SRS Document

1. Project Title: My Room

2. Category: Web Application

3. Introduction

3.1. Purpose

When developing software, the requirements stage involves the creation of Software Requirements Specifications (SRS) (also called a requirements document). This report is created after all requirements have been collected, assessed, and laid out, serving as a foundation for software engineering initiatives. To help clients, determine whether the report (SRS) meets their needs, it is a formal report that also acts as a software representation. The prerequisites for both users and the system are carefully described as well.

The purpose of this SRS is to specify the requirements for the room/apartment finder web application, which will allow users to search for and list properties for lease or sale and make the process of finding rooms/apartments seamless.

3.2. Project Scope

The proposed room/apartment finder web application is a platform that allows users to search for and find rooms or apartments to rent or buy. The room/apartment finder web application will include a login and registration system, a chat feature, a comment feature, a property listing feature, a map showing the location of properties, a search and filtering function, a feature for reporting inappropriate listings, etc. The application will also allow users to upload images of their properties. The application will be developed making it proper responsive and optimized for the use in the desktop websites and for mobile devices as well. It allows users to list their own properties for rent or sale, and to interact with other users through a chat feature.

4. Overall Description

4.1. Product Description

The room/apartment finder web application will be developed as a web-based application that can be accessed from any device with a web browser. It will be developed using React for the frontend and Node.js for the backend and it will use a Mongo DB database to store user and property information.

4.2. User Characteristics

My Room web application will be developed as a web-based application that can be accessed from any device with a web browser. It will be developed using React for the frontend and Node.js for the backend and it will use a Mongo DB database to store user and property information.

4.3. Constraints

There may be constraints on the development of the room/apartment finder web application, such as budget constraints, time constraints, or technical constraints. Some of the constrains are listed below:

i. Product may not be completed on time as the project might take longer than expected.
ii. There could be the problem with budget as some APIs may not be free and should be purchased.
iii. Some frameworks, API, technology may not work on the device that the project is being built on.
iv. There might occur some technical difficulty like some software, frameworks not working properly on the device this project is being developed.

5. System Perspective

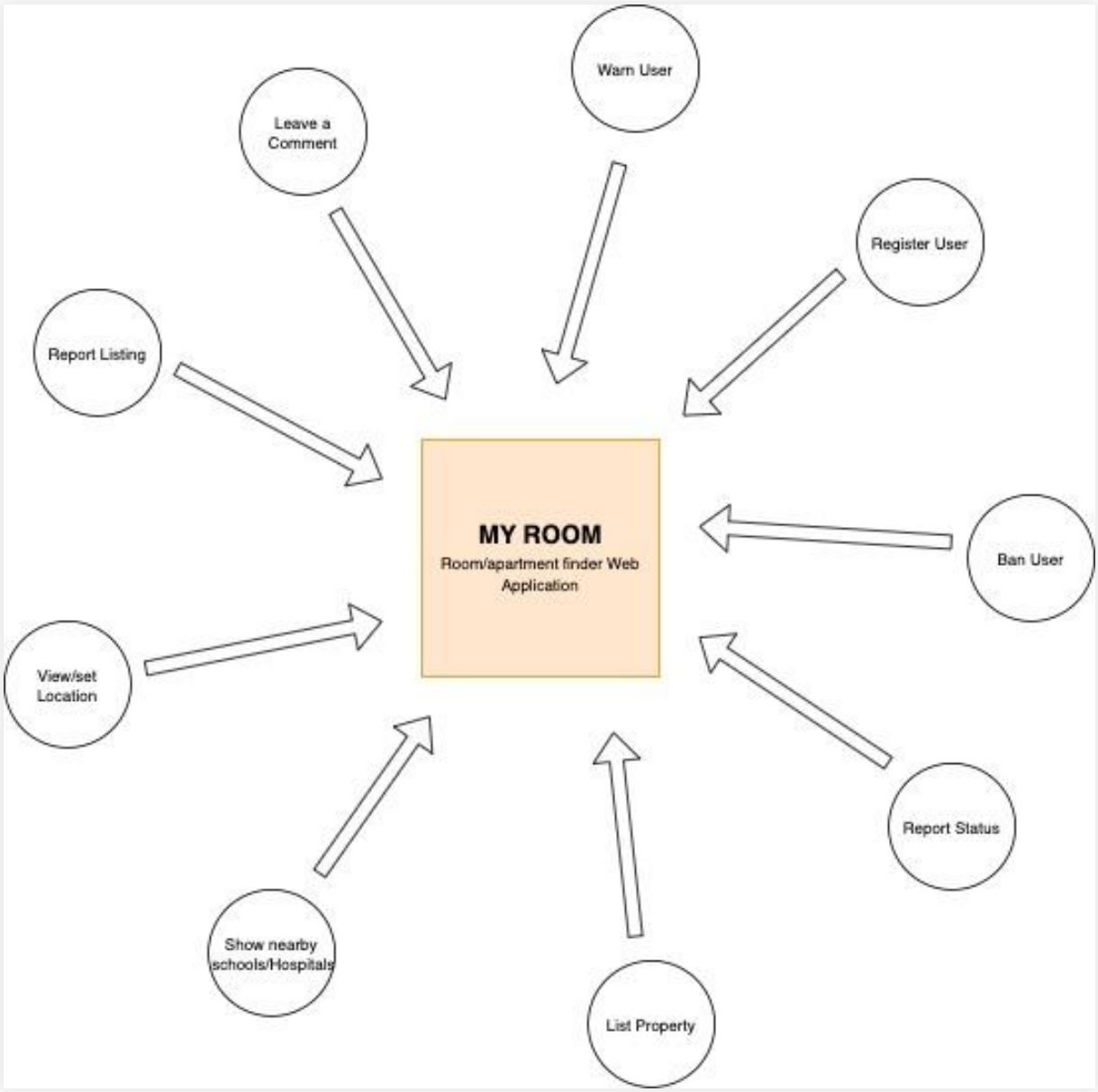


Figure 1: System Perspective

6. System Features

a. Primary Features

i. Online Ad Posting of the property to rent.
ii. Seller can Specify either to lease or to sell.
iii. The user is divided into buyer and seller however they both can sell and buy the property listed.
iv. There will be the expiry date of 30 days on every ad posted.
v. Time passed after the ad is posted will be visible to the user.
vi. The ad posted by the seller will disappear after the expiry time passes and the seller will be notified about the removal of ad.
vii. The seller could further decide to repost the ad is the property is not yet sold out or rented.
viii. Embedded map will be shown in the details of the property added by the seller which is a must and seller cannot post any property without the map showing property's location.
ix. There will be a section for the images of the property including the surrounding of the property, seller should upload the image of the property before making the ad publicly available.
x. Property filter option will be available to the users which will help the user filter the property listed for "Rent Only" "Buy Only".
xi. User can search up for the property they want, further this function enables user to search the property based on nature of the property like location, price range, recent ads, property size, etc.
xii. Buyer will be able to choose the property they want and add it to their Wishlist.
xiii. Seller can specify what luxury or facilities are available in the property they are selling like, furnished or non-furnished, parking space, Water, Electricity, bathroom, etc.

b. Secondary Features

i. Near-by schools/ markets and hospitals can be seen easily in the map for property.
ii. Property filter with pets allowed or not allowed.
iii. Seller could upload videos of the property through which buyer can have virtual tour of the property.
iv. Report feature on each ad so that user can report the ads that are not good for the site.
v. Admin will receive the report notification and can remove that ad or block the user if they violate the terms more than once.

7. External Interface requirements.

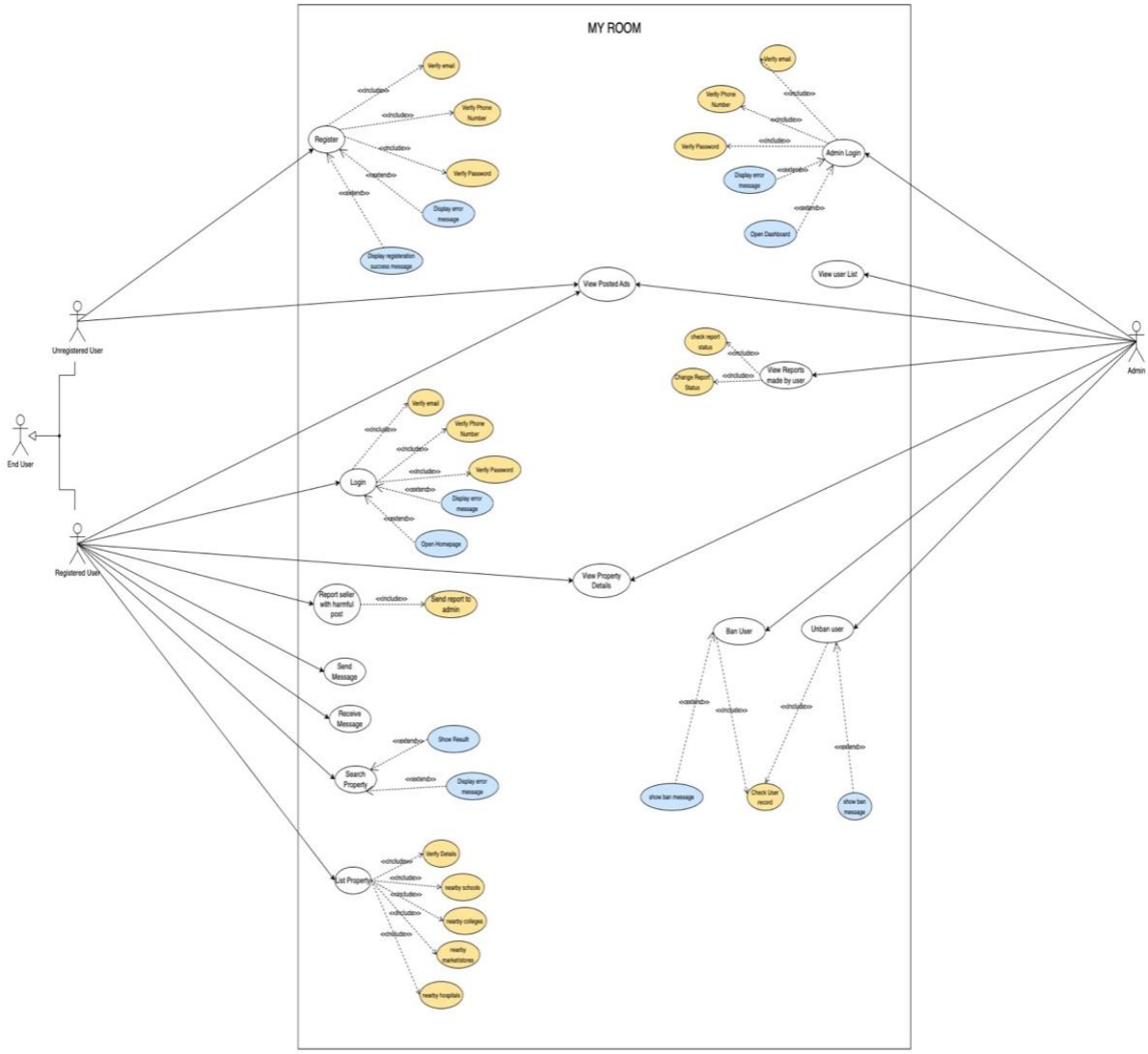
- **User interfaces:** The room/apartment finder web application will have a user interface that allows users to search for properties, list their own properties, chat with other users, and leave comments on property descriptions. The interface will be responsive, meaning it will adjust to the size of the device it is being viewed on. The interface will also include navigation links to allow users to access different parts of the application and will have a search bar for users to enter their search criteria. This whole interface will be as modern minimal and effective as possible.

- **Hardware interfaces:** The room/apartment finder web application will not require any specific hardware interfaces.

- **Software interfaces:** The room/apartment finder web application will interface with the Mongo DB database to store and retrieve user and property information. It will also interface with a mapping library like Google Maps to display the location of properties and nearby schools and hospitals.

- **Communication interfaces:** The room/apartment finder web application will use a real-time communication library like Socket.io to implement the chat feature.

8. Use case Diagram.



9. Flow Chart for the whole application.

9.1. Flow chart of user as seller.

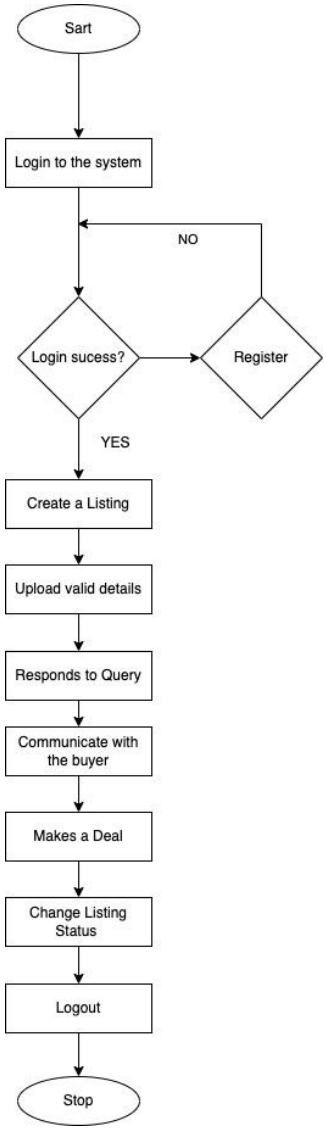


Figure 2: Flowchart of user as seller.

9.2. Flow chart of user as buyer.

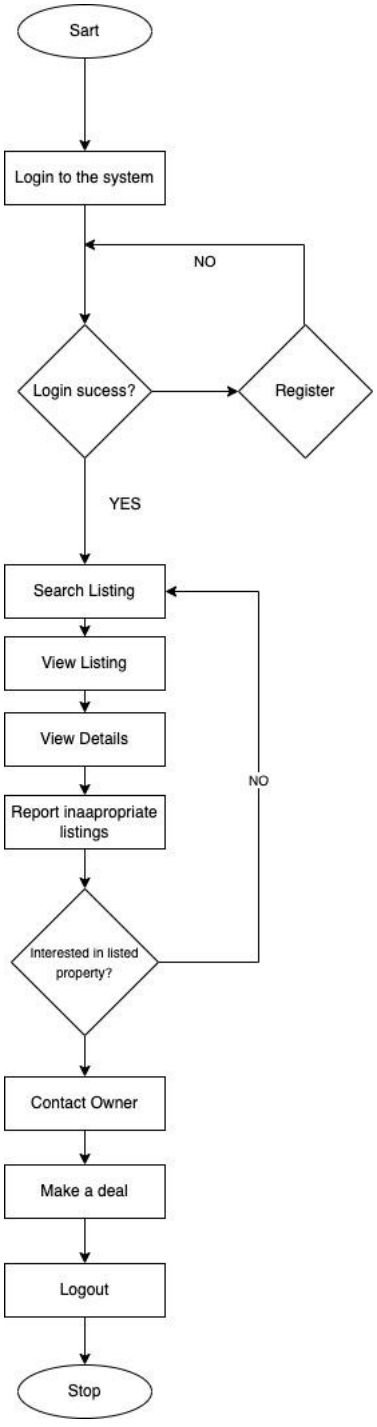


Figure 3: Flowchart of user as buyer.

9.3. Flow chart of Admin.

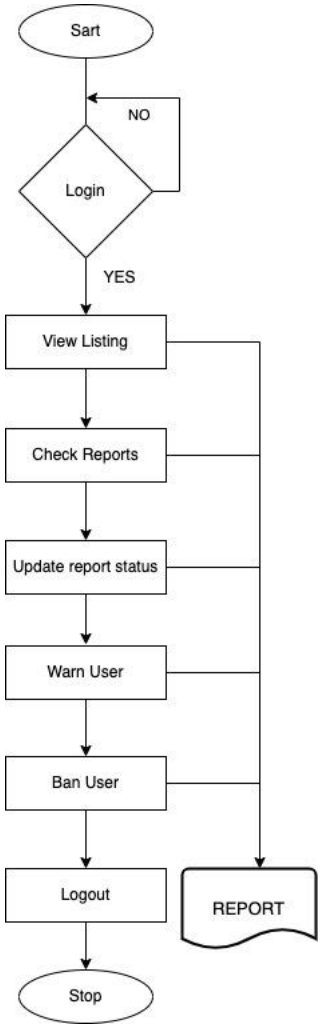


Figure 4: Flowchart of user as buyer.

10. Process Design (UML)

10.1. Actors

Actors: User (General People) and Admin

Table 1: Actors of the web application

10.2. Register

10.2.1. High Level use case

Preconditions: The user does not have an account with My room.

Name: Register
Actor: User
Description: The user opens the registration page and fills the form then, they click the "register" button, and the application checks the phone and email address to ensure it is unique. If it is unique, the application creates a new user record with the provided information and logs the user in. If the email address is already in use, the application displays an error message.

Table 2: High level use case for register

10.2.2.Use case Diagram.

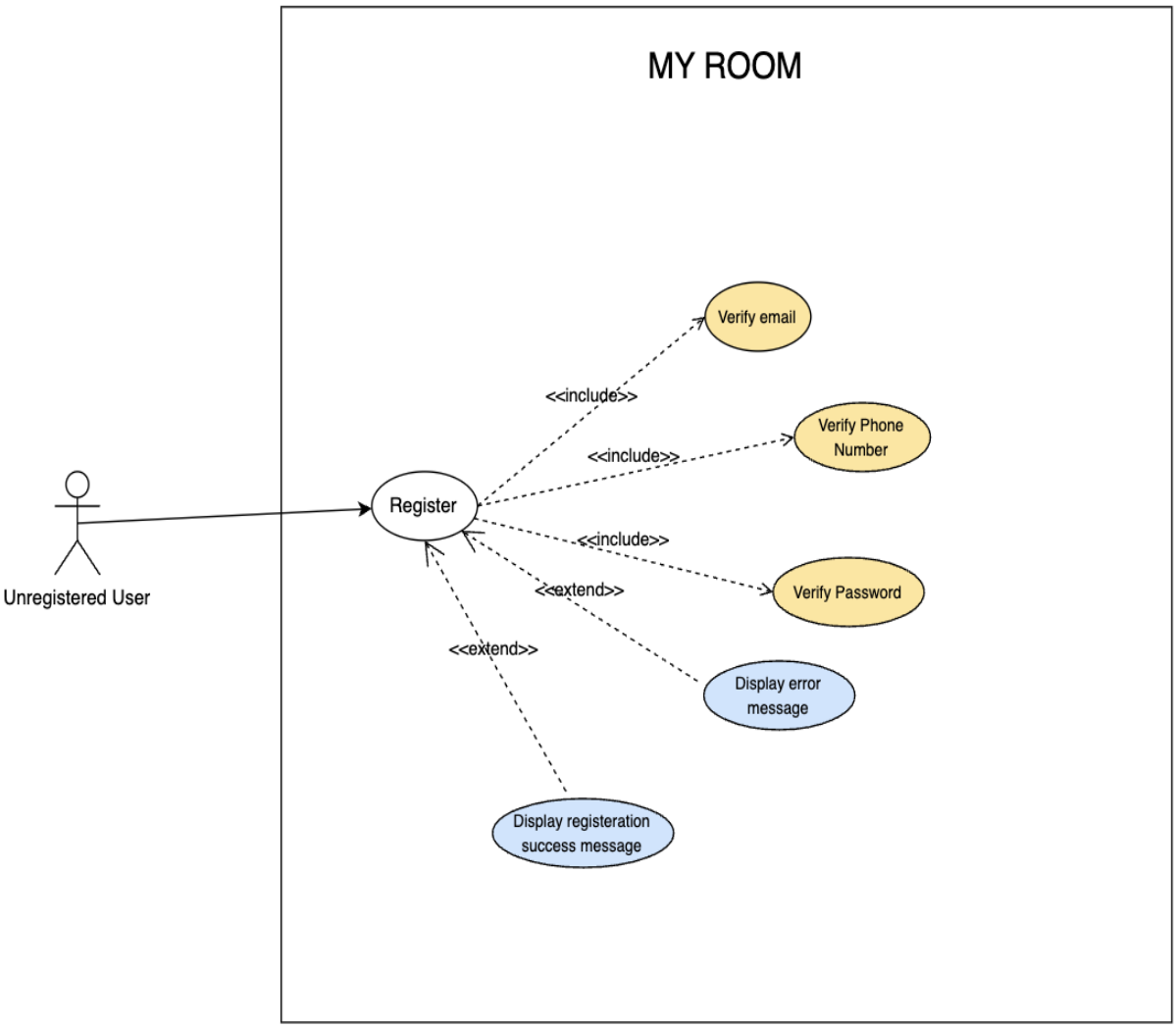


Figure 5: Register Use Case Diagram

10.3. Login

10.3.1.High Level use case

Preconditions: The user has an account with My Room.

Name: Login
Actor: User
Description: The user needs to go to the login page and enter their email address and password. Then, clicks the "login" button and the system will check if their email and password are correct. If they are, the system will let the user log in and take them to the homepage of My Room. If the email and password are not correct, the website will show an error message. **Exceptions:** If the user does not have an account, user can click the "register" button to create a new account.

Table 3: High Level Use case diagram for Login

10.3.2.Login Use case Diagram of User

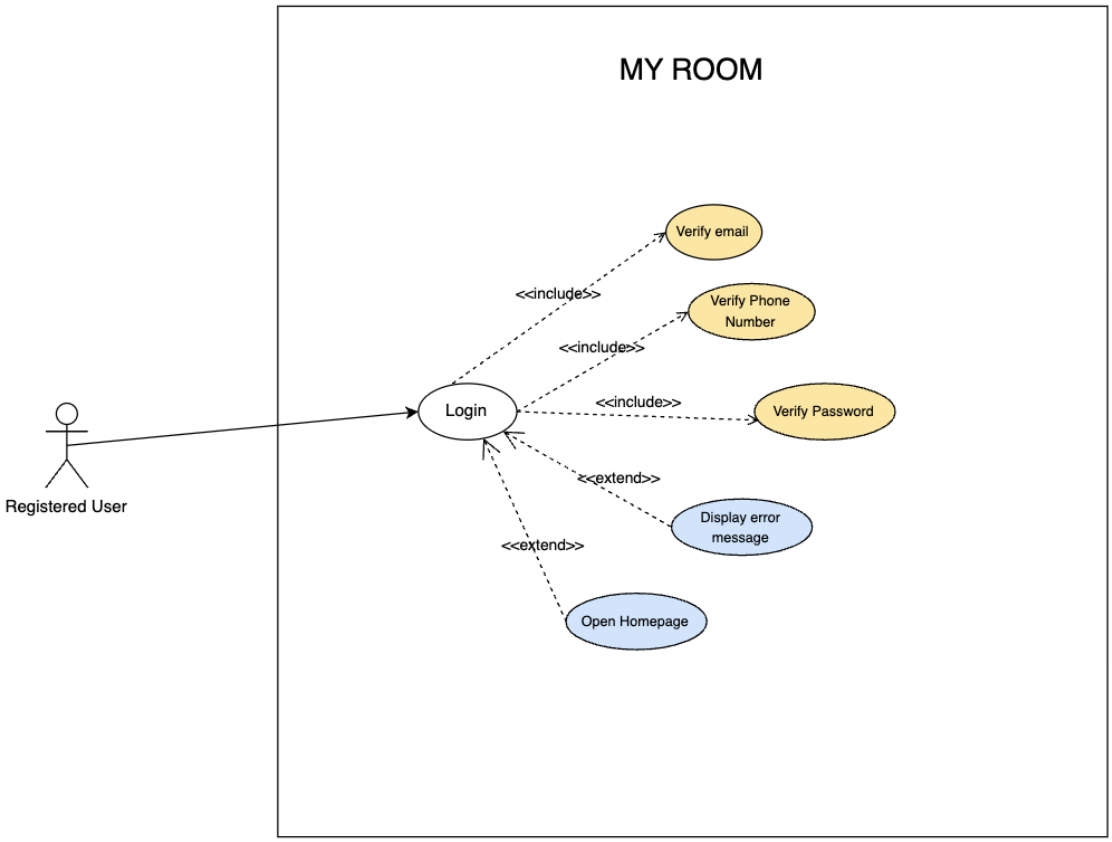


Figure 6: User's Login Use Case diagram

10.4. Property Listing

10.4.1.High Level use case

Name: Property Listing
Actor: User
Description: The user needs to be logged in and click the "list property" button. They can then enter information about their property, like type, rent amount, etc. and further upload images and click "submit". new property listing with the provided information will be created and displayed on the website for other users to see. **Exceptions:** If the user is not logged in, they will be redirected to login page upon clicking on list property button.

Table 4: High Level Use case Diagram for property Listing

10.4.2.Use case Diagram.

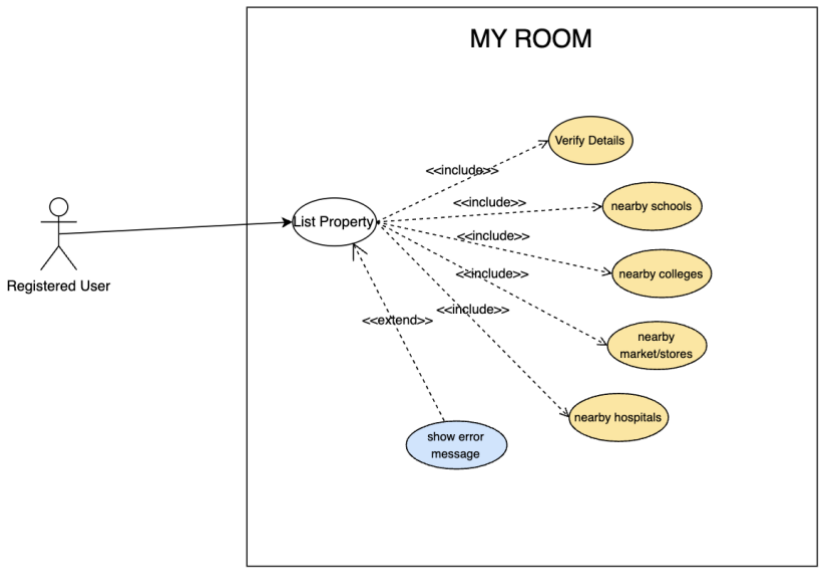


Figure 7: Property Listing Use case Diagram

10.5. Delete Listing

10.5.1.High Level use case

Name: Delete Listing
Actor: User
Description: The user needs to be logged in and select the ad they wish to delete from their dashboard. Then a form will be displayed to the user with two buttons update Ad and Delete ad. Then they can delete the ad by clicking on the delete ad button. **Exceptions:** If the user is not logged in, they will be redirected to login page upon clicking on list property button.

Table 5: High Level Use case Diagram for property Listing

10.5.2.Use case Diagram.

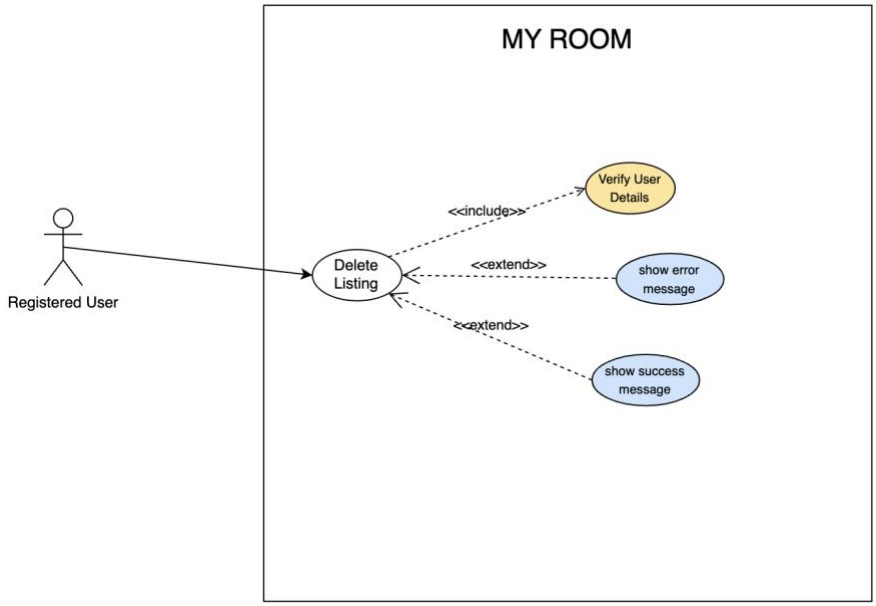


Figure 8: Delete Listing Use case Diagram

10.6. Map View

10.6.1.High Level use case

Name: Map Location
Actor: User
Description: User needs to click on a property listing to view its details. The website will show the property description page, which includes all the details of the property. The user can see a map with the location of the property marked on it in location section.

Table 6: High Level Use case diagram for map Location

10.7. Show Nearby schools, colleges, hospitals, stores.

10.7.1.High Level use case

Name: Show Nearby Schools, Hospitals, College, Stores
Actor: User
Description: the user needs to click on a property listing. They can then click the "view nearby schools/hospitals" toggle button in the map view section, and the website will show a map with the location of the property and nearby schools and hospitals.

Table 7: High Level Use case diagram for show nearby schools/hospitals

10.8. Search Feature

10.8.1.High Level use case

Name: Advance Search
Actor: User
Description: They can then enter their search criteria and click the "search" button. The web application will look for properties that match the search criteria entered by the user and display a list of them on the search results page. If the user does not fill out the search form correctly or the criteria, they entered are not valid, the website will display an error message.

Table 8: High Level Use case diagram for Advance search

10.8.2.Use Case Diagram

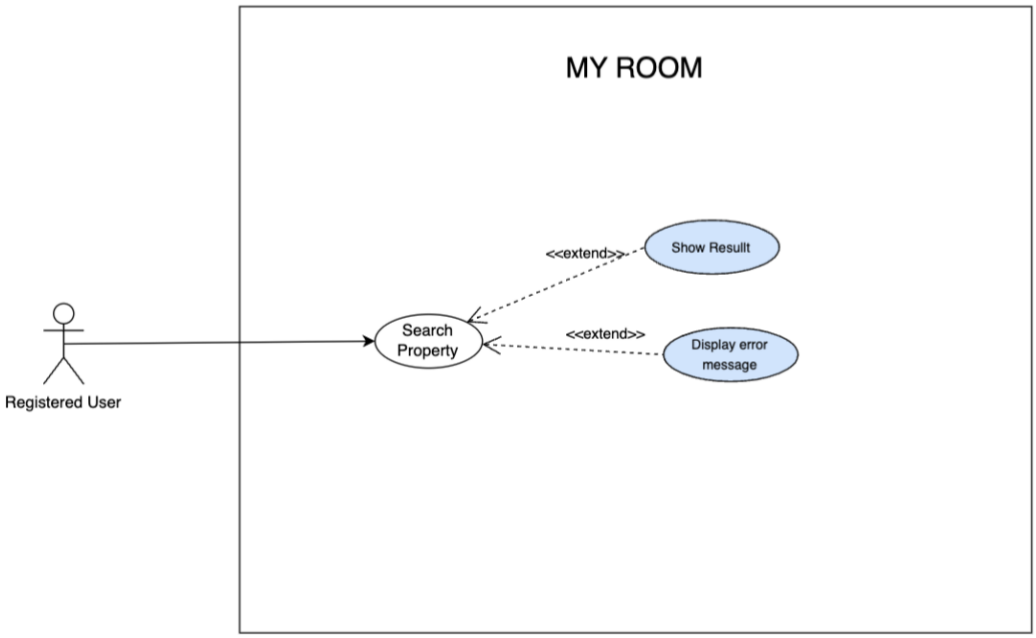


Figure 9: Search Feature Use Case diagram

10.9. Chat Feature

10.9.1.High Level use case

Name: Chat
Actor: User
Description: The user clicks on the chat button of navbar or property description page. Then the application displays the chat window accordingly either list of people or specific person's chat box. The user can then send and receive the messages. **Exceptions:** If the user is not logged in, they will be redirected to login page after clicking the chat button.

Table 9: High Level Use Case Diagram for Chat Feature

10.9.2.Use case diagram.

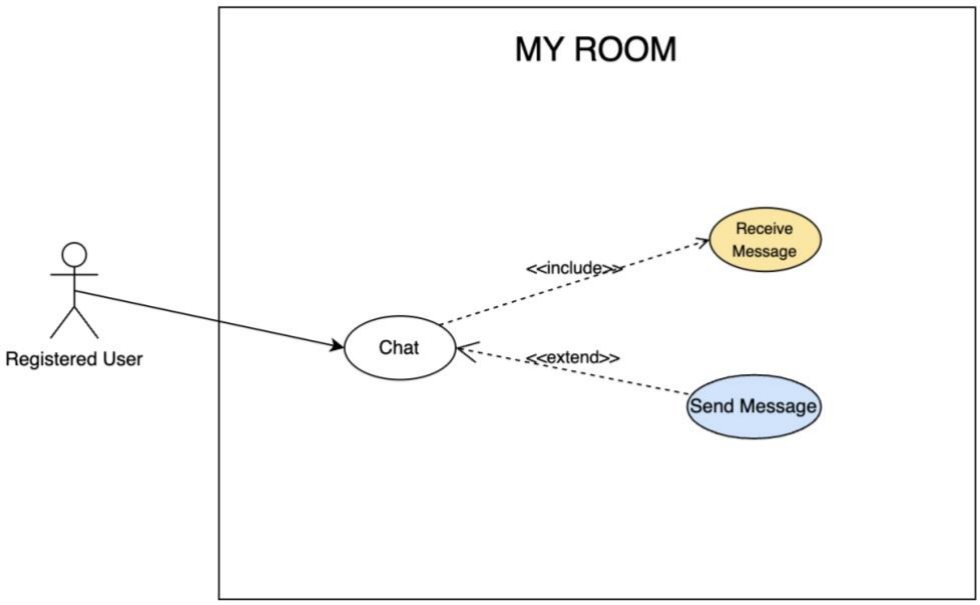


Figure 10: Chat feature Use Case Diagram

10.10. Report Feature

10.10.1. High Level use case

Name: Report
Actor: User
Description: The user must be logged in and visit the Property Description page. They can then click the "report" button, which will open a form for them to type in a reason for the report. After the user submits the form, the application will notify the admin about the report. **Exceptions:** If the user is not logged in, they will be redirected to login page after clicking the chat button.

Table 10: High Level Use Case Diagram for Report Feature

10.10.2. Use Case Diagram

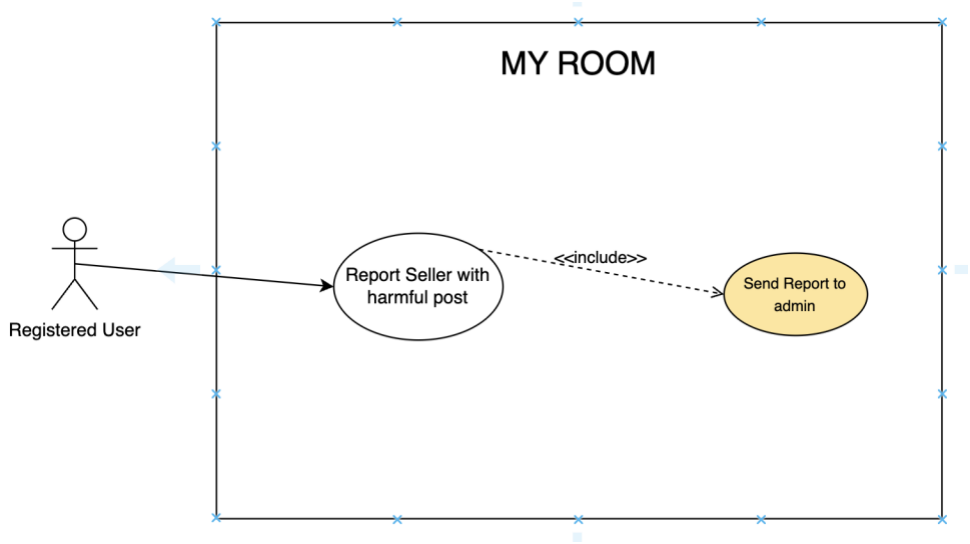


Figure 11: Use Case Diagram for report

10.11. Report Review

10.11.1. High Level use case

Name: Report Review
Actor: Admin
Description: Admin must first log in and click the "row of the table" of the user who is reported. Admin can see the reported users in user list and can select one to review. The admin can then review the report and the associated property listing. They can decide to dismiss the report or take action on the user (e.g., warn the user, ban the user, remove the report). They must click "submit" to save their decision."

Table 11: High Level Use Case Diagram for Report Review Feature

10.11.2. Use case Diagram.

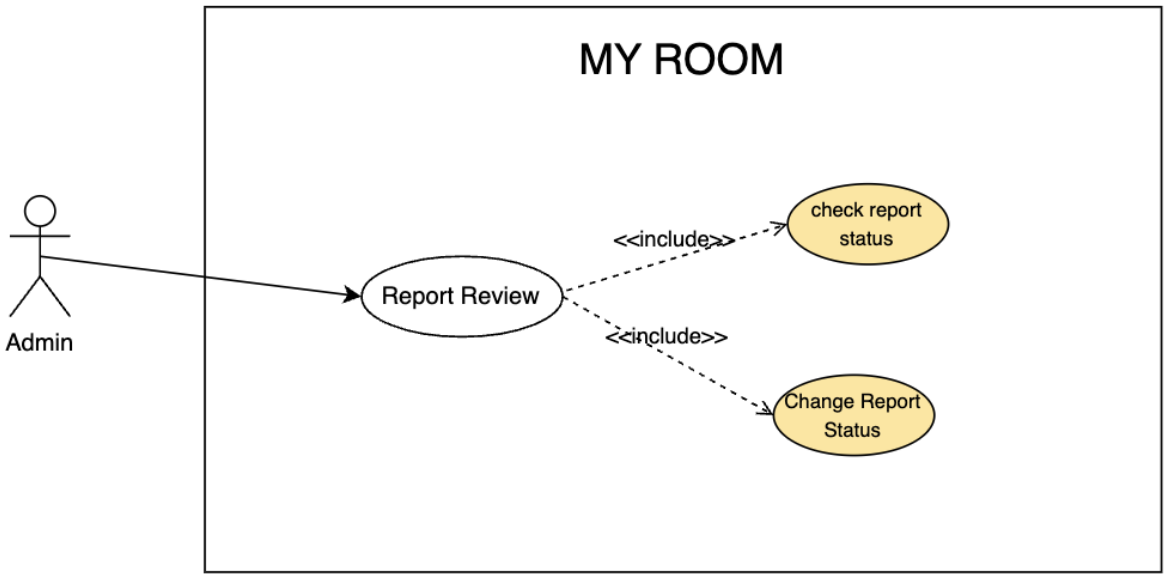


Figure 12: Report Review Use Case Diagram

10.12. Ban User

10.12.1. High Level use case

Name: Ban User
Actor: Admin
Description: Admin must click the "ban user" button on the report review page. A confirmation message will appear and the admin must confirm to ban the user. The application will then ban the user and show a success message."

Table 12: High Level Use Case Diagram for Ban user feature

10.12.2. Use case diagram.

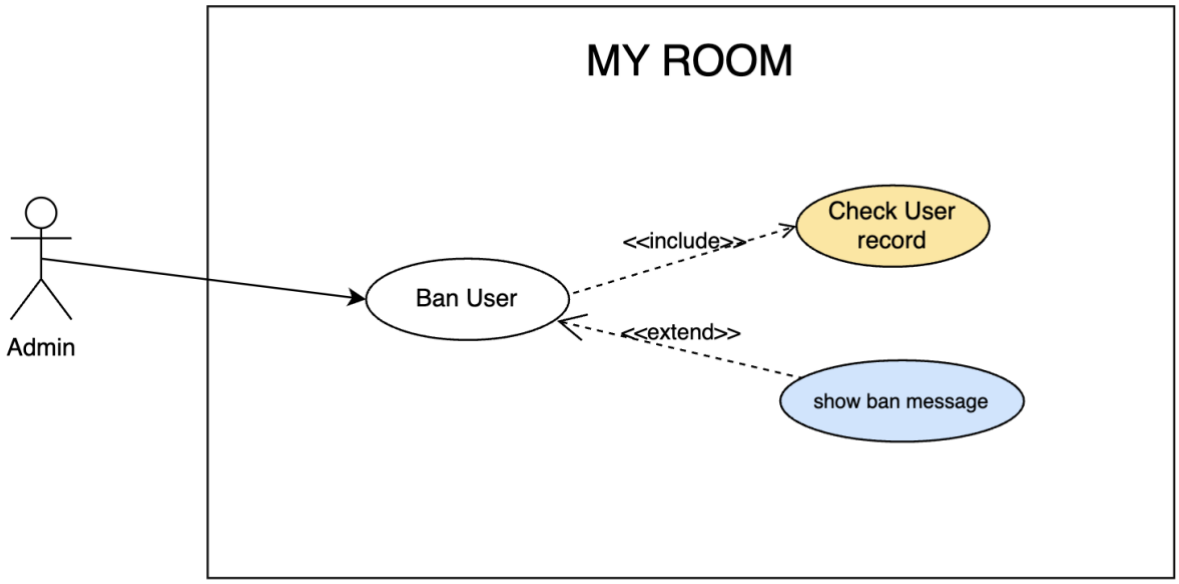


Figure 13: Ban user use case diagram

10.13. Undo Report

10.13.1. High Level use case

Name: Undo report
Actor: Admin
Description: Admin must click the "examined" button on the report review page. A confirmation message will appear and the admin must confirm to ban the user. The application will then ban the user and show a success message."

Table 13: High Level Use Case Diagram for Ban user feature

10.13.2. Use case diagram.

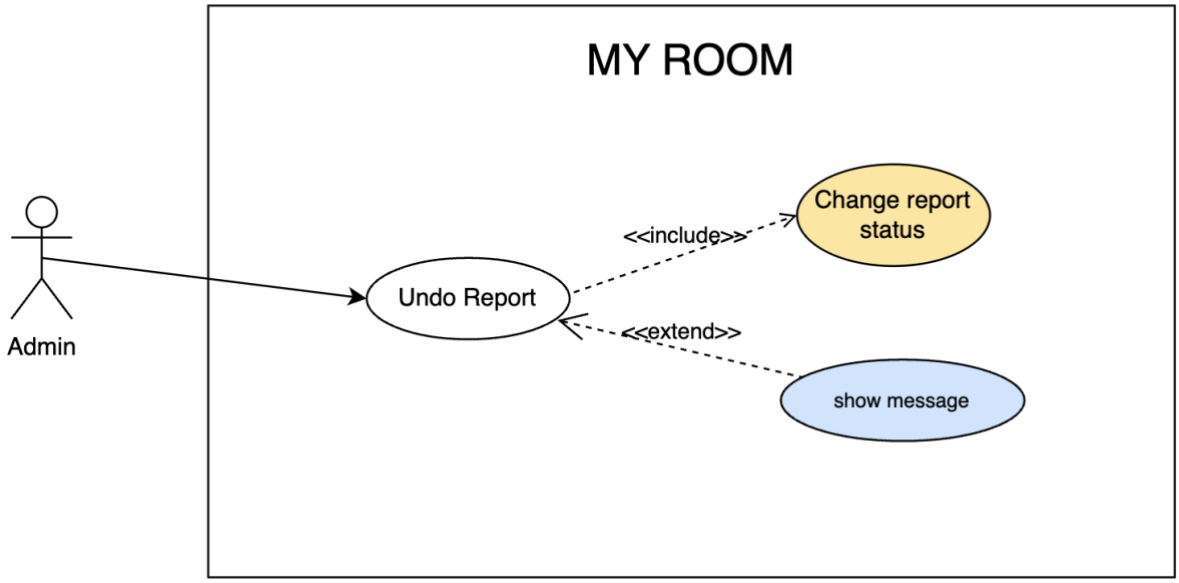


Figure 14: Undo report use Case diagram.

11. Functional Requirements

11.1. Register Feature

<table>
<tr><th>ID</th><th>Requirement Description</th></tr>
<tr><td>FR1</td><td>People can register and create an account to use the web application for listing the property.
<table>
<tr><th>S. Req.ID</th><th>System Requirement</th></tr>
<tr><td>R1</td><td>User can should be able to view the register form.</td></tr>
<tr><td>R2</td><td>Register form should validate the info given by the user</td></tr>
<tr><td>R3</td><td>System should check is the entered phone number and email is unique to the application or not</td></tr>
<tr><td>R4</td><td>Registration successful message should be shown to the user</td></tr>
<tr><td>R5</td><td>If any error is found system shall notify the user with proper message</td></tr>
</table></td></tr>
</table>

Table 14: Register Function Functional Requirements

11.2. Login Feature

<table>
<tr><th>ID</th><th>Requirement Description</th></tr>
<tr><td>FR2</td><td>People can login into the web application after entering their username and password
<table>
<tr><th>S. Req.ID</th><th>System Requirement</th></tr>
<tr><td>R1</td><td>The login form must be accessible from the home page of the application.</td></tr>
<tr><td>R2</td><td>The login form must validate the email and password entered by the user.</td></tr>
<tr><td>R3</td><td>The login form must allow the user to reset their password if they have forgotten it.</td></tr>
<tr><td>R4</td><td>The login form must display an error message if the login attempt is unsuccessful.</td></tr>
<tr><td>R5</td><td>User should be redirected to home page is the login is successful.</td></tr>
</table></td></tr>
</table>

Table 15: Login Feature Functional Requirement

11.3. Property Listing Feature

<table>
<tr><th>ID</th><th>Requirement Description</th></tr>
<tr><td>FR3</td><td>People can list their property for sell/lease after filling the form properly.
<table>
<tr><th>S. Req.ID</th><th>System Requirement</th></tr>
<tr><td>R1</td><td>user interface is needed to allow users to navigate to the listing form from where it is most accessible like navbar.</td></tr>
<tr><td>R2</td><td>User should be able to specify either to lease or to sell.</td></tr>
<tr><td>R3</td><td>Listing form should contain almost all the description that is required before the listing.</td></tr>
<tr><td>R4</td><td>User should be able to select the location of the property.</td></tr>
<tr><td>R5</td><td>User should be able to upload image/video of the property.</td></tr>
<tr><td>R6</td><td>Listing should disappear after the expiry time set by the user or default expiry time.</td></tr>
<tr><td>R7</td><td>Form should have a submit button which validates the form and add the property to the database and listing page.</td></tr>
<tr><td>R8</td><td>Proper Listing should be displayed to the user.</td></tr>
<tr><td>R9</td><td>In case error is detected user should be notified with proper message.</td></tr>
</table></td></tr>
</table>

Table 16: Property Listing feature functional requirement

11.4. Map Location Feature

<table>
<tr><th>ID</th><th>Requirement Description</th></tr>
<tr><td>FR4</td><td>User can view the location of property in map view on property description page.
<table>
<tr><th>S. Req.ID</th><th>System Requirement</th></tr>
<tr><td>R1</td><td>User interface is needed to display the map on the property description page.</td></tr>
<tr><td>R2</td><td>Map should show the location with a proper visible marker</td></tr>
<tr><td>R3</td><td>System is required to enable interactivity on the map, such as the ability to zoom in or out and pan around</td></tr>
</table>
</td></tr>
</table>

Table 17: Map Location Feature Functional Requirement

11.5. Show Nearby Schools and Hospitals Feature

<table>
<tr><th>ID</th><th>Requirement Description</th></tr>
<tr><td>FR5</td><td>User can view the nearby schools and hospitals around the property listed.
<table>
<tr><th>S. Req.ID</th><th>System Requirement</th></tr>
<tr><td>R1</td><td>User interface is needed to display the map that shows nearby schools and hospitals on the property description page.</td></tr>
<tr><td>R2</td><td>System is required to enable interactivity on the map, such as the ability to zoom in or out and pan around</td></tr>
<tr><td>R3</td><td>Multiple markers should be used to show the schools/hospitals</td></tr>
</table>
</td></tr>
</table>

Table 18: Nearby schools/hospitals Functional Requirement

11.6. Advance Search Feature

<table>
<tr><th>ID</th><th>Requirement Description</th></tr>
<tr><td>FR6</td><td>User can search the properties with advance filtering and sorting options.
<table>
<tr><th>S. Req.ID</th><th>System Requirement</th></tr>
<tr><td>R1</td><td>A user interface is needed to search the property and an advance search form should be available to the user.</td></tr>
<tr><td>R2</td><td>Search button should be present to search after selecting different search options</td></tr>
<tr><td>R3</td><td>System should have multiple advance search options like price, rent, buy, etc.</td></tr>
<tr><td>R4</td><td>System should properly show the results of search to the user in appealing way.</td></tr>
</table>
</td></tr>
</table>

Table 19: Advance Search Feature Functional Requirement

11.7. Chat Feature

<table>
<tr><th>ID</th><th>Requirement Description</th></tr>
<tr><td>FR7</td><td>User buyer/seller can chat with each other and inquire about properties.
<table>
<tr><th>S. Req.ID</th><th>System Requirement</th></tr>
<tr><td>R1</td><td>A user interface is needed to direct the user to the chat feature like an icon in navbar and chat button in property description page.</td></tr>
<tr><td>R2</td><td>Users should be able to send and receive message.</td></tr>
<tr><td>R3</td><td>System should have a proper user interface so show the list of previous chats with other user and to start a new conversation</td></tr>
<tr><td>R4</td><td>System is required to support real-time messaging.</td></tr>
</table>
</td></tr>
</table>

Table 20: Chat feature Functional requirement

11.8. Report Feature

<table>
<tr><th>ID</th><th>Requirement Description</th></tr>
<tr><td>FR8</td><td>User can report inappropriate listing from the property description page.
<table>
<tr><th>S. Req.ID</th><th>System Requirement</th></tr>
<tr><td>R1</td><td>A user interface is needed to allow users to report an inappropriate listing from the property description page, and a system is required to store the report in the database.</td></tr>
<tr><td>R3</td><td>System should send the reports made to the admin for the review of the report</td></tr>
</table>
</td></tr>
</table>

Table 21: Report feature functional requirement

11.9. Admin Review

<table>
<tr><th>ID</th><th>Requirement Description</th></tr>
<tr><td>FR9</td><td>Admin can review the reports made by the user and take actions accordingly.
<table>
<tr><th>S. Req.ID</th><th>System Requirement</th></tr>
<tr><td>R1</td><td>Admins should be able to view a list of reports, and a system is required to retrieve the list of reports from the database and display it on the page.</td></tr>
<tr><td>R2</td><td>Admin Should be able to view the contents of reports.</td></tr>
<tr><td>R3</td><td>Admin should be able to take necessary actions to the reports like blacklisting the user, warning the user, etc.</td></tr>
<tr><td>R4</td><td>System should allow the admin to view the status of the report and change the status after reviewing.</td></tr>
</table></td></tr>
</table>

Table 22: Admin review functional requirement

11.10. Expiry date of Ad

<table>
<tr><th>ID</th><th>Requirement Description</th></tr>
<tr><td>FR10</td><td>Listings have the maximum expiry time period of 30 days.
<table>
<tr><th>S. Req.ID</th><th>System Requirement</th></tr>
<tr><td>R1</td><td>System is required to set the maximum expiry date for each listing to 30 days from the time it was created.</td></tr>
<tr><td>R2</td><td>System is required to check for listings that have reached their expiry date and remove them from the application.</td></tr>
<tr><td>R3</td><td>A user interface is needed to allow users to repost their listings after the expiry date, and a system is required to verify the reposted listing before making it live on the application again.</td></tr>
<tr><td>R4</td><td>User interface is needed to display the expiry date of a listing to the user who created it.</td></tr>
</table></td></tr>
</table>

Table 23: Expiry time feature functional requirement

11.11. Ban User Feature

<table>
<tr><th>ID</th><th>Requirement Description</th></tr>
<tr><td>FR11</td><td>Admin can ban the user if user violates some conditions more than once.
<table>
<tr><th>S. Req.ID</th><th>System Requirement</th></tr>
<tr><td>R1</td><td>A user interface is needed to allow admins to initiate the banning process for admin.</td></tr>
<tr><td>R2</td><td>System is required to prevent banned users from accessing the application.</td></tr>
<tr><td>R3</td><td>Interface is needed to allow admins to view a list of banned users for admin.</td></tr>
<tr><td>R4</td><td>Admins should be able to unban a user.</td></tr>
</table></td></tr>
</table>

Table 24: Ban user functional requirement

12. Non-Functional Requirements

- **Performance requirements:** The room/apartment finder web application should have a fast response time and be able to handle many users simultaneously. It should also be able to handle large amounts of data, such as property listings and user information.

- **Testing requirements:** The room/apartment finder web application should be thoroughly tested using black box and white box testing method before it is released to ensure that it is functioning as intended. Testing should include functional testing, performance testing, accuracy testing, etc

- **Free from Malware:** The room/apartment finder web application should be free from malware and viruses there should not by any hidden malware or viruses that will cause harm to the user.

- **Proper responsive:** The application should be fully web responsive so that the user using any mobile or desktop device can use the application with same friendliness in all state of window size.

13. Resource Requirements

Software Requirements

- **VS Code**

Figure 15: Vs code Logo

(vscode, 2022)Visual Studio Code is a lightweight yet capable source code editor that runs on your desktop for Windows, macOS, and Linux. It comes with built-in JavaScript, TypeScript, and Node.js support, as well as a powerful ecosystem of extensions for other languages and runtimes (including C++, C#, Java, Python, PHP, Go, and.NET).

- **HTML**

HTML (Hypertext Markup Language) is the most basic building block of the Web. It defines the meaning and structure of web content (mozilla, 2022). Since this project is a web-based application HTML is a must.

- **CSS**

CSS will be used in this project to give styling to the project. And to make the application responsive and as user friendly as possible.

- **Bootstrap**

Figure 16: (Bootstrap, 2022) Bootstrap Logo

Bootstrap is a free, open-source toolkit for developing websites and web applications. It was created to make it easier to build responsive, mobile-first designs using a set of pre-designed templates and styles. (Andrew Zola, 2022). Bootstrap will be used as the CSS framework to complete the project smoothly.

- **React JS**

Figure 17: (Reactjs ORg, 2022)React Logo

"React.js is a tool developed by Facebook that allows developers to create interactive user interfaces and web applications in a fast and efficient manner. It uses JavaScript and reduces the amount of code needed compared to using vanilla JavaScript. It is open source, which means it is available for anyone to use and modify. (Herbert, 2022) As this project will be a single page application react is going to be used.

- **Node js**

Figure 18: (nodejs, 2022) Nodejs Logo

Node.js is a tool used by developers to build web applications that run on the server side, rather than in a user's browser. It is designed to handle large amounts of data and utilizes an event-driven, asynchronous model. Node.js was created in 2009 and has regularly updated versions, with the latest release being version 15.14 in April 2021. (Taha Sufiyan, 2022)