufficient memory and resources available to accommodate and concurrently serve all the requests and data requested by the users.
- Response time for loading the details on the website should take maximum 30 seconds.
- The website, as of now, is available only in English

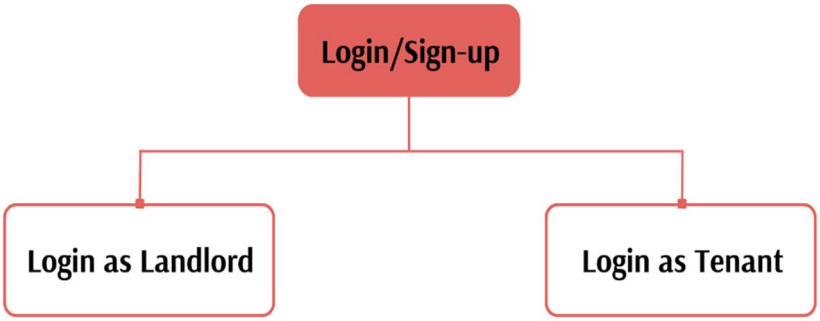
2.4 Assumptions and Dependencies

- The number of users who use this application shall never exceed 10,00,000.
- The number of properties a landlord can list on the website will not exceed 50.
- A person cannot simultaneously sign up as a Tenant and Landlord with same login details.
- Tenants and Landlords are held accountable for the concerned properties. In case of any damage reported to the property due to negligence or other reasons, they will be solely responsible for associated costs or penalties.
- Upcoming new technology shall not affect the execution of the application.
- The landlord and tenants will upload all the details of their profiles or properties accurately.

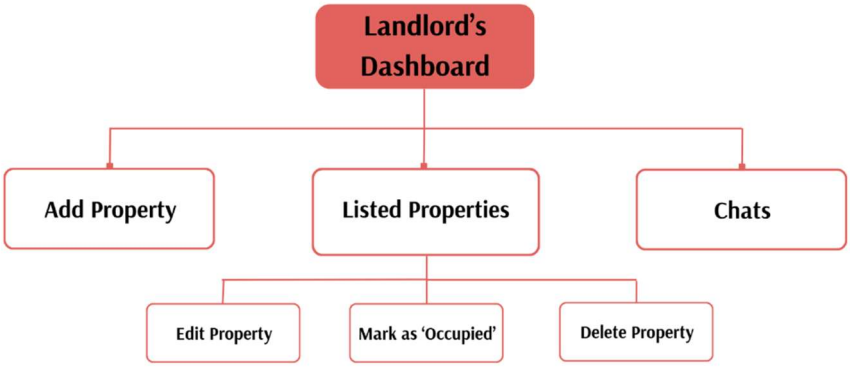
3 Specific Requirements

3.1 External Interface Requirements

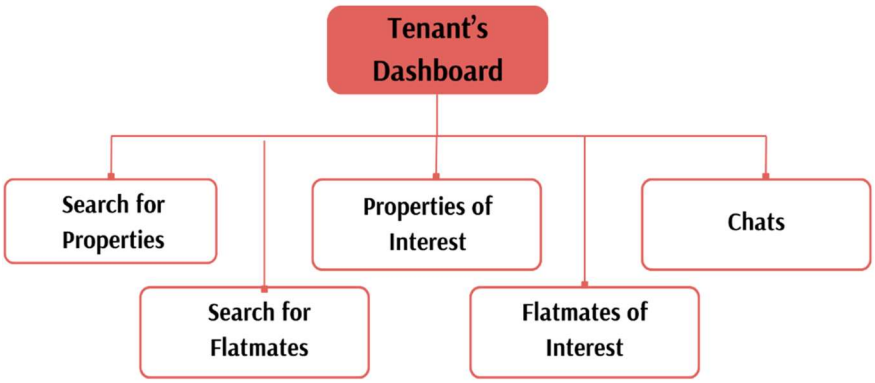
3.1.1 User Interfaces



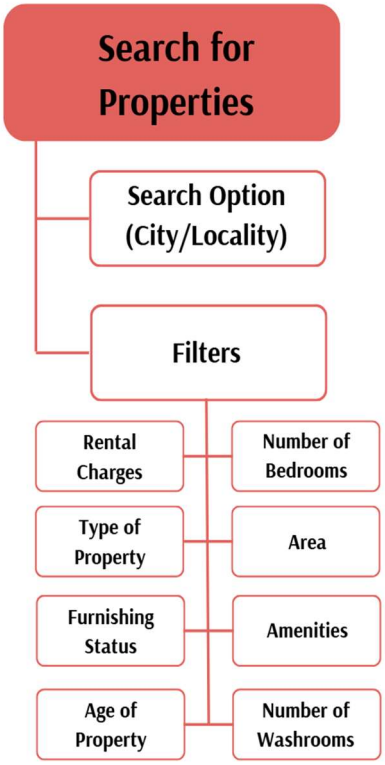
Two different modes of login or sign-up will be available – either as **landlord** or as **tenant**.

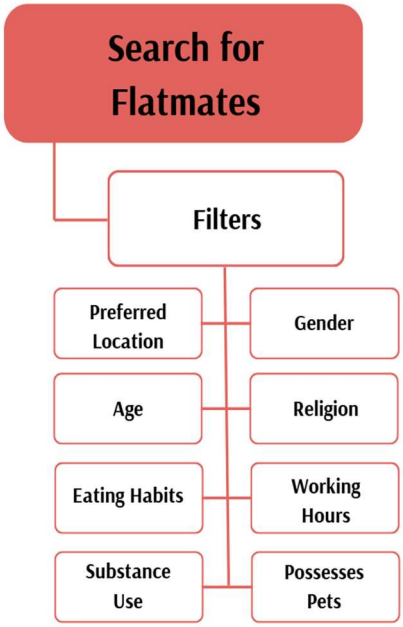


Landlord's dashboard will be visible only after the user has logged in using a landlord's account. It displays various options to **add/modify properties**, along with ongoing **chats** with other people.



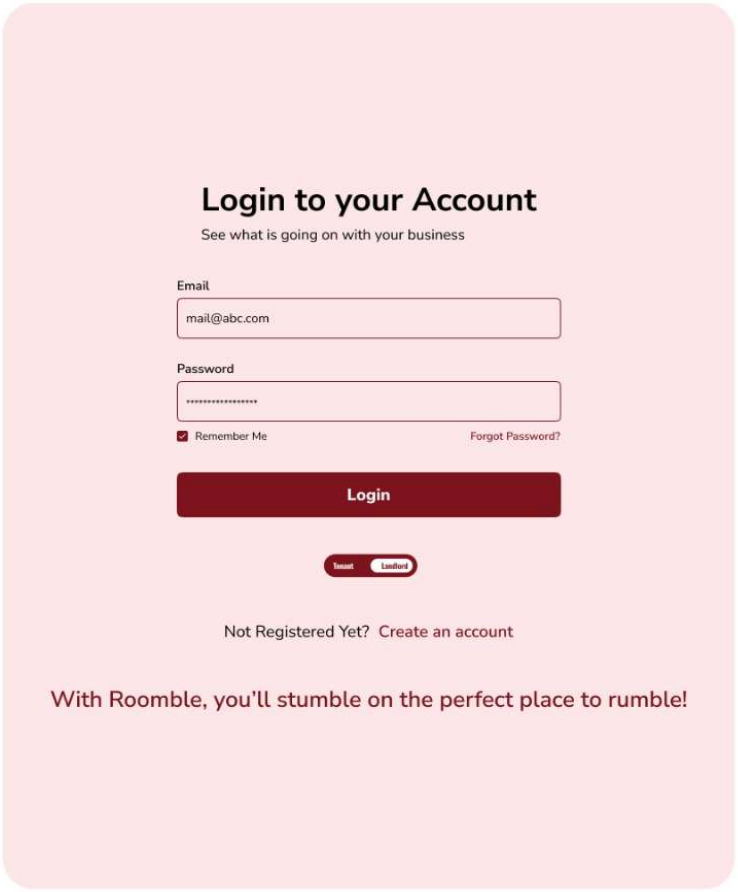
Tenant's dashboard will be visible only after the user has logged in using a tenant's account. It displays various options to **search for properties/flatmates**, a separate list of **properties/flatmates of interest**, along with ongoing **chats** with other people.





While searching for properties or flatmates, the user will be provided with a variety of **filters** that can help them shortlist properties/flatmates best suited according to their preferences.

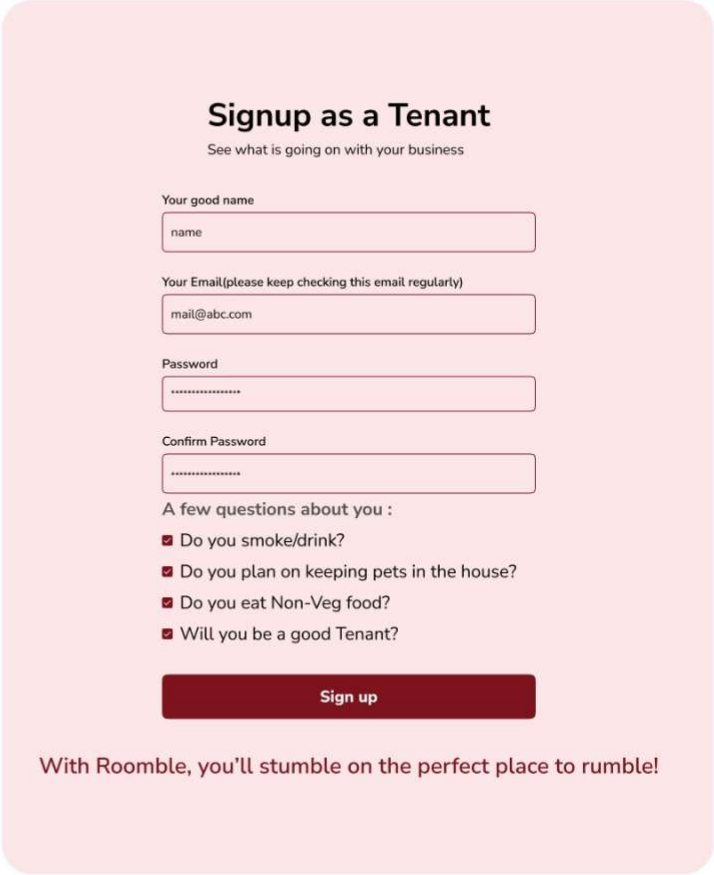
UI DESIGN



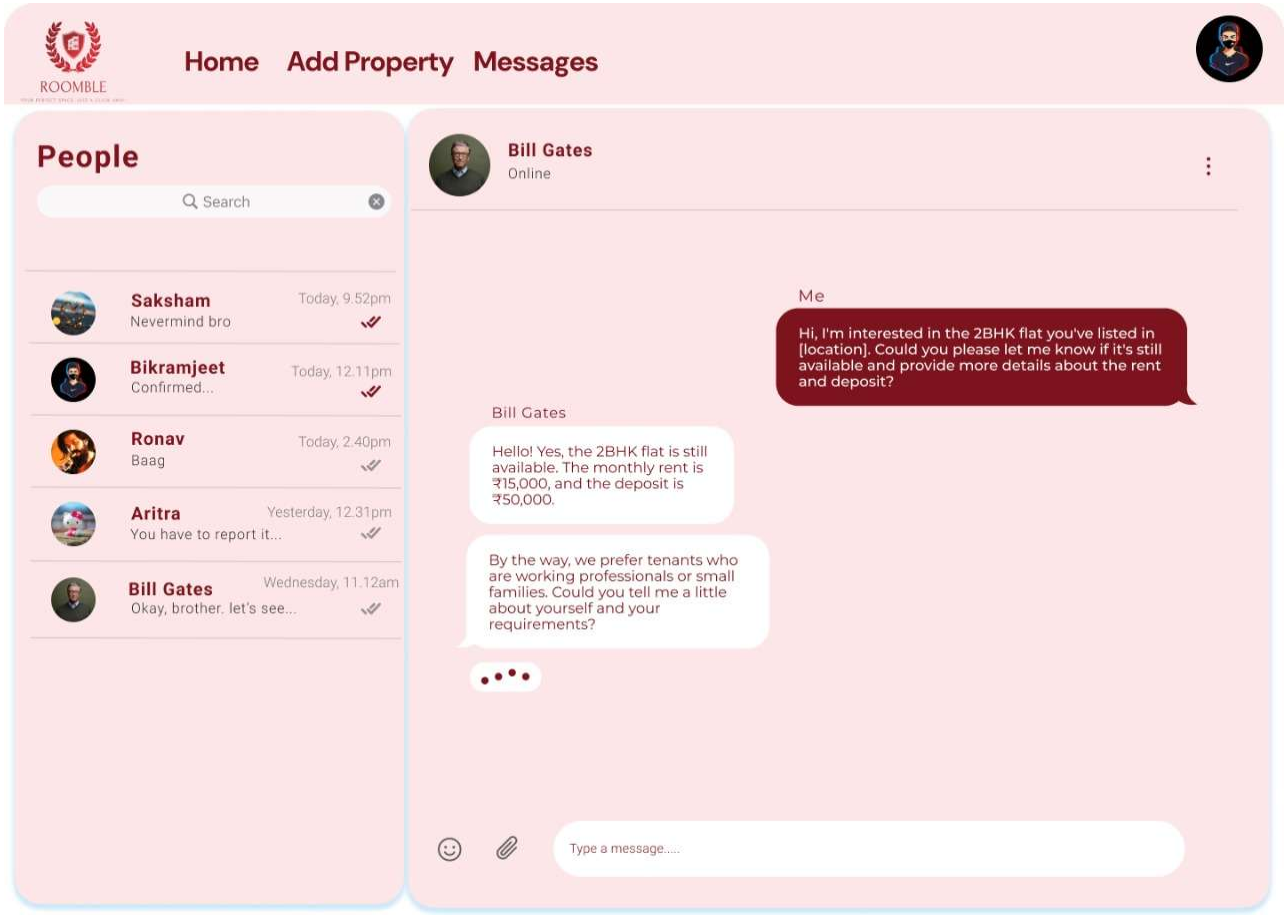
Every Client can maintain 2 accounts, one as a landlord and one as a Tenant.

ROOMBLE

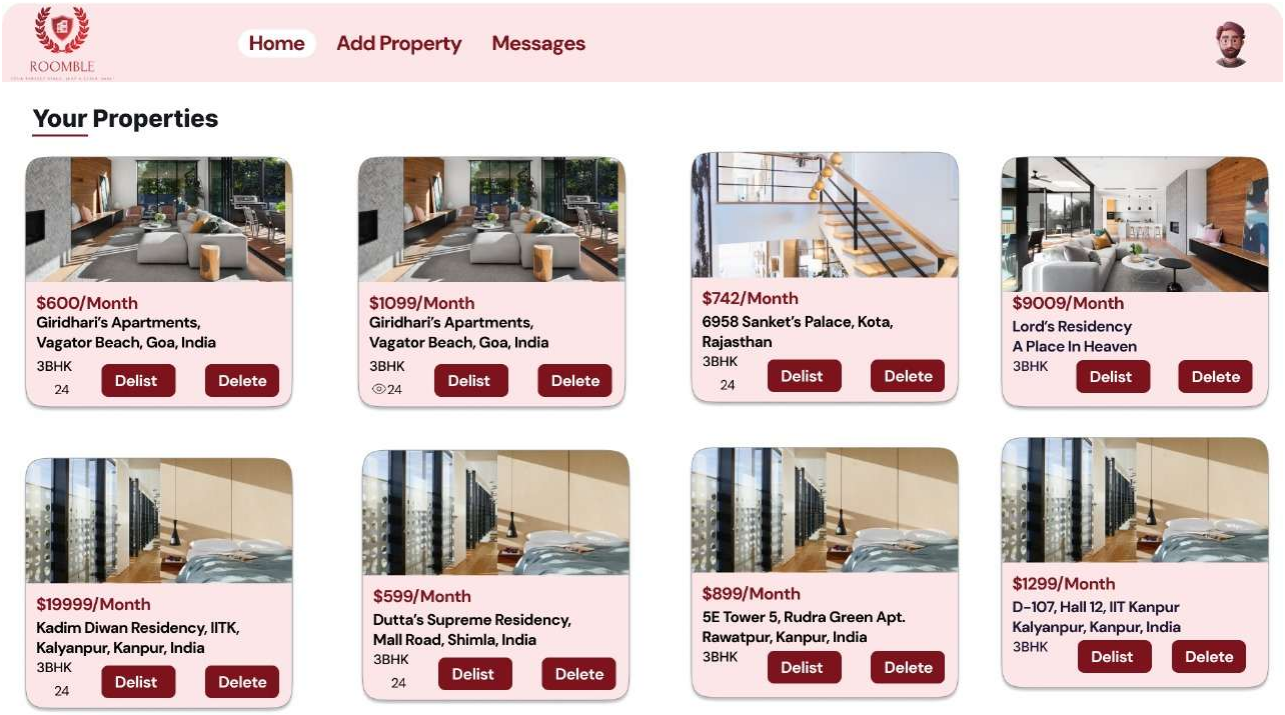
YOUR PERFECT SPACE, JUST A CLICK AWAY!



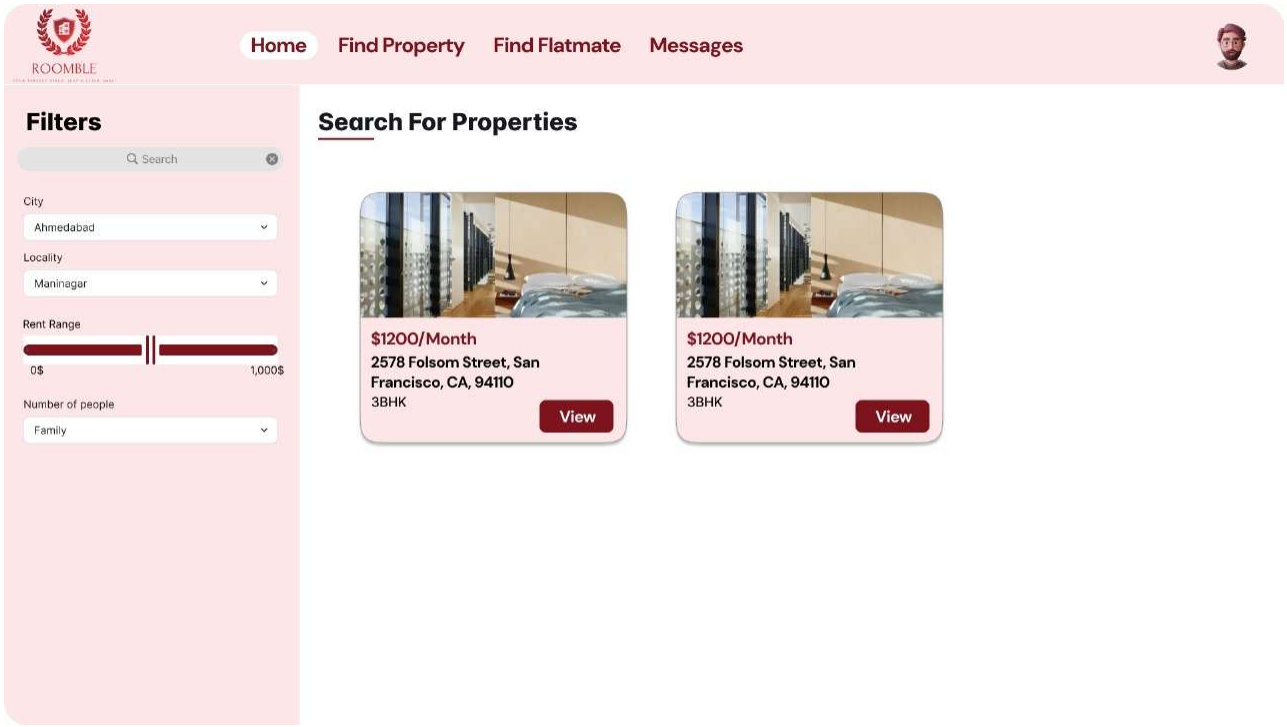
If signing up as Tenant, user will be asked about personal preferences to help find flatmates.



A tenant searching for a Landlord/Flatmate can DM them and have a productive conversation.



This is a typical Landlords Dashboard, where the Landlord can Delist/ List more properties.



This is a typical Tenants Dashboard, where he can see the properties in which he is interested.

3.1.2 Hardware Interfaces

- A device with an active internet connection is a must for accessing the portal. Whether it's a phone, a tablet, a desktop, a laptop, or any other device with internet access, our portal is designed to accommodate your preferred browsing experience. Simply install a browser application and start exploring our website.

- Support for modem and Network/Ethernet Card is needed on the users' devices – that is, appropriate drivers of compatible modem and Ethernet cards are installed for accessing web pages over the internet through their devices.

- GPS must be enabled on the user's device to provide the exact location of the property's address, ensuring accurate property information.

3.1.3 Software Interfaces

- The server-side components of the software system must operate within a Linux operating system environment.

- The client-side components of the software system must operate within typical web browser environments using Secure Sockets Layer (SSL) / Transport Layer Security (TLS) cryptographic protocols at a minimum encryption level of 128 bits. The minimum set of browsers that must be supported is:
    - Apple Safari 7+
    - Google Chrome 44+
    - Microsoft Internet Explorer 10+
    - Mozilla Firefox 40+

- The database and its management systems would be deployed only on the server side. The final client application delivered would not be independent of this. We will be using MongoDB Atlas for building this database.

3.2 Functional Requirements

3.2.1 F1: The system shall provide a convenient platform where landlords can offer their properties for rent.

- Prospective landlords will be able to upload their properties with appropriate details like property name, property type, address, relevant photos, BHK specifications, area in square feet, provided amenities, furnishing status, nearest hospital and airport, and several other attributes, along with the rental charges. They will also be able to specify tenant preferences, e.g. – single/family, no pets, etc.

- Landlords will have the option of editing their listed properties. They can upload updated photos in case of re-paintings, update the BHK and square feet in case of renovation or expansion, edit the provided amenities as a result of additions or developments, or update the nearest landmarks constructed recently.

- The application will enable landlords to delist their properties temporarily in case of current occupancy and also enlist their previously delisted property. They can delete their properties permanently too if so required. This ensures that prospective tenants only view properties which are currently available and improves the dependability of the service.

- Once a tenant finalizes the property and decides to buy it from his side, he will click on an option “Buy Now”. The landlord will then receive a notification, and can confirm from his side if he wants to allot the property. This is like a two-party authentication system which improves the credibility of the application. The system will then automatically delist the property due to current occupancy and store the tenant’s name in the database.

3.2.2 F3: The application shall provide the functionality of finding suitable flatmates with similar interests.

- Initially while making the profile, the tenant will be asked for several details which are essential for determining flatmate compatibility, like top 3 preferred locations for living, gender, age, alcohol use, smoking habits, religion, family/single, vegetarian/non-vegetarian, etc. among many others. Knowledge of these details will help the software to suggest the best possible flatmates for each tenant.

- To facilitate the search for their dream flatmate, the tenant will be shown several available prospective flatmates sorted according to their ‘compatibility score’. This ‘compatibility score’ will be calculated by a sophisticated recommendation algorithm, which will take into account only the most important personal attributes, like top 3 locations, gender, age, smoking habits, substance use, user rating, etc. This will make searching for flatmates way easier for the users.

- The users can further refine their recommendation page by checking several provided filters. These filters shall include the primary personal attributes (used by our recommendation algorithm) along with the other secondary attributes one might look for depending on their personal preferences – like marital status, religion, eating habits, work hours, etc. A wide variety of such qualities will undoubtedly help the users to choose the best possible flatmates for themselves.

- Just like with properties, the users will have the options to mark potential flatmates they are interested in by clicking on an option “Add to Interests”. These ‘interests’ will be displayed in the user dashboard under the heading “Interested Flatmates”. If the user is completely sure that the chosen flat-mate would be a great match for him, then he can choose the option “Confirm match”, following which the chosen flat-mate will receive a notification which he can approve or reject.

- Objective filters alone cannot fully determine how compatible two people will be, they should also have a natural rapport with each other. The system recognizes this essential component and provides the users with the Chat option, which will let both prospective flat mates no each other better and make informed decisions. This will be elaborated in more detail in the next functional requirement.

3.2.3 F4: The software shall promote interaction between prospective tenants, landlords and flatmates.

- All users will be provided with a DM option through which they can easily contact the opposite party and finalize deals, whenever they are looking at a particular property or a prospective flat mate. A Chat page will be there which will show all previous conversations and give the option to reinitiate conversations or start new ones.

- Tenants and landlords can interact seamlessly with each other through this functionality. Tenants can enquire about more intricate details of the property and surrounding environment, they can decide a time to go and visit the property to see it for themselves, they can also bargain with the landlord about the rental price. This DM feature also helps to preserve a written record of the conversation which might prove to be important in the future.

- For prospective flatmates, the DM option is arguably even more important, as it is the closest thing to human connection that software can provide. They can enquire about various details like personal habits, workplace details and perhaps even favourite novels and web series, to find out if they would be a good match for each other. They can also decide upon a time and place to meet and get to know each other even better.

3.2.4 F5: The system shall provide a secure two-way review and rating system for tenants, landlords and flatmates.

- Tenants will be able to rate the properties they have lived in and write an extensive review which might reveal some finer details of the listed property and the landlord. These ratings and reviews will help future users filter their search results and also give them a more comprehensive idea of the respective property.

- Similarly, the landlord will also be able to rate and review their tenant as well. These reviews help convey honest personal experiences which objective filters can never capture completely; hence it will help future landlords decide whether the tenant will be suitable for them. This two-way review and rating system improves the credibility of the service and ensures that both landlords and tenants conform to basic standards and etiquettes during the stay period.

- The review and rating system is applicable not just for landlords-tenants, but flatmates as well. Present or prior flatmates can easily rate each other and write exhaustive reviews detailing pros and cons of the person. This provides a detailed and honest account of how good a particular person is as a flatmate, and will help future users tremendously while choosing their flatmates. This feature thus improves the trustability of our service.

3.2.5 F2: The system will enable tenants to easily book the property of their choice.

- Tenants will have the option of searching for available rental properties by specifying their desired location. The search algorithm will leverage pre-computed distances and property ratings to display the most suitable properties closest to the desired location.

- The system will also provide several filters to the tenants for refinement of search results, pertaining to relevant property attributes – like rental price, BHK, area in square feet, property rating, availability of specific amenities, furnishing status, etc. Tenants can simply check the respective boxes to filter the results and find the best property for them.

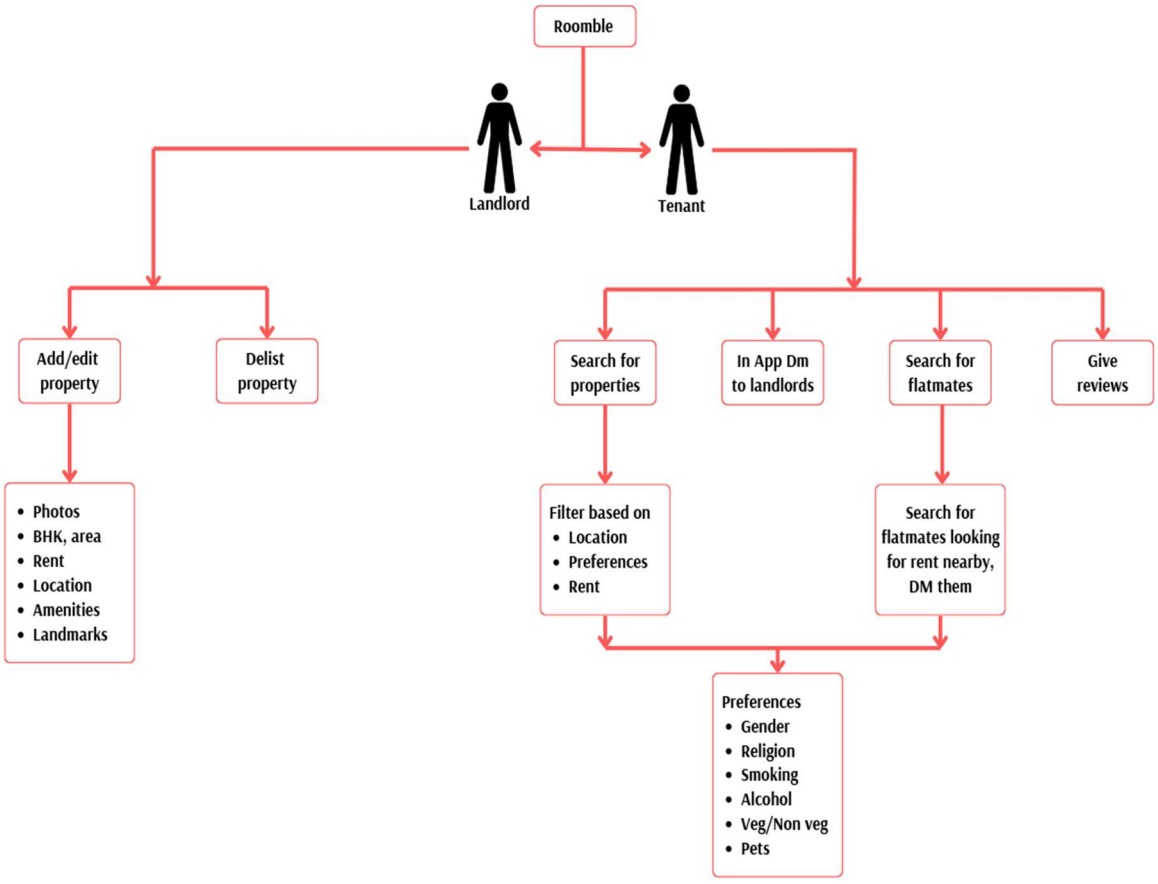
- By clicking on the respective property thumbnail, the tenant will be able to view all details of the property given by the landlord, including photos, nearest airport, railway station details, etc. The tenant will also be able to view the tenant criteria, and check if he or she will be suitable for the property or not. This will provide a true picture of the property from all aspects and prevent chances of future dejection.

- The tenant shall have the facility to save a property they are interested in for quick reference in the future by clicking on “Add to Interests”. This property will then be displayed in the tenant

dashboard under the “Interested Properties” section. The tenant shall also have the option of “Buy Now” if they have checked all details and are sure that they want to go ahead with the deal, and the landlord will then receive a notification as described in the previous functionality (F1).

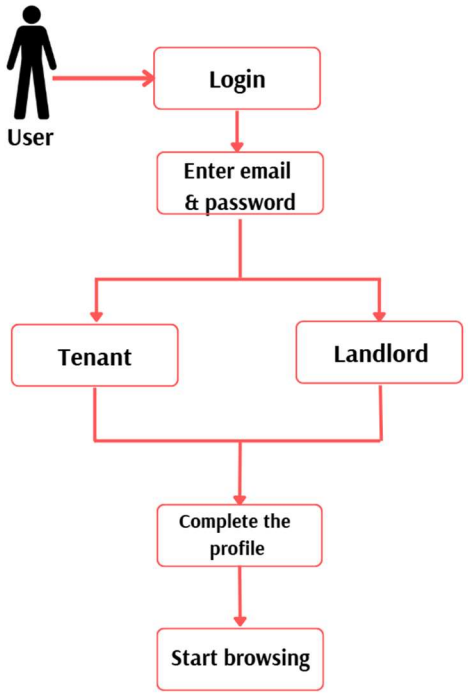
- The tenant will be able to interact with the landlord through the Chat option in the application, to find out if it would be a right match or not and will also be able to bargain. This is covered in further detail in the fourth functional requirement (F4).

3.3 Use Case Model



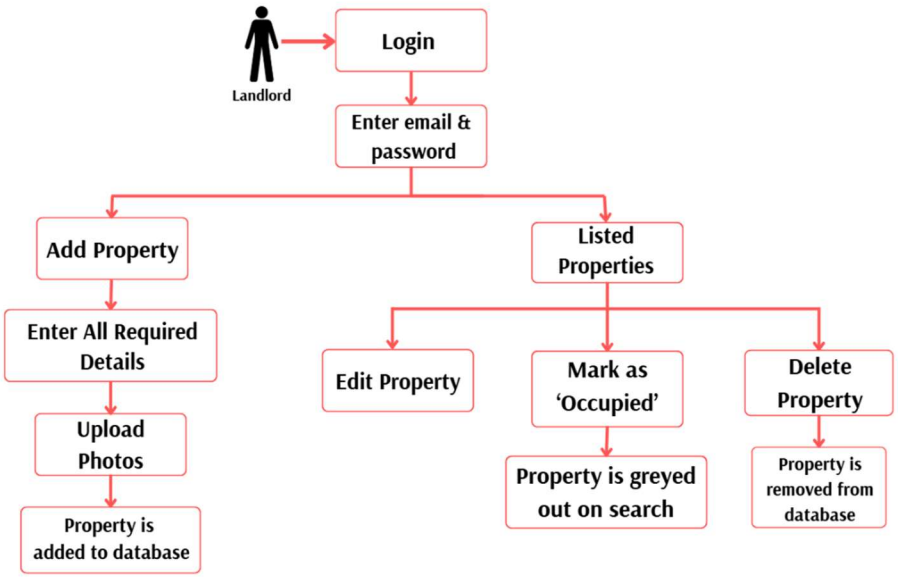
A very Brief Overview of the complete model and what is expected to be done with its help.

3.3.1 Use Case 1 (UC1): Enabling User Account Creation on the platform:



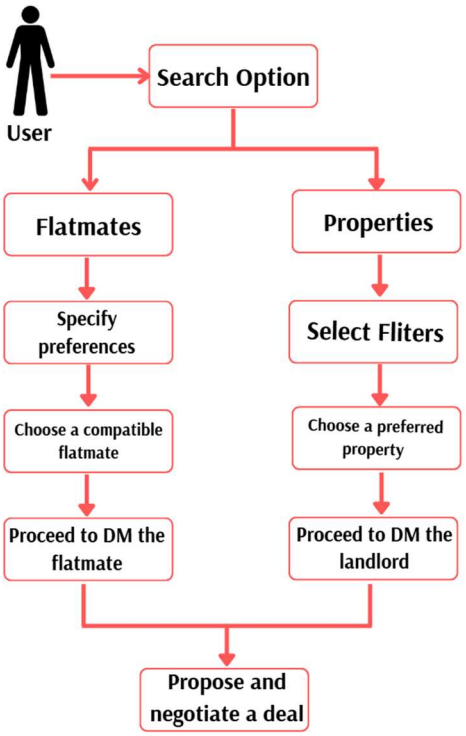
Authors	Hitarth Makawana, Aarsh Jain
Purpose	To allow the user to create an account on the web application enabling them to use it.
Requirements *Traceability*	To create an account a user must have a valid email address and create a unique password so that the admin can authorize the account formation.
Priority	The priority of this use case is high since without creating the account the user won't be able to use the platform.
Pre-Conditions	The user must have a valid email ID as a pre-requisite to making this account.
Post Conditions	After account creation the user's email and the password will be updated in the database owing to which the user can login and use the platform.
Actors	The actors involved would be the users who are looking forward to using the application.

3.3.2 Use Case 2 (UC2): Adding, Editing, Delisting or Deleting Property by Landlord



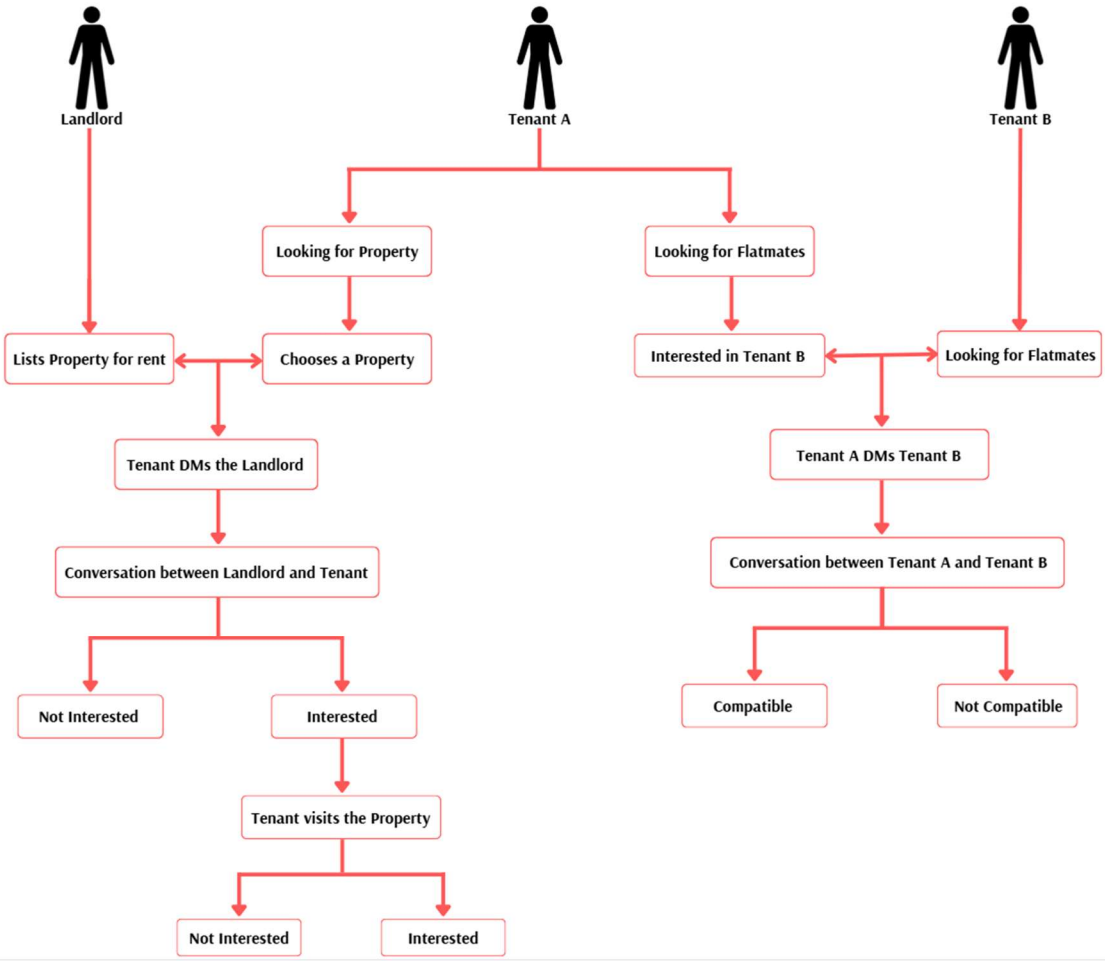
Authors	Ronav Puri
Purpose	To allow the landlord to add, edit, delist or delete their properties.
Requirements Traceability	To add/edit a property, the landlord must have all the necessary details related to the property available with them. To mark a property as 'Occupied', the property must be temporarily unavailable for rent or already occupied.
Priority	The priority of this use case is high since one of the main goals of the platform is to help tenants find a rental property of their choice.
Pre-Conditions	The user must have a valid account as a landlord on the platform to gain access to these operations.
Post Conditions	After adding a property, it will be added to the database, and it will be mapped with the respective landlord. After editing a property, it will be updated in the database. After delisting a property, the property will be greyed out and inaccessible to anyone who searches for it. After deleting a property, it will be removed from our database.
Actors	The actors involved would be the users who are registered as landlords on the platform.

3.3.3 Use Case 3 (UC3): Enabling a user to efficiently search for a property/flatmate



Authors	Bikramjeet Singh
Purpose	To enable users (specifically the tenants) to search and find any rentable property in the area of their choice or to find a flatmate for sharing a rented place
Requirements Traceability	To search for the property, the user must be signed up and logged into the application.
Priority	This use should be considered a high priority since connecting tenants to rentable properties is one of the app's main purposes.
Pre-Conditions	The user must be signed up and logged into the application to use this feature.
Post Conditions	If a user is interested in any property or in contacting a particular person as a prospective roommate, he can connect to the concerned persons through the direct messaging feature of the app.
Actors	The actors involved in this use case are those using the application and seeking a rentable property or a flatmate/roommate.

3.3.4 Use Case 4 (UC4): Efficient Interaction between the two concerned parties



Authors	Aritra Ray and Aritra Ambudh Dutta
Purpose	To facilitate interaction between tenant and landlord and between prospective flatmates
Requirements Traceability	To ensure tenants and landlords can make informed choices and consider a whole lot of factors before committing themselves to a particular property or flatmate
Priority	This use-case has high priority since tenants and landlords will not be able to make the right choices without interacting with the opposite party, in case the system malfunctions.
Pre-Conditions	This use-case has two pre-conditions – 1. **For Tenant-Landlord interaction:** The landlord must have uploaded their property properly and the tenant must be actively searching for properties. 2. **For Flatmate-Flatmate interaction:** Both the prospective flatmates should have completed filling in all

	their preferences in their profile and should be actively searching for flatmates.
Post Conditions	The tenant can go on searching for new landlords or flatmates if he loses interest in the current property or flatmate. If both find each other compatible after meeting, the deal can be finalised in case of property and match-up confirmed in case of flatmate.
Actors	There are 3 actors in this use-case: • **Tenant A –** Searching property and flatmates • **Landlord –** Potential landlord of Tenant A • **Tenant B –** Potential flatmate for Tenant A
Exceptions	If both parties are 100% confident that they are suitable for each other just based on available information without any interaction, then this use-case may not be required.

4 Other Non-functional Requirements

4.1 Performance Requirements

- The software is expected to maintain 99% of the period of service, which does not include shut down due to maintenance breaks and software update windows.
- It should be able to process X simultaneous request per minute, keeping in mind the load capacity of the service-providing server, to ensure smooth functioning and satisfactory user experience.
- The software should be able to respond to the search requests within 5 seconds with the best-matched search results with every search request by the user.
- The software should consist of some alternative navigation system in case the current system becomes faulty/starts malfunctions to preserve the update requests made by the user and prevent loss of the history of the user requests.

4.2 Safety and Security Requirements

4.2.1 User Authentication

- Validates user credentials such as mobile numbers, emails, passwords, and roles (tenant/landlord).
- Cross-checks inputs with a securely stored database to ensure only authorized users gain access.

4.2.2 Privacy and Data Security

- Encrypts personal information using industry-standard protocols to ensure data is secure during storage and transmission.
- Restricts access to sensitive user data, making it invisible to admins and inaccessible over the server without proper verification.
- Facilitates landlord-tenant communication via a DM system, keeping contact details private and secure.

4.2.3 Database Security and Durability

- Preserves data consistency during system downtime, enabling seamless recovery after maintenance.
- Ensures that updating or adding any data does not affect the already stored data.

4.3 Software Quality Attributes

4.3.1 Accessibility

- The software should have a user-friendly interface, which allows the users to navigate through desired functionalities without confusion so that the rental process proceeds with ease.

4.3.2 Maintainability

- The architecture, design, implementation, and documentation of the software must be such that they make the system reduce the maintenance overhead as much as possible and adding a particular minor feature should not take as more than 1 person week while fixing a security bug should be done within 2 person days along with updating the documentation.

4.3.3 Request Load Management

- Since the user requests might vary seasonally hence the database availability might be enough to cater all the requests, i.e. ensure that the software can handle the load as and when the user and property database grows.

4.3.4 Reliability

- The MTTF (mean time to failure) shall be more than one week.
- The software must be tested and validated each time a software update is released or deployed.
- The software must be consistent in providing correct or best matched results according to the prompt filters and search requests.

4.3.5 Portability

- We are using Node.js for designing the backend part of our software and React.js for designing the frontend part of our software, thus our software is portable and can run on most of the modern web browsers and wider screens.

4.3.6 Availability of Server Restoration

- In case of server failure or unexpected malfunction, the software should be restored within 3 hours.

5 Other Requirements

5.1 Authentication

- **Permission for OTP Authentication:** Permission will be required to send OTPs for authentication via automated emails or SMS. This will ensure secure user verification during the signup and login processes.

5.2 Legal Requirements and Copyright

- **User Data Privacy:** Permissions must be sought from users to collect, store, and process personal data, such as contact information and preferences, in accordance with data protection and privacy rules that may be enforced by the Government of the nation like DPDPA, GDPR. User data collected should be stored in the country's regional data center.
- **Copyright:** Copyright of the source code and the documents must be retained by the developers of the system.

5.3 Database Requirements

- **Hosting and Server Requirements:** Permissions or agreements with cloud service providers (e.g., AWS or Google Cloud) may be required for hosting databases and application services to ensure scalability and reliability.
- **Handling Sensitive Data:** All sensitive data (e.g., passwords, personal details) must be encrypted in the database using industry-standard encryption algorithms like AES-256.

Appendix A – Data Dictionary

Appendix A:

Variable/Constants	Type	Description	States/Possible Values	Operations	Functional Requirements
User Type	String	Identifies the type of user accessing the application	Tenant, Landlord	Input during registration/login	Must be validated during user registration or user login.
Password	String	Stores the password for each user, encrypted for protection	String consisting of Numbers, alphabets and symbols	User must create password during signup	During every login user must use the correct password
Roommate Preference	String	Preferences for selecting a roommate.	Examples: Smoker/Non-Smoker, Gender, age, etc.	Input during tenant profile creation	Should help in filtering and sorting with respect to individual preferences
Property Details	String	Complete details about the property as given by the landlord	Includes location, nearby amenities, rent, landlord's particular preferences	Can be used by a prospective tenant to decide between possible rentable properties	Should allow the landlord to list important details about the property as much he/she desires to share
Contact Details	Strings for Numbers and emails	Stores the contact information of the user in case they want to connect with one another	Includes Mobile Numbers and Emails	Can be accessed the one user to connect to another user	Should be authorised, possibly through OTPs
Chat Message	Strings for the text in the chats	Contains messages exchanged during chats	Text Strings	Input by users during their chats	May support encryption and a character limit

Appendix B - Group Log

B. Group Log

S. No.	Date	Timings	Venue	Description
1	09/01/2025	19:00 to 21:30	KD101	Discussed about the possible ideas which included but were not limited to a site for tenants and landlords, an automation system for attendance at hall gates, mental wellness application.
2	11/01/2025	18:00 to 22:00	RM Ground Floor	Finalised the idea about the project and submitted the initial form mentioning a very surface level overview of what we were planning to do.
3	15/01/2025	21:00 to 00:00	RM Ground Floor	Met one of the possible clients (a Ph.D. student currently living as a tenant), and then discussed among ourselves the overall structure of the application and distributed the work for SRS document.
4	18/01/2025	20:00 to 23:00	Hall 12	Discussed the progress on SRS document, the UI designs to be used in the SRS and divided the remainder of the work between 2 teams one for UI and one for description.
5	21/01/2025	20:30 to 22:00	RM Ground Floor	Finalised the UI design and divided the final bits of

				work left intending to finalize the SRS except for a few details in the following meet.
6	23/01/2025	20:00 to 00:00	RM Ground Floor	Finalized the SRS document, including all the remaining sections like index, and made everything coherent.