



COMSATS University, Islamabad Pakistan

Assignment - 06
Project Final Report
For
Renter-Proprietor Coordination System

Submitted By:

Hunia Nadeem CUI/FA20-BCS-024/ISB

Idrees Ahmed Ghazi CUI/ FA20-BCS-025/ISB

Supervisor
Mr. Tehseen Riaz Abbasi

Date of Submission: 19th June, 2022

Original Version 1.0

Bachelor of Science in Computer Science (2020-2024)

The candidate confirms that the work submitted is their own and appropriate credit has been given where reference has been made to the work of others.



COMSATS University, Islamabad Pakistan

Renter-Proprietor Coordination System

A project presented to
COMSATS University, Islamabad

In partial fulfillment
of the requirement for the degree of

Bachelors of Science in Computer Science (20xx-20xx)

By

Hunia Nadeem

CUI/FA20-BCS-024/ISB

Idrees Ahmed Ghazi

CUI/ FA20-BCS-025/ISB

DECLARATION

We hereby declare that this software, neither whole nor as a part has been copied out from any source. It is further declared that we have developed this software and accompanied report entirely on the basis of our personal efforts. If any part of this project is proved to be copied out from any source or found to be reproduction of some other. We will stand by the consequences. No Portion of the work presented has been submitted of any application for any other degree or qualification of this or any other university or institute of learning.

Hunia Nadeem

Idrees Ahmed Ghazi

Executive Summary

Property dealing and renting houses is a daily grind that can drain one mentally and physically. The **Renters/Proprietors Coordination Application** is a smartphone and web-based system that intends to ease the back and forth communication between multiple proprietors and renters. It provides a **single platform** that maintains all the records related to the properties. Automating the post-renting property phase, the problems regarding **payment frauds, delayed communication, and ignored/neglected complaints** will be resolved. **Machine Learning and Artificial Intelligence techniques** will contribute to secure online payments and prompt query resolution through fraud detection systems and bots.

This system provides a plethora of user management and engagement options using **different portals** for tenants and landlords. The tenants will be able to **manage payments, file complaints, and maintain a record** of their rented properties. In contrast, the proprietors will also be able to manage their renters' activities, respond to complaints, and promptly resolve issues through in-app messaging. The app is feasible for all due to internet and smartphone resources availability. It will play a significant part in **easing the post-property renting phase** by saving time, cost, and energy.

Acknowledgement

All praise is to Almighty Allah who bestowed upon us a minute portion of His boundless knowledge by virtue of which we were able to accomplish this challenging task.

We are greatly indebted to our project supervisor “Mr. Tehseen Riaz Abbasi”. Without their personal supervision, advice and valuable guidance, completion of this project would have been doubtful. We are deeply indebted to them for their encouragement and continual help during this work.

And we are also thankful to our parents and family who have been a constant source of encouragement for us and brought us the values of honesty & hard work.

Hunia Nadeem

Idrees Ahmed Ghazi

Abbreviations

API	Application Programming Interface
FAQ	Frequently Asked Questions
FR	Functional Requirements
M	Module
NFR	Non-Functional Requirements
QA	Quality Assurance
RPC System	Renter-Proprietor Coordination System
SDS	Software Design Specification
SRS	Software Require Specification
STP	Software Test Plan
TC	Test Case
UC	Use case
UI	User Interface
URL	Uniform Resource Locator

Table of Contents

1.	Introduction	12
1.1	Project Major Category	12
1.2	Brief	12
1.3	Project Background.....	13
1.4	Related System Analysis/Literature Review	13
1.5	Advantages/Benefits of Proposed System.....	14
1.6	Project Scope.....	14
1.7	Modules	15
<i>1.7.1</i>	<i>Profile Management</i>	<i>16</i>
<i>1.7.2</i>	<i>User Interaction Chat Box.....</i>	<i>17</i>
<i>1.7.3</i>	<i>Rent Payment Portal.....</i>	<i>18</i>
<i>1.7.4</i>	<i>Complaint Management Portal</i>	<i>19</i>
<i>1.7.5</i>	<i>Tenant Management Portal</i>	<i>20</i>
<i>1.7.6</i>	<i>Property Management Portal</i>	<i>21</i>
<i>1.7.7</i>	<i>Push Notification</i>	<i>22</i>
<i>1.7.8</i>	<i>Help and Support</i>	<i>23</i>
1.8	System Limitations/Constraints	23
1.9	Tools and Technologies	24
1.10	Relevance to Course Material	24
1.11	Design and Process Methodology for this Project	25
<i>1.11.1</i>	<i>Process Methodology.....</i>	<i>25</i>
<i>1.11.2</i>	<i>Design Methodology</i>	<i>25</i>
1.12	Team Members Work Division	25
2.	Problem Definition	26

2.1	Problem Statement	26
2.2	Problem Solution for the Proposed System.....	26
2.3	Deliverables and Development Requirements	26
3.	Requirement Analysis	28
3.1	Requirement Elicitation Techniques	28
3.2	Use Cases Diagram(s).....	28
3.3	Detailed Use Case (Tabular- Module Wise)	31
<i>3.3.1 List of Use Cases.....</i>		<i>31</i>
3.4	Functional Requirements.....	111
3.5	Non-Functional Requirements	212
<i>3.5.1 Reliability.....</i>		<i>212</i>
<i>3.5.2 Usability.....</i>		<i>213</i>
<i>3.5.3 Performance.....</i>		<i>213</i>
<i>3.5.4 Security</i>		<i>214</i>
4.	Implementation.....	215
4.1.	Algorithm	249
4.2.	External APIs.....	254
4.3.	User Interface.....	255
5.	Testing and Evaluation	272
6.	Conclusion and Future Work	346
6.1.	Conclusion	346
6.2.	Future Work	346
6.3.	Lessons Learnt	346
6.4.	Work Division	347
7.	References	350
8.	Appendix	351

8.1. Figma UI.....	351
8.2. Figma Prototypes.....	351
8.3. Github Repositories.....	351
8.4. Youtube	351
8.5. Guidelines	351
9. Plagiarism Report	353

Table of Figures

<i>Figure 1: Context Diagram of RPC System.....</i>	15
<i>Figure 2: Use Case Diagram for System</i>	28
<i>Figure 3: MVC Architecture Diagram</i>	215
<i>Figure 4: Entity Relation Diagram of RPC System</i>	217
<i>Figure 5: Activity Diagram of Showing Ads After Logging in</i>	231
<i>Figure 6: Activity Diagram of Muting a User</i>	232
<i>Figure 7: Activity Diagram of Payment System</i>	233
<i>Figure 8: Activity Diagram of Sending/Responding Complaints</i>	234
<i>Figure 9: Activity Diagram of Adding User Property</i>	235
<i>Figure 10: Activity Diagram of Sending Lease Renewal Request</i>	236
<i>Figure 11: Activity Diagram of Setting Ringtone</i>	237
<i>Figure 12: Activity Diagram of Sending Feedback</i>	238
<i>Figure 13: Class Diagram of RPC System</i>	240
<i>Figure 14: Sequence Diagram of Showing Ads after Logging in</i>	241
<i>Figure 15: Activity Diagram of Muting User</i>	242
<i>Figure 16: Sequence Diagram of Payment System.....</i>	243
<i>Figure 17: Sequence Diagram of Sending/Responding Complaints</i>	244
<i>Figure 18: Sequence Diagram of adding Property</i>	245
<i>Figure 19: Sequence Diagram of responding to Lease Renewal</i>	246
<i>Figure 20: Sequence Diagram of setting Ringtone.....</i>	247
<i>Figure 21: Sequence Diagram of Sending Feedback</i>	248
<i>Figure 22: User Interface of Get Started Screen.....</i>	255
<i>Figure 23: User Interface for Login Screen Figure 24: User Interface for Sign Up Screen..</i>	256

Figure 25: User Interface for Role Selection Screen	257
<i>Figure 26: User Interface for Renter Homepage</i>	
<i>Figure 27: User Interface for Renter Main Menu</i>	258
<i>Figure 9: Interface for Renter Complaint Screen</i>	
<i>Figure 10: Interface of Complaint Submission</i>	259
<i>Figure 30: Interface for Proprietor Homepage</i>	
<i>Figure 31: Interface for Proprietor Main Menu</i>	260
<i>Figure 32: Interface for Proprietor Complaint Details</i>	261
<i>Figure 34: User Interface of Side Pane Menu</i>	
<i>Figure 35: User Interface of FAQ menu</i>	262
<i>Figure 36: User Interface of Chat Bot Screen</i>	
<i>Figure 37: User Interface of Live Chat</i>	263
Figure 38: User Interface of Get Started Screen	264
Figure 39: User Interface of Login Screen	264
Figure 40: User Interface of Sign up Screen.....	265
Figure 41: User Interface of Role Selection Screen.....	265
Figure 42: User Interface of Proprietor Menu	266
Figure 43: User Interface of Add property Menu	266
Figure 44: User Interface of Complaint Menu (Proprietor)	267
Figure 45: User Interface of Renter Menu	267
Figure 46: User Interface of Adding Properties through Property Id (Renter).....	268
Figure 47: User Interface of Adding Properties through User Id (Renter)	268
Figure 48: User Interface of Renter Complaint Menu	269
Figure 49: User Interface of Complaint Submission (Renter)	269
Figure 50: User Interface of FAQ menu	270
Figure 51: User Interface of Chat with Bot	270

1. Introduction

Acquiring your own property or renting a convenient place, is a dream many people live for. Though as fascinating as the idea sounds, the process can get really tedious and draining. Not only searching the right property with just the perfect/required environment/ambience is a task, but so is the entire communication process that comes after making this difficult choice. The continuous loop of back-and-forth communication between the landlord and the tenant is both mentally and physically draining. Nonetheless, the insecurity and uncertainty in offline payments lead to some serious issues including data insecurity and payment frauds. In addition to that, time conflicts, resource delays, distance, and commitments also play a major role in causing unnecessary resistance between both the parties.

Fortunately, this issue can easily be resolved by automating the communication process of tenant with the landlord, once the property to be secured has been selected/chosen for. Not only is it smart, but it is also highly effective in resolving avoidable issues like time constraints, distance parameter, and resource delays – enabling a smooth communication channel on both ends. To make this system more reliable and secure, fraud detection system and data insecurity algorithm will be implemented using the concepts of Machine Learning and Artificial Intelligence. Accordingly, to make the system more accessible to all users, it is implemented using smart phone and web-based applications.

1.1 Project Major Category

B- Web Application/Web Application based Information System.

C- Problem Solving and AI

E- Smartphone Application

1.2 Brief

Renting and proprietary work is a very tedious job and is mentally and physically demanding. After a person gets a house on rent, a continuous loop of back-and-forth communication occurs between the landlord and the tenant. Many factors including distance, time conflicts, and commitments can lead to time and resource delays. The insecurity and uncertainty in offline payments lead to further issues including data insecurity and payment fraud.

Property work is an everyday livelihood opted by a vast majority of people. So, to bring simplicity to the entire process, we introduced the **Renter-Proprietor Co-ordination (RPC) System**, an automated environment, capable of resolving these issues. The market has similar systems, but they don't provide a single platform to cater to everything. It is a web and mobile-based application for assisting property dealers. Object-oriented design methodology is used. Other tools like Adobe Illustrator, Figma and MS Office were used for design and presentation purposes. After proper planning and use of such tools we came up with a system that not only automates the overall communication process between the proprietor and renter, but will also help reduce the consumption of extra time, money, and energy that the post-property renting process demands.

1.3 Project Background

Real Estate is a common livelihood or investment choice for people around the globe. As everything has its setbacks, so does real estate procedures. The back and forth consultation, communication and meetings between the renters and proprietors is a tiresome job. It demands much time, energy, resources and cost. Therefore, we have made this process automated by eliminating the chances of having to schedule physical meetings at specific time and location.

The RPC System is a web and mobile based application that eases and simplifies the communication process between tenants and proprietors, enables prompt response to the reported complaints from the renters, secures payment process, and automates the post-renting property phase. Unlike the current applications focused mainly on the pre-renting property phase, people who use our product will have a single platform to perform all their property management tasks, which will save them time, transport cost, and energy.

1.4 Related System Analysis/Literature Review

Following are some existing apps related to our project:

Zillow.com:

Zillow is an online real-estate marketplace which offers services like Buying, Renting, Selling properties. It displays properties on the base of their location and customer reviews. It advertises houses according to the location. Buyers or renters can find houses or properties online without any complex paperwork. The link to their website is: <https://www.zillow.com/>.

Zameen.com:

Zameen is an online property marketplace in Pakistan. It lists hundreds of real-estate dealers, and developers with their properties for sale or for rent purpose. Proprietors can add their properties to be sold or rented and buyers or tenants can look out for them online without facing any issue. The link to their website is: <https://www.zameen.com/>.

Apartments.com:

Apartments.com is also a Real-Estate marketplace on which users can advertise a property or look out for a property on sale. Users can also rent a house. The web link of their website is: <https://www.apartments.com/>.

The **Table 1** provides weakness of these apps and proposed solution that would be done in this project.

Table 1: Related System Analysis with RPC System.

Application Name	Weakness	Proposed Project Solution
• Zillow	• Zillow offers chat box system which does not have a call system nor the voice chat feature	• The application will provide a proper chat system with the call feature and voice chat feature.
• Zameen	• No chat box feature • No online contract feature	• The application will offer a chat box and an online contract which will be credible and secure.
• Apartments	• No Mobile application, only available as a website	• The application will be available on web as well as mobile application.

1.5 Advantages/Benefits of Proposed System

Following are the advantages of our application:

- Automated Coordination between tenants and proprietors.
- Easy to manage ones' properties.
- The app will lessen the gap between tenants and proprietors.
- It keeps a check on timely payments through rent reminders.
- Copy of contract will be available online for easy access and record keeping.
- It has a chat bot which will work based on artificial intelligence for instant automated answers.
- Secured payment system with fraud detection that generates logs to investigate errors.

1.6 Project Scope

The application will be designed to provide coordination among the tenants and proprietors. Many different tools and technologies will be used develop the interface and functionality of the application. The main functionalities of the application are:

Profile Management: The users will have to register themselves first to be able to use the application. A user can manage both, tenant management and renter management by toggling portals.

Chat box: There will be a chat box system where users will be able to communicate each other via, text messages, voice chat, voice call or WhatsApp redirection.

User Contract: Tenants and proprietors will have an online contract in which every required detail of the users will be saved. The users of this application will be able to view the contract details anytime regardless their current location.

Push Notifications: The users will get notifications on their due payments or chat notifications.

Complaint Management: There will also be a complain management system where tenants ‘complaints will be taken through a bot which will generate a ticket id and issues will be resolved by the landlords as soon as possible.

Help and Support: User’s queries will be resolved through an AI bot which will have answers fed into it. Any other user’s query which the bot can’t answer will be answered by the management team by live chat feature.

Payment System: The payment system would analyze all the ongoing payments and will check for the credibility of its source. If any fraud is detected, the system will generate logs through machine learning. The users can also view details of the previous paid rents and due payments. The card information of the users will also be saved.

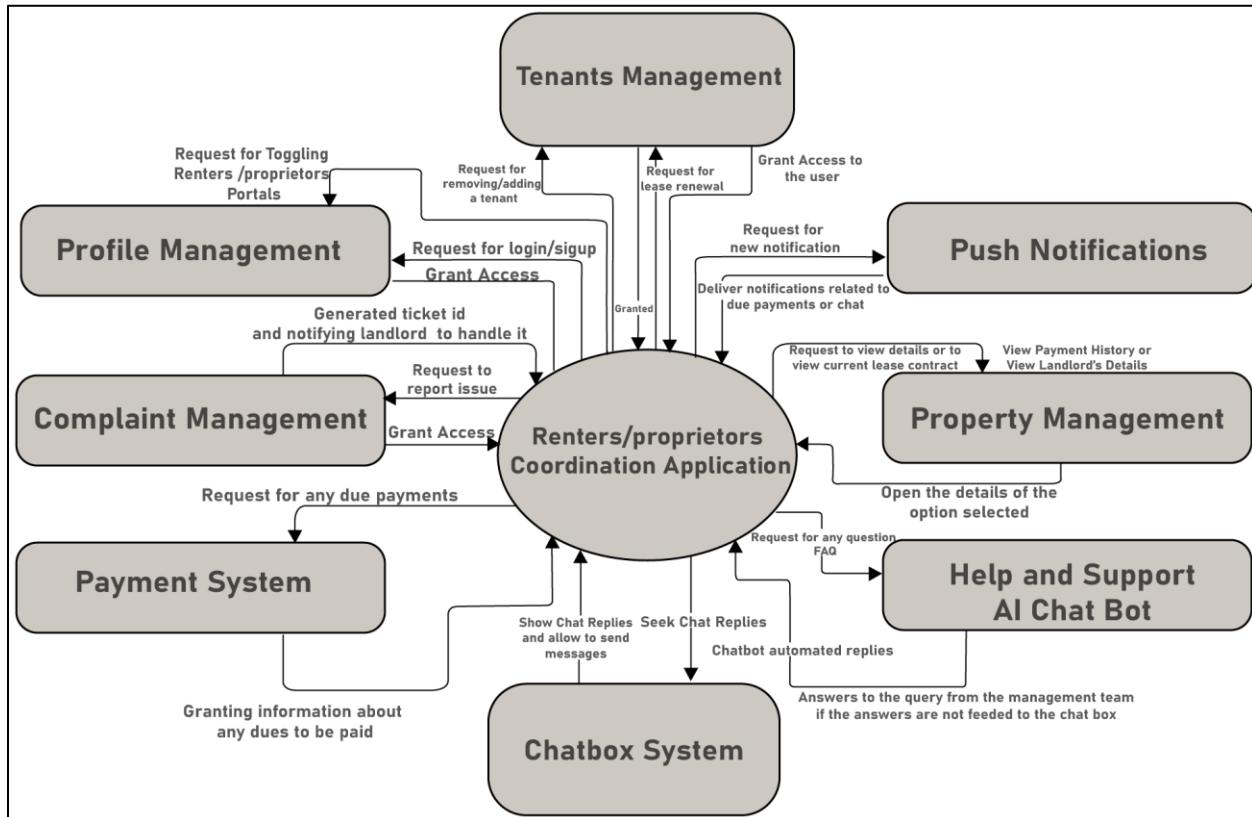


Figure 1: Context Diagram of RPC System

1.7 Modules

The major modules for the RPC System are listed below:

1.7.1 Profile Management

This module focuses on account creation and management of tenants and proprietors. After signing into their accounts, an individual user can toggle between renter and proprietor portals. Each portal will manage its respective domain.

1. Sign Up

The users will have to sign up for a new account in order to use the application.

2. Log In

The users will log in to their accounts when they want to use it.

3. Generate Captcha Images

The system will generate captcha images that will appear on the screen and user will have to select the images according to the question generated to prove his/her human identity.

4. Forget Password

If the users forget their passwords, they can request for a new one by entering their email or phone number.

5. Generate Popup Ads

The app will be monetized, so the advertisements will pop up on the screen.

1. View Profile

The users will be able to view and edit their personal information saved in the application.

2. Logout

The users can logout from their accounts in the application.

3. Change Profile Display

The users can change their profile pictures by either uploading from their device or taking one in real time.

4. View Profile Display

The users can view their profile pictures.

5. Edit Profile

The users will be able to edit their credentials in their profile like their name, contact number, and email etc.

6. Delete Account

The users will be given the option to permanently delete their account from the application. They will be able to stop deletion by logging back in their IDs within 14 days.

7. Disable Account

The users will be given the option to temporarily disable their account from the application.

8. Generate User ID

Whenever a new user account is created, they are assigned with a unique user ID.

1.7.2 User Interaction Chat Box

This module focuses on the interaction between renters and proprietors using in-app chat boxes to maintain separation between their everyday and work conversations. However, they will also be given the option for redirecting to WhatsApp.

1. View Message

The users can view incoming messages from the in-app chat box.

2. Send Message

The renters and proprietors can exchange text messages.

3. Mute Messages

The users can mute conversations.

4. Unmute Messages

The users can unmute muted conversations.

5. Pin Messages

The users can pin important messages on top.

6. Unpin Messages

The users can unpin the pinned messages.

7. Make Voice Call

The users can make voice calls. The application will redirect the user to the Calling application installed in their devices.

8. Redirect to WhatsApp

The users will have the option to get redirecting the WhatsApp app that will give them an additional option to chat freely with voice notes and video calls.

9. Search Text Message

The users can search any piece of word in their text messages.

10. Search User

The users can search other users to send them messages or make them a call.

11. View Chat Media

The users can view pictures, documents, links and videos sent to a specific user in a separate screen-view.

12. Clear Chat

The users can clear the chat box by deleting the messages.

1.7.3 Rent Payment Portal

This portal will ensure the users upgrade to online payments from offline ones. Renters will be given an easy to understand and interactive interface where they can do prompt online payments, keep record of their payments and the related information they would want to access. Safe online payments will be ensured using Machine Learning enabled Fraud Detection Systems which will monitor and notify the development or system team if any anomalies are detected.

1. Pay Rent

The renter will be able to pay their rent online.

2. View Payment History

The renter can view and manage the payments history.

3. Generate Invoice

The renter can generate an invoice of the payments performed.

4. Print Invoice

The renter can print the generated invoice of the payment performed.

5. Save Card Details

During online payment, the renter can save their card details for future use.

6. Request Payment Deadline Extension

The renters can request rent deadline extension from their proprietors.

7. Respond To Payment Deadline Extension Request

The landlord can respond with a Yes [along with new date] or a No to the request of renters about rent deadline extension.

8. View Payment Status

The users can view the payment status of whether it is in-progress, completed or declined.

9. Open Payment Details

The users can view the details of the payments.

10. Edit Payment Details

The proprietor can edit payment details of their properties that is used by the renter for sending rent payments.

11. Buy Premium Account

On buying the premium account, the user will experience smooth app activity without any ad pop ups.

12. Hide Pop-up Ads

After the premium mode of the account activates, the system will hide any occurrence of pop-up Ads.

1.7.4 Complaint Management Portal

This portal will ensure smooth communication between the renters and the proprietors. The renters can start complaint threads related to maintenance or security issues. These complaints can be created, deleted, and can be responded to. Complaint progress is also maintained by updating its status on the portal.

1. Create Complaint

The users can generate complaints in the application.

2. View Complaint Status

The users can view the generated complaints and their status of progress.

3. Delete Complaint

The users can delete the complaints after their creation.

4. Respond to Complaint

The landlord can respond to the complaints and the complaint status will be updated.

5. Edit Complaint Description

The renter can edit their generated complaints within 10 hours of creating a complaint.

1.7.5 Tenant Management Portal

This module will be the main portal for the proprietors or landlords to handle all of their properties being rented out. The landlords can remove and assign new tenants to their properties. Lease can be created and renewed on mutual demands. This feature will ensure that the contract and demands that the renters signed up for are a click away from them. Personal details of the tenants will also be maintained.

1. Add Renter

The landlord can add renters to their properties.

2. Remove Renter

The landlord can remove renters from their properties.

3. View Renters

The landlord can view renters of their properties.

4. Renew Lease

The landlord can create and renew lease after consulting with the renter to keep its softcopy record.

5. View Lease

The landlord can view lease of their properties of each renter.

6. Print Lease

The landlord can print the lease.

7. Add Property

The landlord can add properties to which renters are added later.

8. View Property

The landlords will be able to view their added properties.

9. Delete Property

The landlords can delete the added properties.

10. Edit Property Details

The landlords can edit property details.

11. Generate Property ID

Whenever a property is added by the landlord, a property ID is assigned to it which the renter could use to request connectivity with the landlord.

12. Update Rent

The landlord can update rent amount on the payment portal.

13. Go to Rent Payment Portal

The landlords will have a direct option to go to rent payment portal from tenant management portal for ease of use.

14. Respond to Renter Connectivity Request

The landlords can accept or decline the connectivity requests of renters to their properties.

1.7.6 Property Management Portal

This module will be the main portal for the renters for handling all the properties they are renting. They can view the property's details including lease document, landlord's details, rent tenure, and payment details.

1. View Lease

The renter can view signed lease in the application.

2. View Property Details

The renter can view their rented property details.

3. Check Rent Tenure

The renter can view their tenure of renting the respective property.

4. Go to Payment Portal

The renters will have a direct option to go to rent payment portal from property management portal for ease of use.

5. View Landlord's Details

The renter can view signed lease in the application.

6. Request Lease Renewal

The renter can send a request to their proprietors for lease renewal on new demands.

7. Print Lease

The renter can print the lease directly from the application.

8. Respond to Lease

When the landlord will create or renew lease, the renter can respond to it by accepting or declining it with a reason to negotiate demands through the app.

9. Connect with Landlord

The renter can connect with the landlord by adding the property ID.

1.7.7 Push Notification

The user of the application can turn on the push notifications, so they get a prompts like rent reminders, chat notifications and other application activities.

1. Turn On Rent Reminders

The users can set in-app rent reminders. This will ensure that the landlord does not have to call the renter for payment reminders.

2. Turn off Rent Reminders

The user will be able to turn off the rent reminders.

3. Turn On Chat Notifications

The users can set in-app chat notification. This will ensure that user does not miss any message when the application is not running.

4. Turn off Chat Notifications

The users can turn off chat notifications, which will restrict any upcoming notifications from the chat box.

5. Set Ringtone

The users will be able to set ringtones and can alter their volume levels.

6. Set Vibration

The users can also set vibration intensity of the device.

1.7.8 Help and Support

This module will cater the application users' queries related the application itself. Using artificial intelligence and machine learning, a bot will handle the questions from the users. The users may want to engage with an actual person, so the bot can redirect the user to live chat as well. Contact information of support team will also be available on the portal.

1. Chat with Bot

The bot will handle queries of the users regarding the use of application.

2. Do Live Chat

If users want to engage with a human support person, the bot can redirect them to one.

3. Contact Support Team

The support team's contact information will be available.

4. Change Bot's Language

The Bot's language can be changed by the users. It will be able to support Urdu and English Languages.

5. Send Feedbacks

The user will be able to send feedbacks about the application.

6. View Feedback History

The user can also view the history of the feedback they have sent before.

1.8 System Limitations/Constraints

Following are the limitations of our proposed system:

- There is no option available for video calls.
- The application cannot use or understand languages other than English and Urdu.
- For contacting the support team, the user must have an email app to get redirected to it.

- User must have MasterCard, UnionPay, EasyPaisa, JazzCash or PayPal account for paying rent.

1.9 Tools and Technologies

The tools and technologies to be used are provided in the Table below. The main IDE tool to develop the code will be **Microsoft Visual Studio Code 2019** on which different Technologies like **Flutter**, **HTML-CSS** and **Java Script** will be implemented. **Adobe Illustrator** and **Figma** will be used for design purposes. **Microsoft Word** and **PowerPoint** will be used for documentation and presentation. Firebase used for Data Base Management System.

Table 2: Tools and Technologies of RPC System.

Tools	Tools	Version	Rationale
	MS Visual Studio Code	2019	IDE
	Adobe Photoshop	2022	Design Work
	MS Word	2019	Documentation
	MS Power Point	2019	Presentation
	Adobe Illustrator	2022	Mockups Creation
	Flutter	2.11	SDK
	Figma		Design Work
Technologies	Technology	Version	Rationale
	Dart	2.17	Front-End Language
	Firebase	2019	Query Language
	Html	5	Web Development
	CSS	4.15	Web Front End Design
	JavaScript	ES2015	Back-End Language
	Python	3.8	ML programming language

1.10 Relevance to Course Material

- From coding perspective, we used one of the core courses and concepts of BCS i.e. Object Oriented Programming, and Data Structures and Algorithms.
- From database perspective, we implemented our knowledge about non-relational databases that we learnt in Database System-I course.
- From overall presentation and reports perspective, courses like Information and Communication Technology (ICT) and Reporting Writing Skills (RWS) helped us through.

1.11 Design and Process Methodology for this Project

1.11.1 Process Methodology

The software process methodology that we will use is Incremental Process Model. Our application has limited number of functionalities and most of the requirements are known since the beginning. There is a minor chance that the requirements would be changed in future. Therefore, the most suitable process model we could select is incremental process model.

1.11.2 Design Methodology

The design methodology we will be using is Object Oriented Approach because it increases the reusability of the code, and it would be easier for the team members to work together without any confusion. Therefore, we are using Flutter, HTML, CSS languages which follows object-oriented approach. Also, it is easier to describe the code using UML diagrams. So, OOP is the best approach which fits our framework.

1.12 Team Members Work Division

Table 3: Work Division of RPC System.

Student Name	Student Registration Number	Responsibility/ Modules
• Hunia Nadeem	• FA20-BCS-024	<ul style="list-style-type: none"> • (Module1, Module3, Module6, Module7) Design, Web Front-End, Web Back-End, Documentation, Artificial Intelligence, Data Gathering, Web Database
• Idrees Ahmed Ghazi	• FA20-BCS-025	<ul style="list-style-type: none"> • (Module2, Module4, Module5, Module8) Mobile Database, Documentation, Artificial Intelligence, Mobile Front-End, Mobile Back-End, Design

2. Problem Definition

This chapter highlights the problem faced by people on a daily basis during the property renting procedure. Alongside, the effective solution and our projects' requirements and deliverables are also discussed.

2.1 Problem Statement

Renting and proprietary work is a very tedious job and is mentally and physically demanding. After a person gets a house on rent, a continuous loop of **back-and-forth communication** occurs between the landlord and the tenant. Many factors including distance, time conflicts, and commitments can lead to time and resource delays. The **insecurity** and **uncertainty** in **offline payments** lead to further issues including data insecurity and payment fraud.

Property work is an everyday livelihood opted by a vast majority of people. So, to bring simplicity to the entire process, an **automated system** capable of resolving these issues is a necessity. The market has similar systems, but they don't provide a single platform to cater to everything, further complicating the entire process.

2.2 Problem Solution for the Proposed System

Renters/Proprietors Coordination Application provides a platform for the tenants/landlord to conveniently manage all the activities after renting a land/house. Instead of going back and forth to schedule physical meetings, both the renters and landlords can manage their properties from a single platform. The renters can **view the properties** they have rented, **check the lease**, and **Maintain related records**, whereas the landlords can **Add or remove tenants** and **Create or renew leases**. The online leases provide easy accessibility to the property's records.

The renters can conveniently perform **online payments**. **Fraud detection systems** will secure the online payment process from anomalies as the **Machine Learning system** will monitor them and notify the system's backend engineers if detected. The application will also **generate reminders** for due rents.

The renters can **file complaints** for maintenance issues or safety concerns. These complaints will be handled by bots who will issue a ticket for each complaint to the respective landlord and **update the status** of these tickets after they are processed. It will ensure organized management and smooth handling of complaints. For a prompt response to users' FAQs, a **chatbot** will be set up. This bot will also be able to redirect to live chat on users' demands. The application will allow the user to manage their tenants and their signed landlords separately for **easy and administered use** of the application.

2.3 Deliverables and Development Requirements

Deliverables and development requirements are as follows:

- Project Scope document to define the scope of the system,

- Project Status reports to gather the status of the ongoing project.
- Project Reports including Variance Report, and Comparison reports about the project plan and practical implementation.
- The main goal of the project was to build a web application and mobile phone application for users to coordinate with each other and maintain the data of their properties.
- Time sheet report for the demonstration of work of every team member.
- SRS document (System Requirement Specification)
- SDS document (System Design Specification)
- STP document (Software Test Plan)
- User Manuals for user guidance.
- Final financial report document.

3. Requirement Analysis

Thorough explanation of all the requirements of our project are discussed. System functionalities and quality attributes are discussed in detail as well. This part of the document is very essential for the success of the project.

3.1 Requirement Elicitation Techniques

The Requirement elicitation techniques are given below:

- Personal Experience and Observations.
- Brain Storming
- Mutual Discussion
- Analysis of the document
- Questionnaires
- User case techniques by listing core functionalities of the system.
- Prototyping

3.2 Use Cases Diagram(s)

The Use case diagrams of System, Renters and Proprietors are given below:

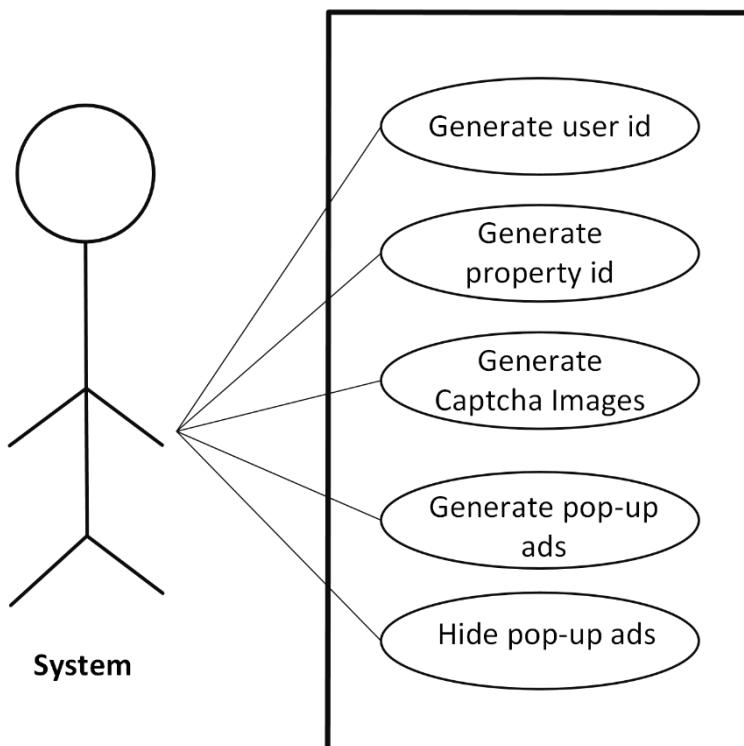


Figure 2: Use Case Diagram for System

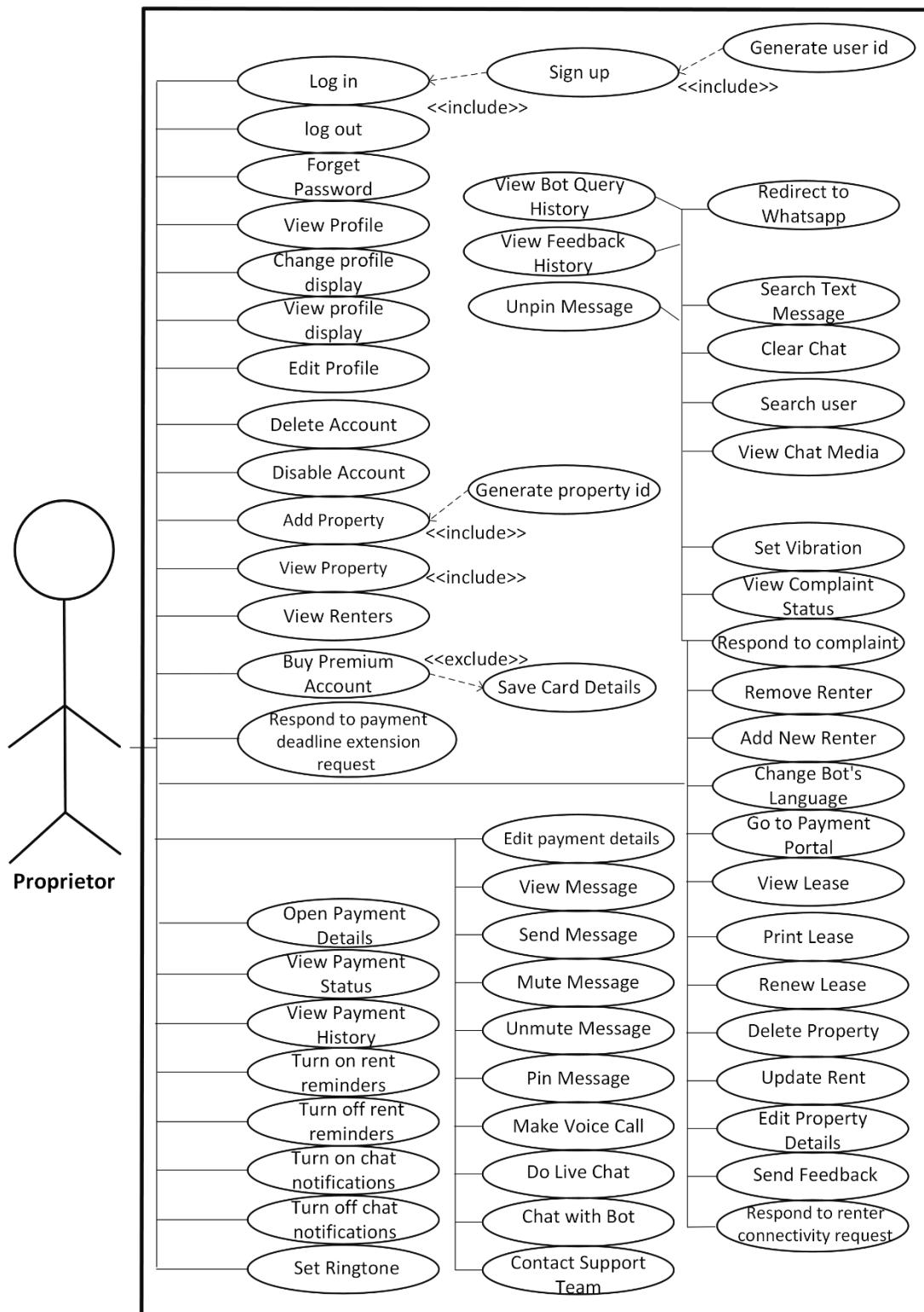


Figure 3: Use Case Diagram for Proprietor

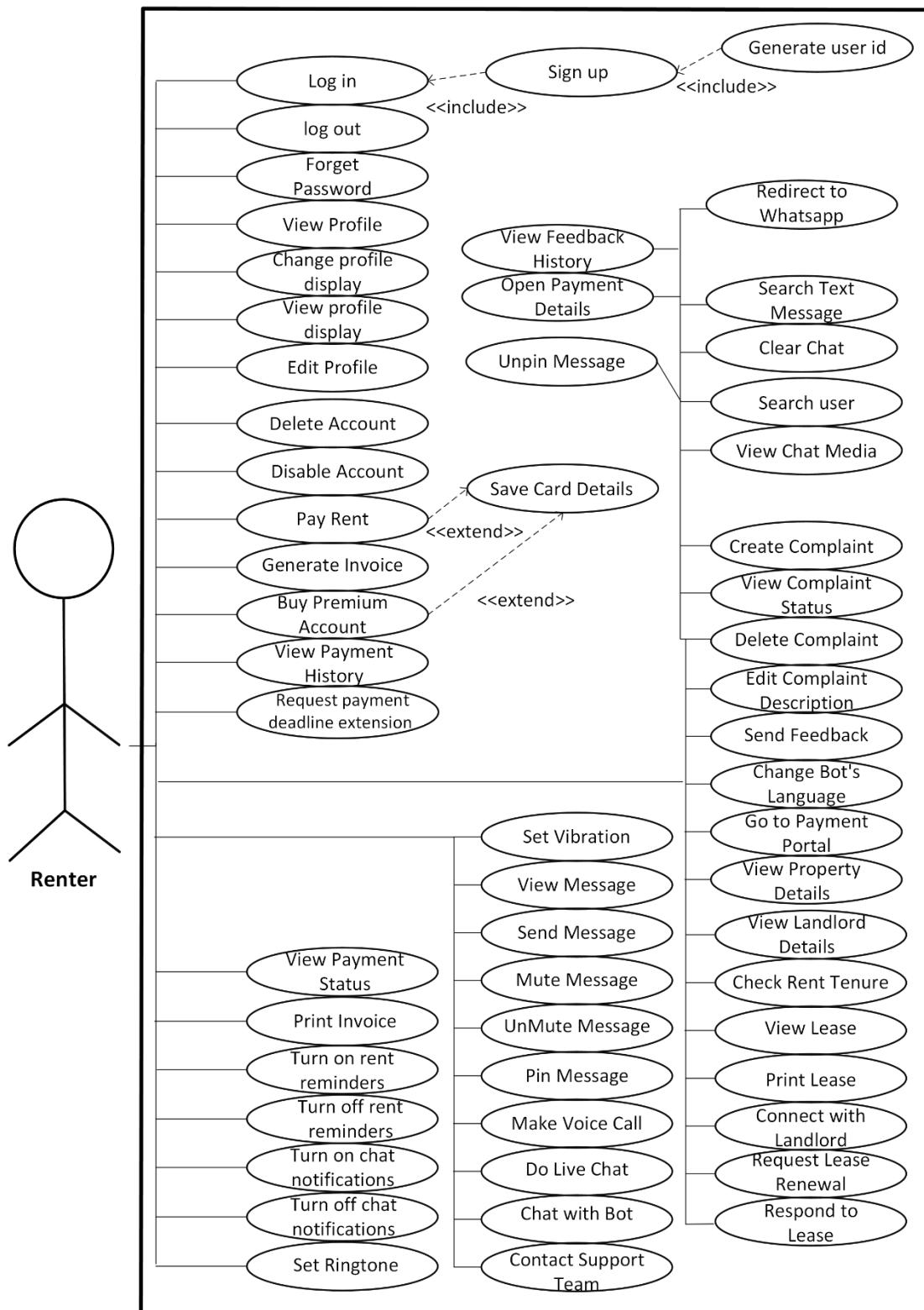


Figure 4: Use Case Diagram for Renter

3.3 Detailed Use Case (Tabular- Module Wise)

3.3.1 List of Use Cases

List of use cases per module is given below:

➤ *Module 1: Profile Management*

Table 4: List of Use Cases (Module 1: Profile Management)

UC-ID	Use Case Name
UC-1.1	Sign up
UC-1.2	Log in
UC-1.3	Log out
UC-1.4	Forget Password
UC-1.5	View Profile
UC-1.6	Edit Profile
UC-1.7	View Profile Display
UC-1.8	Change Profile Display
UC-1.9	Delete Account
UC-1.10	Disable Account
UC-1.11	Generate User Id
UC-1.12	Generate Captcha Images
UC-1.13	Generate Pop-Up Ads

➤ *Module 2: User Interaction Chat Box*

Table 5: List of Use Cases (Module 2: User Interaction Chat Box)

UC-ID	Use Case Name
UC-2.1	View Message
UC-2.2	Send Message
UC-2.3	Mute Message
UC-2.4	Unmute Message
UC-2.5	Pin Message
UC-2.6	Make Voice Call
UC-2.7	Redirect to Whatsapp
UC-2.8	Search Text Message
UC-2.9	Search User
UC-2.10	View Chat Media
UC-2.12	Clear Chat
UC-2.14	Unpin Message

➤ ***Module 3: Rent Payment Portal***

Table 6: List of Use Cases (Module 3: Rent Payment Portal)

UC-ID	Use Case Name
UC-3.1	Pay Rent
UC-3.2	Generate Invoice
UC-3.3	Print Invoice
UC-3.4	View Payment History
UC-3.5	Save Card Details
UC-3.6	Request Payment Deadline Extension
UC-3.7	Respond to Payment Deadline Extension Request
UC-3.8	View Payment Status
UC-3.9	Edit Payment Details
UC-3.10	Buy Premium Account
UC-3.11	Open Payment Details
UC-3.12	Hide Pop-up Ads

➤ ***Module 4: Complaint Management Portal***

Table 7: List of Use Cases (Module 4: Complaint Management Portal)

UC-ID	Use Case Name
UC-4.1	Create Complaint
UC-4.2	View Complaint Status
UC-4.3	Delete Complaint
UC-4.4	Respond to Complaint
UC-4.5	Edit Compliant Description

➤ ***Module 5: Tenant Management Portal***

Table 8: List of Use Cases (Module 5: Tenant Management Portal)

UC-ID	Use Case Name
UC-5.1	Add Property
UC-5.2	View Property
UC-5.3	Delete Property
UC-5.4	Generate Property Id
UC-5.5	View Renters
UC-5.7	Add Renter
UC-5.8	Remove Renter
UC-5.9	View Lease

UC-5.10	Print Lease
UC-5.11	Renew Lease
UC-5.12	Update Rent
UC-5.13	Edit Property Details
UC-5.14	Go to Payment Portal
UC-5.15	Respond to Renter Connectivity Request

➤ **Module 6: Property Management Portal**

Table 9: List of Use Cases (Module 6: Property Management Portal)

UC-ID	Use Case Name
UC-6.1	View Lease
UC-6.2	View Property Details
UC-6.3	View Landlord's Details
UC-6.4	Check Rent Tenure
UC-6.5	Request Lease Renewal
UC-6.6	Print Lease
UC-6.7	Respond to Lease
UC-6.8	Connect With Landlord
UC-6.9	Go to Payment Portal

➤ **Module 7: Push Notification**

Table 10: List of Use Cases (Module 7: Push Notifications)

UC-ID	Use Case Name
UC-7.1	Turn On Rent Reminders
UC-7.2	Turn Off Rent Reminders
UC-7.3	Turn On Chat Notifications
UC-7.4	Turn Off Chat Notifications
UC-7.5	Set Ringtone
UC-7.6	Set Vibration

➤ ***Module 8: Help and Support***

Table 11: List of Use Cases (Module 8: Help and Support)

UC-ID	Use Case Name
UC-8.1	Chat With Bot
UC-8.2	Do Live Chat
UC-8.3	Contact Support Team
UC-8.4	Send Feedback
UC-8.5	Change Bot's Language
UC-8.6	View Bot Query History
UC-8.7	View Feedback History

3.3.2 Tabular Form of Use Cases:

Following are the Use Cases in detailed Tabular Form:

➤ ***Module 1: Profile Management***

The tabular use cases of the module profile management are provided below:

1. Sign Up

The tabular use cases are given below:

Table 12: Tabular Use Case UC-1.1

Use Case ID:	UC-1.1
--------------	--------

Use Case Name:	Sign Up
Actors:	Renter / Proprietor
Description:	Renter/Proprietor will create their account in order to use the application.
Priority:	High
Precondition:	User should have an authentic and valid email address and contact number.
Trigger:	Renter/Proprietor will click on the Sign-up button from main screen.
Include:	None.
Normal Flow:	<ol style="list-style-type: none"> 1. The user will enter authentic credentials (email, CNIC, contact) and password in the sign-up window. 2. Then user will click on the sign-up button. 3. OTP will be generated and sent to the user's contact or email. 4. On correct input of OTP, successful sign-up message will be displayed to the user.
Alternative Flows:	None.
Exceptions:	<ul style="list-style-type: none"> • In step 4 of normal flow, if the user enters an invalid email address, contact number, CNIC and password: <ol style="list-style-type: none"> 1. An appropriate error message will be displayed 2. Asking the user to enter the data and credentials again 3. Then user will provide correct data. Use Case resumes from step 5 of normal flow • In step 7, if the user enters incorrect OTP: <ol style="list-style-type: none"> 1. An appropriate error message will be displayed. 2. User will be asked to enter OTP again. 3. If fails to enter within 30 seconds, OTP will expire. 4. User will have to request another OTP and correctly enter it for successful sign-up.
Post condition:	After successful entry of credentials, new user account will be formed and control will move to home screen from where the user can log in or sign up.
Business Rules:	User should have a verified email address and a correct phone number.
Notes & Issues:	None.
Assumptions:	<ol style="list-style-type: none"> 1. User will have a stable internet connection. 2. User will be able to enter data correctly.

2. Log In

The tabular use cases are given below:

Table 13: Tabular Use Case UC-1.2

Use Case ID:	UC-1.2
Use Case Name:	Log In
Actors:	Renter / Proprietor
Description:	Renter/Proprietor will log in to their account in order to use the application.
Priority:	High
Precondition:	User must have signed up for an account.
Trigger:	Renter/Proprietor will click on the Log-in button from main screen.
Include:	UC-1.1
Normal Flow:	<ol style="list-style-type: none"> 1. The user will enter an authentic email and password used at the time of account creation in the login window. 2. Then user will click on the log in button.
Alternative Flows:	None.
Exceptions:	<p>In step 1 of normal flow, if the user enters an invalid email address and password with which no account was formed:</p> <ol style="list-style-type: none"> 1. An appropriate error message will be displayed 2. Asking the user to enter the credentials again 3. Then user will provide correct data. <p>Use Case resumes from step 2 of normal flow</p>
Post condition:	After logging in, the user will reach to the main menu of the application from where they can choose their role between renter and proprietor.
Business Rules:	User should log in with correct email and password combination.
Notes & Issues:	None.
Assumptions:	<ol style="list-style-type: none"> 1. User will have a stable internet connection. 2. User will be able to enter data correctly.

3. Logout

The tabular use cases are given below:

Table 14: Tabular Use Case UC-1.3

Use Case ID:	UC-1.3
Use Case Name:	Log out
Actors:	Renter / Proprietor
Description:	Renter/Proprietor can log out from their account.
Priority:	High
Precondition:	User must be logged in an account.
Trigger:	Renter/Proprietor will click on the Log-out button from side pane.
Include:	UC-1.2
Normal Flow:	<ol style="list-style-type: none"> 1. After using the application, the user wants to switch off their account. 2. From the main account settings, the user will click on the log out button.
Alternative Flows:	None.
Exceptions:	While logging out, if internet connection is lost the logging out process will be cancelled and user will be redirected to the main menu instead of the home screen. Use case resumes from step 1.
Post condition:	After successfully logging out, the user will be redirected to home screen from where they can sign up or log in again.
Business Rules:	User should have stable internet connection
Notes & Issues:	None.
Assumptions:	<ol style="list-style-type: none"> 1. User will have a stable internet connection. 2. User will be able to find and select the option from the menu correctly.

4. Forget Password

The tabular use cases are given below:

Table 15: Tabular Use Case UC-1.4

Use Case ID:	UC-1.4
Use Case Name:	Forget Password
Actors:	Renter / Proprietor
Description:	Renters and proprietors can change their account passwords when either they forget them or only want to change them.
Priority:	Medium
Preconditions:	User must have signed up for an account.
Trigger:	User will click on the Forget Password button while logging in.
Include:	UC-1.2
Normal Flow:	<ol style="list-style-type: none"> 1. If they forget their password or want to change it, forget password option will be clicked. 2. Then user will enter required credentials in order to verify the account presence. 3. New password will be entered by user. 4. OTP will be generated and sent to the user's contact or email. 5. On correct input of OTP, successful password change message will be displayed to the user.
Alternative Flows:	None.
Exceptions:	<p>In step 2 of normal flow, if the user enters an invalid email with which no account was formed:</p> <ol style="list-style-type: none"> 1. An appropriate error message will be displayed. 2. Asking the user to enter the credentials again. 3. Then user will provide correct data. <p>Use Case resumes from step 3 of normal flow</p>
Post condition:	After successfully verifying the account and changing the password, new log in credentials will be set for the user log in.
Business Rules:	User should have valid email address.
Notes & Issues:	None.
Assumptions:	<ol style="list-style-type: none"> 1. User will have a stable internet connection. 2. User will be able to enter data correctly.

5. View Profile

The tabular use cases are given below:

Table 16: Tabular Use Case UC-1.5

Use Case ID:	UC-1.5
Use Case Name:	View Profile
Actors:	Renter / Proprietor
Description:	Renters and proprietors can view their profiles.
Priority:	Medium
Preconditions:	User must be logged in to their account.
Trigger:	User will select View Profile from side pane.
Include:	UC-1.2
Normal Flow:	1. User's credentials will be displayed on clicking the View Profile button.
Alternative Flows:	None.
Exceptions:	If the internet connection fails, the application will reload the profile window.
Post condition:	User will be viewing the profile credentials.
Business Rules:	User should have stable internet connection.
Notes & Issues:	None.
Assumptions:	<ol style="list-style-type: none"> 1. User will have a stable internet connection. 2. User understands English/Urdu language. 3. User will be able to find and select the option from the menu correctly.

6. Edit Profile

The tabular use cases for profile management are given below:

Table 17: Tabular Use Case UC-1.6

Use Case ID:	UC-1.6
Use Case Name:	Edit Profile
Actors:	Renter / Proprietor
Description:	Renters and proprietors can manage their profiles by changing their personal information.
Priority:	Medium
Preconditions:	User must be logged in to their account.
Trigger:	User will select View Profile from side pane.
Include:	UC-1.5
Normal Flow:	<ol style="list-style-type: none"> 1. The user can make changes to them. 2. OTP will be generated and sent to the user's contact or email. 3. On correct input of OTP, successful data change message will be displayed to the user.
Alternative Flows:	None.
Exceptions:	<ul style="list-style-type: none"> • If the internet connection fails, the application will reload the profile window. • In step 1 of normal flow, if the user enters an invalid credentials like email address or contact number: <ol style="list-style-type: none"> 1. An appropriate error message will be displayed 2. Asking the user to enter the data and credentials again 3. Then user will provide correct data. Use Case resumes from step 2 of normal flow • In step 3, if the user enters incorrect OTP: <ol style="list-style-type: none"> 1. An appropriate error message will be displayed. 2. User will be used to enter OTP again. 3. If fails to enter within 30 seconds, OTP will expire. User will have to request another OTP and correctly enter it for successful sign-up.
Post condition:	New credentials of the user will be updated at their landlords' end as well.
Business Rules:	User should have stable internet connection.
Notes & Issues:	None.
Assumptions:	<ol style="list-style-type: none"> 1. User will have a stable internet connection. 2. User will be able to enter data correctly.

7. View Profile Display

The tabular use cases for profile management are given below:

Table 18: Tabular Use Case UC-1.7

Use Case ID:	UC-1.7
Use Case Name:	View Profile Display
Actors:	Renter / Proprietor
Description:	Renters and proprietors can view their display pictures.
Priority:	Medium
Preconditions:	User must be logged in to their account.
Trigger:	The user will click on the display picture icon from the side pane.
Include:	UC-1.2
Normal Flow:	<ol style="list-style-type: none"> Three options will be provided. Whether view picture, upload picture from device or take a new picture. The user will select view picture. The display picture will be opened.
Alternative Flows:	None.
Exceptions:	If the internet connection fails, the application will reload the display picture till it has been loaded.
Post condition:	View frame of profile picture will be opened.
Business Rules:	User should have stable internet connection.
Notes & Issues:	None.
Assumptions:	<ol style="list-style-type: none"> User will have a stable internet connection. User will be able to find and select the option from the menu correctly.

8. Change Profile Display

The tabular use cases for profile management are given below:

Table 19: Tabular Use Case UC-1.8

Use Case ID:	UC-1.8
Use Case Name:	Change Profile Display
Actors:	Renter / Proprietor
Description:	Renters and proprietors can change their display pictures.
Priority:	Medium
Trigger:	The user will click on the display picture icon from the side pane.
Include:	UC-1.2
Normal Flow:	<ol style="list-style-type: none"> Three options will be provided. Whether view picture, upload picture from device or take a new picture. The user will make a decision. After successfully updating the display picture, display picture will be changed.
Alternative Flows:	None.
Exceptions:	<p>If the internet connection fails:</p> <ol style="list-style-type: none"> Picture will not be uploaded on the application. The application will try to reload itself. On successful reloading, Use Case resumes from step 2 of normal flow. If time outs during reloading process, user will be redirected to the view frame of old display picture.
Post condition:	After successfully changing the display picture, other application users can identify the user by their new display picture.
Business Rules:	User should have stable internet connection.
Notes & Issues:	None.
Assumptions:	<ol style="list-style-type: none"> User will have a stable internet connection. User will be able to enter data correctly.

9. Delete Account

The tabular use cases for profile management are given below:

Table 20: Tabular Use Case UC-1.9

Use Case ID:	UC-1.9
Use Case Name:	Delete Account
Actors:	Renter / Proprietor
Description:	Renters and proprietors can delete their accounts permanently if they do not want to use the application again.
Priority:	High
Preconditions:	User must be logged in to their account.
Trigger:	The user will click on the delete/disable account button from side pane.
Include:	UC-1.2
Normal Flow:	<ol style="list-style-type: none"> 1. The user will select delete account option and give reason for their choice. 2. Then the user will be asked to enter their password and CNIC for verification. 3. After successful entry of the credentials, user will be redirected to home screen from where they can sign up or log in to their account again. 4. The account will be disabled for 14 days and will be deleted afterwards.
Alternative Flows:	None.
Exceptions:	<p>In step 2 of normal flow, if the user fails to enter correct password and CNIC.</p> <ol style="list-style-type: none"> 1. An appropriate error message will be displayed 2. Asking the user to enter the credentials again 3. Then user will provide correct data. <p>Use Case resumes from step 3 of normal flow.</p>
Post condition:	After successfully verifying the account and deleting the account, the account will be disabled for 14 days and then deleted if the user does not log in.
Business Rules:	User should have stable internet connection.
Notes & Issues:	None.
Assumptions:	<ol style="list-style-type: none"> 1. User will have a stable internet connection. 2. User will be able to find and select the option from the menu correctly.

10. Disable Account

The tabular use cases for profile management are given below:

Table 21: Tabular Use Case UC-1.10

Use Case ID:	UC-1.10
Use Case Name:	Disable Account
Actors:	Renter / Proprietor
Description:	Renters and proprietors can disable their accounts temporarily.
Priority:	High
Preconditions:	User must have signed up for an account.
Trigger:	The user will click on the delete/disable account button from side pane.
Include:	UC-1.2
Normal Flow:	<ol style="list-style-type: none"> 1. The user will select disable account option and give reason for their choice. 2. Then the user will be asked to enter their password and CNIC for verification. 3. After successful entry of the credentials, user will be redirected to home screen from where they can sign up or log in to their account again. 4. The account will be disabled for 14 days and can be reactivated by logging in again.
Alternative Flows:	None.
Exceptions:	<p>In step 2 of normal flow, if the user fails to enter correct password and CNIC.</p> <ol style="list-style-type: none"> 1. An appropriate error message will be displayed 2. Asking the user to enter the credentials again 3. Then user will provide correct data. <p>Use Case resumes from step 3 of normal flow</p>
Post condition:	After successfully verifying the account and disabling the account, the account will be disabled for 14 days and can be reactivated by logging in.
Business Rules:	User should have stable internet connection.
Notes & Issues:	None.
Assumptions:	<ol style="list-style-type: none"> 1. User will have a stable internet connection. 2. User will be able to find and select the option from the menu correctly.

11. Generate User ID

The tabular use cases for profile management are given below:

Table 22: Tabular Use Case UC-1.11

Use Case ID:	UC-1.11
Use Case Name:	Generate User ID
Actors:	System
Description:	System will generate user ID on each account creation.
Priority:	High
Preconditions:	Account creation process is performed successfully.
Trigger:	A user creates a new account.
Include:	UC-1.1
Normal Flow:	None.
Alternative Flows:	None.
Exceptions:	If the user fails to provide valid information for sign up, <ol style="list-style-type: none"> 1. Account creation process will be cancelled. 2. An appropriate error message will be displayed 3. Asking the user to enter the credentials again 4. Then user will provide correct data.
Post condition:	User will be displayed a prompt message on successful account creation and user id. User will also receive an email regarding their user ID.
Business Rules:	User should have stable internet connection and verified email address.
Notes & Issues:	None.
Assumptions:	1. User will have a stable internet connection. 2. User will be able to find and select the option from the menu correctly.

12. Generate Captcha Images

The tabular use cases for profile management are given below:

Table 23: Tabular Use Case UC-1.12

Use Case ID:	UC-1.12
Use Case Name:	Generate Captcha Images
Actors:	System
Description:	System will generate captcha images to verify the user before logging in to their account.
Priority:	High
Preconditions:	User wants to log-in to the application
Trigger:	A user selects “I am not a robot” checkbox.
Include:	UC-1.2
Normal Flow:	None.
Alternative Flows:	None.
Exceptions:	If the user fails to select correct images, <ol style="list-style-type: none"> 1. Login will be cancelled. 2. User will have to verify themselves again by triggering the use case.
Post condition:	User will be successfully logged in to their account.
Business Rules:	User should have stable internet connection.
Notes & Issues:	None.
Assumptions:	1. User will have a stable internet connection. 2. User will be able to find and understand the captcha query correctly.

13. Generate Pop-Up Ads

The tabular use cases for profile management are given below:

Table 24: Tabular Use Case UC-1.13

Use Case ID:	UC-1.13
Use Case Name:	Generate Pop-Up Ads
Actors:	System
Description:	System will generate pop-up ads based on user activity to offer app for free.
Priority:	High
Preconditions:	User successfully logs in to the application.
Trigger:	A user selects Login button.
Include:	UC-1.2
Normal Flow:	None.
Alternative Flows:	None.
Exceptions:	None.
Post condition:	User will see ads based on their activity on the user interface.
Business Rules:	Machine Learning.
Notes & Issues:	None.
Assumptions:	1. User will have a stable internet connection.

➤ ***Module 2: User Interaction Chat Box***

The tabular use cases of the module named user interaction chat box are given below:

1. View Message

The tabular use case of Chat box interaction module is given below:

Table 25: Tabular Use Case UC-2.1

Use Case ID:	UC-2.1
Use Case Name:	View Message
Actors:	Renter/Proprietor
Description:	The goal is to enable the Renter/Proprietor to view a message in a chat box.
Priority	High
Preconditions:	<ol style="list-style-type: none"> 1. User must be logged into the application. 2. User must click the inbox icon to open it.
Trigger	When the user clicks the Inbox Icon.
Include	None
Normal Flow:	<ol style="list-style-type: none"> 1. The user will open the web or mobile application 2. The user will sign into his/her account 3. After the home screen appears, the user will click the inbox icon to view incoming messages.
Alternative Flows:	None.
Exceptions:	No Exceptions
Post condition:	The user can view messages.
Business Rules:	User should have a verified email address
Notes and Issues	None
Assumptions:	<ol style="list-style-type: none"> 1. User must have a stable internet connection. 2. User should be able to enter his/her credentials without any error. 3. User should be able to find the inbox icon at the top.

2. Send Message

The tabular use case of Chat box interaction module is given below:

Table 26: Tabular Use Case UC-2.2

Use Case ID:	UC-2.2
Use Case Name:	Send Message
Actors:	Renter/Proprietor
Description:	The goal is to enable the Renter/Proprietor to send messages to each other or reply to chats.
Priority	High
Preconditions:	<ul style="list-style-type: none"> 1. User must be logged into the application. 2. User must click the inbox icon to open it. 3. The user must type a message before sending.
Trigger	When the user clicks the Inbox Icon, then types a text and clicks the send button.
Include	None
Normal Flow:	<ul style="list-style-type: none"> 1. The user will open the web or mobile application 2. The user will sign into his/her account 3. After the home screen appears, the user will click the inbox icon to view incoming messages. 4. The user will type a message in the chat box and click the send button provided.
Alternative Flows:	After the user has typed his/her message, the user can use the “Enter” key to send the message without using the send button.
Exceptions:	Blank Message cannot be sent, no error options will pop up and user’s message will not be sent unless he/she types at least one character.
Post condition:	The user can send message to the recipient.
Business Rules:	None
Notes and Issues	None
Assumptions:	<ul style="list-style-type: none"> 1. User must have a stable internet connection. 2. User should be able to enter his/her credentials without any error. 3. User should be able to find the inbox icon at the top. 4. User should be able to understand and type English Language.

3. Mute Message

The tabular use case of Chat box interaction module is given below:

Table 27: Tabular Use Case UC-2.3

Use Case ID:	UC-2.3
Use Case Name:	Mute Message
Actors:	Renter/Proprietor
Description:	The goal is to enable the Renter/Proprietor to Mute each other from restricting incoming notification from the other user.
Priority	Medium
Preconditions:	<ol style="list-style-type: none"> 1. User must be logged into the application. 2. User must click the inbox icon to open it. 3. The user must select the contact to be muted.
Trigger	When the user clicks the Inbox Icon, then several chats appear in front of the user, then the user can select a user and apply the mute option to trigger the event.
Include	None.
Normal Flow:	<ol style="list-style-type: none"> 1. The user will open the web or mobile application 2. The user will sign into his/her account 3. After the home screen appears, the user will click the inbox icon to open the inbox. 4. The user will have to select a contact who is to be muted 5. After that select the mute option
Alternative Flows:	None
Exceptions:	If the user is not connected to the internet, then he/she cannot mute someone.
Post condition:	The notification will be restricted
Business Rules:	None
Notes and Issues	None
Assumptions:	<ol style="list-style-type: none"> 1. User must have a stable internet connection. 2. User should be able to enter his/her credentials without any error. 3. The User must login into the application. 4. User should be able to find the inbox icon at the top. 5. User should be able to understand English Language.

4. Unmute Message

The tabular use case of Chat box interaction module is given below:

Table 28: Tabular Use Case UC-2.4

Use Case ID:	UC-2.4
Use Case Name:	Unmute Message
Actors:	Renter/Proprietor
Description:	The goal is to enable the Renter/Proprietor to Unmute each other from unrestricting incoming notification from the other user.
Priority	Medium
Preconditions:	<ol style="list-style-type: none"> 1. User must be logged into the application. 2. User must click the inbox icon to open it. 3. The user must select the contact to be Unmuted.
Trigger	When the user clicks the Inbox Icon, then a number of chats appear in front of the user, then the user can select a user and apply the Unmute option to trigger the event.
Include	None
Normal Flow:	<ol style="list-style-type: none"> 1. The user will open the web or mobile application 2. The user will sign into his/her account 3. After the home screen appears, the user will click the inbox icon to open the inbox. 4. The user will have to select a contact who is to be Unmuted 5. After that select the Unmute option
Alternative Flows:	None
Exceptions:	The user must be connected to the internet otherwise the user can be unmuted.
Post condition:	The Notifications will be unrestricted
Business Rules:	None
Notes and Issues	None
Assumptions:	<ol style="list-style-type: none"> 1. User must have a stable internet connection. 2. User should be able to enter his/her credentials without any error. 3. User should be able to find the inbox icon at the top. 4. User should be able to understand English Language.

5. Pin Message

The tabular use case of Chat box interaction module is given below:

Table 29: Tabular Use Case UC-2.5

Use Case ID:	UC-2.5
Use Case Name:	Pin Message
Actors:	Renter/Proprietor
Description:	The goal is to enable the Renter/Proprietor to pin important messages.
Priority	Medium
Preconditions:	<ol style="list-style-type: none"> 1. User must be logged into the application. 2. User must click the inbox icon to open it. 3. The user must select the contact to be pinned.
Trigger	When the user clicks the Inbox Icon, all the chats appear in front of the user, then the user can select a chat and click the pin message option.
Include	None
Normal Flow:	<ol style="list-style-type: none"> 1. The user will open the web or mobile application 2. The user will sign into his/her account 3. After the home screen appears, the user will click the inbox icon to open the inbox. 4. The user will have to select a contact who is to be pinned 5. Then select the pin option
Alternative Flows:	None
Exceptions:	If the app is not responding, then the user cannot be pinned.
Post condition:	The message will be pinned on the top.
Business Rules:	None
Notes and Issues	None
Assumptions:	<ol style="list-style-type: none"> 1. User must have a stable internet connection. 2. The application should be responsive. 3. The User must login into the application. 4. User should be able to find the inbox icon at the top. 5. User should be able to understand English Language.

6. Make Voice Call

The tabular use case of Chat box interaction module is given below:

Table 30: Tabular Use Case UC-2.6

Use Case ID:	UC-2.6
Use Case Name:	Make Voice Call
Actors:	Renter/Proprietor
Description:	The goal is to enable the Renter/Proprietor to call each other on their contact numbers.
Priority	High
Preconditions:	<ol style="list-style-type: none"> 1. User must be logged into the application. 2. User must click the inbox icon to open it. 3. The user must select the Start a Call icon.
Trigger	When the user clicks the Inbox Icon, then several chats appear in front of the user, then the user can select a user and click on the call icon to start a call.
Include	None
Normal Flow:	<ol style="list-style-type: none"> 1. The user will open the web or mobile application 2. The user will sign into his/her account 3. After the home screen appears, the user will click the inbox icon to open the inbox. 4. The user will have to select a contact who is to be Called 5. After that select the Start a Call icon to call the user.
Alternative Flows:	None
Exceptions:	If the user is not connected to the internet, then he/she cannot call someone.
Post condition:	Call notification would be sent to the recipient, and he/she can accept/decline the call request.
Business Rules:	None
Notes and Issues	None
Assumptions:	<ol style="list-style-type: none"> 1. User must have a stable internet connection. 2. User should be able to enter his/her credentials without any error. 3. User should be able to find the inbox icon at the top. 4. User should be able to understand English Language. 5. User should be able to find the call icon at the top.

7. Redirect to WhatsApp

The tabular use case of Chat box interaction module is given below:

Table 31: Tabular Use Case UC-2.7

Use Case ID:	UC-2.7
Use Case Name:	Redirect to WhatsApp
Actors:	Renter/Proprietor
Description:	The goal is to enable the Renter/Proprietor to contact each other on WhatsApp.
Priority	High
Preconditions:	<ul style="list-style-type: none"> 1. User must be logged into the application. 2. User must click the inbox icon to open it. 3. The user must select the WhatsApp icon.
Trigger	When the user clicks the WhatsApp Icon.
Include	None
Normal Flow:	<ul style="list-style-type: none"> 1. The user will open the web or mobile application 2. The user will sign into his/her account 3. After the home screen appears, the user will click the inbox icon to open the inbox. 4. The user will have to select a contact. 5. After that select the Start a WhatsApp icon to be redirected. 6. This will redirect the user to WhatsApp application.
Alternative Flows:	None.
Exceptions:	The user cannot be redirected if the other user is not a WhatsApp user.
Post condition:	User will be able to contact the other user on WhatsApp.
Business Rules:	None
Notes and Issues	None
Assumptions:	<ul style="list-style-type: none"> 1. User must have a stable internet connection. 2. User should be able to enter his/her credentials without any error. 3. User should be able to find the inbox icon at the top. 4. User should be a WhatsApp user.

8. Search Text Message

The tabular use case of Chat box interaction module is given below:

Table 32: Tabular Use Case UC-2.8

Use Case ID:	UC-2.8
Use Case Name:	Search Text Message
Actors:	Renter/Proprietor
Description:	The goal is to enable the Renter/Proprietor to search any piece of word in the text messages.
Priority	Medium
Preconditions:	<ol style="list-style-type: none"> 1. User must be logged into the application. 2. User must click the inbox icon to open it.
Trigger	When the user clicks the search bar and enters any piece of words.
Include	None
Normal Flow:	<ol style="list-style-type: none"> 1. The user will open the web or mobile application 2. The user will sign into his/her account 3. After the home screen appears, the user will click the inbox icon to open the inbox. 4. The user will type the text in the search bar to search it.
Alternative Flows:	None.
Exceptions:	If internet connection is lost, the application will try to re-establish its connection. If it fails to do so within 5 seconds, the inbox window will be reloaded.
Post condition:	Search results matching the piece of words will be displayed.
Business Rules:	None
Notes and Issues	None
Assumptions:	<ol style="list-style-type: none"> 1. User must have a stable internet connection. 2. User should be able to enter his/her credentials without any error. 3. User should be able to find the inbox icon at the top.

9. Search User

The tabular use case of Chat box interaction module is given below:

Table 33: Tabular Use Case UC-2.9

Use Case ID:	UC-2.9
Use Case Name:	Search User
Actors:	Renter/Proprietor
Description:	The goal is to enable the Renter/Proprietor to search other users to send them messages or make them a call.
Priority	Medium
Preconditions:	<ol style="list-style-type: none"> 1. User must be logged into the application. 2. User must click the inbox icon to open it.
Trigger	When the user clicks the search bar and enters user ID.
Include	None
Normal Flow:	<ol style="list-style-type: none"> 1. The user will open the web or mobile application 2. The user will sign into his/her account 3. After the home screen appears, the user will click the inbox icon to open the inbox. 4. The user will type the user ID in the search bar above the contacts to search it.
Alternative Flows:	None
Exceptions:	If internet connection is lost, the application will try to re-establish its connection. If it fails to do so within 5 seconds, the inbox window will be reloaded.
Post condition:	Search results matching the user ID will be displayed.
Business Rules:	None
Notes and Issues	The user must enter a right name to search
Assumptions:	<ol style="list-style-type: none"> 1. User must have a stable internet connection. 2. User should be able to enter his/her credentials without any error. 3. User should be able to find the inbox icon at the top.

10. View Chat Media

The tabular use case of chat box interaction module is given below:

Table 34: Tabular Use Case UC-2.10

Use Case ID:	UC-2.10
Use Case Name:	View Chat Media
Actors:	Renter/Proprietor
Description:	The goal is to enable the Renter/Proprietor to view pictures, documents, links, and videos sent to a specific user in a separate screen-view.
Priority	Medium
Preconditions:	<ol style="list-style-type: none"> 1. User must be logged into the application. 2. User must click the inbox icon to open it.
Trigger	When the user selects a user and clicks on chat menu.
Include	None
Normal Flow:	<ol style="list-style-type: none"> 1. The user will open the web or mobile application 2. The user will sign into his/her account 3. After the home screen appears, the user will click the inbox icon to open the inbox. 4. The user will open a user chat and go to chat details. 5. The user will be able to view the media in a separate view frame.
Alternative Flows:	None
Exceptions:	If internet connection is lost, the application will try to re-establish its connection. If it fails to do so within 5 seconds, the inbox window will be reloaded.
Post condition:	All the media shared between 2 users will be displayed.
Business Rules:	None
Notes and Issues	None
Assumptions:	<ol style="list-style-type: none"> 1. User must have a stable internet connection. 2. User should be able to enter their credentials without any error. 3. User should be able to find the inbox icon at the top. 4. The user must be able to find the chat media icon.

11. Clear Chat

The tabular use case of Chat box interaction module is given below:

Table 35: Tabular Use Case UC-2.12

Use Case ID:	UC-2.12
Use Case Name:	Clear Chat
Actors:	Renter/Proprietor
Description:	The goal is to enable the Renter/Proprietor to clear the chat box by deleting the messages.
Priority	Medium
Preconditions:	<ol style="list-style-type: none"> 1. User must be logged into the application. 2. User must click the inbox icon to open it.
Trigger	User selects the chat and select Clear Chat button.
Include	None.
Normal Flow:	<ol style="list-style-type: none"> 1. The user will open the web or mobile application 2. The user will sign into his/her account 3. After the home screen appears, the user will click the inbox icon to open the inbox. 4. Then open the chat and select Clear Chat.
Alternative Flows:	None.
Exceptions:	None
Post condition:	The chat box will be cleared.
Business Rules:	None.
Notes and Issues	None.
Assumptions:	<ol style="list-style-type: none"> 1. User must have a stable internet connection. 2. User should be able to enter their credentials without any error. 3. User should be able to find the inbox icon at the top.

12. Unpin Message

The tabular use case of Chat box interaction module is given below:

Table 36: Tabular Use Case UC-2.14

Use Case ID:	UC-2.14
Use Case Name:	Unpin Message
Actors:	Renter/Proprietor
Description:	The goal is to enable the Renter/Proprietor to unpin pinned messages.
Priority	Medium
Preconditions:	<ul style="list-style-type: none"> 1. User must be logged into the application. 2. User must click the inbox icon to open it. 3. The user must select the pinned message to be unpinned.
Trigger	When the user clicks the Inbox Icon, all the chats appear in front of the user, then the user can select a pinned chat to unpin it.
Include	UC-2.5
Normal Flow:	None
Alternative Flows:	None
Exceptions:	If the application is not responding, then he/she cannot unpin a message.
Post condition:	The message will be unpinned.
Business Rules:	None
Notes and Issues	None
Assumptions:	<ul style="list-style-type: none"> 1. User must have a stable internet connection. 2. User should be able to enter his/her credentials without any error. 3. The User must login into the application. 4. User should be able to find the inbox icon at the top. 5. User should be able to understand English Language.

➤ **Module 3: Rent Payment Portal**

The tabular use cases of rent payment portal are provided below:

1. Pay Rent

The tabular use case of Rent Payment Portal is given below:

Table 37: Tabular Use Case UC-3.1

Use Case ID:	UC-3.1
Use Case Name:	Pay Rent
Actors:	Renter
Description:	Renters can pay their rents online through the application.
Priority:	High
Preconditions:	User must be logged in to their account through Renter Portal.
Trigger:	Renter clicks on Pay Rent button.
Include:	UC-1.2
Normal Flow:	<ol style="list-style-type: none"> 1. User selects Renter Portal. 2. User goes to the Rent Payment Portal 3. Selects Pay Rent. 4. Proceeds to the selection of payment method and entering valid information required for the chosen method.
Alternative Flows:	Instead of going to the Rent Payment Portal directly from main menu, the renter can go to the portal through Property Management Portal for paying rent. Use case steps 3-4 will be followed in normal flow.
Exceptions:	<ul style="list-style-type: none"> • In step 4 of normal flow, if the user fails to enter correct card details, <ol style="list-style-type: none"> 1. An appropriate error message will be displayed 2. Asking the user to enter the credentials again 3. Then user will provide correct data. 4. In case the system detects suspicious behavior, the renter will be forcibly signed out. • If internet connection is lost, the application will try to re-establish the connection so payments are done. If it fails to do so within 5 seconds, the user will be redirected to the last menu that was opened.
Post condition:	Rent Payment Status will be updated and the landlord will get notified about the payment.
Business Rules:	User should have stable internet connection and valid card details.
Notes & Issues:	None.
Assumptions:	<ol style="list-style-type: none"> 1. User will have a stable internet connection. 2. User will be able to find and select the option from the menu correctly. 3. User has basic knowledge of online payments.

2. Generate Invoice

The tabular use case of Rent Payment Portal is given below:

Table 38: Tabular Use Case UC-3.2

Use Case ID:	UC-3.2
Use Case Name:	Generate Invoice
Actors:	Renter
Description:	Renters can generate invoice of their payment history.
Priority:	Medium
Preconditions:	User must be logged in to their account through Renter Portal.
Trigger:	Renter clicks on Generate Invoice button.
Include:	UC-1.2
Normal Flow:	<ol style="list-style-type: none"> 1. User selects Renter Portal. 2. User goes to the Rent Payment Portal 3. Selects Generate Invoice. 4. A pdf of invoice will become ready for downloading.
Alternative Flows:	Instead of going to the Rent Payment Portal directly from main menu, the user can go to the portal through Property Management Portal as well. Use case steps 3-4 will be followed in normal flow.
Exceptions:	If internet connection is lost, the application will try to re-establish its connection. If it fails to do so within 5 seconds, the user will be redirected to the last menu that was opened.
Post condition:	Renter will have the information of their payment history in the form of a softcopy of invoice.
Business Rules:	User should have stable internet connection.
Notes & Issues:	None.
Assumptions:	<ol style="list-style-type: none"> 1. User will have a stable internet connection. 2. User will be able to find and select the option from the menu correctly. 3. User has basic knowledge of online payments.

3. Print Invoice

The tabular use case of Rent Payment Portal is given below:

Table 39: Tabular Use Case UC-3.3

Use Case ID:	UC-3.3
Use Case Name:	Print Invoice
Actors:	Renter
Description:	Renters can print invoice of their payment history.
Priority:	Low
Preconditions:	User must be logged in to their account through Renter Portal.
Trigger:	Renter clicks on Print Invoice button.
Include:	UC-1.2
Normal Flow:	<ol style="list-style-type: none"> 1. User selects Renter Portal. 2. User goes to the Rent Payment Portal 3. Selects Print Invoice. 4. A pdf of invoice will become ready for printing.
Alternative Flows:	Instead of going to the Rent Payment Portal directly from main menu, the user can go to the portal through Property Management Portal as well. Use case steps 3-4 will be followed in normal flow.
Exceptions:	<ul style="list-style-type: none"> • If internet connection is lost, the application will try to re-establish its connection. If it fails to do so within 5 seconds, the user will be redirected to the last menu that was opened. • If printer is not connected, it will wait for 5 seconds. After the time is reached, the print command will be cancelled.
Post condition:	Renter will have the information of their payment history in the form of a hardcopy of invoice.
Business Rules:	User should have stable internet connection and printer connected to the device.
Notes & Issues:	None.
Assumptions:	<ol style="list-style-type: none"> 1. User will have a stable internet connection. 2. User will be able to find and select the option from the menu correctly. 3. User has basic knowledge of online payments. 4. User has a printer.

4. View Payment History

The tabular use case of Rent Payment Portal is given below:

Table 40: Tabular Use Case UC-3.4

Use Case ID:	UC-3.4
Use Case Name:	View Payment History
Actors:	Renter/Proprietor
Description:	Renter/Proprietor can view their payments history.
Priority:	Medium
Preconditions:	User must be logged in to their account through Renter/Proprietor Portal.
Trigger:	Renter clicks on Rent Payment Portal button.
Include:	UC-1.2
Normal Flow:	None.
Alternative Flows:	Instead of going to the Rent Payment Portal directly from main menu, the user can go to the portal through Property or Tenant Management Portal as well.
Exceptions:	If internet connection is lost, the application will try to re-establish its connection. If it fails to do so within 5 seconds, the user will be redirected to the last menu that was opened.
Post condition:	Renter will have the information of their payment history. Proprietor will have the information about their received payment history.
Business Rules:	User should have stable internet connection.
Notes & Issues:	None.
Assumptions:	<ol style="list-style-type: none"> 1. User will have a stable internet connection. 2. User will be able to find and select the option from the menu correctly. 3. User has basic knowledge of online payments.

5. Save Card Details

The tabular use case of Rent Payment Portal is given below:

Table 41: Tabular Use Case UC-3.5

Use Case ID:	UC-3.5
Use Case Name:	Save Card Details
Actors:	Renter/Proprietor
Description:	Renters and proprietors can save their card details at the time of online payment.
Priority:	Low
Preconditions:	User must be logged in to their account through Renter/Proprietor Portal.
Trigger:	When the user clicks on send payment.
Include:	UC-3.1, UC-3.10
Normal Flow:	<ol style="list-style-type: none"> Proceeds to the selection of payment method and entering valid information required for the chosen method. Clicks on Send Payment and get a prompt asking whether to save card details or not.
Alternative Flows:	Instead of going to the Rent Payment Portal directly from main menu, the user can go to the portal through Property Management Portal as well. Use case steps 1-2 will be followed in normal flow.
Exceptions:	If internet connection is lost, the application will try to re-establish its connection. If it fails to do so within 5 seconds, the user will be redirected to the last menu that was opened.
Post condition:	User's card details will be saved and will be used next time online payment is initiated through the application.
Business Rules:	User should have stable internet connection and valid card details.
Notes & Issues:	None.
Assumptions:	<ol style="list-style-type: none"> User will have a stable internet connection. User will be able to find and select the option from the menu correctly. User has basic knowledge of online payments.

6. Request Payment Deadline Extension

The tabular use case of Rent Payment Portal is given below:

Table 42: Tabular Use Case UC-3.6

Use Case ID:	UC-3.6
Use Case Name:	Request Payment Deadline Extension
Actors:	Renter
Description:	Renter can request rent deadline extension from the respective landlord.
Priority:	Medium
Preconditions:	User must be logged in to their account through Renter Portal.
Trigger:	Renter clicks Request Deadline Extension button next to the payment status tab.
Include:	UC-1.2
Normal Flow:	1. None.
Alternative Flows:	Instead of going to the Rent Payment Portal directly from main menu, the user can go to the portal through Property Management Portal as well.
Exceptions:	If internet connection is lost, the application will try to re-establish its connection. If it fails to do so within 5 seconds, the user will be redirected to the last menu that was opened.
Post condition:	Proprietor will receive deadline extension request.
Business Rules:	User should have stable internet connection.
Notes & Issues:	None.
Assumptions:	<ol style="list-style-type: none"> 1. User will have a stable internet connection. 2. User will be able to find and select the option from the menu correctly. 3. User has basic knowledge of online payments.

7. Respond To Payment Deadline Extension Request

The tabular use case of Rent Payment Portal is given below:

Table 43: Tabular Use Case UC-3.7

Use Case ID:	UC-3.7
Use Case Name:	Respond To Payment Deadline Extension Request
Actors:	Proprietor
Description:	Proprietor can respond to the renter's request of rent deadline extension.
Priority:	Medium
Preconditions:	User must be logged in to their account through Proprietor Portal.
Trigger:	Proprietor clicks Respond To Payment Deadline Extension Request button next to the payment status tab.
Include:	UC-1.2
Normal Flow:	None.
Alternative Flows:	Instead of going to the Rent Payment Portal directly from main menu, the user can go to the portal through Tenant Management Portal as well.
Exceptions:	If internet connection is lost, the application will try to re-establish its connection. If it fails to do so within 5 seconds, the user will be redirected to the last menu that was opened.
Post condition:	Proprietor will receive deadline extension request.
Business Rules:	User should have stable internet connection.
Notes & Issues:	None.
Assumptions:	<ol style="list-style-type: none"> 1. User will have a stable internet connection. 2. User will be able to find and select the option from the menu correctly. 3. User has basic knowledge of online payments.

8. View Payment Status

The tabular use case of Rent Payment Portal is given below:

Table 44: Tabular Use Case UC-3.8

Use Case ID:	UC-3.8
Use Case Name:	View Payment Status
Actors:	Renter/Proprietor
Description:	Renter/Proprietor can view the payment's status of individual renters.
Priority:	Medium
Preconditions:	User must be logged in to their account through Renter/Proprietor Portal.
Trigger:	User clicks the Rent Payment Portal.
Include:	UC-1.2
Normal Flow:	1. Payment Status will be displayed on the home screen.
Alternative Flows:	Instead of going to the Rent Payment Portal directly from main menu, the user can go to the portal through Property or Tenant Management Portal as well.
Exceptions:	If internet connection is lost, the application will try to re-establish its connection. If it fails to do so within 5 seconds, the user will be redirected to the last menu that was opened.
Post condition:	Renter and Proprietor will have the information of the payment status of individual renters.
Business Rules:	User should have stable internet connection.
Notes & Issues:	None.
Assumptions:	<ol style="list-style-type: none"> 1. User will have a stable internet connection. 2. User will be able to find and select the option from the menu correctly. 3. User has basic knowledge of online payments.

9. Edit Payment Details

The tabular use case of Rent Payment Portal is given below:

Table 45: Tabular Use Case UC-3.9

Use Case ID:	UC-3.9
Use Case Name:	Edit Payment Details
Actors:	Proprietor
Description:	Proprietor can edit their payment details that will be used by the renters for paying rents.
Priority:	Medium
Preconditions:	User must be logged in to their account through Renter Portal.
Trigger:	User clicks the Rent Payment Portal.
Include:	UC-1.2
Normal Flow:	1. Selects Edit Payment Details.
Alternative Flows:	Instead of going to the Rent Payment Portal directly from main menu, the user can go to the portal through Tenant Management Portal as well.
Exceptions:	If internet connection is lost, the application will try to re-establish its connection. If it fails to do so within 5 seconds, the user will be redirected to the last menu that was opened.
Post condition:	Bank details with respect to the property ID will be changed.
Business Rules:	User should have stable internet connection and verified card details.
Notes & Issues:	None.
Assumptions:	<ol style="list-style-type: none"> 1. User will have a stable internet connection. 2. User will be able to find and select the option from the menu correctly. 3. User has basic knowledge of online payments.

10. Buy Premium Account

The tabular use case of Rent Payment Portal is given below:

Table 46: Tabular Use Case UC-3.10

Use Case ID:	UC-3.10
Use Case Name:	Buy Premium Account
Actors:	Renter/Proprietor
Description:	Renter/Proprietor can get premium account to experience smooth application experience.
Priority:	High
Preconditions:	User must be logged in to their account through Renter Portal.
Trigger:	User clicks the Rent Payment Portal.
Include:	UC-1.2
Normal Flow:	<ol style="list-style-type: none"> 1. Selects Buy Premium Account. 2. Proceeds to the selection of payment method and entering valid information required for the chosen method.
Alternative Flows:	Instead of going to the Rent Payment Portal directly from main menu, the user can go to the portal through Property or Tenant Management Portal as well. Use case steps 1-2 will be followed in normal flow.
Exceptions:	<ul style="list-style-type: none"> • In step 2 of normal flow, if the user fails to enter correct card details, <ol style="list-style-type: none"> 1. An appropriate error message will be displayed 2. Asking the user to enter the credentials again 3. Then user will provide correct data. 4. In case the system detects suspicious behavior, the renter will be forcibly signed out. • If internet connection is lost, the application will try to re-establish the connection so payments are done. If it fails to do so within 5 seconds, the user will be redirected to the last menu that was opened.
Post condition:	Pop-up ads will be removed for faster and smoother experience.
Business Rules:	User should have stable internet connection and verified card details.
Notes & Issues:	None.
Assumptions:	<ol style="list-style-type: none"> 1. User will have a stable internet connection. 2. User will be able to find and select the option from the menu correctly. 3. User has basic knowledge of online payments.

11. Open Payment Details

The tabular use case of Rent Payment Portal is given below:

Table 47: Tabular Use Case UC-3.11

Use Case ID:	UC-3.11
Use Case Name:	Open Payment Details
Actors:	Renter/Proprietor
Description:	Renter/Proprietor can view individual detailed information of any payment.
Priority:	Medium
Preconditions:	User must be logged in to their account through Renter/Proprietor Portal.
Trigger:	Renter clicks on Rent Payment Portal button.
Include:	UC-3.4
Normal Flow:	None.
Alternative Flows:	Instead of going to the Rent Payment Portal directly from main menu, the user can go to the portal through Property or Tenant Management Portal as well.
Exceptions:	If internet connection is lost, the application will try to re-establish its connection. If it fails to do so within 5 seconds, the user will be redirected to the last menu that was opened.
Post condition:	Users will have detailed view of payment details of an individual transaction.
Business Rules:	User should have stable internet connection.
Notes & Issues:	None.
Assumptions:	<ol style="list-style-type: none"> 1. User will have a stable internet connection. 2. User will be able to find and select the option from the menu correctly. 3. User has basic knowledge of online payments.

12. Hide Pop-up Ads

The tabular use case of Rent Payment Portal is given below:

Table 48: Tabular Use Case UC-3.12

Use Case ID:	UC-3.12
Use Case Name:	Hide Pop-Up Ads
Actors:	System
Description:	System will hide pop-up ads to provide smooth app experience.
Priority:	High
Preconditions:	User buys premium account.
Trigger:	A user clicks Send Payment button after entering required payment credentials.
Include:	UC-3.10
Normal Flow:	None.
Alternative Flows:	None.
Exceptions:	None.
Post condition:	User will not see ads on their screens anymore.
Business Rules:	Machine Learning.
Notes & Issues:	None.
Assumptions:	1. User will have a stable internet connection.

➤ ***Module 4: Complaint Management Portal***

The tabular use cases of complaint management portal are provided below:

1. Create Complaint

The tabular use case of complaint management portal module is given below:

Table 49: Tabular Use Case of UC-4.1

Use Case ID:	UC-4.1
Use Case Name:	Make Complaint
Actors:	Renter
Description:	The goal is to enable the renter to send complaints to the landlord which needs to be resolved as soon as possible.
Priority	High
Preconditions:	<ol style="list-style-type: none"> 1. User must be logged into the application. 2. The user must select complaint option to register a complaint.
Trigger	When the user clicks the complaint option provided in the interface of the Complaint Portal.
Include	None.
Normal Flow:	<ol style="list-style-type: none"> 1. The user will open the web or mobile application 2. The user will sign into his/her account 3. After the home screen appears, the user will click the complaint option to add a complaint. 4. The user will have to select the complaint type 5. The user will have to provide description of what the complaint is about.
Alternative Flows:	None.
Exceptions:	If the user is not connected to the internet, then he/she cannot create a complaint.
Post condition:	The Renter will register a complaint which will be received by the landlord.
Business Rules:	High.
Notes & Issues	High.
Assumptions:	<ol style="list-style-type: none"> 1. User must have a stable internet connection. 2. User should be able to enter his/her credentials without any error. 3. User should be able to find the complaint option 4. User should be able to understand English Language. 5. The user should be able to type the complaint.

2. View Complaint Status

The tabular use case of complaint management portal module is given below:

Table 50: Tabular Use Case of UC-4.2

Use Case ID:	UC-4.2
Use Case Name:	View Complaint
Actors:	Renter
Description:	The goal is to enable the renter to view complaint.
Priority	High.
Preconditions:	<ol style="list-style-type: none"> 1. User must be logged into the application. 2. The user must select the View Complaint option.
Trigger	When the user clicks the complaint option provided in the interface of the Complaint Portal.
Include	None.
Normal Flow:	<ol style="list-style-type: none"> 1. The user will open the web or mobile application 2. The user will sign into his/her account 3. After the home screen appears, the user will click the complaint option to view a complaint.
Alternative Flows:	None.
Exceptions:	If the user is not connected to the internet, then he/she cannot view a complaint.
Post condition:	The user will be able to view the registered complaint.
Business Rules:	None.
Notes and Issues	None.
Assumptions:	<ol style="list-style-type: none"> 1. User must have a stable internet connection. 2. The user must have registered a complaint before.

3. Delete Complaint

The tabular use case of complaint management portal module is given below:

Table 51: Tabular Use Case of UC-4.3

Use Case ID:	UC-4.3
Use Case Name:	Delete Complaint
Actors:	Renter
Description:	The goal is to enable the renter to delete complaint.
Priority	High.
Preconditions:	<ol style="list-style-type: none"> 1. User must be logged into the application. 2. The user must select the View Complaint option.
Trigger	When the user clicks the complaint option provided in the interface of the Complaint Portal.
Include	None.
Normal Flow:	<ol style="list-style-type: none"> 1. The user will open the web or mobile application 2. The user will sign into his/her account 3. After the home screen appears, the user will click the complaint option to view a complaint. 4. Then the user will select the complaint and click the delete option provided in the interface. <p>Then the user will delete the complaint.</p>
Alternative Flows:	None.
Exceptions:	If the user is not connected to the internet, then he/she cannot delete a complaint.
Post condition:	The user will be able to delete the complaint
Business Rules:	None
Notes and Issues	None
Assumptions:	<ol style="list-style-type: none"> 1. User must have a stable internet connection. 2. The user must understand the English language. 3. The user must have registered a complaint before.

4. Respond to Complain

The tabular use case of complaint management portal module is given below:

Table 52: Tabular Use Case of UC-4.4

Use Case ID:	UC-4.4
Use Case Name:	Respond to a Complaint
Actors:	Proprietor
Description:	The goal is to enable the renter to delete complaint.
Priority	High
Preconditions:	<ol style="list-style-type: none"> 1. User must be logged into the application. 2. The user must select the View Complaint option.
Trigger	When the user clicks the complaint option provided in the interface of the Complaint Portal.
Include	None
Normal Flow:	<ol style="list-style-type: none"> 1. The user will open the web or mobile application 2. The user will sign into his/her account 3. After the home screen appears, the user will click the complaint option to view a complaint. 4. The user will start resolving the complaint if the reasons are valid as soon as possible 5. Then the user will select the complaint and click the respond option provided in the interface.
Alternative Flows:	The renter can delete the complaint if the issue is resolved If the user is not connected to the internet, then he/she cannot delete a complaint.
Exceptions:	If the user is not connected to the internet, then he/she cannot delete a complaint.
Post condition:	The user will be able to resolve the registered complaint.
Business Rules:	None.
Notes and Issues	None.
Assumptions:	<ol style="list-style-type: none"> 1. User must have a stable internet connection. 2. The user must understand the English language. 3. The user must have registered a complaint before.

5. Edit Compliant Description

The tabular use case of complaint management portal module is given below:

Table 53: Tabular Use Case of UC-4.5

Use Case ID:	UC-4.5
Use Case Name:	Edit Complaint Description
Actors:	Renter
Description:	The goal is to enable the renter to edit the description of the complaint.
Priority	Medium.
Preconditions:	<ol style="list-style-type: none"> 1. User must be logged into the application. 2. The user must select the View Complaint option.
Trigger	When the user clicks the complaint option provided in the interface of the Complaint Portal.
Include	None.
Normal Flow:	<ol style="list-style-type: none"> 1. The user will open the web or mobile application 2. The user will sign into his/her account 3. After the home screen appears, the user will click the complaint option to view a complaint. 4. Then the user will select the complaint and click the edit option provided in the interface.
Alternative Flows:	If the user is not connected to the internet, then he/she cannot edit a complaint.
Exceptions:	None
Post condition:	The user will be able to view and edit the registered complaint.
Business Rules:	None.
Notes and Issues	None.
Assumptions:	<ol style="list-style-type: none"> 1. User must have a stable internet connection. 2. The user must understand the English language. 3. The user must have registered a complaint before.

➤ **Module 5: Tenant Management Portal**

The tabular use cases of tenant management portal are provided in the following:

1. Add Property

The tabular use case of tenant management portal module is given below:

Table 54: Tabular Use Case of UC-5.1

Use Case ID:	UC-5.1
Use Case Name:	Add a property
Actors:	Proprietor
Description:	The goal is to enable the Proprietor to add a property to be rented.
Priority	High
Preconditions:	<ol style="list-style-type: none"> 1. User must be logged into the application. 2. The user must be a landlord. 3. The user must have properties to be rented. 4. The user selects the Add a property option.
Trigger	When the user clicks the Add a property option, then this event will be triggered.
Include	None.
Normal Flow:	<ol style="list-style-type: none"> 1. The user will open the web or mobile application 2. The user will sign into his/her account 3. After the home screen appears, will move to the tenant management where he will have multiple options to select. 4. The Proprietor will select Add a property option.
Alternative Flows:	None
Exceptions:	If the user is not connected to the internet, then he/she cannot Add a property.
Post condition:	The details of property which the proprietor owns will be added to the system
Business Rules:	None
Notes and Issues	None
Assumptions:	<ol style="list-style-type: none"> 1. User must have a stable internet connection. 2. The user must understand the English language. 3. The user must have an owned property before.

2. View Property

The tabular use case of tenant management portal module is given below:

Table 55: Tabular Use Case of UC-5.2

Use Case ID:	UC-5.2
Use Case Name:	View a property
Actors:	Proprietor
Description:	The goal is to enable the Proprietor to view a property to be rented.
Priority	High
Preconditions:	<ul style="list-style-type: none"> 1. User must be logged into the application. 2. The user must be a landlord. 3. The user must have properties to be rented. 4. The user selects the View a property option.
Trigger	When the user clicks the View a property option, then this event will be triggered.
Include	None
Normal Flow:	<ul style="list-style-type: none"> 1. The user will open the web or mobile application 2. The user will sign into his/her account 3. After the home screen appears, will move to the tenant management where he will have multiple options to select. 4. The Proprietor will select View a property option.
Alternative Flows:	None.
Exceptions:	If the user is not connected to the internet, then he/she cannot view a property.
Post condition:	The Proprietor will be able to view every detail of the property added.
Business Rules:	None
Notes and Issues	None
Assumptions:	<ul style="list-style-type: none"> 1. User must have a stable internet connection. 2. The user must understand the English language. 3. The user must have added an owned property before.

3. Delete Property

The tabular use case of tenant management portal module is given below:

Table 56: Tabular Use Case of UC-5.3

Use Case ID:	UC-5.3
Use Case Name:	Delete a property
Actors:	Proprietor
Description:	The goal is to enable the Proprietor to Delete a property to be rented.
Priority	High
Preconditions:	<ol style="list-style-type: none"> 1. User must be logged into the application. 2. The user must be a landlord. 3. The user must have added properties to be rented. 4. The user selects the Delete a property option.
Trigger	When the user clicks the Delete a property option, then this event will be triggered.
Include	None.
Normal Flow:	<ol style="list-style-type: none"> 1. The user will open the web or mobile application 2. The user will sign into his/her account 3. After the home screen appears, will move to the tenant management where he will have multiple options to select. 4. The Proprietor will select Delete a property option. 5. After that the user will select the property to be Deleted
Alternative Flows:	None
Exceptions:	If the user is not connected to the internet, then he/she cannot Delete a property.
Post condition:	The Proprietor will be able to Delete the added property.
Business Rules:	None.
Notes and Issues	None.
Assumptions:	<ol style="list-style-type: none"> 1. User must have a stable internet connection. 2. The user must understand the English language. 3. The user must have added an owned property before.

4. Generate Property Id

The tabular use case of tenant management portal module is given below:

Table 57: Tabular Use Case of UC-5.4

Use Case ID:	UC-5.4
Use Case Name:	Generate Property ID
Actors:	System
Description:	The System will generate property ID for each property.
Priority	High
Preconditions:	Property adding process is performed successfully.
Trigger	A person adds a new property.
Include	UC-5.1
Normal Flow:	None
Alternative Flows:	None.
Exceptions:	<p>In step 4 of normal flow, if the user fails to authorize the property,</p> <ol style="list-style-type: none"> 1. Property Addition process will be cancelled. 2. An appropriate error message will be displayed 3. Asking the user to enter the credentials again 4. Then user will provide correct data. <p>Use Case resumes from step 5 of normal flow</p>
Post condition:	Property is assigned with a property ID.
Business Rules:	User should have stable internet connection and verified email address.
Notes and Issues	None
Assumptions:	<ol style="list-style-type: none"> 1. User will have a stable internet connection. 2. User will be able to find and select the option from the menu correctly

5. View Renters

The tabular use case of tenant management portal module is given below:

Table 58: Tabular Use Case of UC-5.5

Use Case ID:	UC-5.5
Use Case Name:	View Renters
Actors:	Proprietor
Description:	The Proprietor will be able to view all the renters of his properties.
Priority	High
Preconditions:	<ol style="list-style-type: none"> 1. The user has logged into the account. 2. The user has open the tenant portal.
Trigger	The user wants to view a profile.
Include	<ol style="list-style-type: none"> 1. The user will open the web/mobile application. 2. User successfully signed up for the application. 3. User will be click on view profile of the renter to view their profile.
Normal Flow:	None.
Alternative Flows:	None.
Exceptions:	If internet connection is lost, the application will try to re-establish its connection. If it fails to do so within 5 seconds, the user will be redirected to the last menu that was opened.
Post condition:	
Business Rules:	User should have stable internet connection and verified email address.
Notes and Issues	None.
Assumptions:	<ol style="list-style-type: none"> 1. User will have a stable internet connection. 2. User will be able to understand Urdu and English language. 3. User will be able to find and select the option from the menu correctly.

6. Add Renter

The tabular use case of tenant management portal module is given below:

Table 59: Tabular Use Case of UC-5.7

Use Case ID:	UC-5.7
Use Case Name:	Add Renter
Actors:	Proprietor
Description:	The proprietor will be able to add a new renter of a property.
Priority	High
Preconditions:	<ol style="list-style-type: none"> 1. The user has logged into the account 2. The user wants to add the renter's details.
Trigger	The user clicks on add a new renter option.
Include	None
Normal Flow:	<ol style="list-style-type: none"> 1. The user will open the web/mobile application. 2. User successfully signed up for the application. 3. User will open the proprietor portal. 4. The proprietor will choose the property and add a renter by entering their user ID. 5. The new renter will be added to the property.
Alternative Flows:	None
Exceptions:	<p>If internet connection is lost, the application will try to re-establish its connection. If it fails to do so within 5 seconds, the user will be redirected to the last menu that was opened. If wrong user ID is entered, an error message will be displayed asking for correct user ID.</p>
Post condition:	New Renter will be added.
Business Rules:	User should have stable internet connection and verified email address.
Notes and Issues	None.
Assumptions:	<ol style="list-style-type: none"> 1. User will have a stable internet connection. 2. User will be able to understand Urdu and English language. 3. User will be able to find and select the option from the menu correctly. 4. A property would be added before adding a renter before.

7. Remove Renter

The tabular use case of tenant management portal module is given below:

Table 60: Tabular Use Case of UC-5.8

Use Case ID:	UC-5.8
Use Case Name:	Remove Renter
Actors:	Proprietor
Description:	The proprietor will be able to remove renter of a property.
Priority	High
Preconditions:	<ul style="list-style-type: none"> 1. The user has logged into the account 2. The user wants to remove the renter's details.
Trigger	The user wants to remove a renter.
Include	None.
Normal Flow:	<ul style="list-style-type: none"> 1. The user will open the web/mobile application. 2. User successfully signed up for the application. 3. User will open the proprietor portal 4. The renter will be selected and then Remove Renter button will be pressed. 5. After proper authentication and 0 outstanding dues, the renter will be removed from the property.
Alternative Flows:	None.
Exceptions:	<p>If internet connection is lost, the application will try to re-establish its connection. If it fails to do so within 5 seconds, the user will be redirected to the last menu that was opened.</p> <p>If outstanding dues are not zero, the renter will be given a deadline of 2 days to clear dues otherwise legal action will be taken.</p>
Post condition:	Renter will be removed.
Business Rules:	User should have stable internet connection and verified email address.
Notes and Issues	None
Assumptions:	<ul style="list-style-type: none"> 1. User will have a stable internet connection. 2. User will be able to understand Urdu and English language. 3. User will be able to find and select the option from the menu correctly. 4. A renter should be existing before removal.

8. View Lease

The tabular use case of tenant management portal module is given below:

Table 61: Tabular Use Case of UC-5.9

Use Case ID:	UC-5.9
Use Case Name:	View Lease
Actors:	Proprietor
Description:	The goal is to allow proprietors to view lease.
Priority	High
Preconditions:	User must be logged in to their account through Proprietor Portal.
Trigger	On selecting the Tenant Management Portal, the use case will be initiated.
Include	None
Normal Flow:	<ol style="list-style-type: none"> 1. The user will open the web/mobile application. 2. User successfully logged in to their account. 3. User selects Proprietor Portal. 4. User goes to the Tenant Management Portal 5. Selects View Lease. 6. A pdf of lease will open after downloading into the device.
Alternative Flows:	None
Exceptions:	If internet connection is lost, the application will try to re-establish its connection. If it fails to do so within 5 seconds, the user will be redirected to the last menu that was opened.
Post condition:	Proprietor will have the information of their rent contract in the form of a softcopy of lease.
Business Rules:	User should have stable internet connection and printer connected to the device.
Notes and Issues	None
Assumptions:	<ol style="list-style-type: none"> 1. User will have a stable internet connection. 2. User will be able to find and select the option from the menu correctly. 3. User has basic knowledge of lease. 4. Lease contract should be made before.

9. Print Lease

The tabular use case of tenant management portal module is given below:

Table 62: Tabular Use Case of UC-5.10

Use Case ID:	UC-5.10
Use Case Name:	Print Lease
Actors:	Proprietor
Description:	The goal is to allow proprietors to print lease.
Priority	Medium
Preconditions:	User must be logged in to their account through Proprietor Portal.
Trigger	On selecting the Tenant Management Portal, the use case will be initiated.
Include	None.
Normal Flow:	<ol style="list-style-type: none"> 1. The user will open the web/mobile application. 2. User successfully logged in to their account. 3. User selects Proprietor Portal. 4. User goes to the Tenant Management Portal 5. Selects Print Lease. 6. A pdf of lease will become ready for printing.
Alternative Flows:	None.
Exceptions:	<p>If internet connection is lost, the application will try to re-establish its connection. If it fails to do so within 5 seconds, the user will be redirected to the last menu that was opened.</p> <p>If printer is not connected, it will wait for 5 seconds, After the time is reached, the print command will be cancelled.</p>
Post condition:	Renter will have the information of their rent contract in the form of a hardcopy of lease.
Business Rules:	User should have stable internet connection and printer connected to the device.
Notes and Issues	None
Assumptions:	<ol style="list-style-type: none"> 1. User will have a stable internet connection. 2. User will be able to find and select the option from the menu correctly. 3. User has basic knowledge of online payments. 4. User should have a printer.

10. Renew Lease

The tabular use case of tenant management portal module is given below:

Table 63: Tabular Use Case of UC-5.11

Use Case ID:	UC-5.11
Use Case Name:	Renew Lease
Actors:	Proprietor
Description:	The goal is to allow proprietors to renew lease on mutual demands.
Priority	Medium
Preconditions:	User must be logged in to their account through Proprietor Portal.
Trigger	On selecting the Tenant Management Portal, the use case will be initiated.
Include	None.
Normal Flow:	<ol style="list-style-type: none"> 1. The user will open the web/mobile application. 2. User successfully logged in to their account. 3. User selects Proprietor Portal. 4. User goes to the Tenant Management Portal 5. Selects Renew Lease. 6. After entering all the necessary details, the new lease will be sent to the renter. If they don't accept it, the old lease will not be replaced.
Alternative Flows:	None
Exceptions:	If internet connection is lost, the application will try to re-establish its connection. If it fails to do so within 5 seconds, the user will be redirected to the last menu that was opened.
Post condition:	Renter will receive a lease which they can accept or decline by giving reasons and negotiate over chat or phone.
Business Rules:	User should have stable internet connection and printer connected to the device.
Notes and Issues	None
Assumptions:	<ol style="list-style-type: none"> 1. User will have a stable internet connection. 2. User will be able to find and select the option from the menu correctly. 3. User has basic knowledge of lease.

11. Update Rent

The tabular use case of tenant management portal module is given below:

Table 64: Tabular Use Case of UC-5.12

Use Case ID:	UC-5.12
Use Case Name:	Update Rent
Actors:	Proprietor
Description:	Medium
Priority	The user wants to update the payment of the rent
Preconditions:	<ul style="list-style-type: none"> 1. The user has logged into the account 2. The user wants update rent.
Trigger	The Proprietor will be able to update the rent anytime.
Include	None.
Normal Flow:	<ul style="list-style-type: none"> 1. The user will open the web/mobile application. 2. User successfully signed up for the application. 3. User will open the proprietor menu 4. The user will have a list of renters. 5. User will be click on view profile and check their due rent. 6. The landlord can custom set up their rents
Alternative Flows:	None
Exceptions:	If internet connection is lost, the application will try to re-establish its connection. If it fails to do so within 5 seconds, the user will be redirected to the last menu that was opened.
Post condition:	The landlord will update the rent.
Business Rules:	User should have stable internet connection and verified email address.
Notes and Issues	None
Assumptions:	<ul style="list-style-type: none"> 1. User will have a stable internet connection. 2. User will be able to understand Urdu and English language. 3. User will be able to find and select the option from the menu correctly.

12. Edit Property Details

The tabular use case of tenant management portal module is given below:

Table 65: Tabular Use Case of UC-5.13

Use Case ID:	UC-5.13
Use Case Name:	Edit Property Details
Actors:	Proprietor
Description:	The goal is to allow proprietor to edit the added property details.
Priority	Medium
Preconditions:	User must be logged in to their account through Tenant Portal.
Trigger	On selecting the edit property details, the use case will be initiated.
Include	None.
Normal Flow:	<ol style="list-style-type: none"> 1. The user will open the web/mobile application. 2. User successfully logged in to their account. 3. User selects Proprietor Portal. 4. User goes to the Tenant Management Portal 5. Selects property, opens it and clicks Edit Property Details.
Alternative Flows:	None
Exceptions:	<p>In step 5 of normal flow, if the user fails to authorize the property,</p> <ol style="list-style-type: none"> 1. Property Addition process will be cancelled. 2. An appropriate error message will be displayed 3. Asking the user to enter the credentials again <p>Then user will provide correct data.</p>
Post condition:	Property Details that the renter could view as well will be changed.
Business Rules:	User should have stable internet connection and verified card details.
Notes and Issues	None
Assumptions:	<ol style="list-style-type: none"> 1. User will have a stable internet connection. 2. User will be able to find and select the option from the menu correctly. 3. User has basic knowledge of online payments.

13. Go to Payment Portal

The tabular use case of tenant management portal module is given below:

Table 66: Tabular Use Case of UC-5.14

Use Case ID:	UC-5.14
Use Case Name:	Go to Payment Portal
Actors:	Proprietor
Description:	The Proprietor can directly go to the payment portal as a shortcut.
Priority	Low
Preconditions:	<ul style="list-style-type: none"> 1) The user has logged into the account 2) The user is in the tenant management portal. 3) The user can directly go to the payment menu
Trigger	The user wants to go to the payment menu.
Include	None
Normal Flow:	<ul style="list-style-type: none"> 1. The user will open the web/mobile application. 2. User successfully signed up for the application. 3. User will open the proprietor menu 4. The user wants to go to the payment menu
Alternative Flows:	Instead of going to the Rent Payment Portal directly from main menu, the user can go to the portal through Tenant Management Portal as well. Use case steps 1-3 and 5 will be followed in normal flow.
Exceptions:	
Post condition:	The user will move to the payment portal.
Business Rules:	User should have stable internet connection and verified email address.
Notes and Issues	None
Assumptions:	<ul style="list-style-type: none"> 1) User will have a stable internet connection. 2) User will be able to understand Urdu and English language. 3) User will be able to find and select the option from the menu correctly.

14. Respond to Renter Connectivity Request

The tabular use case of tenant management portal module is given below:

Table 67: Tabular Use Case of UC-5.15

Use Case ID:	UC-5.15
Use Case Name:	Respond Renter Connectivity Request
Actors:	Proprietor
Description:	The Proprietor can respond to the request of renter as Yes/No
Priority	High
Preconditions:	<ul style="list-style-type: none"> 1) The user has logged into the account 2) The user is in the tenant management portal. 3) The user has new notification in the request section where the renter's contract request is found.
Trigger	The renter sends a request then this use case will be triggered.
Include	None.
Normal Flow:	<ul style="list-style-type: none"> 1) The user will open the web/mobile application. 2) User successfully signed up for the application. 3) User will open the proprietor menu 4) The user will open notifications where the renter's request is seen.
Alternative Flows:	None.
Exceptions:	Unstable Internet may cause errors and delays in accepting and delaying requests
Post condition:	The proprietor will respond to the request as Yes/No
Business Rules:	User should have stable internet connection and verified email address.
Notes and Issues	None
Assumptions:	<ul style="list-style-type: none"> 1) User will have a stable internet connection. 2) User will be able to understand Urdu and English language. 3) User will be able to find and select the option from the menu correctly.

➤ ***Module 6: Property Management Portal***

The tabular use cases of property management portal are provided in the following:

1. View Lease

The tabular use case of property management portal module is given below:

Table 68: Tabular Use Case of UC-6.1

Use Case ID:	UC-6.1
Use Case Name:	View Lease
Actors:	Renter
Description:	Renters can view lease.
Priority:	Medium
Preconditions:	User must be logged in to their account through Renter Portal.
Trigger:	Renter selects Property Management Portal.
Include:	UC-1.2
Normal Flow:	<ol style="list-style-type: none"> 1. Selects View Lease. 2. A pdf of lease will open after downloading into the device.
Alternative Flows:	None.
Exceptions:	If internet connection is lost, the application will try to re-establish its connection. If it fails to do so within 5 seconds, the user will be redirected to the last menu that was opened.
Post condition:	Renter will have the information of their rent contract in the form of a softcopy of lease.
Business Rules:	User should have stable internet connection.
Notes & Issues:	None.
Assumptions:	<ol style="list-style-type: none"> 1. User will have a stable internet connection. 2. User will be able to find and select the option from the menu correctly. 3. User has basic knowledge of lease.

2. View Property Details

The tabular use case of property management portal module is given below:

Table 69: Tabular Use Case of UC-6.2

Use Case ID:	UC-6.2
Use Case Name:	View Property Details
Actors:	Renter
Description:	The renters can view their rented properties' details.
Priority:	Medium
Preconditions:	User must be logged in to their account through Renter Portal.
Trigger:	Renter selects Property Management Portal.
Include:	UC-1.2
Normal Flow:	<ol style="list-style-type: none"> 1. User selects specific property. 2. Pop up options appear. 3. User selects View Property Details.
Alternative Flows:	None.
Exceptions:	If internet connection is lost, the application will try to re-establish its connection. If it fails to do so within 5 seconds, the user will be redirected to the last menu that was opened.
Post condition:	Renter will have the information about their rented property.
Business Rules:	User should have stable internet connection.
Notes & Issues:	None.
Assumptions:	<ol style="list-style-type: none"> 1. User will have a stable internet connection. 2. User will be able to find and select the option from the menu correctly. 3. User can understand English/Urdu language.

3. View Landlord's Details

The tabular use case of property management portal module is given below:

Table 70: Tabular Use Case of UC-6.3

Use Case ID:	UC-6.3
Use Case Name:	View Landlord's Details
Actors:	Renter
Description:	The renters can view their rented property's landlord's details.
Priority:	Medium
Preconditions:	User must be logged in to their account through Renter Portal.
Trigger:	Renter selects Property Management Portal.
Include:	UC-1.2
Normal Flow:	<ol style="list-style-type: none"> 1. User selects specific property. 2. Pop up options appear. 3. Selects View Landlord's Details.
Alternative Flows:	None.
Exceptions:	If internet connection is lost, the application will try to re-establish its connection. If it fails to do so within 5 seconds, the user will be redirected to the last menu that was opened.
Post condition:	Renter will have the information about their rented property's landlord.
Business Rules:	User should have stable internet connection.
Notes & Issues:	None.
Assumptions:	<ol style="list-style-type: none"> 1. User will have a stable internet connection. 2. User will be able to find and select the option from the menu correctly. 3. User can understand English/Urdu language.

4. Check Rent Tenure

The tabular use case of property management portal module is given below:

Table 71: Tabular Use Case of UC-6.4

Use Case ID:	UC-6.4
Use Case Name:	Check Rent Tenure
Actors:	Renter
Description:	The renters can view the tenure they have spent in renting a property.
Priority:	Medium
Preconditions:	User must be logged in to their account through Renter Portal.
Trigger:	Renter selects Property Management Portal.
Include:	UC-1.2
Normal Flow:	<ol style="list-style-type: none"> 1. User goes to the Property Management Portal 2. Tenure will be visible on the home page.
Alternative Flows:	None.
Exceptions:	If internet connection is lost, the application will try to re-establish its connection. If it fails to do so within 5 seconds, the user will be redirected to the last menu that was opened.
Post condition:	Renter will have the information about their rented property.
Business Rules:	User should have stable internet connection.
Notes & Issues:	None.
Assumptions:	<ol style="list-style-type: none"> 1. User will have a stable internet connection. 2. User will be able to find and select the option from the menu correctly. 3. User can understand English/Urdu language.

5. Request Lease Renewal

The tabular use case of property management portal module is given below:

Table 72: Tabular Use Case of UC-6.5

Use Case ID:	UC-6.5
Use Case Name:	Request Lease Renewal
Actors:	Renter
Description:	The use case will allow renters to request lease renewal on some demands from the landlord.
Priority:	Medium
Preconditions:	User must be logged in to their account through Renter Portal.
Trigger:	Renter selects Property Management Portal.
Include:	UC-1.2
Normal Flow:	<ol style="list-style-type: none"> 1. User selects specific property. 2. Pop up options appear. 3. Selects Request Lease Renewal. 4. The renter will write the demands and send it to the respective proprietor.
Alternative Flows:	None.
Exceptions:	If internet connection is lost, the application will try to re-establish its connection. If it fails to do so within 5 seconds, the user will be redirected to the last menu that was opened.
Post condition:	The proprietor will receive a request of lease renewal.
Business Rules:	User should have stable internet connection.
Notes & Issues:	None.
Assumptions:	<ol style="list-style-type: none"> 1. User will have a stable internet connection. 2. User will be able to find and select the option from the menu correctly. 3. User can understand English/Urdu language.

6. Print Lease

The tabular use case of property management portal module is given below:

Table 73: Tabular Use Case of UC-6.6

Use Case ID:	UC-6.6
Use Case Name:	Print Lease
Actors:	Renter
Description:	The renter can print lease.
Priority:	Low
Preconditions:	User must be logged in to their account through Renter Portal.
Trigger:	Renter selects Property Management Portal.
Include:	UC-1.2
Normal Flow:	<ol style="list-style-type: none"> 1. User selects specific property. 2. Pop up options appear. 3. Selects Print Lease. 4. A pdf of lease will become ready for printing.
Alternative Flows:	None.
Exceptions:	<ul style="list-style-type: none"> • If internet connection is lost, the application will try to re-establish its connection. If it fails to do so within 5 seconds, the user will be redirected to the last menu that was opened. • If printer is not connected, it will wait for 5 seconds. After the time is reached, the print command will be cancelled.
Post condition:	Renter will have the information of their rent contract in the form of a hardcopy of lease.
Business Rules:	User should have stable internet connection and printer connected to the device.
Notes & Issues:	None.
Assumptions:	<ol style="list-style-type: none"> 1. User will have a stable internet connection. 2. User will be able to find and select the option from the menu correctly. 3. User has basic knowledge of online payments. 4. User has a printer.

7. Respond to Lease

The tabular use case of property management portal module is given below:

Table 74: Tabular Use Case of UC-6.7

Use Case ID:	UC-6.7
Use Case Name:	Respond to Lease
Actors:	Renter
Description:	The use case will allow renters to accept or decline the lease proposed by the proprietor.
Priority:	High
Preconditions:	User must be logged in to their account through Renter Portal.
Trigger:	Renter selects Property Management Portal.
Include:	UC-1.2
Normal Flow:	<ol style="list-style-type: none"> 1. User goes to the Property Management Portal 2. User selects specific property. 3. Pop up options appear. 4. Selects Respond to Lease. 5. The renter will accept or decline the offer. While declining, they must write the reasons and send it to the respective proprietor.
Alternative Flows:	None.
Exceptions:	If internet connection is lost, the application will try to re-establish its connection. If it fails to do so within 5 seconds, the user will be redirected to the last menu that was opened.
Post condition:	The lease generated by the proprietor will receive an answer from the renter.
Business Rules:	User should have stable internet connection.
Notes & Issues:	None.
Assumptions:	<ol style="list-style-type: none"> 1. User will have a stable internet connection. 2. User will be able to find and select the option from the menu correctly. 3. User can understand English/Urdu language.

8. Connect with Landlord

The tabular use case of property management portal module is given below:

Table 75: Tabular Use Case of UC-6.8

Use Case ID:	UC-6.8
Use Case Name:	Connect with Landlord
Actors:	Renter
Description:	The renters can connect with the landlord and become part of their property.
Priority:	High
Preconditions:	User must be logged in to their account through Renter Portal.
Trigger:	Renter selects Connect with Landlord from the main menu of Property Management Portal.
Include:	UC-1.2
Normal Flow:	<ol style="list-style-type: none"> 1. User goes to the Property Management Portal 2. Selects Connect with Landlord. 3. The renter will enter property ID in the pop up window. A request will be sent to the proprietor.
Alternative Flows:	None.
Exceptions:	<ul style="list-style-type: none"> • If internet connection is lost, the application will try to re-establish its connection. If it fails to do so within 5 seconds, the user will be redirected to the last menu that was opened. • If invalid property ID is added, the application will show error message.
Post condition:	The proprietor will receive a request of connectivity.
Business Rules:	User should have stable internet connection and valid property ID.
Notes & Issues:	None.
Assumptions:	<ol style="list-style-type: none"> 1. User will have a stable internet connection. 2. User will be able to find and select the option from the menu correctly. 3. User can understand English/Urdu language.

9. Go to Payment Portal

The tabular use case of property management portal module is given below:

Table 76: Tabular Use Case of UC-6.9

Use Case ID:	UC-6.9
Use Case Name:	Go to Payment Portal
Actors:	Renter
Description:	The Renter can directly go to the payment portal as a shortcut.
Priority:	Low
Preconditions:	<ul style="list-style-type: none"> 1. The user has logged into the account 2. The user is in the property management portal. 3. The user can directly go to the payment menu
Trigger:	The user wants to go to the payment menu.
Include:	UC-1.2
Normal Flow:	<ul style="list-style-type: none"> 1. The user will open the web/mobile application. 2. User successfully signed up for the application. 3. User will open the proprietor menu 4. The user selects Payment Portal
Alternative Flows:	None.
Exceptions:	Unstable Internet may cause errors and delays.
Post condition:	The user will move to the payment portal.
Business Rules:	User should have stable internet connection and verified email address.
Notes & Issues:	None.
Assumptions:	<ul style="list-style-type: none"> 1. User will have a stable internet connection. 2. User will be able to understand Urdu and English language. 3. User will be able to find and select the option from the menu correctly.

➤ **Module 7: Push Notification**

The tabular use cases of the module push notifications is provided in the following:

1. Turn On Rent Reminders

The tabular use case of push notification module is given below:

Table 77: Tabular Use Case of UC-7.1

Use Case ID:	UC-7.1
Use Case Name:	Turn On Rent Reminders
Actors:	Renter/Proprietor
Description:	The users can receive rent reminders on their application when the application is not open.
Priority:	Medium
Preconditions:	User must be logged in to their account and the rent reminders are turned off beforehand.
Trigger:	User clicks on the Notification Settings Button from side pane.
Include:	UC-1.2
Normal Flow:	<ol style="list-style-type: none"> 1. Notification settings will be displayed. 2. The user can then turn on the rent reminders.
Alternative Flows:	None.
Exceptions:	If internet connection is lost, the application will try to re-establish its connection. If it fails to do so within 5 seconds, the user will be redirected to the last menu that was opened.
Post condition:	The user will be notified when due date is near or when the rent has been paid.
Business Rules:	User should have stable internet connection.
Notes & Issues:	None.
Assumptions:	<ol style="list-style-type: none"> 1. User will have a stable internet connection. 2. User will be able to find and select the option from the menu correctly. 3. User can understand English/Urdu language. 4. Rent Reminders are turned off.

2. Turn Off Rent Reminders

The tabular use case of push notification module is given below:

Table 78: Tabular Use Case of UC-7.2

Use Case ID:	UC-7.2
Use Case Name:	Turn Off Rent Reminders
Actors:	Renter/Proprietor
Description:	The users can stop receiving rent reminders on their application.
Priority:	Medium
Preconditions:	User must be logged in to their account and the rent reminders are turned on beforehand.
Trigger:	User clicks on the Notification Settings Button from side pane.
Include:	UC-1.2
Normal Flow:	<ol style="list-style-type: none"> 1. Notification settings will be displayed. 2. The user can then turn off the rent reminders.
Alternative Flows:	None.
Exceptions:	If internet connection is lost, the application will try to re-establish its connection. If it fails to do so within 5 seconds, the user will be redirected to the last menu that was opened.
Post condition:	The user will not be notified when due date is near or when the rent has been paid.
Business Rules:	User should have stable internet connection.
Notes & Issues:	None.
Assumptions:	<ol style="list-style-type: none"> 1. User will have a stable internet connection. 2. User will be able to find and select the option from the menu correctly. 3. User can understand English/Urdi language. 4. Rent Reminders are turned off.

3. Turn On Chat Notifications

The tabular use case of push notification module is given below:

Table 79: Tabular Use Case of UC-7.3

Use Case ID:	UC-7.3
Use Case Name:	Turn On Chat Notifications
Actors:	Renter/Proprietor
Description:	The users can turn on chat notifications on their application when the application is not open.
Priority:	Medium
Preconditions:	User must be logged in to their account and the chat notifications are turned off beforehand.
Trigger:	User clicks on the Notification Settings Button from side pane.
Include:	UC-1.2
Normal Flow:	<ol style="list-style-type: none"> 1. Notification settings will be displayed. 2. The user can then turn on the chat notifications.
Alternative Flows:	None.
Exceptions:	If internet connection is lost, the application will try to re-establish its connection. If it fails to do so within 5 seconds, the user will be redirected to the last menu that was opened.
Post condition:	The user will be notified when someone messages them.
Business Rules:	User should have stable internet connection.
Notes & Issues:	None.
Assumptions:	<ol style="list-style-type: none"> 1. User will have a stable internet connection. 2. User will be able to find and select the option from the menu correctly. 3. User can understand English/Urdu language. 4. Chat Notifications are turned off.

4. Turn Off Chat Notifications

The tabular use case of push notification module is given below:

Table 80: Tabular Use Case of UC-7.4

Use Case ID:	UC-7.4
Use Case Name:	Turn Off Chat Notifications
Actors:	Renter/Proprietor
Description:	The users can turn off chat notifications on their application.
Priority:	Medium
Preconditions:	User must be logged in to their account and the chat notifications are turned on beforehand.
Trigger:	User clicks on the Notification Settings Button from side pane.
Include:	UC-1.2
Normal Flow:	<ol style="list-style-type: none"> 1. Notification settings will be displayed. 2. The user can then turn on the chat notifications.
Alternative Flows:	None.
Exceptions:	If internet connection is lost, the application will try to re-establish its connection. If it fails to do so within 5 seconds, the user will be redirected to the last menu that was opened.
Post condition:	The user will not be notified when someone messages them.
Business Rules:	User should have stable internet connection.
Notes & Issues:	None.
Assumptions:	<ol style="list-style-type: none"> 1. User will have a stable internet connection. 2. User will be able to find and select the option from the menu correctly. 3. User can understand English/Urdu language. 4. Chat Notifications are turned off.

5. Set Ringtone

The tabular use case of push notification module is given below:

Table 81: Tabular Use Case of UC-7.5

Use Case ID:	UC-7.5
Use Case Name:	Set Ringtone
Actors:	Renter/Proprietor
Description:	The users can set ringtone of notifications of their own choice.
Priority:	Low
Preconditions:	User must be logged in to their account and the notifications are turned on.
Trigger:	User clicks on the Notification Settings Button from side pane.
Include:	UC-1.2
Normal Flow:	<ol style="list-style-type: none"> 1. Notification settings will be displayed. 2. The user can then set a ringtone.
Alternative Flows:	None.
Exceptions:	If internet connection is lost, the application will try to re-establish its connection. If it fails to do so within 5 seconds, the user will be redirected to the last menu that was opened.
Post condition:	The user will be notified with the selected ringtone.
Business Rules:	User should have stable internet connection.
Notes & Issues:	None.
Assumptions:	<ol style="list-style-type: none"> 1. User will have a stable internet connection. 2. User will be able to find and select the option from the menu correctly. 3. User can understand English/Urdu language. 4. Chat Notifications are turned on.

6. Set Vibration

The tabular use case of push notification module is given below:

Table 82: Tabular Use Case of UC-7.6

Use Case ID:	UC-7.6
Use Case Name:	Set Vibration
Actors:	Renter/Proprietor
Description:	The use case will allow users to set vibration of notifications.
Priority:	Low
Preconditions:	User must be logged in to their account and the notifications are turned on.
Trigger:	User clicks on the Notification Settings Button from side pane.
Include:	UC-1.2
Normal Flow:	<ol style="list-style-type: none"> 1. Notification settings will be displayed. 2. The user can then set vibration.
Alternative Flows:	None.
Exceptions:	If internet connection is lost, the application will try to re-establish its connection. If it fails to do so within 5 seconds, the user will be redirected to the last menu that was opened.
Post condition:	The user will be notified with the selected intensity of vibration.
Business Rules:	User should have stable internet connection.
Notes & Issues:	None.
Assumptions:	<ol style="list-style-type: none"> 1. User will have a stable internet connection. 2. User will be able to find and select the option from the menu correctly. 3. User can understand English/Urdu language. 4. Chat Notifications are turned on.

➤ ***Module 8: Help and Support***

The tabular use cases of help and support are provided in the following:

1. Chat With Bot

The tabular use case of help and support module is given below:

Table 83: Tabular Use Case of UC-8.1

Use Case ID:	UC-8.1
Use Case Name:	Chat With Bot
Actors:	Renter/Proprietor
Description:	The goal is to enable the users to chat with bot to get the automated answers of the AI Bot.
Priority	High
Preconditions:	<ol style="list-style-type: none"> 1. User must be logged into the application. 2. The user can be a renter or a proprietor. 3. The user must select the FAQ option tab.
Trigger	When the user clicks the Bot at the lower section of the interface.
Include	None
Normal Flow:	<ol style="list-style-type: none"> 1. The user will open the web or mobile application 2. The user will sign into his/her account 3. After the home screen appears, move to the FAQ section. 4. The AI implemented bot will answer user's queries.
Alternative Flows:	If the user is not connected to the internet, then he/she cannot chat with bot.
Exceptions:	None
Post condition:	The user will be able to chat with the bot.
Business Rules:	None
Notes and Issues	The bot might not answer the query if it is not fed into automated answers.
Assumptions:	<ol style="list-style-type: none"> 1. User must have a stable internet connection. 2. The user must understand the English language. 3. The user's query exists in the answers fed into the bot.

2. Do Live Chat

The tabular use case of help and support module is given below:

Table 84: Tabular Use Case of UC-8.2

Use Case ID:	UC-8.2
Use Case Name:	Do Live Chat
Actors:	Renter/Proprietor
Description:	The goal is to enable the users to chat with the management team if it is live.
Priority	High
Preconditions:	<ul style="list-style-type: none"> 1. User must be logged into the application. 2. The user can be a renter or a proprietor. 3. The user selects the Live Chat box and types a message to be sent there.
Trigger	When the user clicks the Live Chat option.
Include	None
Normal Flow:	<ul style="list-style-type: none"> 1. The user will open the web or mobile application 2. The user will sign into his/her account 3. After the home screen appears, move to the FAQ section. 4. Then, the user will click on the live chat option to get answers.
Alternative Flows:	None
Exceptions:	If the user is not connected to the internet, then he/she cannot chat with the live chat user.
Post condition:	The user will get reply from the live chat as soon as possible.
Business Rules:	User should have a stable internet connection
Notes and Issues	None
Assumptions:	<ul style="list-style-type: none"> 1. User must have a stable internet connection. 2. The user must understand the English language. 3. The Live chat is online at that moment.

3. Contact Support Team

The tabular use case of help and support module is given below:

Table 85: Tabular Use Case of UC-8.3

Use Case ID:	UC-8.3
Use Case Name:	Contact Support Team
Actors:	Renter/Proprietor
Description:	The goal is to enable the users to chat with the support team by sending a message.
Priority	Low
Preconditions:	User must be logged into the application.
Trigger	When the user clicks the Contact Us option provided in the interface.
Include	None.
Normal Flow:	<ol style="list-style-type: none"> 1. The user will open the web or mobile application 2. The user will sign into his/her account 3. After the home screen appears, move to the FAQ section. 4. Then select Contact us option.
Alternative Flows:	The user can directly contact the support team through email if he/she knows the email address of the support team.
Exceptions:	<ol style="list-style-type: none"> 1. If the user is not connected to the internet, then: 2. The message will be failed to send. 3. The message might reload if the internet facility comes back. 4. The message is blank.
Post condition:	The message will be sent to the management team and the query will be answered within 3-4 hours.
Business Rules:	User should have a stable internet connection.
Notes and Issues	None
Assumptions:	<ol style="list-style-type: none"> 1. User must have a stable internet connection. 2. The user must understand the English language. 3. The user must be able to find the correct menu.

4. Send Feedback

The tabular use case of help and support module is given below:

Table 86: Tabular Use Case of UC-8.4

Use Case ID:	UC-8.4
Use Case Name:	Send Feedback
Actors:	Renter/Proprietor
Description:	The goal is to enable the users to send feedback to the management team about the app.
Priority	High
Preconditions:	<ol style="list-style-type: none"> 1. User must be logged into the application. 2. The user must select the feedback option available in the interface of the app.
Trigger	When the user wants to send the feedback.
Include	None.
Normal Flow:	<ol style="list-style-type: none"> 1. The user will open the web or mobile application 2. The user will sign into his/her account 3. After the home screen appears, move to the feedback section, then select the feedback option to add description. 4. The user can also rate the app while sending feedback.
Alternative Flows:	None
Exceptions:	<ol style="list-style-type: none"> 1. If the user is not connected to the internet, then: 2. The feedback will be failed to be sent. 3. The feedback might reload if the internet facility comes back. 4. The feedback is blank. 5. No rating star is selected.
Post condition:	The feedback would be sent to the management team which will help them make their application better and get notified about any bugs
Business Rules:	User should have a stable internet connection.
Notes and Issues	None
Assumptions:	<ol style="list-style-type: none"> 1. User must have a stable internet connection. 2. The user must understand the English language. 3. The user must be able to find the correct menu.

5. Change Bot's Language

The tabular use case of help and support module is given below:

Table 87: Tabular Use Case of UC-8.5

Use Case ID:	UC-8.5
Use Case Name:	Bot Language
Actors:	Renter/Proprietor
Description:	The goal is to enable the users to set a language for the bot which would be easy to understand.
Priority	Low
Preconditions:	<ul style="list-style-type: none"> 1. User must be logged into the application. 2. The user must go to the setting option to change the language.
Trigger	When the user clicks the change language button.
Include	None
Normal Flow:	<ul style="list-style-type: none"> 1. The user will open the web or mobile application 2. The user will sign into his/her account 3. After the home screen appears, move to the settings menu, then select the language option to select.
Alternative Flows:	None
Exceptions:	If the user is not connected to the internet, then bot's language will not change.
Post condition:	The feedback would be sent to the management team which will help them make their application better and get notified about any bugs
Business Rules:	None
Notes and Issues	None
Assumptions:	<ul style="list-style-type: none"> 1. The user must understand the English language. 2. The user must be able to find the correct menu.

6. View Feedback History

The tabular use case of help and support module is given below:

Table 88: Tabular Use Case of UC-8.7

Use Case ID:	UC-8.7
Use Case Name:	View Feedback History
Actors:	Renter/Proprietor
Description:	The user can view the history of the feedbacks provided.
Priority	Low
Preconditions:	<ol style="list-style-type: none"> 1. User must be logged into the application. 2. The user must go to the help and support option to view the feedback history.
Trigger	When the user clicks the view history button.
Include	UC-8.6
Normal Flow:	None
Alternative Flows:	None
Exceptions:	The user must be connected to the internet otherwise the page will fail to load and move to the previous menu.
Post condition:	The user will be able to view the history of the feedbacks given by the user.
Business Rules:	None
Notes and Issues	None.
Assumptions:	<ol style="list-style-type: none"> 1. The user must understand the English language. 2. The user must be able to find the correct menu.

3.4 Functional Requirements

The Functional Requirements of the system are listed below:

➤ ***Module 1 (Profile Management):***

1) Sign-Up

Table 89: Description of FR-1.1.1

Identifier	FR-1.1.1
Title	Get Email Address
Requirement	Renter and Proprietor will have to enter a credible email address.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to get the user registered for their account.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 90: Description of FR-1.1.2

Identifier	FR-1.1.2
Title	Get Contact Number
Requirement	Renter and Proprietor will have to enter a credible phone number.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to get the user registered for their account.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 91: Description of FR-1.1.3

Identifier	FR-1.1.3
Title	Get First Name
Requirement	Renter and Proprietor will have to enter their first name.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to get the user registered for their account.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 92: Description of FR-1.1.4

Identifier	FR-1.1.4
Title	Get Last Name
Requirement	Renter and Proprietor will have to enter their last name.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to get the user registered for their account.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 93: Description of FR-1.1.5

Identifier	FR-1.1.5
Title	Get Password
Requirement	Renter and Proprietor will have to enter a password.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to get the user registered for their account.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 94: Description of FR-1.1.6

Identifier	FR-1.1.6
Title	Confirm Password
Requirement	Renter and Proprietor will have to enter the password again for password confirmation.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to get the user registered for their account successfully.
Business Rule	N/A
Dependencies	FR-1.1.5
Priority	High

Table 95: Description of FR-1.1.7

Identifier	FR-1.1.7
Title	Get CNIC
Requirement	Renter and Proprietor will have to enter CNIC in a proper format.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to get the user registered for their account.
Business Rule	N/A
Dependencies	N/A
Priority	High

2) Log-In

Table 96: Description of FR-1.2.1

Identifier	FR-1.2.1
Title	Get Email Address
Requirement	Renter and Proprietor will have to enter a credible email that they provided while signing up.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to get the user successfully login to their account.
Business Rule	N/A
Dependencies	UC-1.1
Priority	High

Table 97: Description of FR-1.2.2

Identifier	FR-1.2.2
Title	Get Password
Requirement	Renter and Proprietor will have to enter a password that they provided while signing up.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to get the user successfully login to their account.
Business Rule	N/A
Dependencies	UC-1.1
Priority	High

Table 98: Description of FR-1.2.3

Identifier	FR-1.2.3
Title	Get Captcha Verification
Requirement	Renter and Proprietor will have to select appropriate images as asked for.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to verify the user before logging in to their account.
Business Rule	N/A
Dependencies	UC-1.1
Priority	High

3) Log out

Table 99: Description of FR-1.3.1

Identifier	FR-1.3.1
Title	Log out
Requirement	Renter and Proprietor can log out from their account.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to log out from the account.
Business Rule	N/A
Dependencies	UC-1.1, UC-1.2
Priority	Medium

4) Forget Password

Table 100: Description of FR-1.4.1

Identifier	FR-1.4.1
Title	Get Email
Requirement	Renter and Proprietor will have to enter their valid email whose password needs to be recovered.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to verify the account's existence by sending an OTP confirmation.
Business Rule	N/A
Dependencies	UC-1.1, FR-1.1.1
Priority	Medium

Table 101: Description of FR-1.4.2

Identifier	FR-1.4.2
Title	Get Phone Number
Requirement	Renter and Proprietor will have to enter their valid phone number whose password needs to be recovered.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to verify the account's existence by sending an OTP confirmation.
Business Rule	N/A
Dependencies	UC-1.1, FR-1.1.2
Priority	Medium

Table 102: Description of FR-1.4.3

Identifier	FR-1.4.3
Title	Send Verification Code
Requirement	The System will send a verification code at the provided email account or the contact number.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to send an OTP confirmation code at the provided credentials.
Business Rule	N/A
Dependencies	UC-1.1, FR-1.4.1, FR-1.4.2
Priority	Medium

Table 103: Description of FR-1.4.4

Identifier	FR-1.4.4
Title	Type Verification Code
Requirement	Renter and Proprietor will have to type a verification code sent to the provided email account or the contact number.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to verify the ownership of the account.
Business Rule	N/A
Dependencies	UC-1.1, FR-1.4.1, FR-1.4.2
Priority	Medium

Table 104: Description of FR-1.4.5

Identifier	FR-1.4.5
Title	New Password
Requirement	Renter and Proprietor will have to type a new password for the account.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to set a new password.
Business Rule	N/A
Dependencies	UC-1.1
Priority	Medium

Table 105: Description of FR-1.4.6

Identifier	FR-1.4.6
Title	Confirm Password
Requirement	Renter and Proprietor will have to type the password again to check if it matches the new password.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to set a new Password
Business Rule	N/A
Dependencies	UC-1.1
Priority	Medium

5) View Profile

Table 106: Description of FR-1.5.1

Identifier	FR-1.5.1
Title	View Profile
Requirement	Renter and Proprietor can view their user profile to check their credentials.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to view personal and account details which may be visible to others as well.
Business Rule	N/A
Dependencies	UC-1.2
Priority	Low

Table 107: Description of FR-1.5.2

Identifier	FR-1.5.2
Title	Share Profile
Requirement	Renter and Proprietor can share their user profile on other social media platforms.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to share account's redirect link for easy access to the account.
Business Rule	N/A
Dependencies	UC-1.2
Priority	Low

Table 108: Description of FR-1.5.3

Identifier	FR-1.5.3
Title	Copy Contact
Requirement	Renter and Proprietor can copy their provided contact numbers from the profile.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to copy provided contact number and use it.
Business Rule	N/A
Dependencies	UC-1.2
Priority	Low

Table 109: Description of FR-1.5.4

Identifier	FR-1.5.4
Title	Copy User ID
Requirement	Renter and Proprietor can directly copy their account's user ID from the profile.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to copy user ID and use it.
Business Rule	N/A
Dependencies	UC-1.2
Priority	Low

6) Edit Profile***Table 110: Description of FR-1.6.1***

Identifier	FR-1.6.1
Title	Change Email
Requirement	Renter and Proprietor will enter a new and valid email address
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to allow the user to update their profile.
Business Rule	N/A
Dependencies	UC-1.5
Priority	Low

Table 111: Description of FR-1.6.2

Identifier	FR-1.6.2
Title	Change Phone Number
Requirement	Renter and Proprietor will enter a new and valid phone number.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to allow the user to update their profile.
Business Rule	N/A
Dependencies	UC-1.5
Priority	Low

Table 112: Description of FR-1.6.3

Identifier	FR-1.6.3
Title	Change First Name
Requirement	Renter and Proprietor can change their first name.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to allow the user to update their profile.
Business Rule	N/A
Dependencies	UC-1.5
Priority	Low

Table 113: Description of FR-1.6.4

Identifier	FR-1.6.4
Title	Change Last Name
Requirement	Renter and Proprietor can change their last name.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to allow the user to update their profile.
Business Rule	N/A
Dependencies	UC-1.5
Priority	Low

Table 114: Description of FR-1.6.5

Identifier	FR-1.6.5
Title	Change Password
Requirement	Renter and Proprietor will have to enter a password.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to allow the user to update their profile.
Business Rule	N/A
Dependencies	UC-1.5
Priority	Low

Table 115: Description of FR-1.6.6

Identifier	FR-1.6.6
Title	Confirm Password
Requirement	Renter and Proprietor will have to enter the password again for password confirmation.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to allow the user to update their profile.
Business Rule	N/A
Dependencies	UC-1.5, FR-1.6.5
Priority	Medium

Table 116: Description of FR-1.6.7

Identifier	FR-1.6.7
Title	Send Verification Code
Requirement	The System will send a verification code at the provided email account or the contact number.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to send an OTP confirmation code at the provided credentials.
Business Rule	N/A
Dependencies	UC-1.5, FR-1.6.1, FR-1.6.2
Priority	Medium

Table 117: Description of FR-1.6.8

Identifier	FR-1.6.8
Title	Type Verification Code
Requirement	Renter and Proprietor will have to type a verification code sent to the provided email account or the contact number.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to verify the ownership of the account.
Business Rule	N/A
Dependencies	UC-1.5, FR-1.6.1, FR-1.6.2
Priority	Medium

7) View Profile Display

Table 118: Description of FR-1.7.1

Identifier	FR-1.7.1
Title	View Profile Display
Requirement	Renter and Proprietor can view their profile picture.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to view profile picture which is visible to others.
Business Rule	N/A
Dependencies	UC-1.8
Priority	Low

Table 119: Description of FR-1.7.2

Identifier	FR-1.7.2
Title	Zoom in
Requirement	Renter and Proprietor can zoom into their profile picture.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to view profile picture by zooming in.
Business Rule	N/A
Dependencies	UC-1.8
Priority	Low

Table 120: Description of FR-1.7.3

Identifier	FR-1.7.3
Title	Zoom out
Requirement	Renter and Proprietor can zoom out of their profile picture.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to view profile picture by zooming out.
Business Rule	N/A
Dependencies	UC-1.8
Priority	Low

8) Change Profile Display

Table 121: Description of FR-1.8.1

Identifier	FR-1.8.1
Title	Take Picture
Requirement	The system will open the camera of the device to capture picture for profile display.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to set up a new picture on user's profile.
Business Rule	N/A
Dependencies	UC-1.2, UC-1.7
Priority	Medium

Table 122: Description of FR-1.8.2

Identifier	FR-1.8.2
Title	Upload Picture
Requirement	Renter and Proprietor can upload an existing photo from the gallery for profile display.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to set up a new picture on user's profile
Business Rule	N/A
Dependencies	UC-1.2, UC-1.7
Priority	Medium

9) Delete Account

Table 123: Description of FR-1.9.1

Identifier	FR-1.9.1
Title	Confirm Account Deletion
Requirement	Renter and Proprietor will click on confirm message prompt button to confirm account deletion.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to delete the account.
Business Rule	N/A
Dependencies	UC-1.2
Priority	High

Table 124: Description of FR-1.9.2

Identifier	FR-1.9.2
Title	Select Reason for Account Deletion
Requirement	Renter and Proprietor will select the appropriate reason from a list of provided reasons for account deletion.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to delete the account.
Business Rule	N/A
Dependencies	UC-1.2
Priority	High

Table 125: Description of FR-1.9.3

Identifier	FR-1.9.3
Title	Write Reason for Account Deletion
Requirement	Renter and Proprietor will write an appropriate reason for account deletion.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to delete the account.
Business Rule	N/A
Dependencies	UC-1.2
Priority	Medium

10) Disable Account

Table 126: Description of FR-1.10.1

Identifier	FR-1.10.1
Title	Confirm Account Disablement
Requirement	Renter and Proprietor will click on confirm message prompt button to confirm account disablement.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to disable the account.
Business Rule	N/A
Dependencies	UC-1.2
Priority	High

Table 127: Description of FR-1.10.2

Identifier	FR-1.10.2
Title	Select Reason for Account Disablement
Requirement	Renter and Proprietor will select the appropriate reason from a list of provided reasons for account disablement.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to disable the account.
Business Rule	N/A
Dependencies	UC-1.2
Priority	High

Table 128: Description of FR-1.10.3

Identifier	FR-1.10.3
Title	Write Reason for Account Disablement
Requirement	Renter and Proprietor will write an appropriate reason for account disablement.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to disable the account.
Business Rule	N/A
Dependencies	UC-1.2
Priority	Medium

11) Generate User ID

Table 129: Description of FR-1.11.1

Identifier	FR-1.11.1
Title	Generate User ID
Requirement	The system will generate unique User ID on account creation.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to give user account unique identifier.
Business Rule	N/A
Dependencies	UC-1.1
Priority	High

12) Generate Captcha Images

Table 130: Description of FR-1.12.1

Identifier	FR-1.12.1
Title	Select Similar Images
Requirement	Renter and Proprietor have to select images based on the query asked.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to verify human activity on the application.
Business Rule	N/A
Dependencies	UC-1.1
Priority	High

Table 131: Description of FR-1.12.2

Identifier	FR-1.12.2
Title	Enter Answer
Requirement	Renter and Proprietor will enter their selection.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to verify human activity on the application.
Business Rule	N/A
Dependencies	UC-1.1
Priority	High

13) Generate Pop-up Ads

Table 132: Description of FR-1.13.1

Identifier	FR-1.13.1
Title	Generate Pop-up Ads
Requirement	The system will generate pop-up ads based on user activity.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to offer apps for free while still earning money from the advertisers.
Business Rule	Machine Learning
Dependencies	UC-1.1
Priority	High

➤ **Module 2 (User Interaction Chat box):**

1) View Message

Table 133: Description of FR-2.1.1

Identifier	FR-2.1.1
Title	View Message
Requirement	The user will be able to View a message.
Source	Idrees Ghazi (Developer 2)
Rationale	To View a message in the chat box.
Business Rule	N/A
Dependencies	UC-1.2
Priority	High

Table 134: Description of FR-2.1.2

Identifier	FR-2.1.2
Title	Select Message
Requirement	The user will be able to select a message.
Source	Idrees Ghazi (Developer 2)
Rationale	To copy a text, a message is selected before.
Business Rule	N/A
Dependencies	UC-1.2
Priority	Medium

Table 135: Description of FR-2.1.3

Identifier	FR-2.1.3
Title	Message Details
Requirement	The user will be able to see message details i.e., time sent, seen time, etc.
Source	Idrees Ghazi (Developer 2)
Rationale	To Check the details of the text message
Business Rule	N/A
Dependencies	UC-1.2
Priority	Medium

Table 136: Description of FR-2.1.4

Identifier	FR-2.1.4
Title	Copy Message
Requirement	The user will be able copy important messages.
Source	Idrees Ghazi (Developer 2)
Rationale	To copy a text from the chat.
Business Rule	No
Dependencies	UC-1.2
Priority	Medium

Table 137: Description of FR-2.1.5

Identifier	FR-2.1.5
Title	Paste Message
Requirement	The user will be able paste messages in the chat box or the search box.
Source	Idrees Ghazi (Developer 2)
Rationale	To paste a text.
Business Rule	N/A
Dependencies	UC-1.2
Priority	Medium

2) Send Message

Table 138: Description of FR-2.2.1

Identifier	FR-2.2.1
Title	Type a Message
Requirement	The user will be able type a message in the provided chat box other.
Source	Idrees Ghazi (Developer 2)
Rationale	To communicate with each other
Business Rule	N/A
Dependencies	UC-1.2
Priority	High

Table 139: Description of FR-2.2.2

Identifier	FR-2.2.2
Title	Send Message
Requirement	The user will be able send a message by clicking the button provided.
Source	Idrees Ghazi (Developer 2)
Rationale	To communicate with each other
Business Rule	N/A
Dependencies	UC-1.2
Priority	High

Table 140: Description of FR-2.2.3

Identifier	FR-2.2.3
Title	Attach Image
Requirement	The user will be attaching an image by clicking the attach image icon.
Source	Idrees Ghazi (Developer 2)
Rationale	To send images/media
Business Rule	N/A
Dependencies	UC-1.2N/A
Priority	Medium

Table 141: Description of FR-2.2.4

Identifier	FR-2.2.4
Title	Attach Document
Requirement	The user will be able send a document by clicking the document icon.
Source	Idrees Ghazi (Developer 2)
Rationale	To send different types of documents
Business Rule	N/A
Dependencies	UC-1.2
Priority	Medium

3) Mute Message

Table 142: Description of FR-2.3.1

Identifier	FR-2.3.1
Title	Mute Message
Requirement	The user will be able to mute someone and avoid incoming notifications from the muted user.
Source	Idrees Ghazi (Developer 2)
Rationale	To mute the user.
Business Rule	No
Dependencies	UC-1.2
Priority	High

4) Unmute Message

Table 143: Description of FR-2.4.1

Identifier	FR-2.4.1
Title	Unmute Message
Requirement	The user will be able to Unmute the muted user.
Source	Idrees Ghazi (Developer 2)
Rationale	To unmute the muted user.
Business Rule	N/A
Dependencies	UC-1.2
Priority	High

5) Pin Message

Table 144: Description of FR-2.5.1

Identifier	FR-2.5.1
Title	Pin Message
Requirement	The user will be able to pin important messages.
Source	Idrees Ghazi (Developer 2)
Rationale	To Pin a message.
Business Rule	N/A
Dependencies	UC-1.2
Priority	High

6) Make Voice Call

Table 145: Description of FR-2.6.1

Identifier	FR-2.6.1
Title	Contact Number
Requirement	The user will be able to make a call on the contact number of the other user.
Source	Idrees Ghazi (Developer 2)
Rationale	To communicate each other using a voice call feature.
Business Rule	N/A
Dependencies	UC-1.2
Priority	High

7) Redirect to WhatsApp

Table 146: Description of FR-2.7.1

Identifier	FR-2.7.1
Title	Redirect to WhatsApp
Requirement	The user will be able to get him/herself redirected to the WhatsApp application by clicking the WhatsApp icon beside the contact number.
Source	Idrees Ghazi (Developer 2)
Rationale	To communicate each other on WhatsApp.
Business Rule	N/A
Dependencies	UC-1.2
Priority	Medium

8) Search Text Message

Table 147: Description of FR-2.8.1

Identifier	FR-2.8.1
Title	Write Message.
Requirement	The user should be able to write a message to be search.
Source	Idrees Ghazi (Developer 2)
Rationale	To search a message in the chat.
Business Rule	N/A
Dependencies	UC-1.2
Priority	High

Table 148: Description of FR-2.8.2

Identifier	FR-2.8.2
Title	Enter Message.
Requirement	The user should be able to enter a message to be searched.
Source	Idrees Ghazi (Developer 2)
Rationale	To search a message in the chat.
Business Rule	N/A
Dependencies	UC-1.2
Priority	High

9) Search User

Table 149: Description of FR-2.9.1

Identifier	FR-2.9.1
Title	Write user id/name.
Requirement	The user should be able to write the user id or name of the contact to be searched.
Source	Idrees Ghazi (Developer 2)
Rationale	To search a user in chat contacts.
Business Rule	N/A
Dependencies	UC-1.2
Priority	High

Table 150: Description of FR-2.9.2

Identifier	FR-2.9.2
Title	Enter user id/name.
Requirement	The user should be able to enter the user id or name of the contact to be searched.
Source	Idrees Ghazi (Developer 2)
Rationale	To search a user in chat contacts.
Business Rule	N/A
Dependencies	UC-1.2
Priority	High

10) View Chat Media

Table 151: Description of FR-2.10.1

Identifier	FR-2.10.1
Title	View Media
Requirement	The user of the system should be able to view chat media by clicking on the View Media option.
Source	Idrees Ghazi (Developer 2)
Rationale	To view media content of the chat.
Business Rule	N/A
Dependencies	UC-1.2
Priority	Medium

Table 152: Description of FR-2.10.2

Identifier	FR-2.10.2
Title	Download Media
Requirement	The user of the system should be able to save media to the gallery of their device
Source	Idrees Ghazi (Developer 2)
Rationale	To save media content of the chat into the gallery.
Business Rule	No
Dependencies	UC-1.2
Priority	Medium

11) Clear Chat

Table 153: Description of FR-2.12.1

Identifier	FR-2.12.1
Title	Delete Chat
Requirement	The user will have to click the delete option to delete a chat
Source	Idrees Ghazi (Developer 2)
Rationale	To remove chat.
Business Rule	N/A
Dependencies	UC-1.2
Priority	Medium

Table 154: Description of FR-2.12.2

Identifier	FR-2.12.2
Title	Delete Message
Requirement	The user will be able to click the delete option to delete a message
Source	Idrees Ghazi (Developer 2)
Rationale	To remove messages.
Business Rule	N/A
Dependencies	UC-1.2
Priority	Medium

Table 155: Description of FR-2.12.3

Identifier	FR-2.12.3
Title	Confirm Deletion
Requirement	Before the deletion of the message, a confirmation option will pop up to check if the user really wants to delete the message so the user will be able to confirm deletion
Source	Idrees Ghazi (Developer 2)
Rationale	To get confirmation from the user.
Business Rule	N/A
Dependencies	UC-1.2
Priority	High

12) Unpin Message

Table 156: Description of FR-2.14.1

Identifier	FR-2.14.1
Title	Unpin Message
Requirement	The user will be able to unpin messages.
Source	Idrees Ghazi (Developer 2)
Rationale	To unpin message.
Business Rule	N/A
Dependencies	UC-1.2, UC-2.5
Priority	High

➤ **Module 3 (Rent Payment Portal):**

1) Pay Rent

Table 157: Description of FR-3.1.1

Identifier	FR-3.1.1
Title	Get Account Number
Requirement	Renter will have to enter an account number from which the payment is to be made.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to pay the due rent online safely.
Business Rule	<ul style="list-style-type: none"> • The account number must be valid according to the requirements of IBAN and it must be 17 digit. • The Payment method must be Visa, MasterCard, UnionPay and JazzCash.
Dependencies	UC-1.2
Priority	High

Table 158: Description of FR-3.1.2

Identifier	FR-3.1.2
Title	Get Account Holder Name
Requirement	Renter will have to enter the name of the card holder.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to pay the due rent online safely.
Business Rule	The Payment method must be Visa, MasterCard and UnionPay.
Dependencies	UC-1.2
Priority	High

Table 159: Description of FR-3.1.3

Identifier	FR-3.1.3
Title	Get Billing Address
Requirement	The Renter will have to enter appropriate address of the card holder.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to pay the due rent online safely.
Business Rule	The Payment method must be Visa, MasterCard and UnionPay.
Dependencies	UC-1.2
Priority	High

Table 160: Description of FR-3.1.4

Identifier	FR-3.1.4
Title	Get Country
Requirement	The Renter will have to enter the country of the card holder.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to pay the due rent.
Business Rule	The Payment method must be Visa, MasterCard and UnionPay.
Dependencies	UC-1.2
Priority	High

Table 161: Description of FR-3.1.5

Identifier	FR-3.1.5
Title	Get Expiry
Requirement	The Renter will have to enter the expiry date of the card.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to pay the due rent.
Business Rule	The Payment method must be Visa, MasterCard and UnionPay.
Dependencies	UC-1.2
Priority	High

Table 162: Description of FR-3.1.6

Identifier	FR-3.1.6
Title	Get CVV
Requirement	The Renter will have to enter the CVV 3 digit code of the card.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to pay the due rent.
Business Rule	The Payment method must be Visa, MasterCard and UnionPay.
Dependencies	UC-1.2
Priority	High

Table 163: Description of FR-3.1.7

Identifier	FR-3.1.7
Title	Get Username
Requirement	The Renter will have to enter their username of PayPal Account.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to pay the due rent.
Business Rule	The Payment method must be PayPal.
Dependencies	UC-1.2
Priority	High

Table 164: Description of FR-3.1.8

Identifier	FR-3.1.8
Title	Get Password
Requirement	The Renter will have to enter their password of PayPal Account.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to pay the due rent.
Business Rule	The Payment method must be PayPal.
Dependencies	UC-1.2
Priority	High

Table 165: Description of FR-3.1.9

Identifier	FR-3.1.9
Title	Get Phone Number
Requirement	The Renter will have to enter their phone number registered for JazzCash and EasyPaisa.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to pay the due rent.
Business Rule	The Payment method must be JazzCash or EasyPaisa.
Dependencies	UC-1.2
Priority	High

Table 166: Description of FR-3.1.10

Identifier	FR-3.1.10
Title	Get CNIC
Requirement	The Renter will have to enter their CNIC on which EasyPaisa account is registered.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to pay the due rent.
Business Rule	The Payment method must be EasyPaisa.
Dependencies	UC-1.2
Priority	High

Table 167: Description of FR-3.1.11

Identifier	FR-3.1.11
Title	Payment method
Requirement	The Renter will select a payment method from payment options.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to pay the due rent.
Business Rule	Choice is between Visa, MasterCard, PayPal, UnionPay, JazzCash and EasyPaisa.
Dependencies	UC-1.2
Priority	Medium

2) Generate Invoice

Table 168: Description of FR-3.2.1

Identifier	FR-3.2.1
Title	Generate Invoice
Requirement	The Renter can generate invoice of their payment history.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to generate payment record for a specific property.
Business Rule	N/A
Dependencies	UC-1.2
Priority	Medium

3) Print Invoice

Table 169: Description of FR-3.3.1

Identifier	FR-3.3.1
Title	Save as PDF
Requirement	The Renter can save the invoice document in pdf format.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to save a copy of invoice in hardcopy form.
Business Rule	A fully functional printer with proper connections with the computer is required.
Dependencies	UC-1.2
Priority	Low

Table 170: Description of FR-3.3.2

Identifier	FR-3.3.2
Title	Select Page Number
Requirement	The Renter can select the pages by their number to get them printed.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to select specific pages.
Business Rule	A fully functional printer with proper connections with the computer is required.
Dependencies	UC-1.2
Priority	Medium

Table 171: Description of FR-3.3.3

Identifier	FR-3.3.3
Title	Select Printer
Requirement	The Renter can select a printer from connected printers.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to select specific printer form a list of available printers.
Business Rule	A fully functional printer with proper connections with the computer is required.
Dependencies	UC-1.2
Priority	Medium

Table 172: Description of FR-3.3.4

Identifier	FR-3.3.4
Title	Zoom in
Requirement	Renter can zoom into the document preview.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to view document preview by zooming in.
Business Rule	N/A
Dependencies	UC-1.2
Priority	Low

Table 173: Description of FR-3.3.5

Identifier	FR-3.3.5
Title	Zoom out
Requirement	Renter can zoom out of the document preview.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to view document preview by zooming out.
Business Rule	N/A
Dependencies	UC-1.2
Priority	Low

Table 174: Description of FR-3.3.6

Identifier	FR-3.3.6
Title	Select Page Orientation
Requirement	Renter can change page orientation of the document to be printed.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to set page orientation.
Business Rule	N/A
Dependencies	UC-1.2
Priority	Low

Table 175: Description of FR-3.3.7

Identifier	FR-3.3.7
Title	Select Page Size
Requirement	Renter can change page size of the document to be printed.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to set page size.
Business Rule	N/A
Dependencies	UC-1.2
Priority	Low

Table 176: Description of FR-3.3.8

Identifier	FR-3.3.8
Title	Select Color of Printing
Requirement	Renter can choose to print document in black & white or RGB.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to set color of printed document.
Business Rule	A fully functional printer that can print in black & white and RGB is required.
Dependencies	UC-1.2
Priority	Low

Table 177: Description of FR-3.3.9

Identifier	FR-3.3.9
Title	Select Number of Copies
Requirement	Renter can set number of copies of the document to be printed.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to select number of copies of printed document.
Business Rule	N/A
Dependencies	UC-1.2
Priority	Low

4) View Payment History

Table 178: Description of FR-3.4.1

Identifier	FR-3.4.1
Title	View Payment History
Requirement	Renter and Proprietor can view their payment history of received and paid cash.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to view record of payments.
Business Rule	N/A
Dependencies	UC-1.2
Priority	Low

5) Open Payment Details

Table 179: Description of FR-3.5.1

Identifier	FR-3.5.1
Title	View Payment Details
Requirement	Renter and Proprietor can view the advanced details of a specific payment done.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to view additional details of payments.
Business Rule	N/A
Dependencies	UC-1.2, FR-3.4.1
Priority	Low

6) Save Card Details

Table 180: Description of FR-3.6.1

Identifier	FR-3.6.1
Title	Save Card Details
Requirement	Renter/Proprietor can save their payment procedure details while doing payment.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to save payment details for future use.
Business Rule	N/A
Dependencies	UC-1.2
Priority	Low

7) Request for Payment Deadline Extension

Table 181: Description of FR-3.7.1

Identifier	FR-3.7.1
Title	Get Date
Requirement	The Renter can select a date to request for rent payment deadline extension.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to request the landlord for payment deadline extension.
Business Rule	N/A
Dependencies	UC-1.2
Priority	Medium

Table 182: Description of FR-3.7.2

Identifier	FR-3.7.2
Title	Get Request Description
Requirement	The Renter can add a request description about the reason for requesting rent payment deadline extension.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to request the landlord for payment deadline extension.
Business Rule	N/A
Dependencies	UC-1.2
Priority	Medium

Table 183: Description of FR-3.7.3

Identifier	FR-3.7.3
Title	Send Request
Requirement	The Renter will send the request for payment deadline extension to the proprietor.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to request the landlord for payment deadline extension.
Business Rule	N/A
Dependencies	UC-1.2
Priority	Medium

8) Respond to Payment Deadline Extension

Table 184: Description of FR-3.8.1

Identifier	FR-3.8.1
Title	Accept Request
Requirement	The Proprietor can accept the request of the renter regarding payment deadline extension.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to respond the renter for payment deadline extension.
Business Rule	N/A
Dependencies	UC-1.2, UC-3.7
Priority	Medium

Table 185: Description of FR-3.8.2

Identifier	FR-3.8.2
Title	Decline Request
Requirement	The Proprietor can decline the request of the renter regarding payment deadline extension.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to respond the renter for payment deadline extension.
Business Rule	N/A
Dependencies	UC-1.2, UC-3.7
Priority	Medium

9) View Payment Status

Table 186: Description of FR-3.9.1

Identifier	FR-3.9.1
Title	View Payment Status
Requirement	Renter and Proprietor can view the rent payment status whether paid or not.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to keep check of rent payments.
Business Rule	N/A
Dependencies	UC-1.2
Priority	Low

10) Edit Payment Details

Table 187: Description of FR-3.10.1

Identifier	FR-3.10.1
Title	Change Account Number
Requirement	Proprietor will have to enter a new and valid account number to which payment must be made.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to receive the due rent online safely.
Business Rule	<ul style="list-style-type: none"> • The account number must be valid according to the requirements of IBAN and it must be 17 digit. • The Payment method must be Visa, MasterCard, UnionPay and JazzCash.
Dependencies	UC-1.2, UC-3.1
Priority	Medium

Table 188: Description of FR-3.10.2

Identifier	FR-3.10.2
Title	Change Phone Number
Requirement	The Proprietor will have to enter a new and valid phone number registered for JazzCash and EasyPaisa.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to receive the due rent online safely.
Business Rule	The Payment method must be JazzCash or EasyPaisa.
Dependencies	UC-1.2
Priority	Medium

Table 189 Description of FR-3.10.3

Identifier	FR-3.10.3
Title	Change CNIC
Requirement	The Renter will have to enter a new and valid CNIC on which EasyPaisa account is registered.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to receive the due rent online safely.
Business Rule	The Payment method must be EasyPaisa.
Dependencies	UC-1.2
Priority	Medium

Table 102: Description of FR-3.10.4

Identifier	FR-3.10.4
Title	Change Encrypted PayPal ID
Requirement	The Renter will have to enter a new and valid Encrypted PayPal ID.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to receive the due rent online safely.
Business Rule	The Payment method must be PayPal.
Dependencies	UC-1.2
Priority	Medium

11) Buy Premium Account

Table 190: Description of FR-3.11.1

Identifier	FR-3.11.1
Title	Get Account Number
Requirement	Renter and Proprietor will have to enter an account number from which the payment is to be made.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to buy premium account.
Business Rule	<ul style="list-style-type: none"> • The account number must be valid according to the requirements of IBAN and it must be 17 digit. • The Payment method must be Visa, MasterCard, UnionPay and JazzCash.
Dependencies	UC-1.2
Priority	Medium

Table 191: Description of FR-3.11.2

Identifier	FR-3.11.2
Title	Get Account Holder Name
Requirement	Renter and Proprietor will have to enter the name of the card holder.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to buy premium account.
Business Rule	The Payment method must be Visa, MasterCard and UnionPay.
Dependencies	UC-1.2
Priority	Medium

Table 192: Description of FR-3.11.3

Identifier	FR-3.11.3
Title	Get Billing Address
Requirement	Renter and Proprietor will have to enter appropriate address of the card holder.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to buy premium account.
Business Rule	The Payment method must be Visa, MasterCard and UnionPay.
Dependencies	UC-1.2
Priority	Medium

Table 193: Description of FR-3.11.4

Identifier	FR-3.11.4
Title	Get Country
Requirement	Renter and Proprietor will have to enter the country of the card holder.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to buy premium account.
Business Rule	The Payment method must be Visa, MasterCard and UnionPay.
Dependencies	UC-1.2
Priority	Medium

Table 194: Description of FR-3.11.5

Identifier	FR-3.11.5
Title	Get Expiry
Requirement	Renter and Proprietor will have to enter the expiry date of the card.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to buy premium account.
Business Rule	The Payment method must be Visa, MasterCard and UnionPay.
Dependencies	UC-1.2
Priority	Medium

Table 195: Description of FR-3.11.6

Identifier	FR-3.11.6
Title	Get CVV
Requirement	Renter and Proprietor will have to enter the CVV 3 digit code of the card.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to buy premium account.
Business Rule	The Payment method must be Visa, MasterCard and UnionPay.
Dependencies	UC-1.2
Priority	Medium

Table 196: Description of FR-3.11.7

Identifier	FR-3.11.7
Title	Get Username
Requirement	Renter and Proprietor will have to enter their username of PayPal Account.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to buy premium account.
Business Rule	The Payment method must be PayPal.
Dependencies	UC-1.2
Priority	Medium

Table 197: Description of FR-3.11.8

Identifier	FR-3.11.8
Title	Get Password
Requirement	Renter and Proprietor will have to enter their password of PayPal Account.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to buy premium account.
Business Rule	The Payment method must be PayPal.
Dependencies	UC-1.2
Priority	Medium

Table 198: Description of FR-3.11.9

Identifier	FR-3.11.9
Title	Get Phone Number
Requirement	Renter and Proprietor will have to enter their phone number registered for JazzCash and EasyPaisa.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to buy premium account.
Business Rule	The Payment method must be JazzCash or EasyPaisa.
Dependencies	UC-1.2
Priority	Medium

Table 199: Description of FR-3.11.10

Identifier	FR-3.11.10
Title	Get CNIC
Requirement	Renter and Proprietor will have to enter their CNIC on which EasyPaisa account is registered.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to buy premium account.
Business Rule	The Payment method must be EasyPaisa.
Dependencies	UC-1.2
Priority	Medium

Table 200: Description of FR-3.11.11

Identifier	FR-3.11.11
Title	Payment method
Requirement	Renter and Proprietor will select a payment method from payment options.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to buy premium account.
Business Rule	Choice is between Visa, MasterCard, PayPal, UnionPay, JazzCash and EasyPaisa.
Dependencies	UC-1.2
Priority	Medium

12) Hide Pop-up Ads*Table 201: Description of FR-3.12.1*

Identifier	FR-3.12.1
Title	Hide Pop-up Ads
Requirement	The system will remove pop-up ads from the application after purchasing premium account.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to provide smooth app experience.
Business Rule	Premium Account
Dependencies	UC-3.11
Priority	High

➤ ***Module 4 (Complaint Management Portal):***

1) Create Complaint

Table 202: Description of FR-4.1.1

Identifier	FR-4.1.1
Title	Write Complaint
Requirement	The user will write the complaint issue to the landlord in a text box provided
Source	Idrees Ghazi (Developer 2)
Rationale	To report an issue to be resolved by the landlord.
Business Rule	N/A
Dependencies	UC-1.2
Priority	High

Table 203: Description of FR-4.1.2

Identifier	FR-4.1.2
Title	Select Type
Requirement	The user can select the type of the complaint to be registered
Source	Idrees Ghazi (Developer 2)
Rationale	To report an issue to be resolved by the landlord.
Business Rule	N/A
Dependencies	UC-1.2
Priority	High

Table 204: Description of FR-4.1.3

Identifier	FR-4.1.3
Title	Send Complaint
Requirement	The user can send the complaint written by selecting the Send option available.
Source	Idrees Ghazi (Developer 2)
Rationale	To report an issue to be resolved by the landlord.
Business Rule	N/A
Dependencies	UC-1.2
Priority	High

2) View Complaint Status

Table 205: Description of FR-4.2.1

Identifier	FR-4.2.1
Title	View Complaint Status
Requirement	The landlord and renter can view the status of the complaints registered.
Source	Idrees Ghazi (Developer 2)
Rationale	To view the status of the complaint.
Business Rule	N/A
Dependencies	UC-1.2
Priority	High

3) Delete Complaint

Table 206: Description of FR-4.3.1

Identifier	FR-4.3.1
Title	Confirm Deletion
Requirement	The user will have to confirm the deletion of complaint.
Source	Idrees Ghazi (Developer 2)
Rationale	To confirm complaint deletion.
Business Rule	N/A
Dependencies	UC-1.2
Priority	High

4) Respond to Complaint

Table 207: Description of FR-4.4.1

Identifier	FR-4.4.1
Title	View Complaint
Requirement	The landlord can view the complaint registered by the renters.
Source	Idrees Ghazi (Developer 2)
Rationale	To respond to the complaints of the renters.
Business Rule	N/A
Dependencies	UC-1.2
Priority	High

Table 208: Description of FR-4.4.2

Identifier	FR-4.4.2
Title	Mark as in-progress
Requirement	The landlord can mark the complaint as in-progress and start resolving it.
Source	Idrees Ghazi (Developer 2)
Rationale	To respond to the complaints of the renters.
Business Rule	N/A
Dependencies	UC-1.2
Priority	High

Table 209: Description of FR-4.4.3

Identifier	FR-4.4.3
Title	Mark as Completed
Requirement	The landlord can mark the complaint as completed after resolving it.
Source	Idrees Ghazi (Developer 2)
Rationale	To respond to the complaints of the renters.
Business Rule	N/A
Dependencies	UC-1.2
Priority	High

Table 210: Description of FR-4.4.4

Identifier	FR-4.4.4
Title	Decline
Requirement	The landlord can decline the complaint if the reason that renter provided is invalid.
Source	Idrees Ghazi (Developer 2)
Rationale	To respond to the complaints of the renters.
Business Rule	N/A
Dependencies	UC-1.2
Priority	High

Table 211: Description of FR-4.4.5

Identifier	FR-4.4.5
Title	Write Reason
Requirement	The Landlord can write the reason for the declination of the complaint from the Renter
Source	Idrees Ghazi (Developer 2)
Rationale	To provide a reason of declining the complaint of renter
Business Rule	N/A
Dependencies	UC-1.2
Priority	Medium

Table 212: Description of FR-4.4.6

Identifier	FR-4.4.6
Title	Send Reason
Requirement	The Landlord can send the reason for the declination of the complaint from the Renter
Source	Idrees Ghazi (Developer 2)
Rationale	To provide a reason of declining the complaint of renter
Business Rule	N/A
Dependencies	UC-1.2
Priority	Medium

5) Edit Complaint Description

Table 213: Description of FR-4.5.1

Identifier	FR-4.5.1
Title	Write Complaint
Requirement	The Renter can edit the Reason of the Complaint he provided before.
Source	Idrees Ghazi (Developer 2)
Rationale	To send the description of complaint
Business Rule	N/A
Dependencies	UC-4.1
Priority	Medium

Table 214: Description of FR-4.5.2

Identifier	FR-4.5.2
Title	Change Type
Requirement	The Renter can change the type of complaint from the selection box provided.
Source	Idrees Ghazi (Developer 2)
Rationale	To send the description of complaint
Business Rule	N/A
Dependencies	UC-1.2, UC-4.1
Priority	Medium

Table 215: Description of FR-4.5.3

Identifier	FR-4.5.3
Title	Confirm Edit
Requirement	The Renter will be able to confirm the changes made.
Source	Idrees Ghazi (Developer 2)
Rationale	To send the description of complaint
Business Rule	N/A
Dependencies	UC-1.2, UC-4.1
Priority	Medium

➤ ***Module 5 (Tenant Management Portal):***

1) View Property

Table 216: Description of FR-5.1.1

Identifier	FR-5.1.1
Title	View Property
Requirement	The Landlords can view their properties added.
Source	Idrees Ghazi (Developer 2)
Rationale	To view a property.
Business Rule	N/A
Dependencies	UC-1.2, UC-5.10
Priority	High

Table 217: Description of FR-5.1.2

Identifier	FR-5.1.2
Title	Copy Property id
Requirement	The Landlords can copy the property ids of their properties
Source	Idrees Ghazi (Developer 2)
Rationale	To copy a property id.
Business Rule	N/A
Dependencies	UC-1.2, UC-5.10
Priority	Medium

2) View Renters

Table 218: Description of FR-5.2.1

Identifier	FR-5.2.1
Title	View Renters
Requirement	The Landlords can view their renters.
Source	Idrees Ghazi (Developer 2)
Rationale	To view an added renter
Business Rule	N/A
Dependencies	UC-1.2, UC-5.4
Priority	Medium

Table 219: Description of FR-5.2.2

Identifier	FR-5.2.2
Title	Select Renters
Requirement	The Landlord can select a specific renter for different purposes.
Source	Idrees Ghazi (Developer 2)
Rationale	To select an added renter
Business Rule	N/A
Dependencies	UC-1.2, UC-5.4
Priority	Low

Table 220: Description of FR-5.2.3

Identifier	FR-5.2.3
Title	Download Renter List
Requirement	The Landlord can download the renter list and can have it saved in an MS Excel file.
Source	Idrees Ghazi (Developer 2)
Rationale	To save renters list.
Business Rule	N/A
Dependencies	UC-1.2, UC-5.4
Priority	Low

3) Add New Renter

Table 221: Description of FR-5.4.1

Identifier	FR-5.4.1
Title	User ID
Requirement	The landlord can enter a new tenant by adding a new user id.
Source	Idrees Ghazi (Developer 2)
Rationale	To add a new renter of the property.
Business Rule	N/A
Dependencies	UC-1.2
Priority	High

4) Remove Renter

Table 222: Description of FR-5.5.1

Identifier	FR-5.5.1
Title	Confirm Removal
Requirement	The landlord will be able confirm the removal of a specific renter on demand.
Source	Idrees Ghazi (Developer 2)
Rationale	To confirm removing a renter.
Business Rule	N/A
Dependencies	UC-1.2
Priority	High

5) View Lease

Table 223: Description of FR-5.6.1

Identifier	FR-5.6.1
Title	View Lease
Requirement	The landlord will be able to view the lease contract
Source	Idrees Ghazi (Developer 2)
Rationale	To view lease contract.
Business Rule	N/A
Dependencies	UC-1.2, UC-5.4
Priority	High

Table 224: Description of FR-5.6.2

Identifier	FR-5.6.2
Title	Download Lease
Requirement	The landlord will be able to download the lease contract
Source	Idrees Ghazi (Developer 2)
Rationale	To download lease contract.
Business Rule	N/A
Dependencies	UC-1.2, UC-5.4
Priority	High

6) Print Lease

Table 225: Description of FR-5.7.1

Identifier	FR-5.7.1
Title	Save as PDF
Requirement	The user can save the contract in the pdf format
Source	Idrees Ghazi (Developer 2)
Rationale	To save a copy of lease contract.
Business Rule	N/A
Dependencies	UC-1.2, UC-5.4
Priority	High

Table 226 Description of FR-5.7.2

Identifier	FR-5.7.2
Title	Select Page Number
Requirement	The user can select the number of pages to be printed.
Source	Idrees Ghazi (Developer 2)
Rationale	To select specific pages to be printed.
Business Rule	N/A
Dependencies	UC-1.2, UC-5.4
Priority	Medium

Table 227: Description of FR-5.7.3

Identifier	FR-5.7.3
Title	Select Printer
Requirement	The user can select a printer using the select printer tab.
Source	Idrees Ghazi (Developer 2)
Rationale	To select specific printer options.
Business Rule	N/A
Dependencies	UC-1.2, UC-5.4
Priority	High

Table 228: Description of FR-5.7.4

Identifier	FR-5.7.4
Title	Select Printer
Requirement	The user can select a printer using the select printer tab.
Source	Idrees Ghazi (Developer 2)
Rationale	To select specific printer options.
Business Rule	N/A
Dependencies	UC-1.2, UC-5.4
Priority	Medium

Table 229: Description of FR-5.7.5

Identifier	FR-5.7.5
Title	Zoom In
Requirement	The user will be able to zoom into the lease contract.
Source	Idrees Ghazi (Developer 2)
Rationale	To Zoom In.
Business Rule	N/A
Dependencies	UC-1.2, UC-5.4
Priority	Medium

Table 230: Description of FR-5.7.6

Identifier	FR-5.7.6
Title	Zoom Out
Requirement	The user will be able to zoom out of the lease contract.
Source	Idrees Ghazi (Developer 2)
Rationale	To Zoom out.
Business Rule	N/A
Dependencies	UC-1.2, UC-5.4
Priority	Medium

Table 231: Description of FR-5.7.7

Identifier	FR-5.7.7
Title	Set Page Size
Requirement	The user will be able to set page size of the lease contract.
Source	Idrees Ghazi (Developer 2)
Rationale	To set page size of the lease contract.
Business Rule	N/A
Dependencies	UC-1.2, UC-5.4
Priority	Medium

Table 232: Description of FR-5.7.8

Identifier	FR-5.7.8
Title	Set Color
Requirement	The user will be able to set color of the lease contract.
Source	Idrees Ghazi (Developer 2)
Rationale	To set color of the lease contract.
Business Rule	N/A
Dependencies	UC-1.2, UC-5.4
Priority	Medium

Table 233: Description of FR-5.7.9

Identifier	FR-5.7.9
Title	Number of copies
Requirement	The user will be able to select the number of copies to be printed of the lease contract.
Source	Idrees Ghazi (Developer 2)
Rationale	To set number of copies to be printed of the lease contract.
Business Rule	N/A
Dependencies	UC-1.2, UC-5.4
Priority	High

7) Renew Lease

Table 234: Description of FR-5.8.1

Identifier	FR-5.8.1
Title	Lease Description
Requirement	The user will be able to provide the new lease description.
Source	Idrees Ghazi (Developer 2)
Rationale	To renew the lease contract.
Business Rule	N/A
Dependencies	UC-1.2, UC-5.4
Priority	Medium

Table 235: Description of FR-5.8.2

Identifier	FR-5.8.2
Title	Request Lease confirmation
Requirement	The landlord will be able to send request for lease renewal.
Source	Idrees Ghazi (Developer 2)
Rationale	To renew the lease contract.
Business Rule	N/A
Dependencies	UC-1.2, UC-5.4
Priority	Medium

8) Delete Property

Table 236: Description of FR-5.9.1

Identifier	FR-5.9.1
Title	Confirm Deletion
Requirement	The landlord will be able to confirm the deletion of the property.
Source	Idrees Ghazi (Developer 2)
Rationale	To confirm property deletion.
Business Rule	N/A
Dependencies	UC-1.2, UC-5.10
Priority	High

9) Add Property

Table 237: Description of FR-5.10.1

Identifier	FR-5.10.1
Title	Add Picture
Requirement	The landlord will be able to add a picture of the property.
Source	Idrees Ghazi (Developer 2)
Rationale	To add a new property.
Business Rule	N/A
Dependencies	UC-1.2
Priority	High

Table 238: Description of FR-5.10.2

Identifier	FR-5.10.2
Title	Add Property Details
Requirement	The landlord will be able to add details of the new property.
Source	Idrees Ghazi (Developer 2)
Rationale	To add a new property.
Business Rule	N/A
Dependencies	UC-1.2
Priority	High

Table 239: Description of FR-5.10.3

Identifier	FR-5.10.3
Title	Number of Renters
Requirement	The landlord will be able to add the number of renters who rented the property.
Source	Idrees Ghazi (Developer 2)
Rationale	To specify renters of the new property.
Business Rule	N/A
Dependencies	UC-1.2
Priority	High

Table 240: Description of FR-5.10.4

Identifier	FR-5.10.4
Title	Add Account Number
Requirement	The Landlord will be able to add an account number
Source	Idrees Ghazi (Developer 2)
Rationale	To add details with the new property.
Business Rule	N/A
Dependencies	UC-1.2
Priority	High

Table 241: Description of FR-5.10.5

Identifier	FR-5.10.5
Title	Add PayPal Id
Requirement	The Landlord will be able to add a PayPal ID where the rent is to be paid by the renter
Source	Idrees Ghazi (Developer 2)
Rationale	To add details with the new property.
Business Rule	N/A
Dependencies	UC-1.2
Priority	High

10) Update Rent

Table 242: Description of FR-5.11.1

Identifier	FR-5.11.1
Title	Enter Rent Amount
Requirement	The Landlord will be able to Enter the Rent Amount.
Source	Idrees Ghazi (Developer 2)
Rationale	To update rent.
Business Rule	N/A
Dependencies	UC-1.2, UC-5.4
Priority	Medium

Table 243: Description of FR-5.11.2

Identifier	FR-5.11.2
Title	Confirm Updation.
Requirement	The Landlord will be able to confirm the Updation of rent.
Source	Idrees Ghazi (Developer 2)
Rationale	To update rent.
Business Rule	N/A
Dependencies	UC-1.2, UC-5.4
Priority	Medium

11) Edit Property Details

Table 244: Description of FR-5.12.1

Identifier	FR-5.12.1
Title	Add New Picture
Requirement	The Landlord will be able to add a new picture for the property
Source	Idrees Ghazi (Developer 2)
Rationale	To Edit the Details of the property.
Business Rule	N/A
Dependencies	UC-1.2, UC-5.10
Priority	Medium

Table 245: Description of FR-5.12.2

Identifier	FR-5.12.2
Title	Edit Property Details
Requirement	The Landlord will be able to edit the description of the property.
Source	Idrees Ghazi (Developer 2)
Rationale	To Edit the Details of the property.
Business Rule	N/A
Dependencies	UC-1.2, UC-5.10
Priority	Medium

Table 246: Description of FR-5.12.3

Identifier	FR-5.12.3
Title	Number of Renters
Requirement	The Landlord will be able to edit the number of renters of the property.
Source	Idrees Ghazi (Developer 2)
Rationale	To Edit the Details of the property.
Business Rule	N/A
Dependencies	UC-1.2, UC-5.10
Priority	Medium

12) Go to Payment Portal

Table 247: Description of FR-5.13.1

Identifier	FR-5.13.1
Title	Go to Payment Portal
Requirement	The Landlord will be able to directly go to the payment portal from the tenant menu.
Source	Idrees Ghazi (Developer 2)
Rationale	To go to payment portal directly.
Business Rule	N/A
Dependencies	UC-1.2
Priority	Medium

13) Respond to Renter Connectivity Request

Table 248: Description of FR-5.14.1

Identifier	FR-5.14.1
Title	Accept
Requirement	The Landlord will be able to accept the request of connectivity from the renter.
Source	Idrees Ghazi (Developer 2)
Rationale	To accept the renter's request and get connected with the renter.
Business Rule	The proprietor must add a property before adding a renter
Dependencies	UC-1.2
Priority	High

Table 249: Description of FR-5.14.2

Identifier	FR-5.14.2
Title	Decline
Requirement	The Landlord will be able to Decline the request of connectivity from the renter.
Source	Idrees Ghazi (Developer 2)
Rationale	To decline the renter's request.
Business Rule	The proprietor must add a property before adding a renter
Dependencies	UC-1.2
Priority	High

➤ ***Module 6 (Property Management Portal):***

1) View Lease

Table 250: Description of FR-6.1.1

Identifier	FR-6.1.1
Title	View Lease
Requirement	The Renter will be able to view the lease contract.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to view lease contract.
Business Rule	N/A
Dependencies	UC-5.4
Priority	Medium

2) View Property Details

Table 251: Description of FR-6.2.1

Identifier	FR-6.2.1
Title	View Property Details
Requirement	The Renter can view the details of the properties they have rented.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to view rented properties' details.
Business Rule	N/A
Dependencies	UC-5.4
Priority	Medium

3) View Landlord's Details

Table 252: Description of FR-6.3.1

Identifier	FR-6.3.1
Title	View Landlord's Details
Requirement	The Renter can view the details of landlord of the properties they have rented.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to view rented properties' landlord's details.
Business Rule	N/A
Dependencies	UC-5.4, UC-5.10
Priority	Medium

4) Check Rent Tenure

Table 253: Description of FR-6.4.1

Identifier	FR-6.4.1
Title	Check Rent Tenure
Requirement	The Renter can view their rent tenure for each rented property.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to keep a check on the tenure spent at one property.
Business Rule	N/A
Dependencies	UC-5.4
Priority	Medium

5) Request Lease Change

Table 254: Description of FR-6.5.1

Identifier	FR-6.5.1
Title	Write Reason of Request
Requirement	The Renter must provide a credible reason for their request in the text box.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to request for lease change.
Business Rule	N/A
Dependencies	UC-5.14
Priority	High

Table 255: Description of FR-6.5.2

Identifier	FR-6.5.2
Title	Send Request
Requirement	The Renter will have to click the send button to send the request.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to request for lease change.
Business Rule	N/A
Dependencies	UC-5.14
Priority	High

6) Print Lease

Table 256: Description of FR-6.6.1

Identifier	FR-6.6.1
Title	Save as PDF
Requirement	The Renter can save the lease document in pdf format.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to save a copy of lease in hardcopy form.
Business Rule	A fully functional printer with proper connections with the computer is required.
Dependencies	UC-1.2
Priority	Low

Table 257: Description of FR-6.6.2

Identifier	FR-6.6.2
Title	Select Page Number
Requirement	The Renter can select the pages by their number to get them printed.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to select specific pages.
Business Rule	A fully functional printer with proper connections with the computer is required.
Dependencies	UC-1.2
Priority	Medium

Table 258: Description of FR-6.6.3

Identifier	FR-6.6.3
Title	Select Printer
Requirement	The Renter can select a printer from connected printers.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to select specific printer form a list of available printers.
Business Rule	A fully functional printer with proper connections with the computer is required.
Dependencies	UC-1.2
Priority	Medium

Table 259: Description of FR-6.6.4

Identifier	FR-6.6.4
Title	Zoom in
Requirement	Renter can zoom into the document preview.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to view document preview by zooming in.
Business Rule	N/A
Dependencies	UC-1.2
Priority	Low

Table 260: Description of FR-6.6.5

Identifier	FR-6.6.5
Title	Zoom out
Requirement	Renter can zoom out of the document preview.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to view document preview by zooming out.
Business Rule	N/A
Dependencies	UC-1.2
Priority	Low

Table 261: Description of FR-6.66

Identifier	FR-6.6.6
Title	Select Page Orientation
Requirement	Renter can change page orientation of the document to be printed.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to set page orientation.
Business Rule	N/A
Dependencies	UC-1.2
Priority	Low

Table 262: Description of FR-6.6.7

Identifier	FR-6.6.7
Title	Select Page Size
Requirement	Renter can change page size of the document to be printed.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to set page size.
Business Rule	N/A
Dependencies	UC-1.2
Priority	Low

Table 263: Description of FR-6.6.8

Identifier	FR-6.6.8
Title	Select Color of Printing
Requirement	Renter can choose to print document in black & white or RGB.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to set color of printed document.
Business Rule	A fully functional printer that can print in black & white and RGB is required.
Dependencies	UC-1.2
Priority	Low

Table 264: Description of FR-6.6.9

Identifier	FR-6.6.9
Title	Select Number of Copies
Requirement	Renter can set number of copies of the document to be printed.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to select number of copies of printed document.
Business Rule	N/A
Dependencies	UC-1.2
Priority	Low

7) Respond to Lease

Table 265: Description of FR-6.7.1

Identifier	FR-6.7.1
Title	Accept Request
Requirement	The Renter can accept lease renewal request from the landlord.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to finalize a lease contract.
Business Rule	N/A
Dependencies	UC-5.8
Priority	High

Table 266: Description of FR-6.7.2

Identifier	FR-6.7.2
Title	Decline Request
Requirement	The renter can decline lease renewal request from the landlord.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to finalize a lease contract.
Business Rule	N/A
Dependencies	UC-5.8
Priority	High

Table 267: Description of FR-6.7.3

Identifier	FR-6.7.3
Title	Write Reason
Requirement	The renter will write the reason for declining lease renewal request.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to finalize a lease contract.
Business Rule	N/A
Dependencies	UC-5.8
Priority	High

Table 268: Description of FR-6.7.4

Identifier	FR-6.7.4
Title	Send Reason
Requirement	The renter will send the reason for declining lease renewal request.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to finalize a lease contract.
Business Rule	N/A
Dependencies	UC-5.8
Priority	High

8) Connect with Landlord

Table 269: Description of FR-6.8.1

Identifier	FR-6.8.1
Title	Get Property ID
Requirement	The renter will enter a property id to connect with the landlord.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to connect renters with their respective landlords.
Business Rule	N/A
Dependencies	UC-1.2
Priority	High

Table 270: Description of FR-6.8.2

Identifier	FR-6.8.2
Title	Get User ID
Requirement	The renter will enter a user id to connect with the landlord.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to connect renters with their respective landlords.
Business Rule	N/A
Dependencies	UC-1.2
Priority	High

9) Go to Payment Portal

Table 271: Description of FR-6.9.1

Identifier	FR-6.9.1
Title	Go to Payment Portal
Requirement	The Renter will be able to directly go to the payment portal from the property menu.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to go to payment portal directly.
Business Rule	N/A
Dependencies	UC-1.2
Priority	Low

➤ **Module 7 (Push Notifications):**

1) Turn On Rent Reminders

Table 272: Description of FR-7.1.1

Identifier	FR-7.1.1
Title	Turn On Rent Reminders
Requirement	Renter and Proprietor can switch on rent reminders to get notified about pending payments.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to get reminders from application about pending dues.
Business Rule	N/A
Dependencies	UC-1.2, FR-7.2.1
Priority	Medium

2) Turn Off Rent Reminders

Table 273: Description of FR-7.2.1

Identifier	FR-7.2.1
Title	Turn Off Rent Reminders
Requirement	Renter and Proprietor can switch off rent reminders to get notified about pending payments.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to stop getting reminders from application about pending dues.
Business Rule	N/A
Dependencies	UC-1.2, FR-7.1.1
Priority	Medium

3) Turn On Chat Notifications

Table 274: Description of FR-7.3.1

Identifier	FR-7.3.1
Title	Turn On Chat Notifications
Requirement	Renter and Proprietor can switch on chat notifications to get notified about unread messages.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to get reminders from application about unread messages.
Business Rule	N/A
Dependencies	UC-1.2, FR-7.4.1
Priority	Medium

4) Turn Off Chat Notifications

Table 275: Description of FR-7.4.1

Identifier	FR-7.4.1
Title	Turn Off Chat Notifications
Requirement	Renter and Proprietor can switch off chat notifications to get notified about unread messages.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to stop getting reminders from application about unread messages.
Business Rule	N/A
Dependencies	UC-1.2, FR-7.3.1
Priority	Medium

5) Set Ringtone

Table 276: Description of FR-7.5.1

Identifier	FR-7.5.1
Title	Set Volume
Requirement	Renter and Proprietor can set ringtone volume.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to turn volume up or down.
Business Rule	N/A
Dependencies	UC-1.2
Priority	Low

Table 277: Description of FR-7.5.2

Identifier	FR-7.5.2
Title	Set Audio
Requirement	Renter and Proprietor can choose ringtone.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to set up a ringtone.
Business Rule	N/A
Dependencies	UC-1.2
Priority	Low

6) Set Vibration

Table 278: Description of FR-7.6.1

Identifier	FR-7.6.1
Title	Set Intensity
Requirement	Renter and Proprietor can set the intensity of vibration.
Source	Hunia Nadeem (Developer 1)
Rationale	The purpose is to set up intensity of vibration whenever a notification pops up.
Business Rule	N/A
Dependencies	UC-1.2, FR-7.1.1, FR-7.3.1
Priority	Low

➤ **Module 8 (Help and Support):**

1) Chat with Bot

Table 279: Description of FR-8.1.1

Identifier	FR-8.1.1
Title	Display Auto generated messages
Requirement	The user will be able to view auto generated messages by bot.
Source	Idrees Ghazi (Developer 2)
Rationale	To Answer user's query
Business Rule	N/A
Dependencies	UC-1.2
Priority	Medium

Table 280: Description of FR-8.1.2

Identifier	FR-8.1.2
Title	Display Reply
Requirement	The user will be able to view replies of the bot if the query does not exist in auto answered messages.
Source	Idrees Ghazi (Developer 2)
Rationale	To Answer user's query
Business Rule	N/A
Dependencies	UC-1.2
Priority	Medium

Table 281: Description of FR-8.1.3

Identifier	FR-8.1.3
Title	Write Question
Requirement	The user will be able to write a question who's answered is not provided by the application.
Source	Idrees Ghazi (Developer 2)
Rationale	To Write a Question separately.
Business Rule	N/A
Dependencies	UC-1.2
Priority	Medium

Table 282: Description of FR-8.1.4

Identifier	FR-8.1.4
Title	Enter Question
Requirement	The user will be able to enter a question after writing it in the box provided.
Source	Idrees Ghazi (Developer 2)
Rationale	To send the question.
Business Rule	N/A
Dependencies	UC-1.2
Priority	Medium

2) Do Live Chat

Table 283: Description of FR-8.2.1

Identifier	FR-8.2.1
Title	Enter Message
Requirement	The user will be able to type a message to the live chat if it is online.
Source	Idrees Ghazi (Developer 2)
Rationale	To Answer user's query
Business Rule	N/A
Dependencies	UC-1.2
Priority	Medium

3) Contact Support Team

Table 284: Description of FR-8.3.1

Identifier	FR-8.3.1
Title	Idrees Ghazi (Developer 2)
Requirement	The user will be able to type a message to the provided email to contact the support team.
Source	Renter/Proprietor
Rationale	To Answer user's query
Business Rule	N/A
Dependencies	UC-1.2
Priority	Medium

4) Send Feedback

Table 285: Description of FR-8.4.1

Identifier	FR-8.4.1
Title	Write Feedback
Requirement	The user will be able to write feedback in the text box provided
Source	Idrees Ghazi (Developer 2)
Rationale	To send feedback to the support team about the app
Business Rule	N/A
Dependencies	UC-1.2
Priority	Medium

Table 286: Description of FR-8.4.2

Identifier	FR-8.4.2
Title	Give Rating
Requirement	The user will be able to give rating out of 5.
Source	Idrees Ghazi (Developer 2)
Rationale	To send feedback
Business Rule	N/A
Dependencies	UC-1.2
Priority	Medium

5) Select Language

Table 287: Description of FR-8.5.1

Identifier	FR-8.5.1
Title	Select Language
Requirement	The user can select the any language for the bot including Urdu and English.
Source	Idrees Ghazi (Developer 2)
Rationale	To Select A Language
Business Rule	N/A
Dependencies	UC-1.2
Priority	Medium

6) View Feedback History

Table 288: Description of FR-8.7.1

Identifier	FR-8.7.1
Title	View Feedback History
Requirement	The user will be able to view the feedback history.
Source	Idrees Ghazi (Developer 2)
Rationale	To view feedback history.
Business Rule	N/A
Dependencies	UC-1.2, UC-8.1
Priority	Medium

3.5 Non-Functional Requirements

The non-functional requirements which would affect the quality of the system are described below:

3.5.1 Reliability

- **MTBF (Mean Time between Failures):**
The estimated mean time between failures will be 24 **hours ÷ 4 Errors**. So the estimated MTBF value is **6**.
- **Consequences of the Software Failure:**
The software can fail in multiple ways:
 - Interface Bug (i.e., Buttons not working correctly)
 - Payment Reminder Bug (i.e., Payment reminder did not receive on time)
 - Late Chat Reply Notification Bug.
 - App not responding bug.

The users can report the bug by contacting the management of the Software and each bug will be fixed as soon as possible to maintain the rating of the Application.

Even if the software fails due to an error, the user will not lose his/her data. Every detail of the user's data will be recovered because the backup system of software will always update the backup file after every change the user made in his/her profile. So in case the system fails, the backup program will execute automatically to recover all the data.

- **Protection from Failure:**

At the testing phase of the Software development, the software would be thoroughly checked before going on to the next phase. So, there would be a **10%** chance that an error could be faced by any user.

- **Strategy of Error Detection:**

The feedback feature of the system will allow the user to report bugs which will get the management notified about the errors and it would be fixed within **42 hours**.

- **Strategy for Correction:**

As the design methodology that will be used is Object Oriented, so even if the error shows up, it will be easier to detect the line of code where the bug is present. So, the average time to fix an error would be **3-4 hours**.

3.5.2 Usability

- A new user of the software will take only **20-30 minutes** to understand the use of the system.
- The software will be easy to use as the users will not have to add their credentials repeatedly because their credentials will be saved when the user adds them for the first time on choice.
- With a user-friendly interface, users can easily switch menus and understand it without any problem because every menu has a simple interface.
- The actions of the user would be confirmed using **OTP** which would be sent on user's email address or phone number before the app makes any changes on the user's contracts or saved information. This OTP will be received within **5 seconds**.
- The lost data of the user can be recovered through the backup system in less than **15 minutes**.

3.5.3 Performance

- **Response Time**

The system will have a minimum response time of **1.5 second** and maximum response time of **5 seconds**. After reaching 5 seconds limit, user will receive a prompt message and previous activity will be restored.

- **Throughput**

The application will work on minimum Internet speed of **4 mbps**.

- **Memory Utilization**

The average memory required by the application will be **50 MBs**.

- **Degradation Modes**

If the system crashes, it will work towards restoring the previous app activity. If it is not achieved within **5 seconds**, the app will restart.

3.5.4 Security

The system will ensure the security of data and information of users. The online payment process will be secured from any fraud. Proper authentication of users will be simultaneously performed as they try to access their accounts, taking less than **1 minute**.

4. Architecture and Design

This chapter thoroughly discusses about the system architecture and design of our Renter-Proprietor Coordination System.

4.1. System Architecture

The System has a **Model View Controller (MVC) architecture**. The system will have static pages or menus which will be viewed by the user. All the independent pages of the View will be connected to the Controller of the system which will manage the sequences of instructions added by the user. The Model of the system is the database which will be used to store data. The model will update anything on the view based on the queries. The Controller will take the query to the model in order to fetch data.

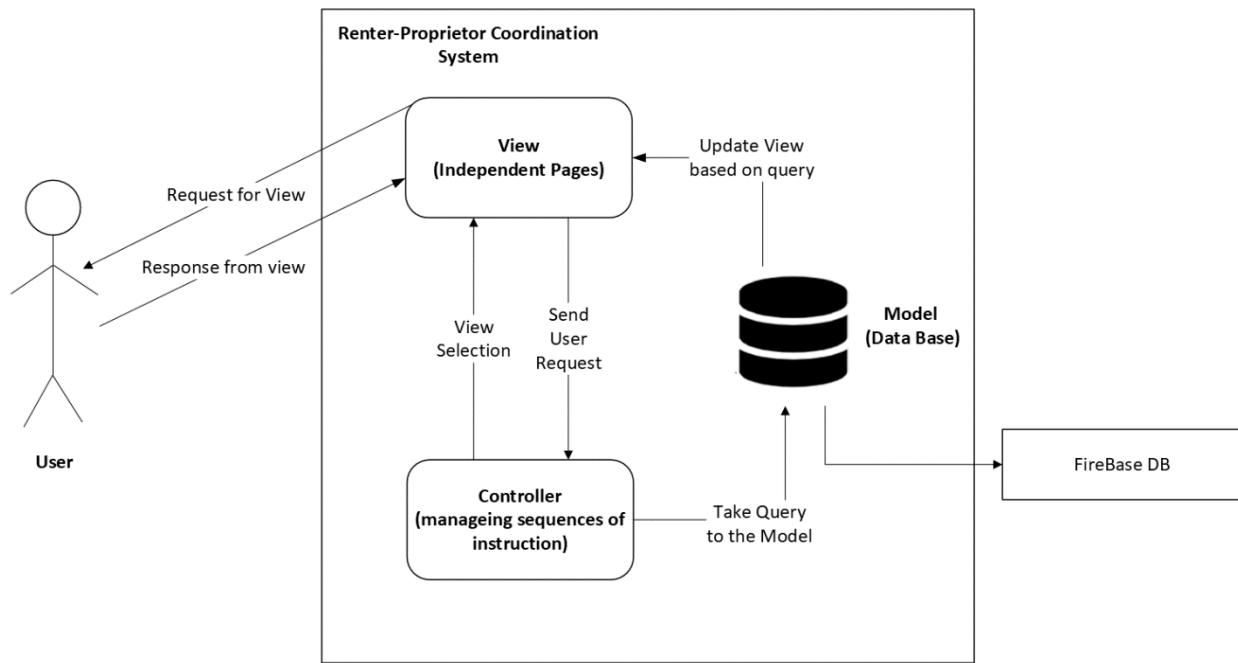


Figure 3: MVC Architecture Diagram

4.2. Design Methodology

The design methodology we will be using is **Object Oriented Approach** because it increases the reusability of the code, and it would be easier for the team members to work together without any confusion. Therefore, we are using Flutter, HTML, CSS languages which follows object-oriented approach. Also, it is easier to describe the code using UML diagrams. So, OOP is the best approach which fits our framework.

4.3. Data Representation

4.3.1. ERD

The Entity-Relationship Diagram of Renter Proprietor System is given below:

The main entities of the ERD are given below:

1. User
2. Renter
3. Proprietor
4. Lease
5. Chat
6. Live Chat
7. Notification Settings
8. Notification
9. Feedback
10. Payment
11. Complaint
12. Chatbot
13. Property
14. Rent

The Relations of the entities are given below:

1. User can either be a Renter or a Proprietor, Renter and Proprietor inherit User.
2. Many renters can be proprietors and many proprietors can be renters.
3. One renter can send many complaints and one proprietor can accept many complaints.
4. A user has only one notification setting.
5. One notification setting have only one notification
6. Many users can send many feedbacks, many feedbacks can have many users.
7. Renters and proprietors may have many lease contracts.
8. One user can have many chats.
9. One user can have many properties.
10. One property can have many rents
11. Many rents have one payment.
12. One user can pay bills using only one payment.
13. One user can use only one Chatbot.

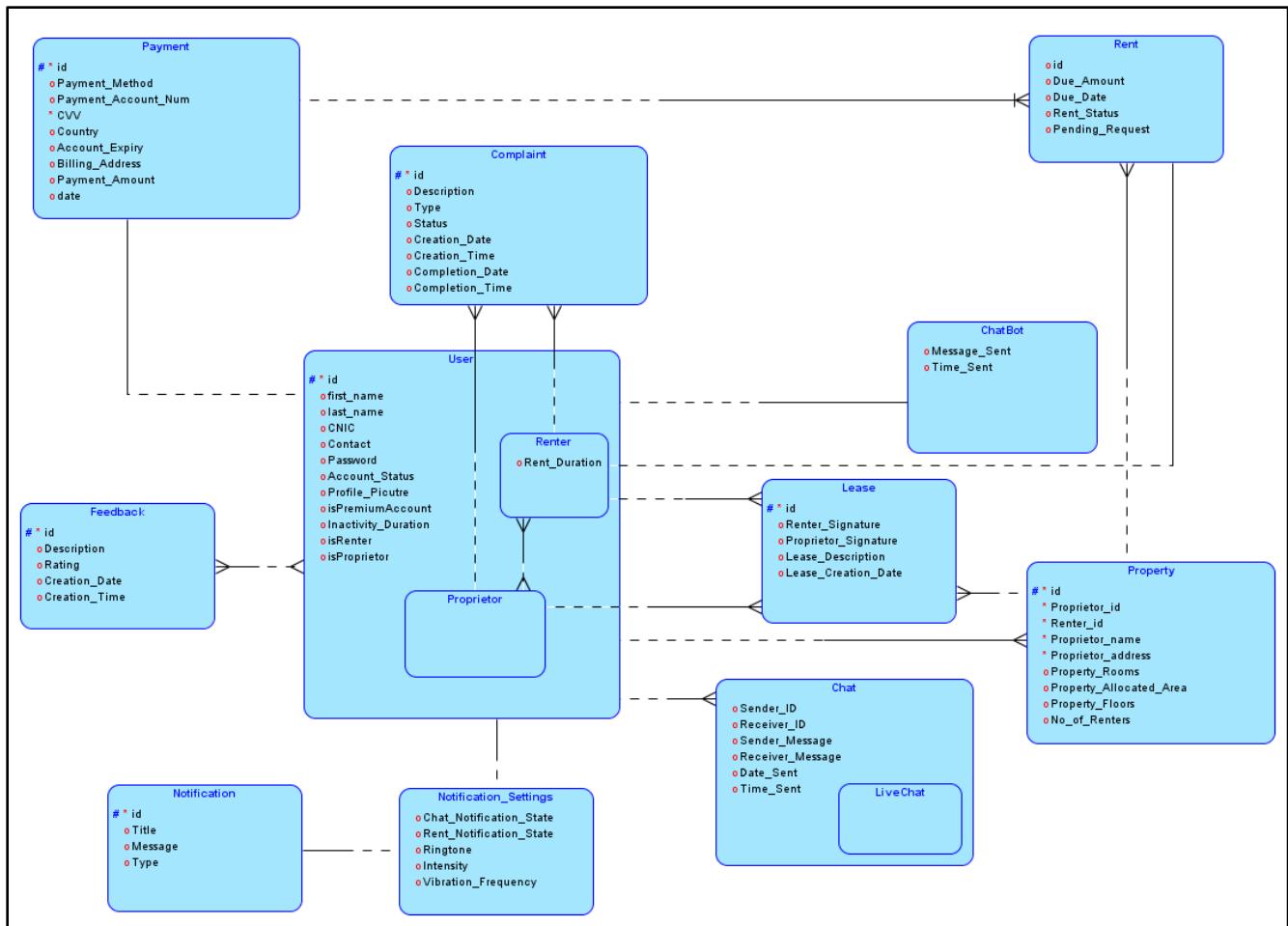


Figure 4: Entity Relation Diagram of RPC System

4.3.2. JSON Schema

Following is the JSON schema of RPC [Renter-Proprietor Coordination] System:

```
{
    "title": "Payment",
    "type": "object",
    "properties": {
        "Payment_ID": {
            "type": "String",
            "description": "It will contain the ID of payment"
        },
        ...
    }
}
```

```
“Payment_Method”: {  
    “type”: “String”,  
    “description”: “It will contain the method of online payment”.  
},  
  
“Payment_Account_Number”: {  
    “type”: “int”,  
    “description”: “It will contain account number”.  
},  
  
“CVV”: {  
    “type”: “int”,  
    “description”: “It will contain the CVV code of account”.  
},  
  
“Country”: {  
    “type”: “String”,  
    “description”: “It will contain the country of account holder”.  
},  
  
“Account_Expiry”: {  
    “type”: “Date”,  
    “description”: “It will contain the date of expiry of bank card”.  
},  
  
“Billing_Address”: {  
    “type”: “String”,  
    “description”: “It will contain the address of account holder”.  
},  
  
“Date”: {
```

```
    "type": "Date",
    "description": "It will contain the date of payment".
  },
  "Payment_Amount": {
    "type": "float",
    "description": "It will contain the amount of payment".
  },
}
```

```
{
  "title": "Complaint",
  "type": "object",
  "properties": {
    "Complaint_ID": {
      "type": "String",
      "description": "It will contain the ID of complaint"
    },
    "Type": {
      "type": "String",
      "description": "It will contain the type of complaint".
    },
    "Description": {
      "type": "String",
      "description": "It will contain description of complaint".
    }
  }
}
```

```

    "Status": {
        "type": "boolean",
        "description": "It will contain the progress status of complaint".
    },
    "Date": {
        "type": "Date",
        "description": "It will contain the date of complaint".
    },
    "Time": {
        "type": "Time",
        "description": "It will contain the time of complaint".
    },
}

{
    "title": "User",
    "type": "object",
    "properties": {
        "ID": {
            "type": "int",
            "description": "It will contain the ID of user"
        },
        "Profile_Image": {

```

```
    "type": "image",
    "description": "It will contain the profile picture of user".
  },
  "CNIC": {
    "type": "String",
    "description": "It will contain the CNIC of user".
  },
  "First_Name": {
    "type": "String",
    "description": "It will contain the first name of user".
  },
  "Last_Name": {
    "type": "String",
    "description": "It will contain the last name of user".
  },
  "Contact": {
    "type": "int",
    "description": "It will contain the contact number of user".
  },
  "Email": {
    "type": "String",
    "description": "It will contain the email of user".
  },
  "Password": {
    "type": "String",

```

```

    "description": "It will contain the password of user".

    },

    "isPremiumAccount": {

        "type": "boolean",

        "description": "It will contain whether account is premium or not.

    },

    "isRenter": {

        "type": "boolean",

        "description": "It will contain whether user is renter.

    },

    "isProprietor": {

        "type": "boolean",

        "description": "It will contain whether user is proprietor.

    }

}

{

    "title": "Feedback",

    "type": "object",

    "properties": {

        "ID": {

            "type": "String",

            "description": "It will contain the ID of feedback"

        },

        "Description": {


```

```
        "type": "String",
        "description": "It will contain description of feedback".
    },
    "Rating": {
        "type": "int",
        "description": "It will contain the rating from 1-5".
    },
    "Date": {
        "type": "Date",
        "description": "It will contain the date of feedback".
    },
    "Time": {
        "type": "Time",
        "description": "It will contain the time of feedback".
    },
},
{
    "title": "Notification",
    "type": "object",
    "properties": {
        "ID": {
            "type": "String",
            "description": "It will contain the ID of notification"
        },
    }
}
```

```

“Type”: {

    “type”: “String”,
    “description”: “It will contain the type of notification”.
},

“Message”: {

    “type”: “String”,
    “description”: “It will contain message of notification”.
},

“Title”: {

    “type”: “String”,
    “description”: “It will contain the Title of notification”.
},

}

{

    “title”: “Notification_Settings”,
    “type”: “object”,
    “properties”: {

        “Chat_Notification_State”: {

            “type”: “boolean”,
            “description”: “It will contain whether chat notification is on or off”
        },
        “Rent_Notification_State”: {

            “type”: “boolean”,
            “description”: “It will contain whether rent notification is on or off”
        }
    }
}

```

},

“Ringtone”: {

“type”: “File”,

“description”: “It will contain the ringtone”.

},

“Intensity”: {

“type”: “int”,

“description”: “It will contain volume of notification”.

},

“Vibration_Frequency”: {

“type”: “float”,

“description”: “It will contain the frequency of vibration”.

},

}

{

“title”: “Rent”,

“type”: “object”,

“properties”: {

“ID”: {

“type”: “String”,

“description”: “It will contain the ID of rent”

},

“Due_Amount”: {

```

    "type": "float",
    "description": "It will contain the amount of due rent".
  },
  "Due_Date": {
    "type": "Date",
    "description": "It will contain Date of due rent".
  },
  "Rent_Status": {
    "type": "boolean",
    "description": "It will contain the progress status of rent payment".
  },
}

{
  "title": "ChatBot",
  "type": "object",
  "properties": {
    "Message_Sent": {
      "type": "String",
      "description": "It will contain the message sent to bot"
    },
    "Time": {
      "type": "Time",
      "description": "It will contain the time of message sent to bot".
    }
  }
}

```

}

{

“title”: “Lease”,

“type”: “object”,

“properties”: {

“ID”: {

“type”: “String”,

“description”: “It will contain the ID of lease”

},

“Description”: {

“type”: “String”,

“description”: “It will contain description of lease”.

},

“Renter_Signature”: {

“type”: “image”,

“description”: “It will contain the verification signature of renter”.

},

“Proprietor_Signature”: {

“type”: “image”,

“description”: “It will contain the verification signature of Proprietor”.

},

“Creation_Date”: {

“type”: “Date”,

“description”: “It will contain the date of lease finalization”.

},

}

{

“title”: “Property”,

“type”: “object”,

“properties”: {

“ID”: {

“type”: “String”,

“description”: “It will contain the ID of property”

},

“Property_Name”: {

“type”: “String”,

“description”: “It will contain name of property”.

},

“Property_Address”: {

“type”: “String”,

“description”: “It will contain address of property”.

},

“Property_Rooms”: {

“type”: “int”,

“description”: “It will contain number of rooms in the property”.

},

```

“Property_Allocation_Area”: {

    “type”: “float”,
    “description”: “It will contain area of the property”.

},

“Property_Floors”: {

    “type”: “int”,
    “description”: “It will contain number of floors in the property”.

},

“Number_of_Renters”: {

    “type”: “int”,
    “description”: “It will contain number of renters allowed to rentthe
property”.
},

}

{

    “title”: “Chat”,
    “type”: “object”,
    “properties”: {

        “Sender_ID”: {

            “type”: “String”,
            “description”: “It will contain the ID of message sender”

        },
        “Receiver_ID”: {

            “type”: “String”,
            “description”: “It will contain the ID of message receiver”
        }
    }
}

```

```
        },  
  
        “Sender_Message”: {  
            “type”: “String”,  
            “description”: “It will contain the sender message”  
        },  
  
        “Receiver_Message”: {  
            “type”: “String”,  
            “description”: “It will contain the receiver message”  
        },  
  
        “Date_Sent”: {  
            “type”: “Date”,  
            “description”: “It will contain the date of message sent”.  
        },  
  
        “Time_Sent”: {  
            “type”: “Time”,  
            “description”: “It will contain the time of message sent”.  
        },  
    }  

```

4.4. Process Flow

Following are some of the activity diagrams of the application “**Renter-Proprietor Coordination System**”:

- **Module 1: Profile Management**

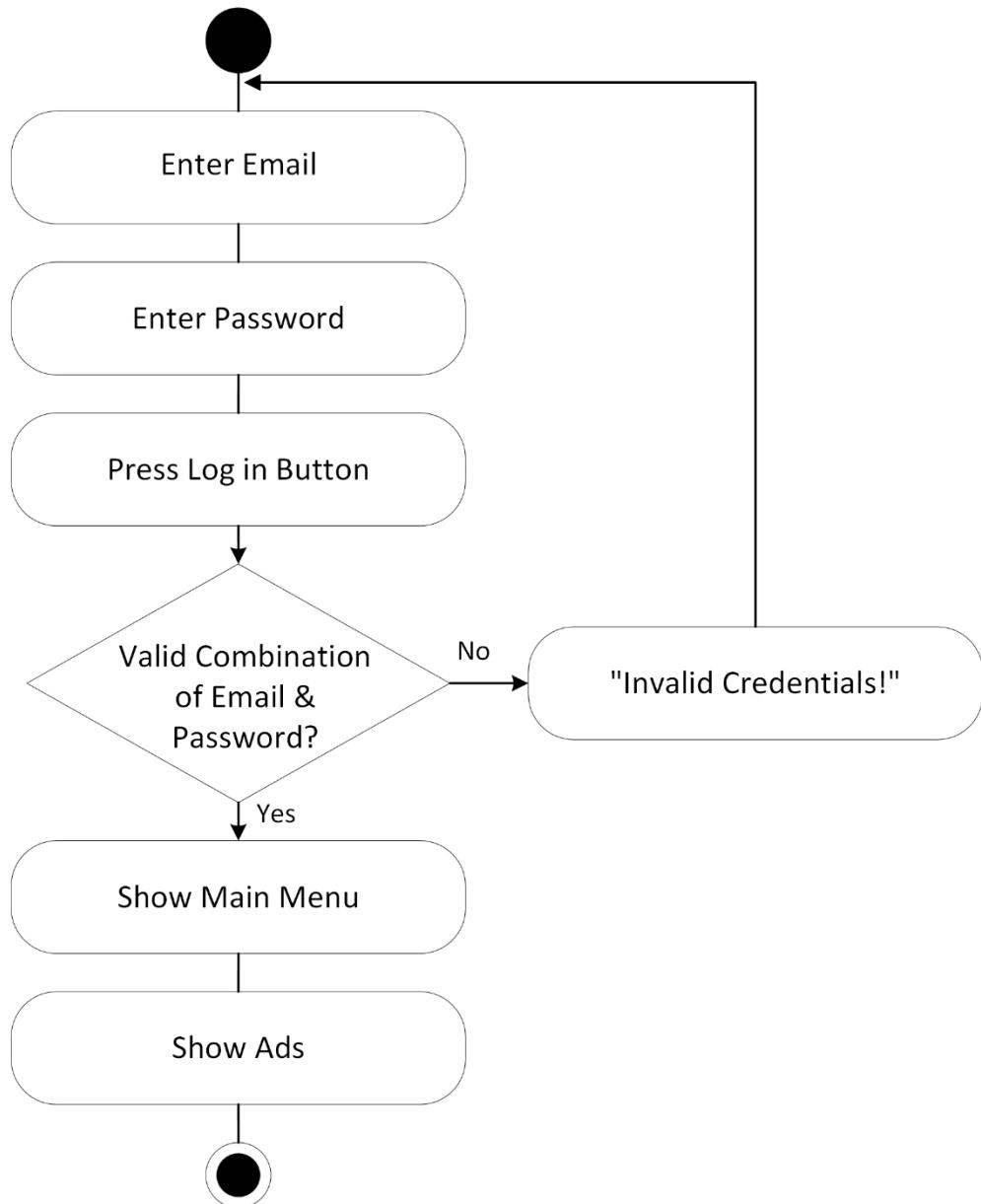


Figure 5: Activity Diagram of Showing Ads After Logging in

- **Module 2: User Interaction Chat Box**

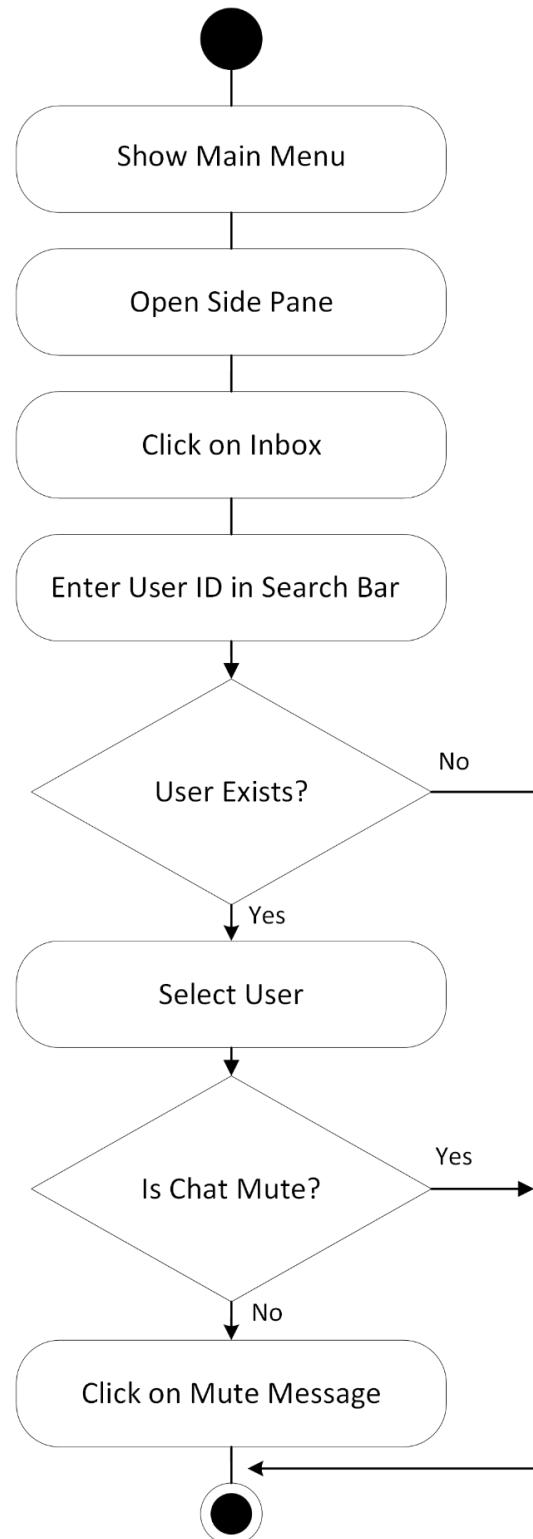


Figure 6: Activity Diagram of Muting a User

- Module 3: Rent Payment Portal

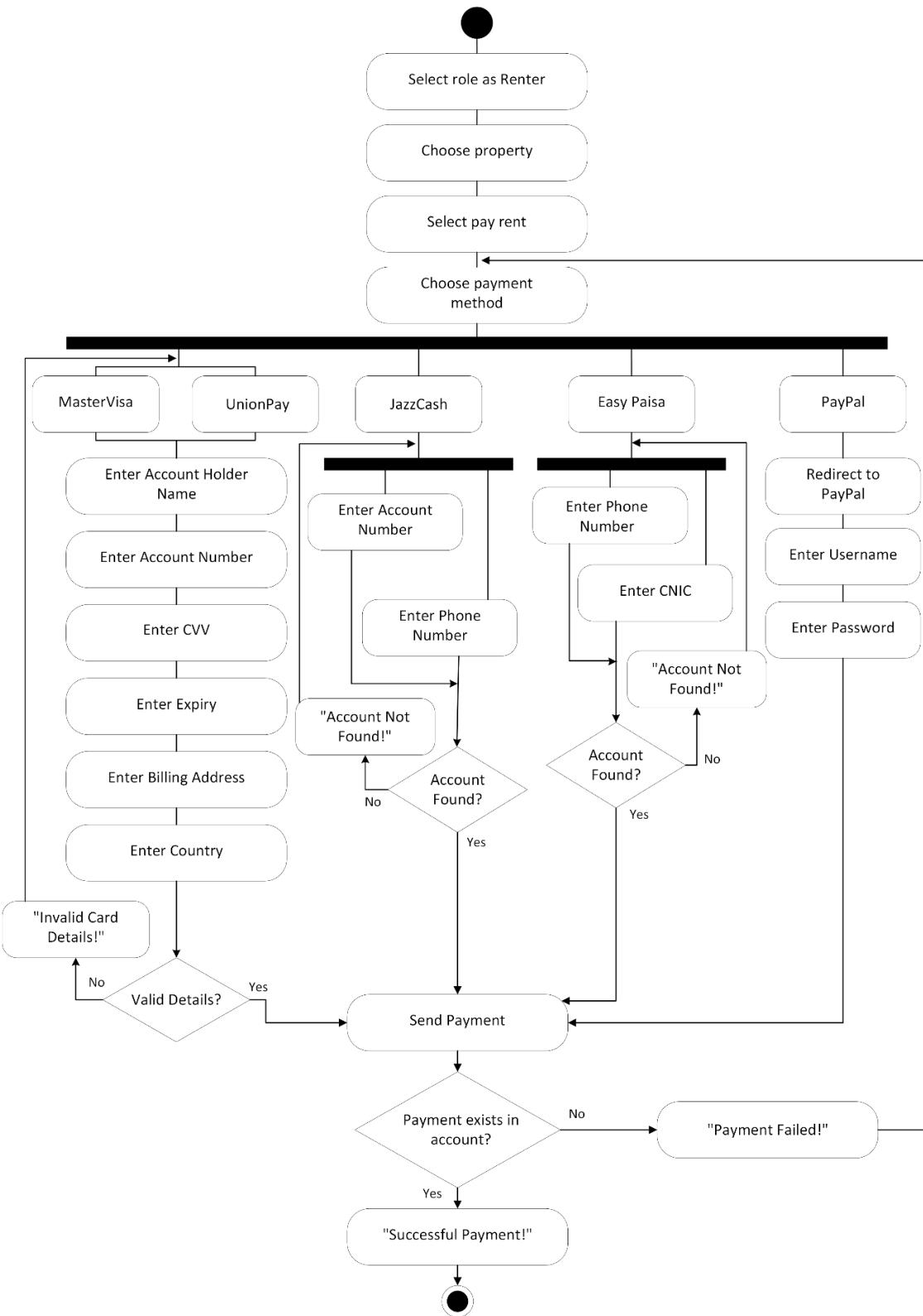


Figure 7: Activity Diagram of Payment System

- Module 4: Complaint Management Portal

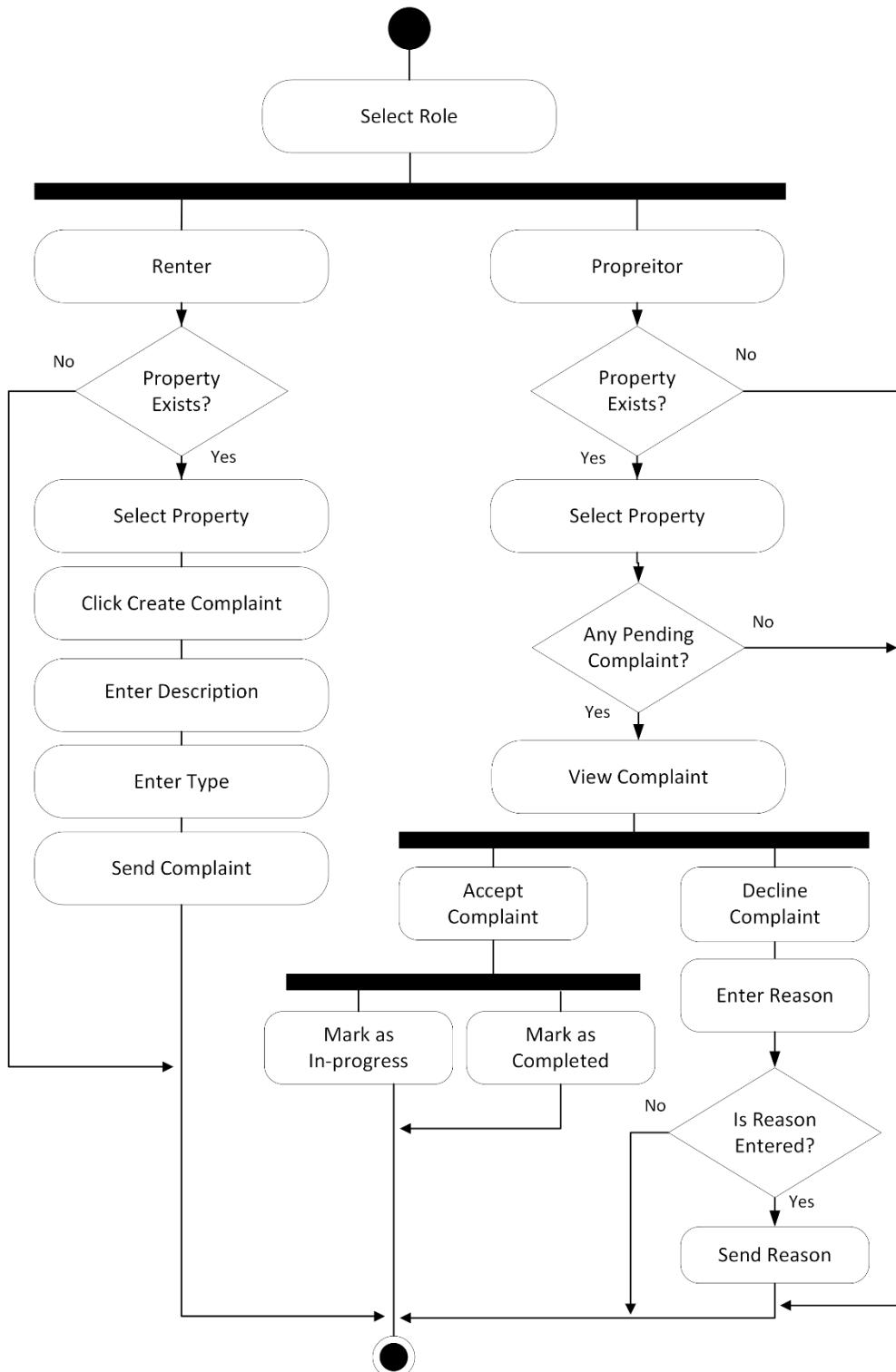


Figure 8: Activity Diagram of Sending/Responding Complaints

- Module 5: Tenant Management Portal

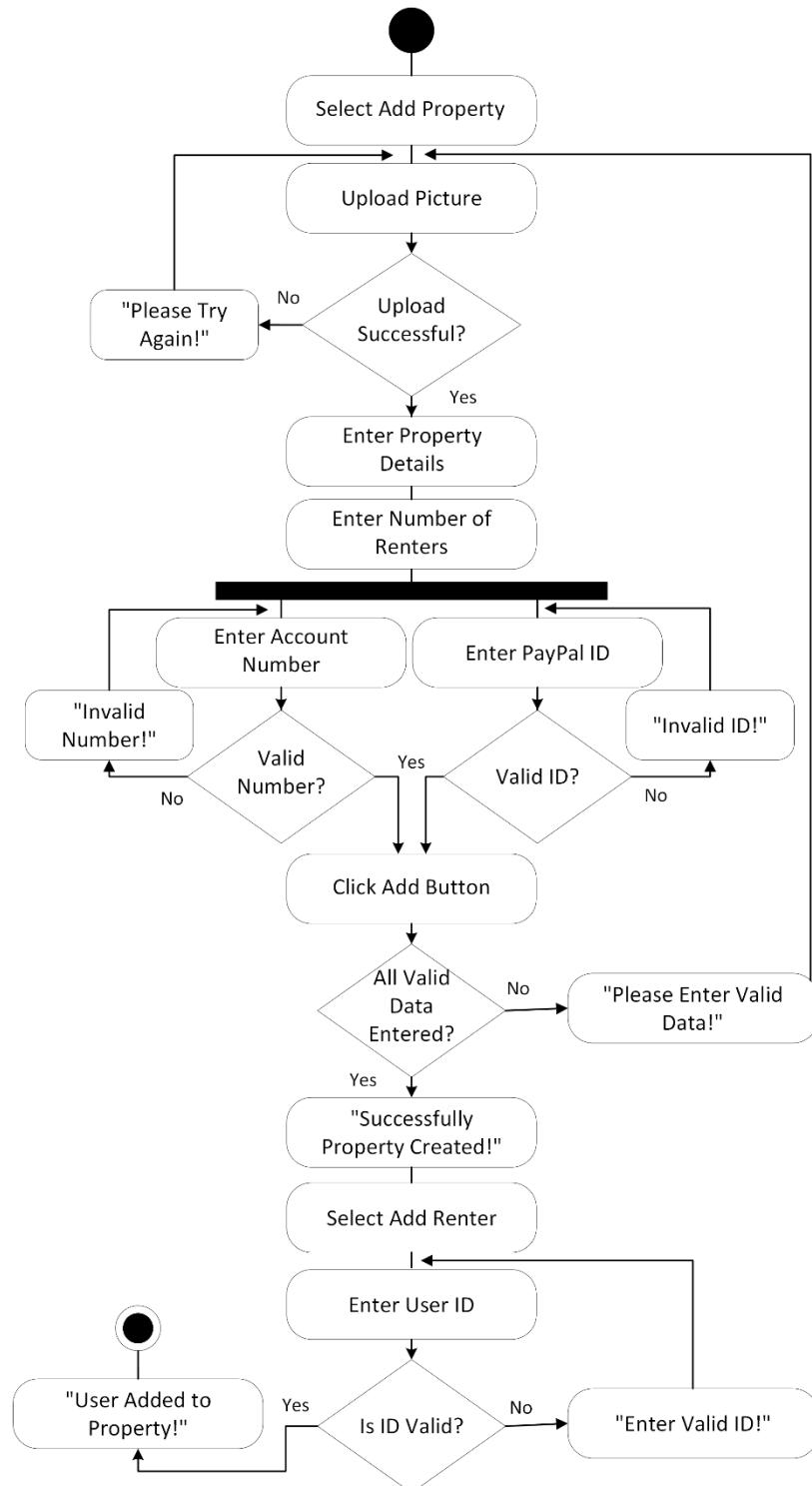


Figure 9: Activity Diagram of Adding User Property

- Module 6: Property Management Portal

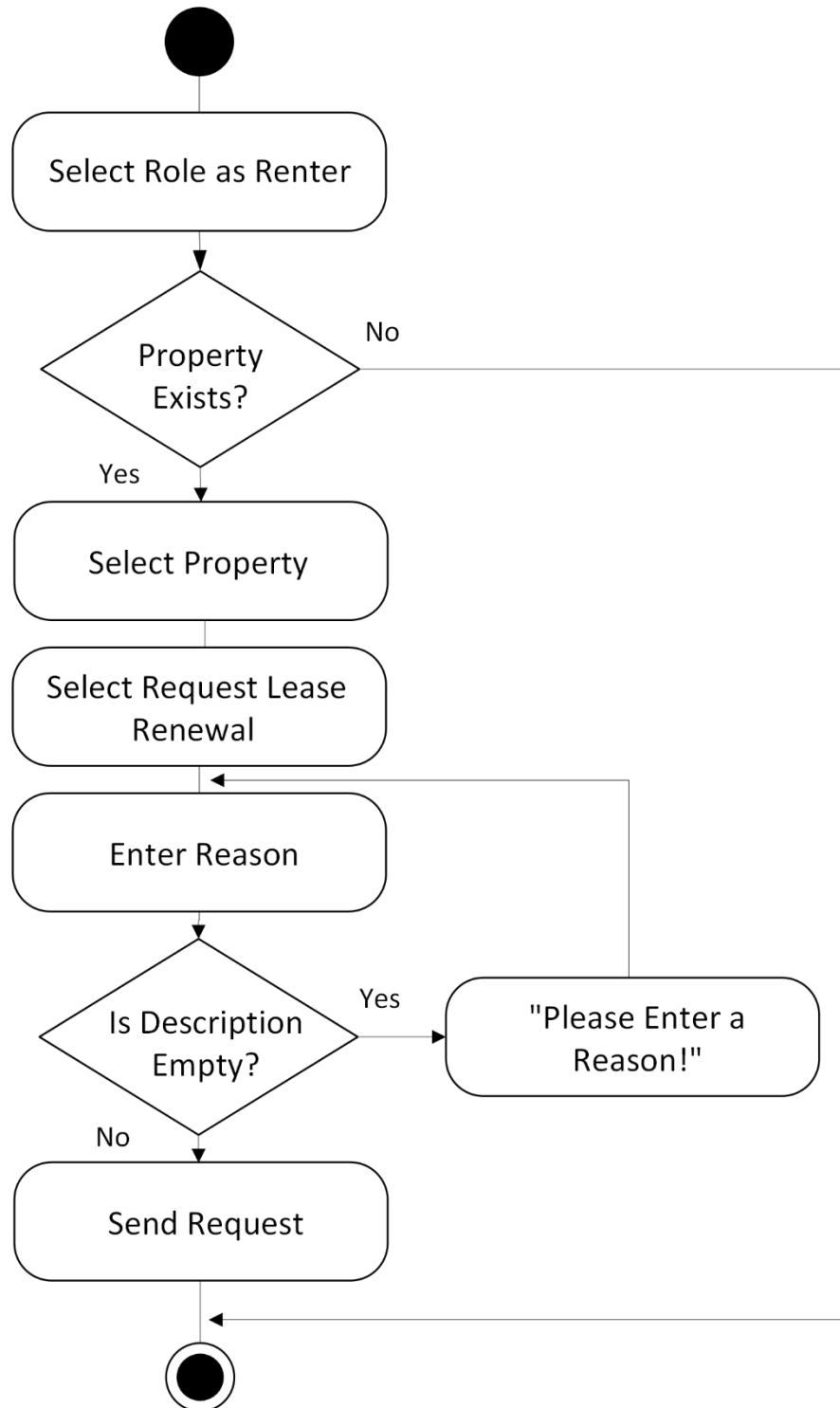


Figure 10: Activity Diagram of Sending Lease Renewal Request

- **Module 7: Push Notification**

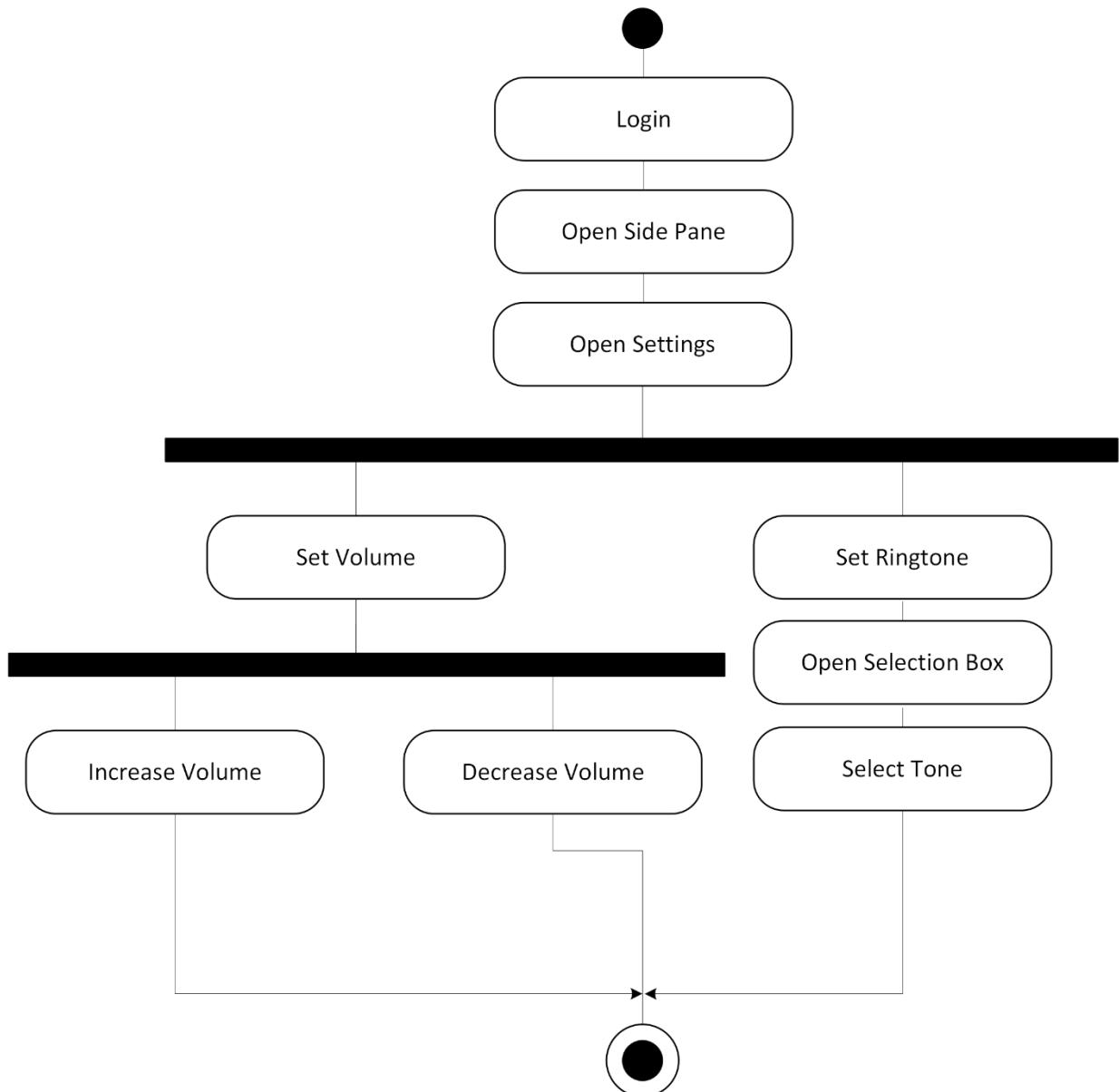


Figure 11: Activity Diagram of Setting Ringtone

- Module 8: Help and Support

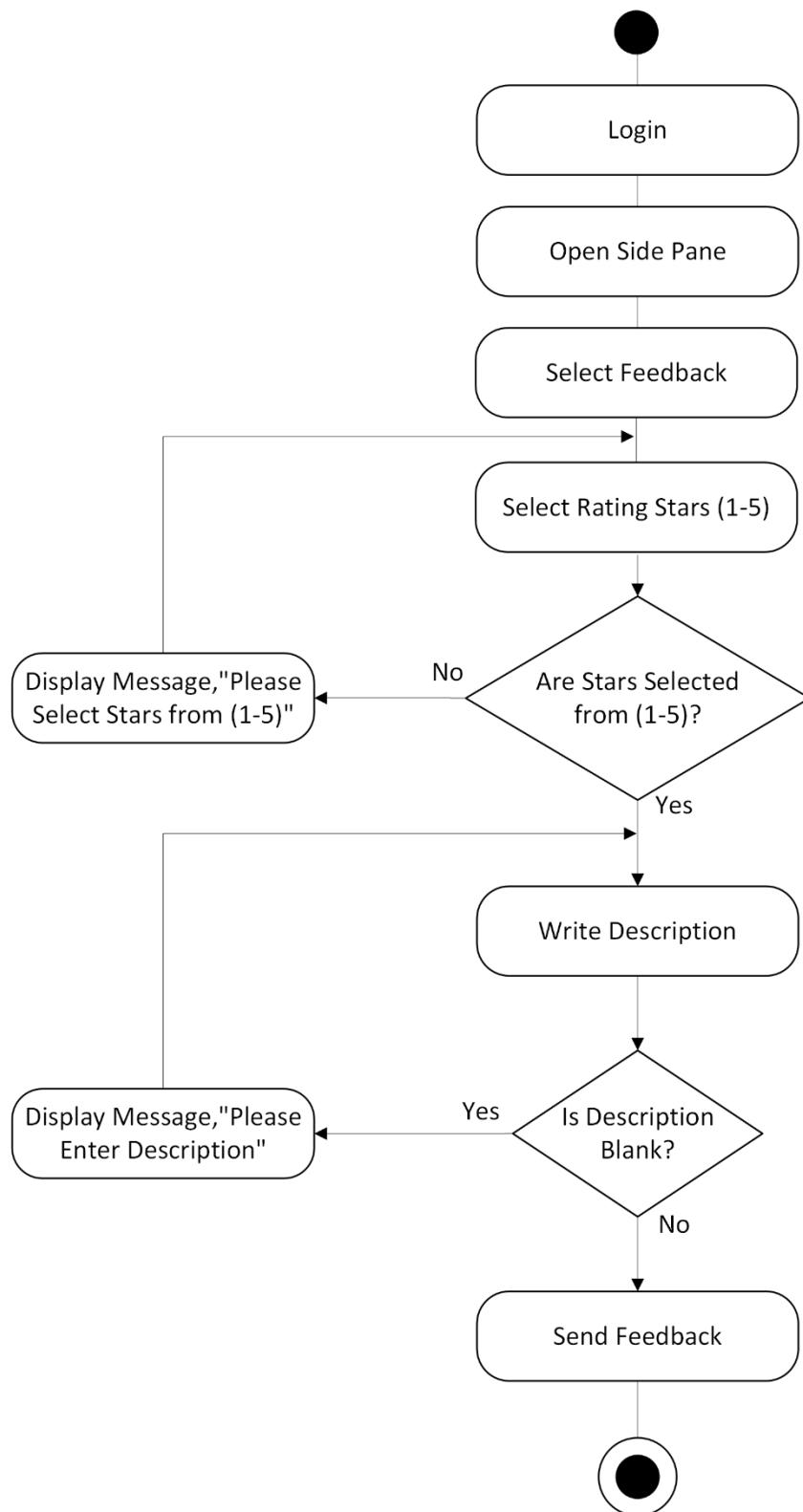


Figure 12: Activity Diagram of Sending Feedback

4.5. Design Models

4.5.1. Class Diagram

The class diagram consists of 20 classes which fetch or insert data from database. The proposed system has MVC architecture at application level so separate classes have been developed for model and controller. Following are the class for Renter-Proprietor Coordination System:

1. Controller
2. User
3. Renter
4. Proprietor
5. Request
6. Property
7. Lease
8. Rent
9. Payment
10. Complaint
11. Chat
12. Live Chat
13. Chat History
14. Ads
15. Chatbot
16. Chatbot Connection
17. Notification
18. Notification Setting
19. Feedback
20. About Us

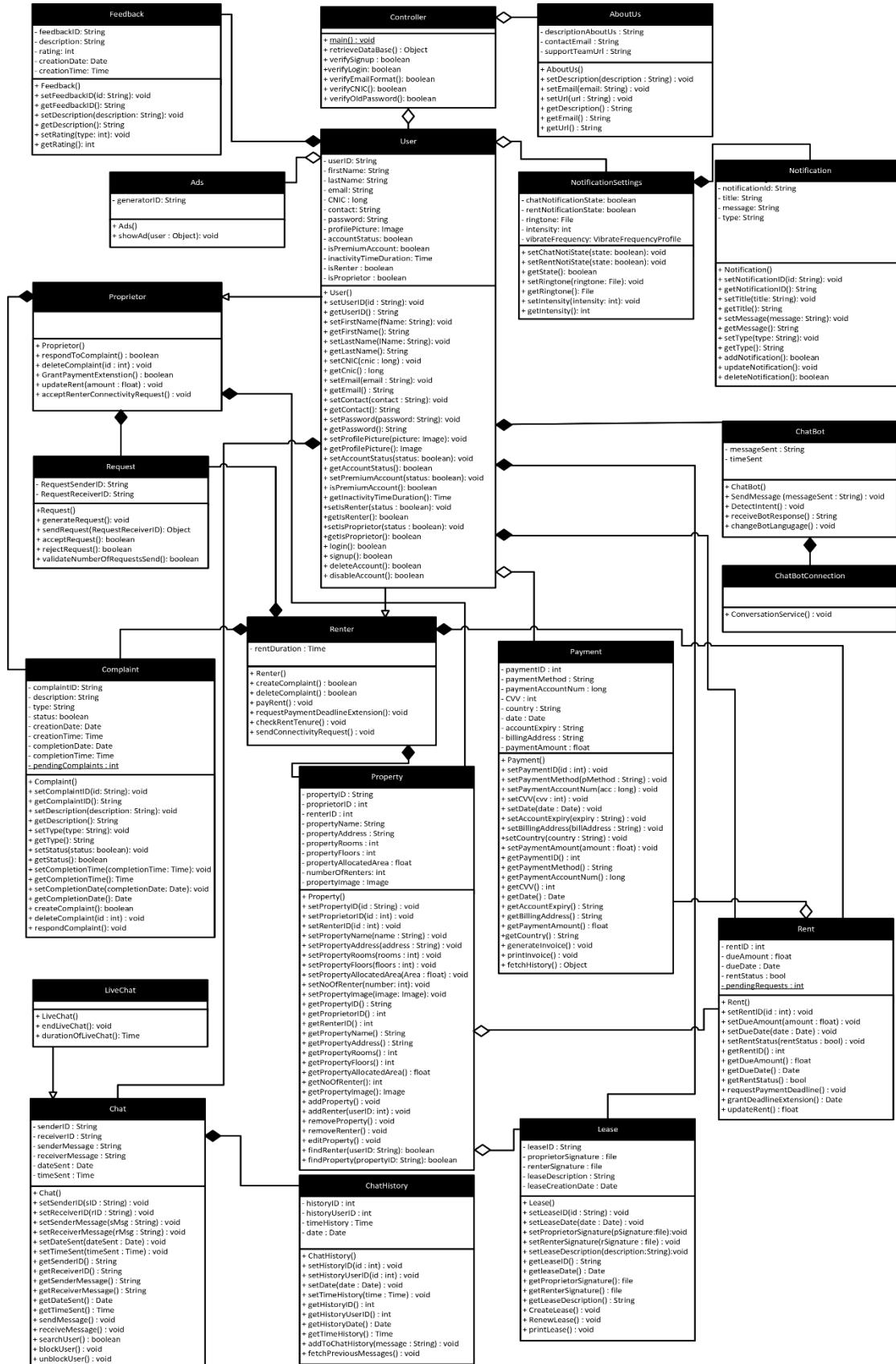


Figure 13: Class Diagram of RPC System

4.5.2. Sequence Diagram

5. Module 1: Profile Management

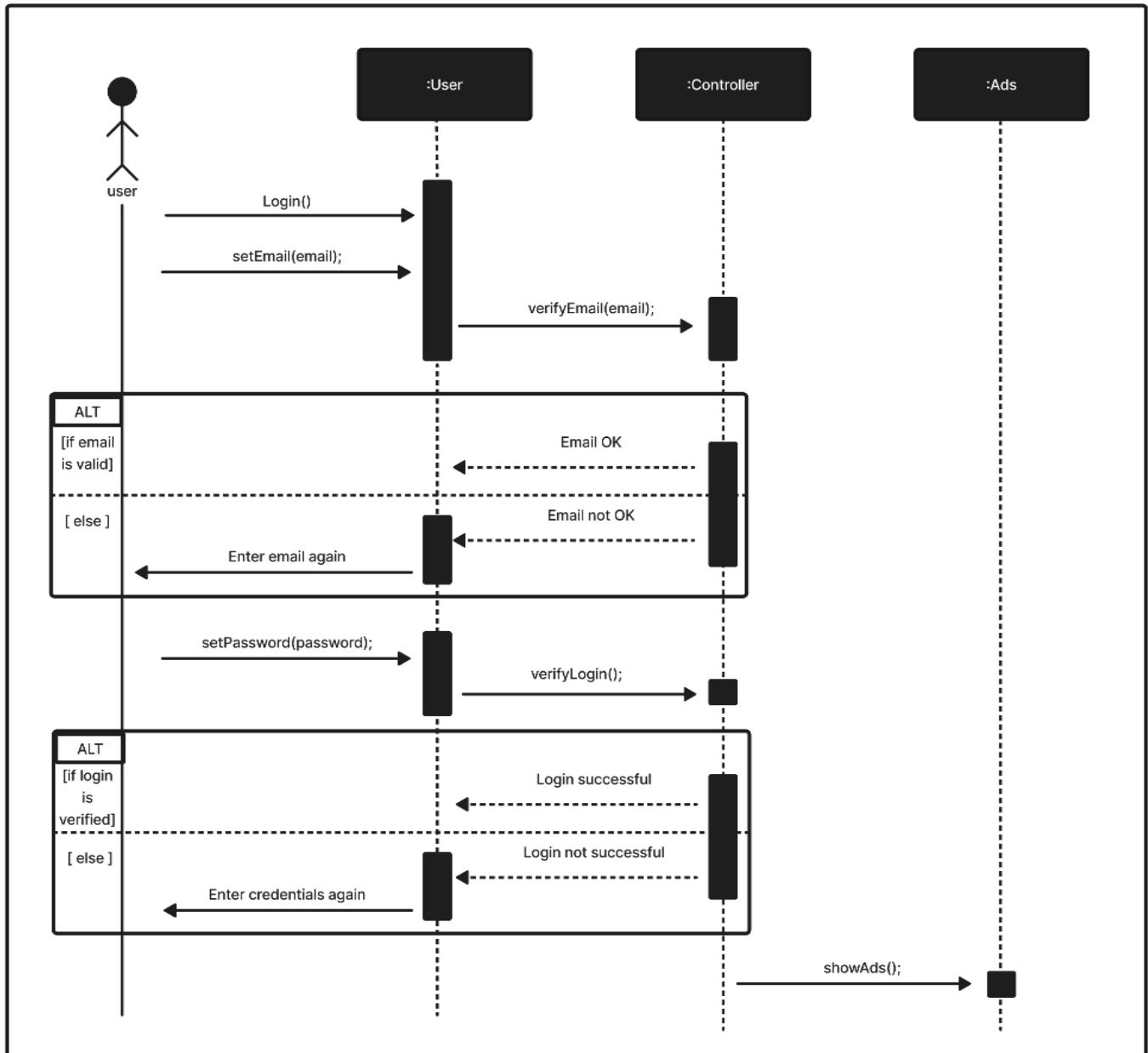


Figure 14: Sequence Diagram of Showing Ads after Logging in

6. Module 2: User Interaction Chat Box

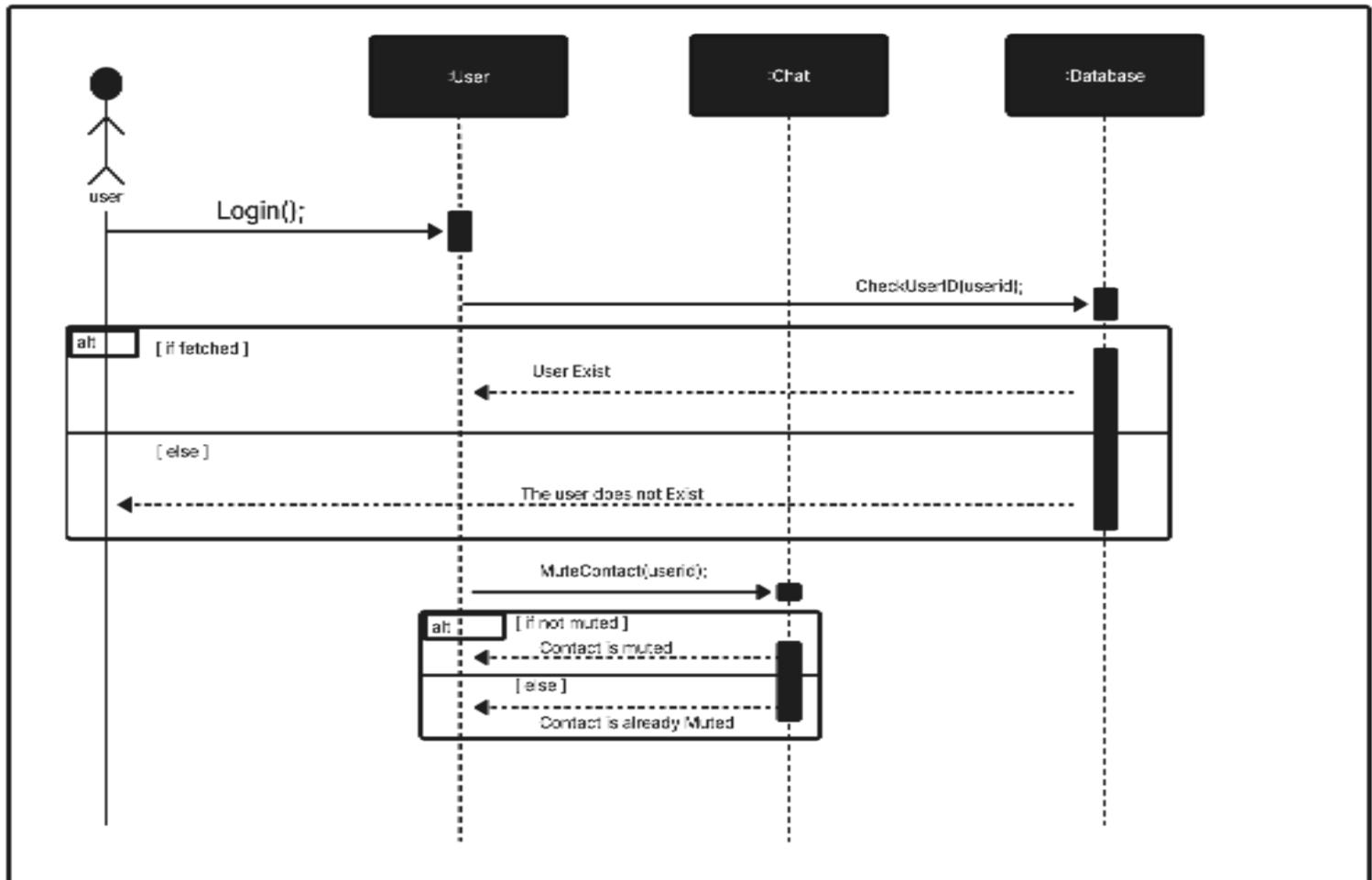


Figure 15: Activity Diagram of Muting User

7. Module 3: Rent Payment Portal

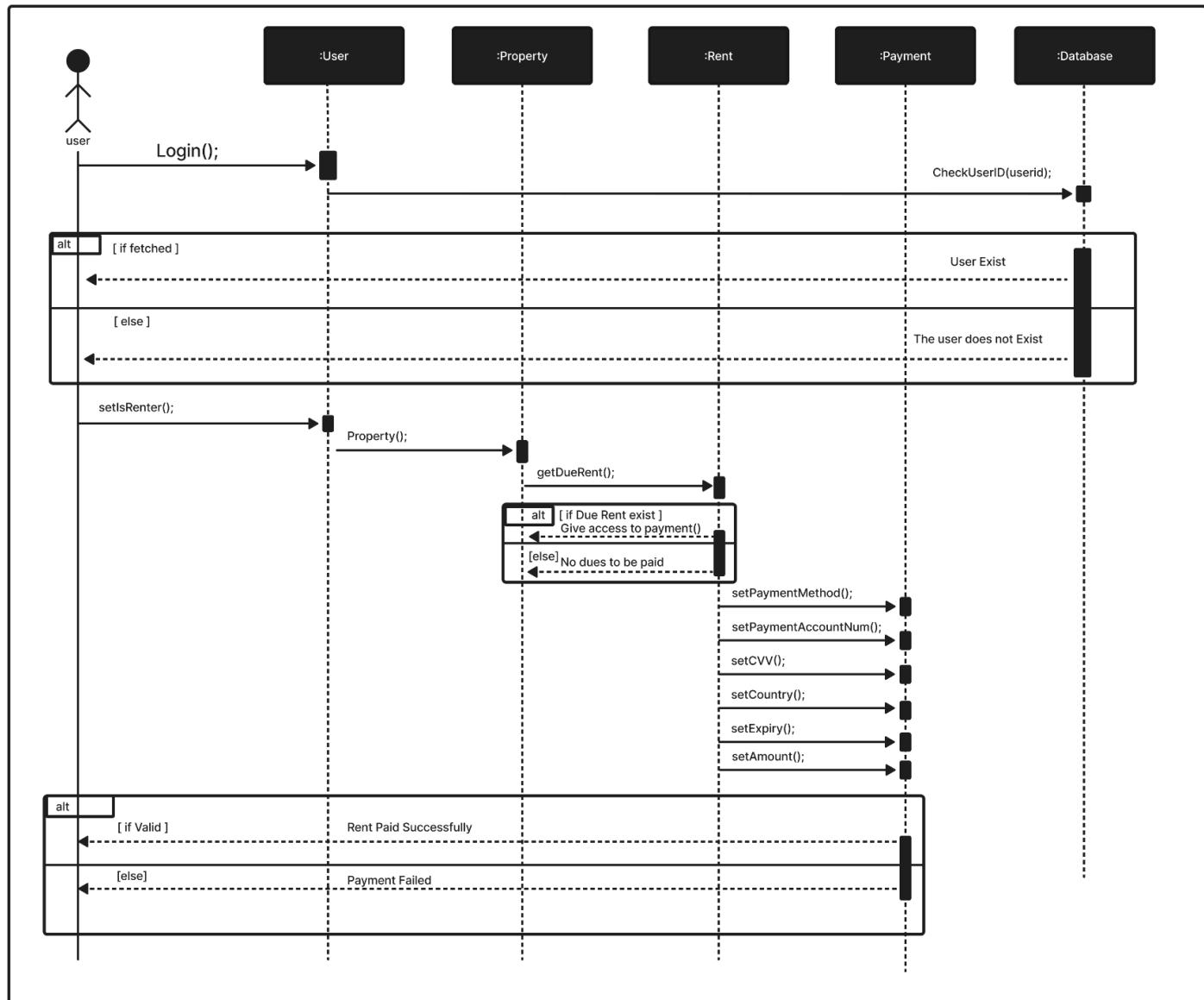


Figure 16: Sequence Diagram of Payment System

8. Module 4: Complaint Management Portal

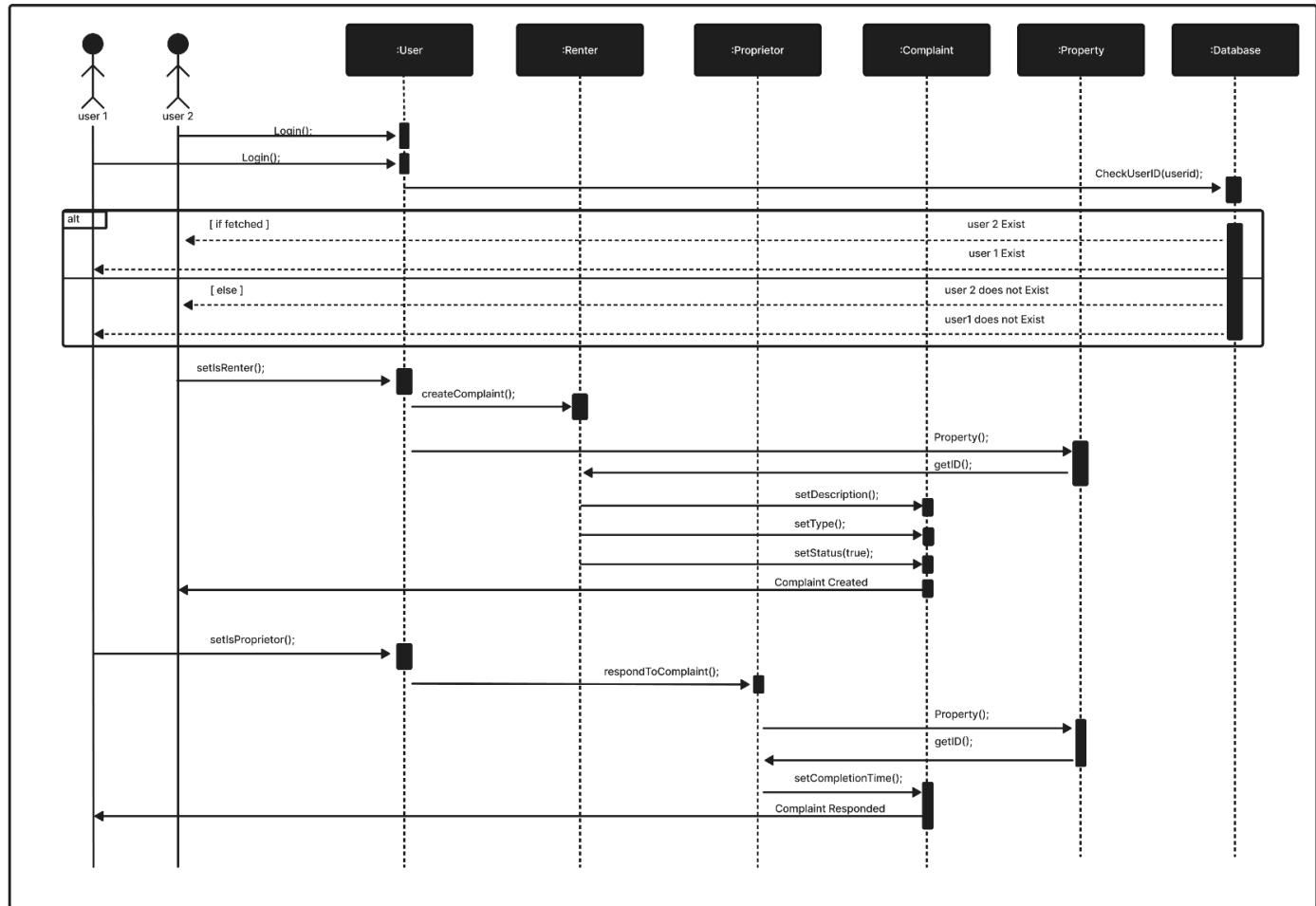


Figure 17: Sequence Diagram of Sending/Responding Complaints

9. Module 5: Tenant Management Portal

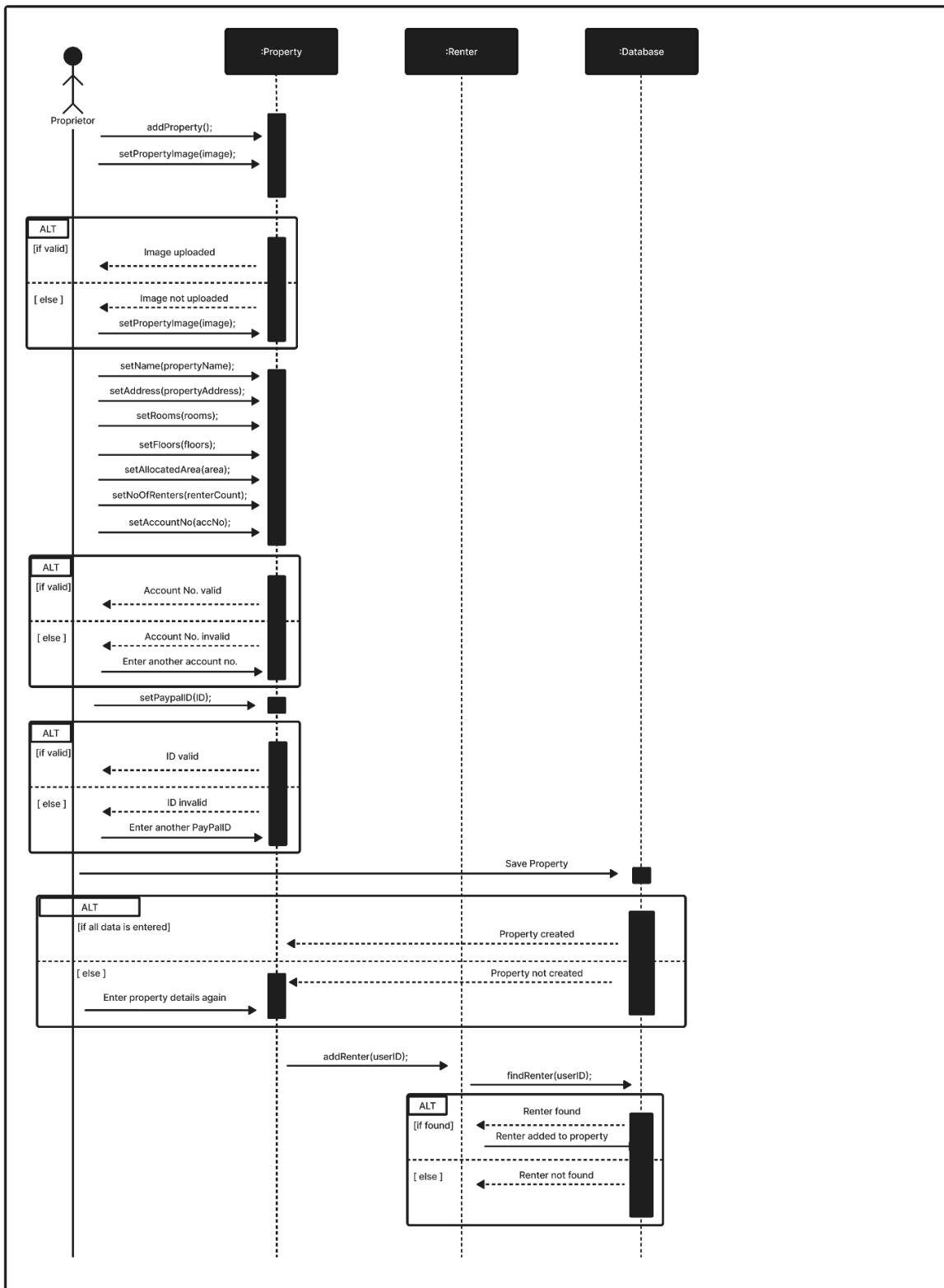


Figure 18: Sequence Diagram of adding Property

10. Module 6: Property Management Portal

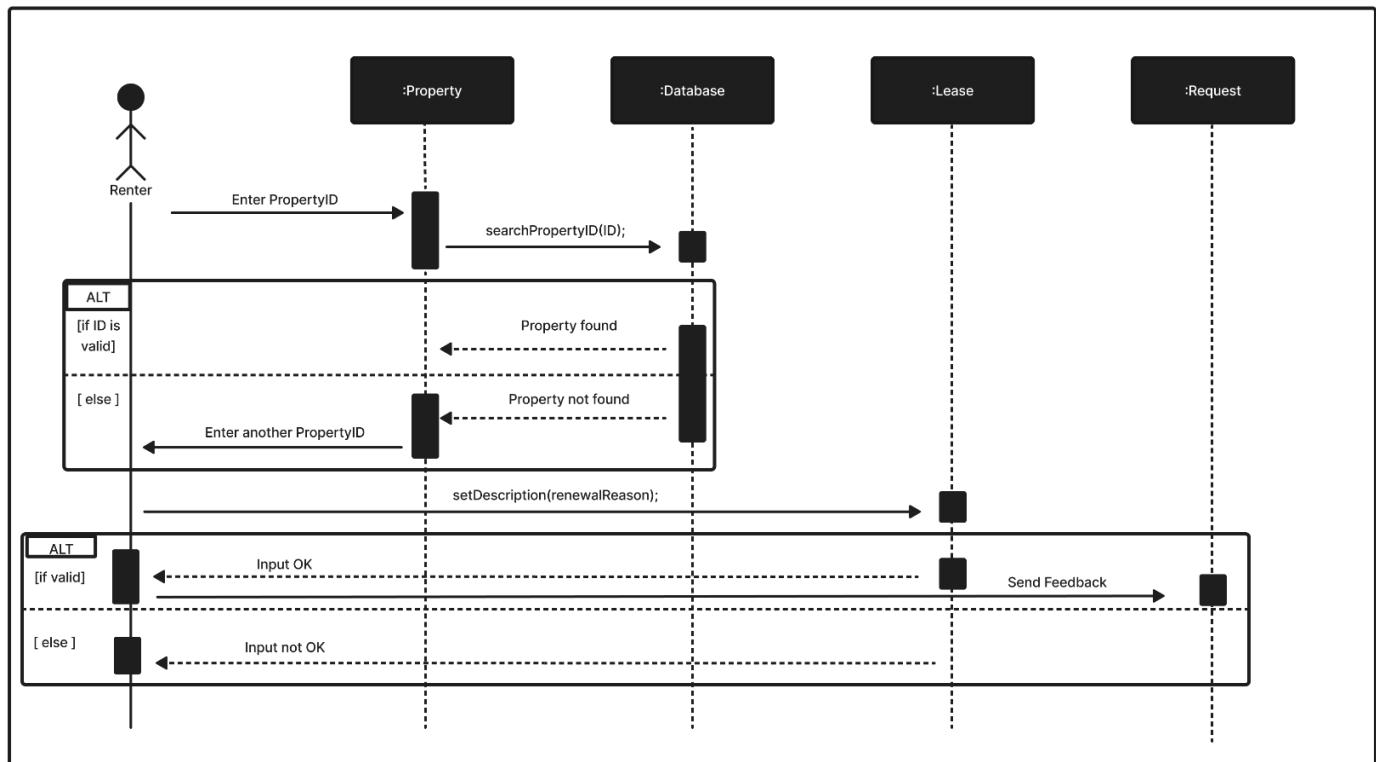


Figure 19: Sequence Diagram of responding to Lease Renewal

11. Module 7: Push Notification

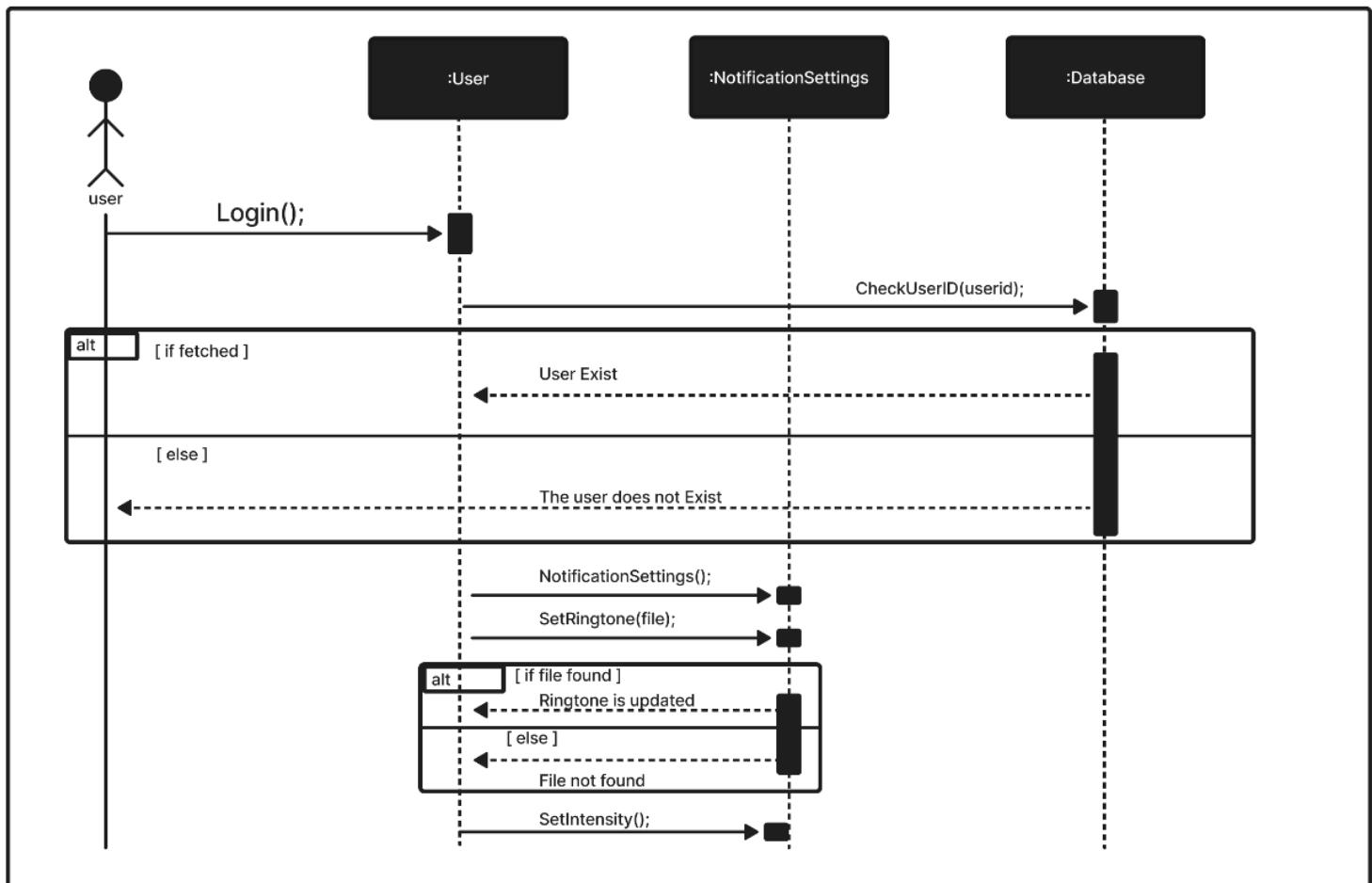


Figure 20: Sequence Diagram of setting Ringtone

12. Module 8: Help and Support

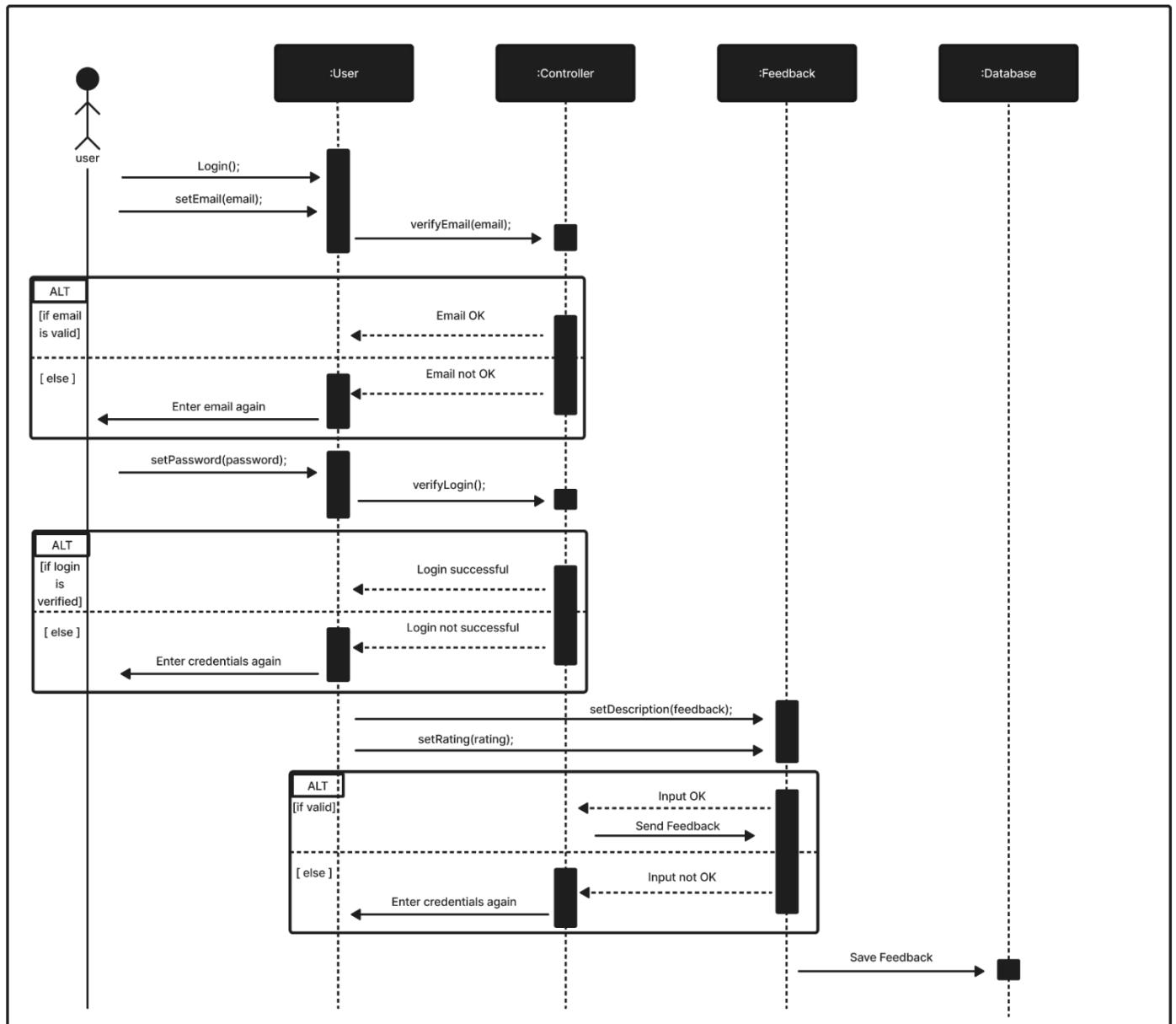


Figure 21: Sequence Diagram of Sending Feedback

5. Implementation

This chapter thoroughly discusses the implementation procedures used in the RPC System. Our project is divided into 8 modules. Algorithm used for their major implementation are discussed. Alongside, the external APIs (Application Programming Interface), like FAQ Bot API, Google Ads API, reCaptacha API, are also discussed. Lastly, the user interface that was designed using Figma is also presented.

5.1. Algorithm

1. Log In and Generation of Ads

```

DO
  INPUT email
  IF(verifyEmail NOT EQUAL true)
    DISPLAY "Invalid Email"
  WHILE (verifyEmail EQUAL false)
    INPUT password
    IF(account NOT found)
      DISPLAY "Invalid login credentials! Please try again!"
    ELSE
      DISPLAY "Successfully logged in!"
      GENERATE ads
      DISPLAY ads
  
```

2. Sending Text Message

```

IF(login)
  DO
    INPUT message
    INPUT receiverID
    WHILE(message NOT null OR valid receiverID)
  
```

SEND message request to receiver end

ELSE

DISPLAY "Please login"

3. Pay Rent

DISPLAY "1: MasterVisa 2: UnionPay 3: JazzCash 4: EasyPaisa 5: PayPal"

INPUT choice

IF(choice EQUAL 1 OR 2)

DO

INPUT accountHolderName, accountNumber, CVV, expiry, billingAddress, country

IF(account NOT found)

DISPLAY "Enter valid account details!"

WHILE(account found)

ELSE IF(choice EQUAL 3)

DO

INPUT accountHolderName OR phoneNumber

IF(account NOT found)

DISPLAY "Enter valid account details!"

WHILE(account NOT found)

ELSE IF(choice EQUAL 4)

DO

INPUT CNIC OR phoneNumber

IF(account NOT found)

DISPLAY "Enter valid account details!"

WHILE(account NOT found)

ELSE IF(choice EQUAL 5)
 CONNECT to PayPal API
ELSE
 DISPLAY "Invalid choice entered!"

4. Creating Complaint

IF(propertyID found)
 DO
 INPUT complaintMessage
 INPUT complaint type
 WHILE(complaintMessage OR complaintType INVALID)
 DISPLAY "Message Sent!"
 INCREMENT pendingComplaints by 1
 ELSE
 DISPLAY "Property not found!"

5. Add Property and Renter to New Property

CHOOSE addProperty
 DO
 UPLOAD profileImage
 IF(upload NOT successful)
 DISPLAY "Please try again!"
 WHILE(upload NOT successful)
 INPUT propertyName, propertyAddress, propertyFloors, propertyAllocatedArea, noOfRenters
 DO
 INPUT AccountDetails
 IF(AccountDetails NOT valid)

```
DISPLAY "Please enter valid account details!"  
WHILE(AccountDetails NOT valid)  
    GENERATE propertyID  
    DISPLAY "Successfully property created!"  
    DO  
        INPUT userID  
        IF(userID NOT found)  
            DISPLAY "Invalid userID!"  
            DISPLAY "Do you want to try again? y/n"  
            INPUT choice  
            WHILE(userID NOT found OR choice EQUAL y)  
                DISPLAY "User added to property!"
```

6. Renter Connectivity with Landlord

```
DO  
    INPUT propertyID  
    IF(propertyID NOT found)  
        DISPLAY "Invalid propertyID!"  
        DISPLAY "Do you want to try again? y/n"  
        INPUT choice  
        WHILE(propertyID NOT found OR choice EQUAL y)  
            SEND request  
            INCREMENT pendingRequests by 1  
            DISPLAY "Request successfully sent!"
```

7. Notification Settings

IF(chatNotification)

DISPLAY chat notification settings

CHANGE settings

ELSE IF(rentNotification)

DISPLAY rent notification settings

CHANGE settings

IF(setRingtone)

SELECT ringtone

IF(intensity)

INPUT vibration intensity level

SET new vibration level

8. Send Feedback

IF(login)

DO

INPUT feedbackMessage

INPUT rating

WHILE(feedbackMessage NOT null OR rating NOT 0)

ELSE

DISPLAY "Please login"

5.2. External APIs

Following are the API used by RPC System [Renter-Proprietor Coordination System]:

Name of API	Description	Purpose
Faqbot API	Faqbot is an AI powered chatbot for customer support. It turns FAQ content into a helper bot.	Eliminates the need for a human to be available 24/7 to resolve redundant and common queries.
Whatsapp API	WhatsApp Business API powers communication with customers all over the world in a simple, secure, and reliable way.	Provides connection between our application and Whatsapp for easy communication.
reCaptcha API	reCAPTCHA is a free service that protects site from spam and abuse. It uses advanced risk analysis techniques to tell humans and bots apart.	Provides secure and safe access to our application.
Google Ads API	The Google Ads API lets advertisers manage large Google Ads accounts and lets developers make applications that use these ads.	A way to provides user free services but the development team still gets revenue out of it.

5.3. User Interface

- Mobile User Interface



Ready to manage your
properties through a click?

Get Started!

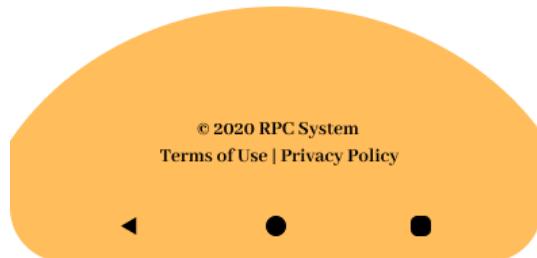


Figure 22: User Interface of Get Started Screen

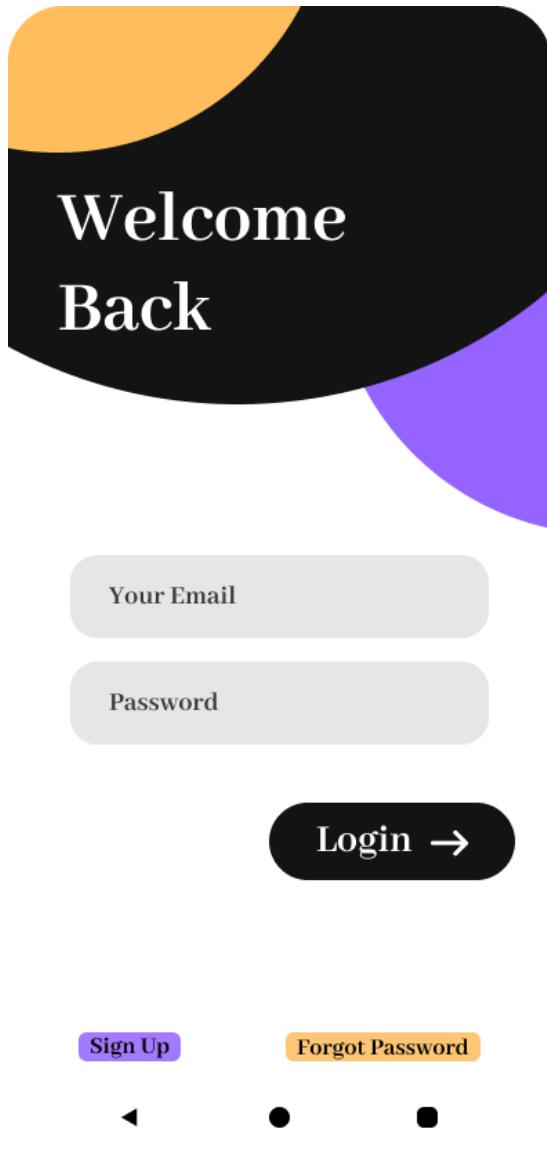


Figure 23: User Interface for Login Screen

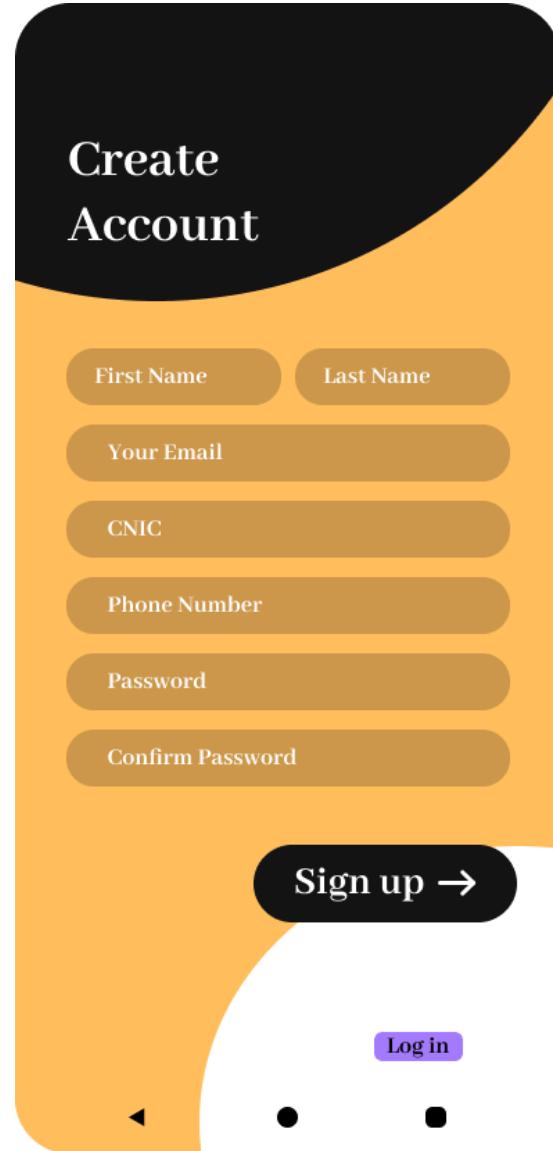


Figure 24: User Interface for Sign Up Screen

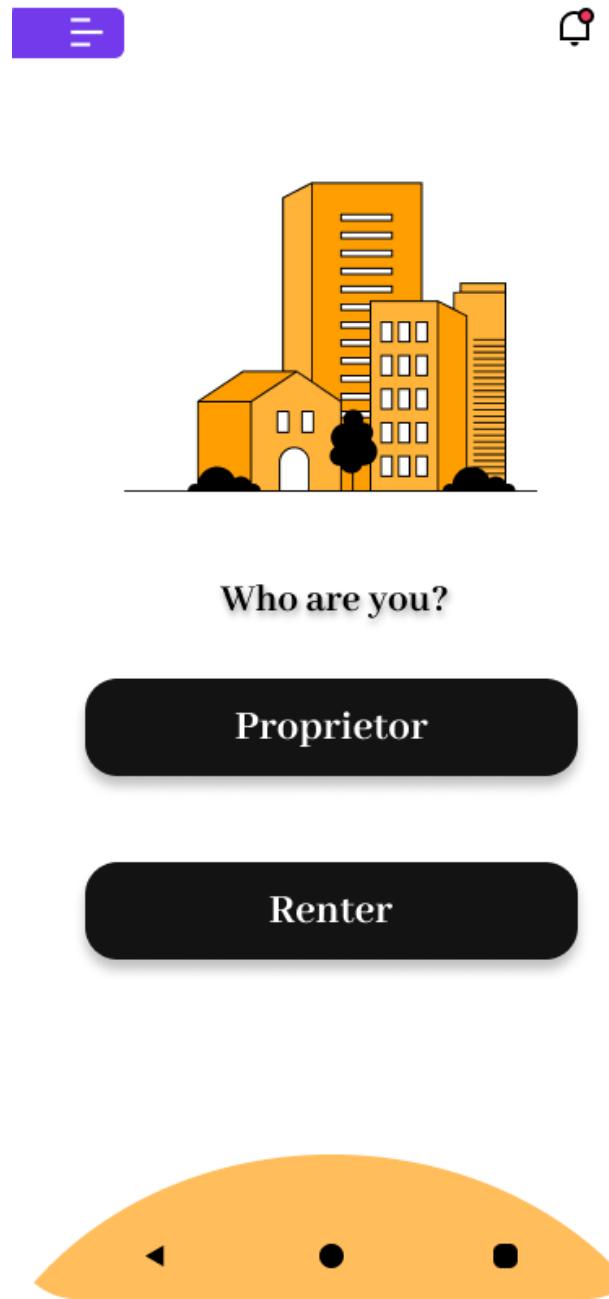


Figure 25: User Interface for Role Selection Screen

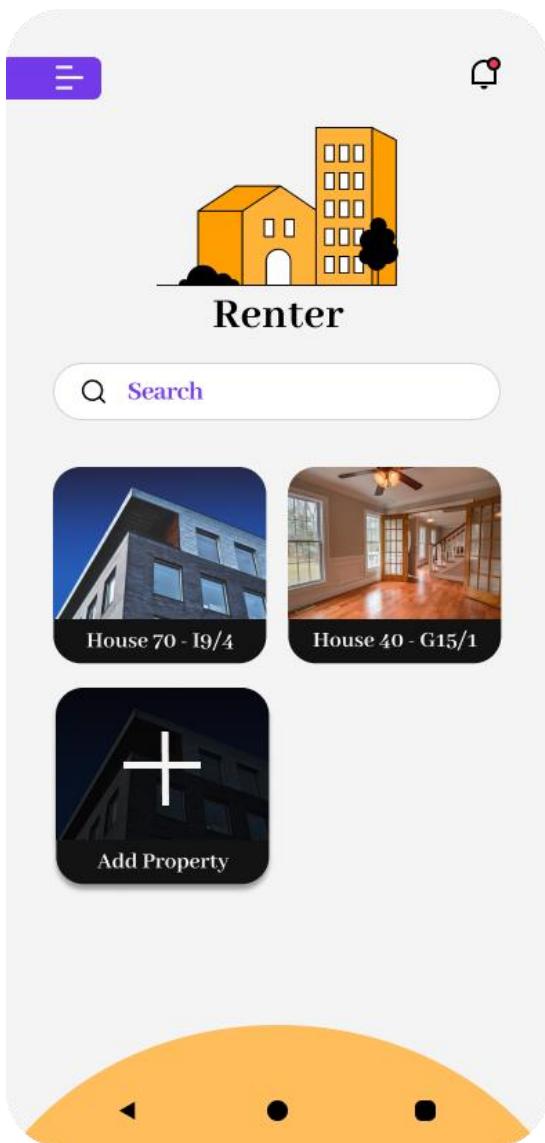


Figure 26: User Interface for Renter Homepage
Menu

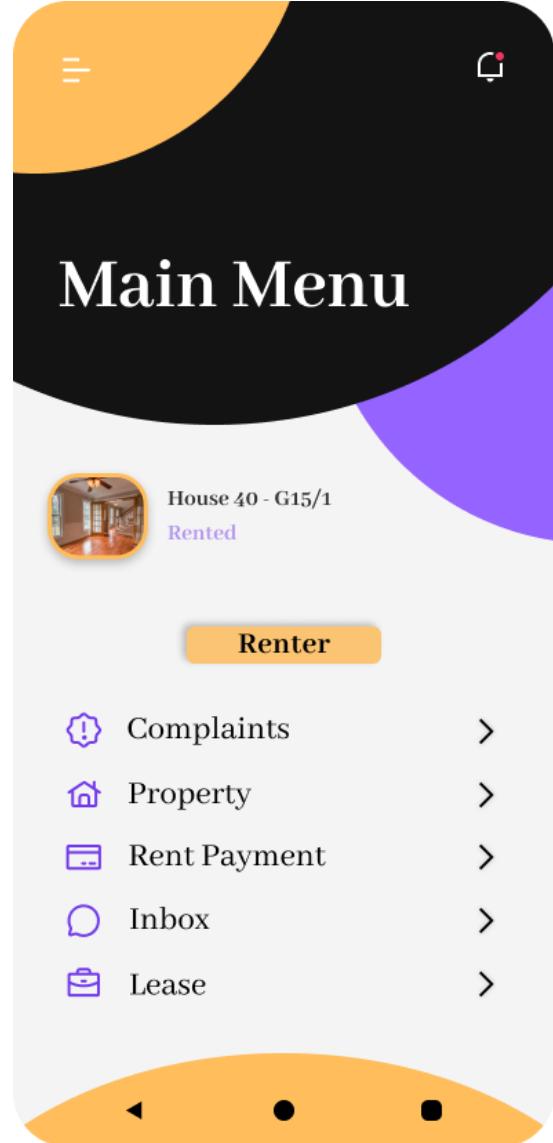


Figure 27: User Interface for Renter Main
Menu

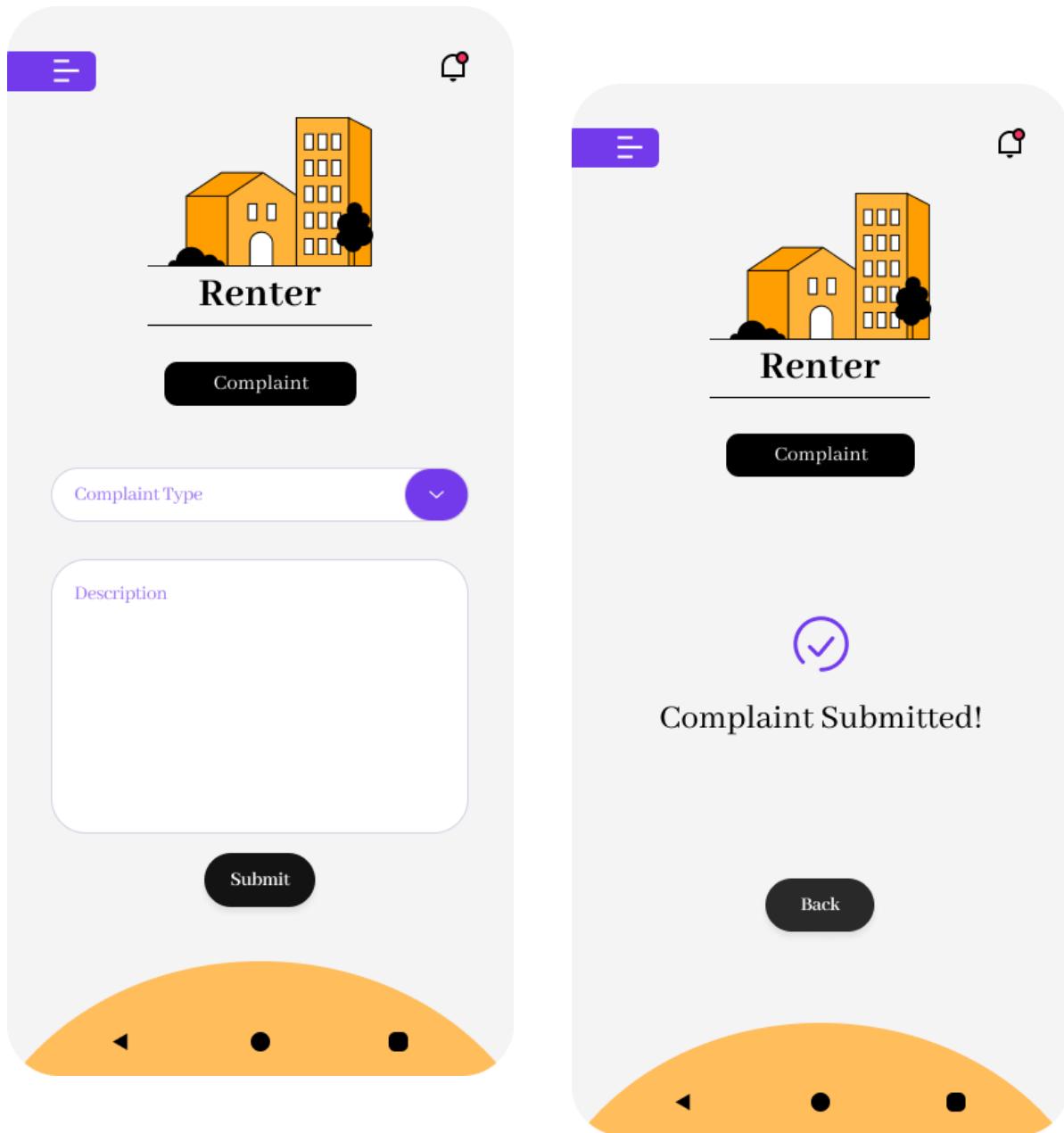


Figure 28: Interface for Renter Complaint Screen
Figure 29: Interface of Complaint Submission

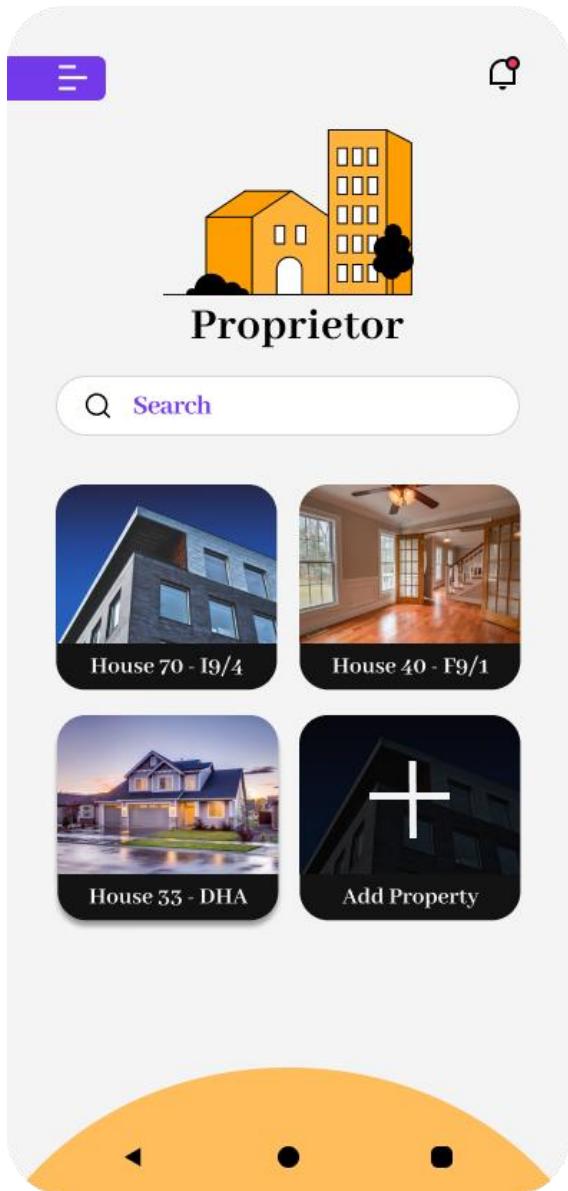


Figure 30: Interface for Proprietor Homepage

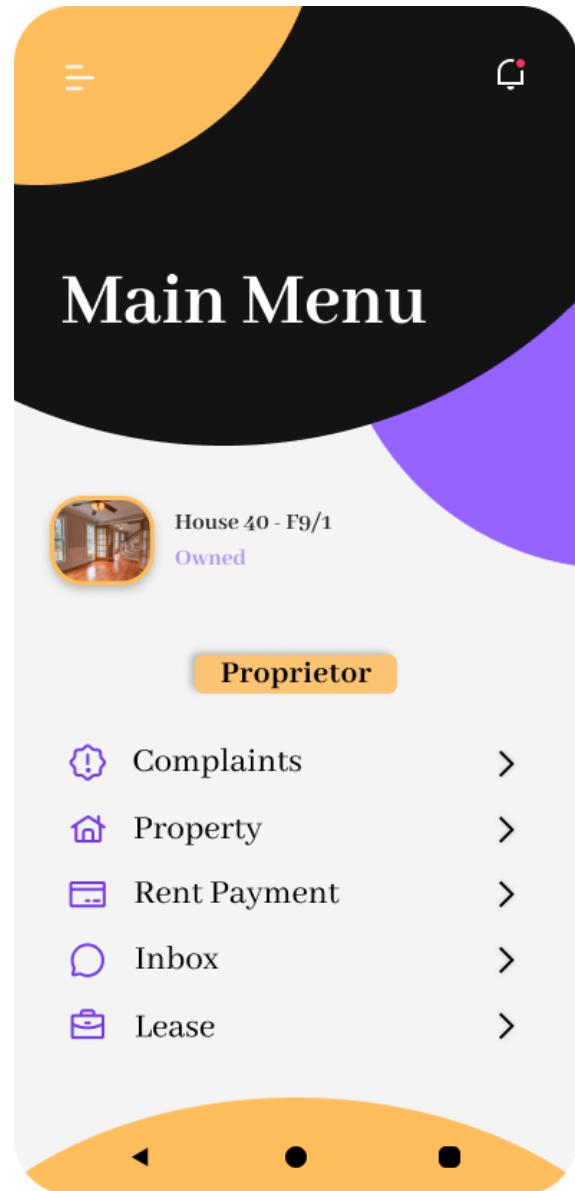


Figure 31: Interface for Proprietor Main Menu

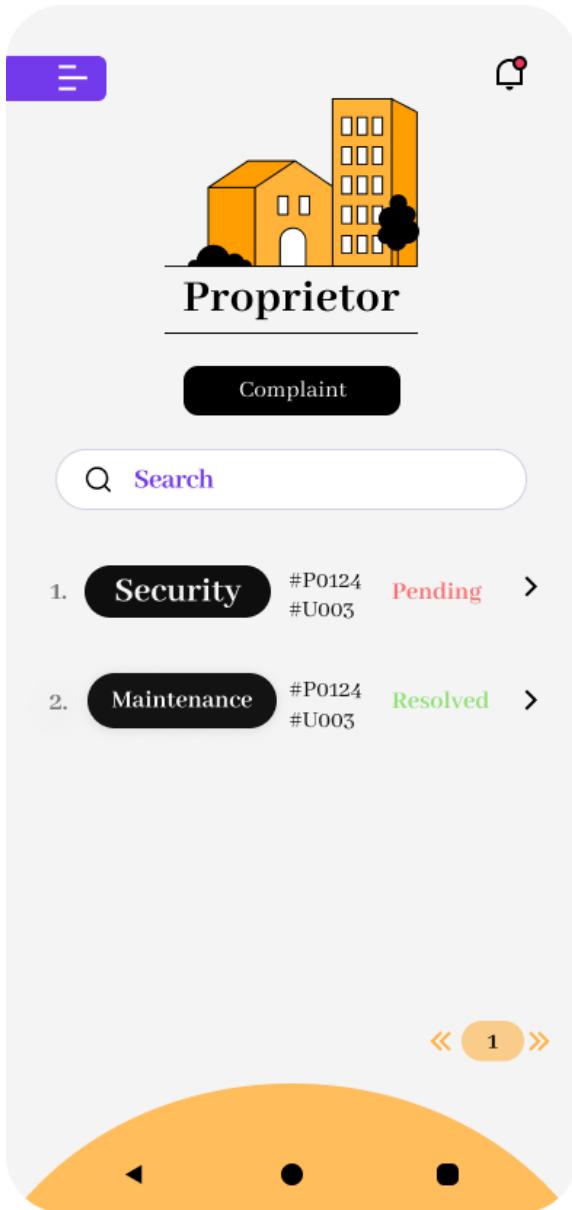


Figure 32: Interface for Proprietor Complaint

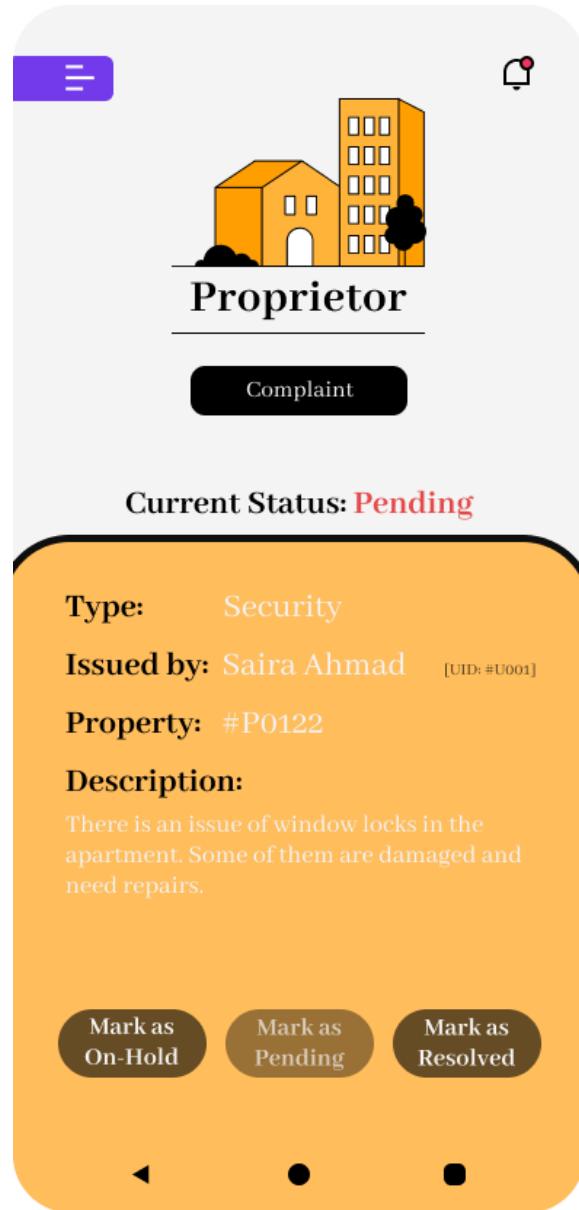


Figure 33: Interface for Proprietor Complaint Details

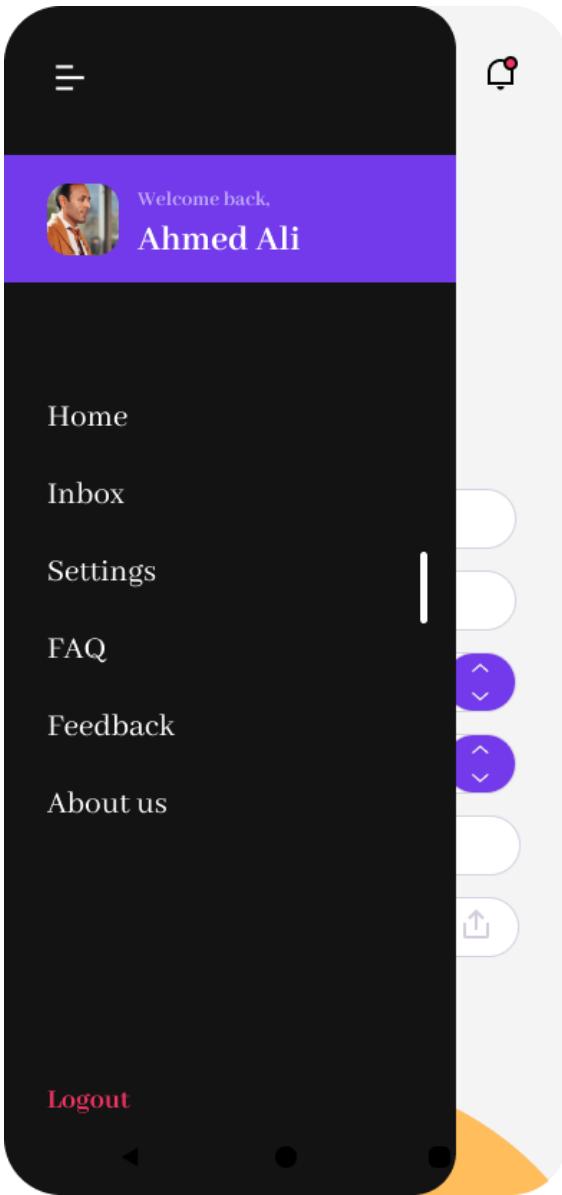


Figure 34: User Interface of Side Pane Menu

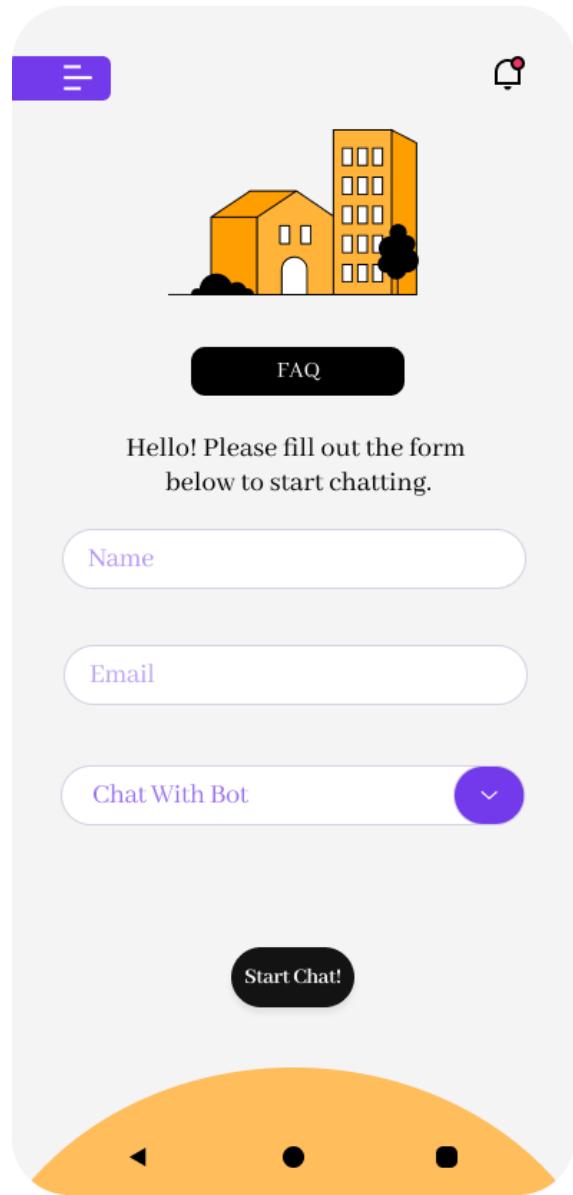


Figure 35: User Interface of FAQ menu

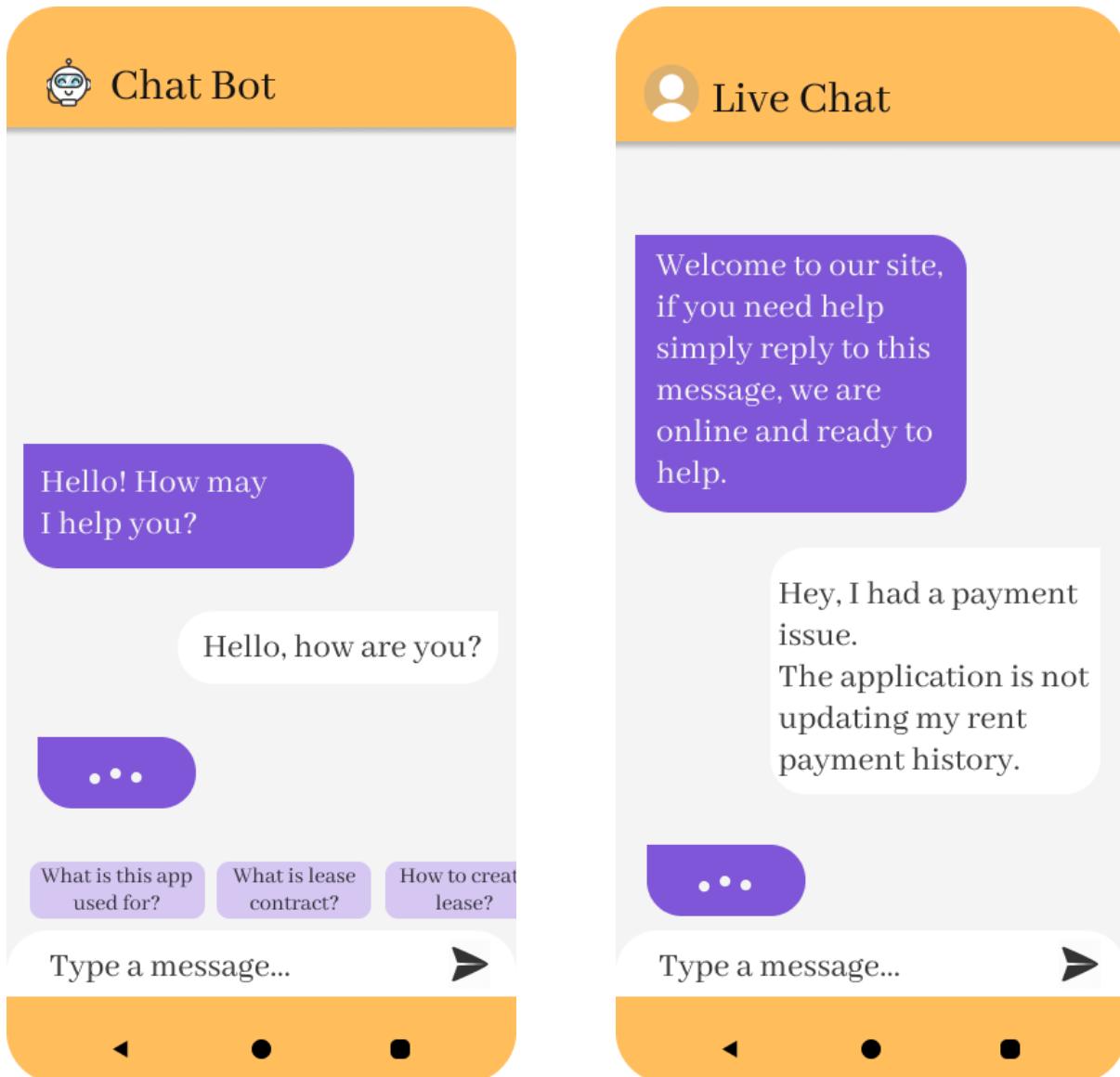


Figure 36: User Interface of Chat Bot Screen

Figure 37: User Interface of Live Chat

- Web User Interface

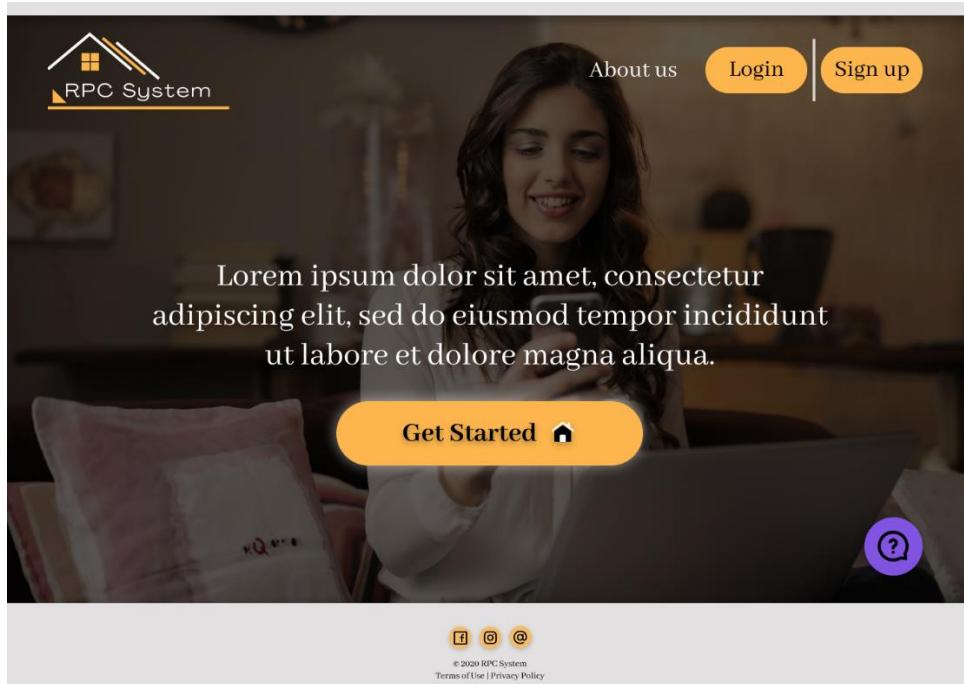


Figure 38: User Interface of Get Started Screen

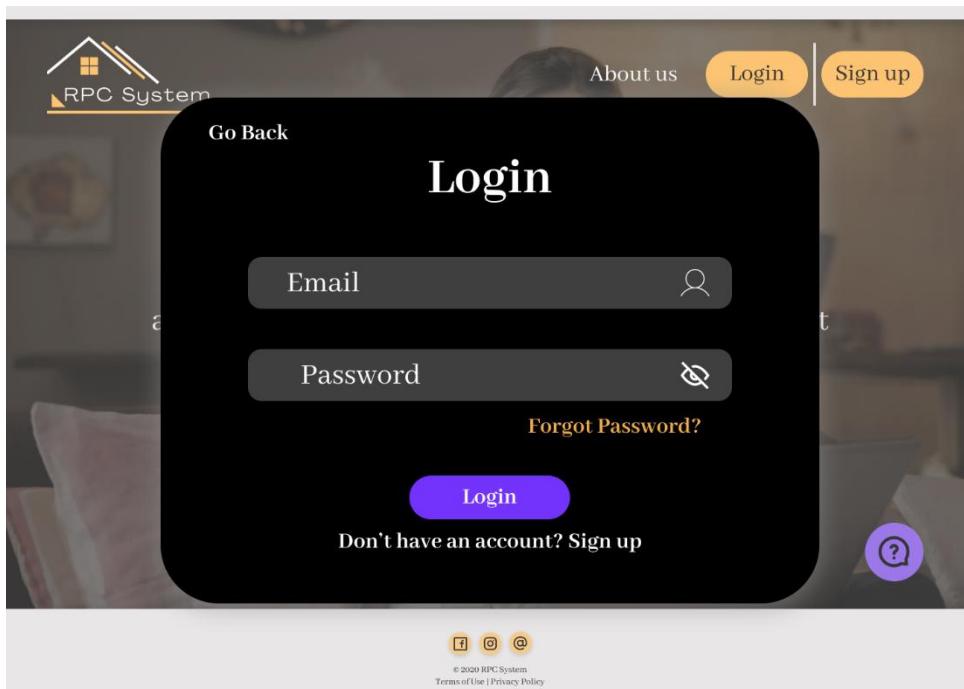


Figure 39: User Interface of Login Screen

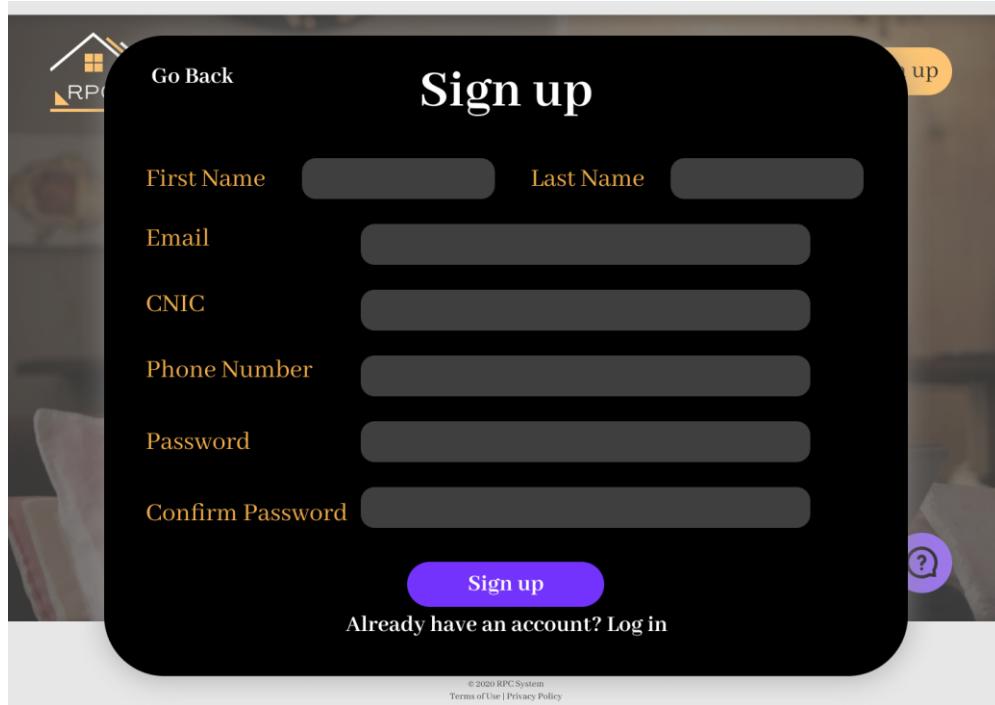


Figure 40: User Interface of Sign up Screen

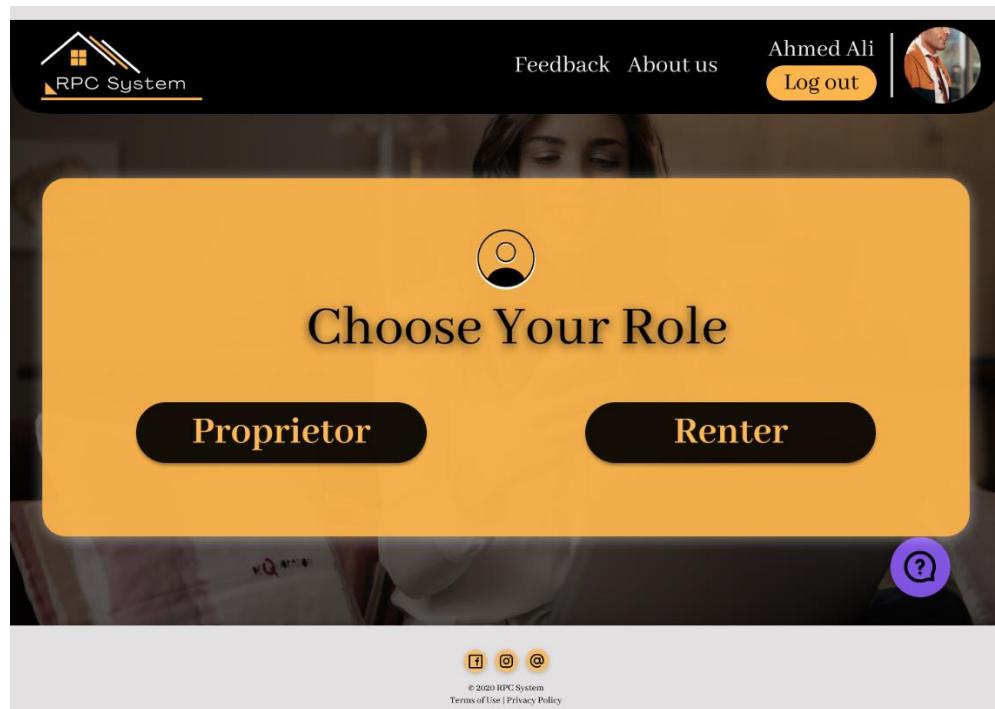


Figure 41: User Interface of Role Selection Screen

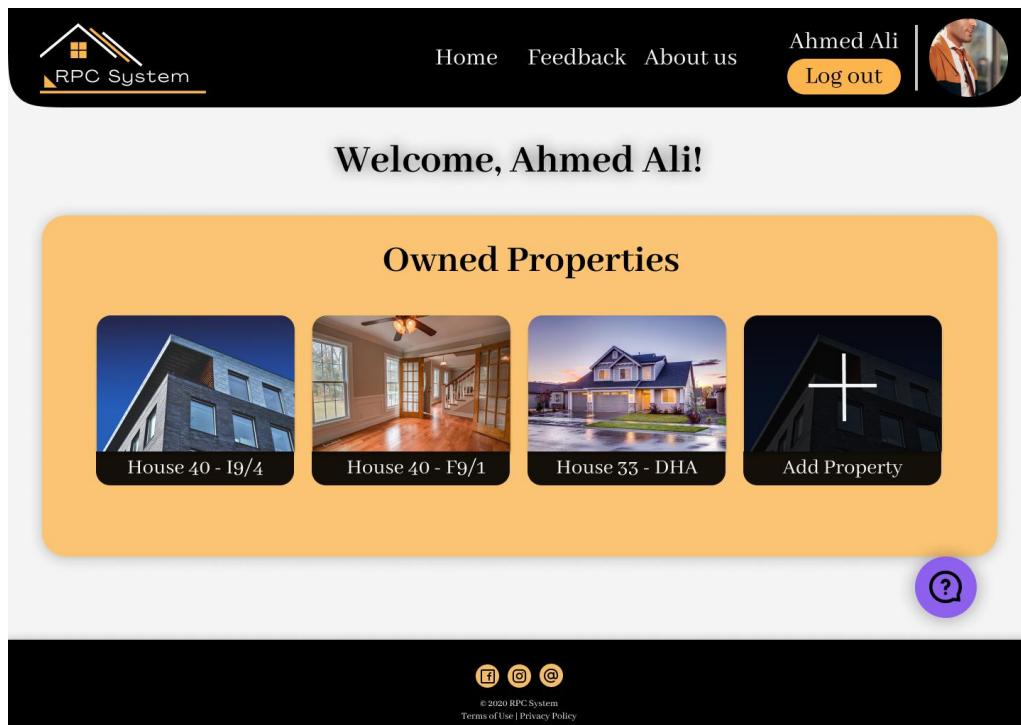


Figure 42: User Interface of Proprietor Menu

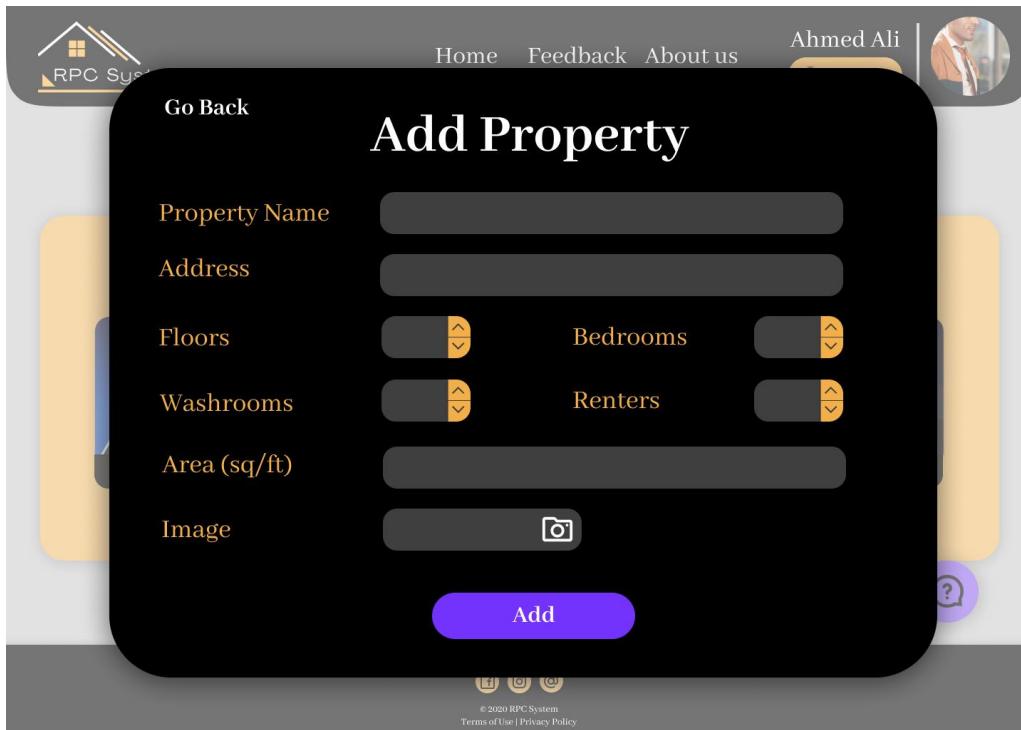


Figure 43: User Interface of Add property Menu

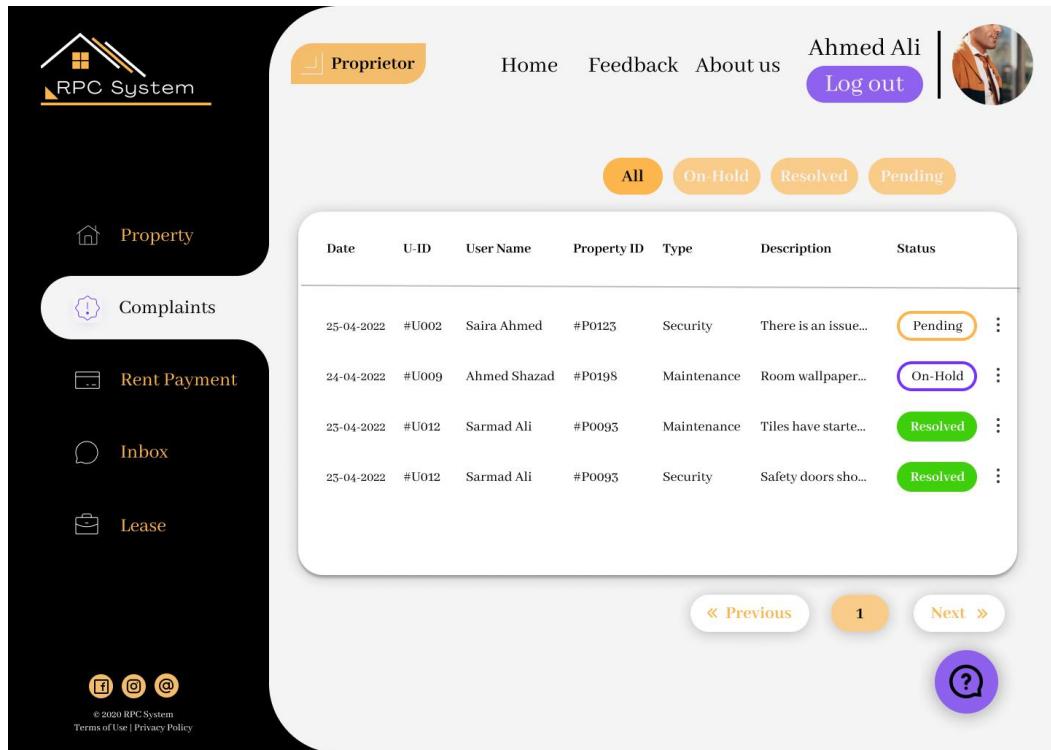


Figure 44: User Interface of Complaint Menu (Proprietor)

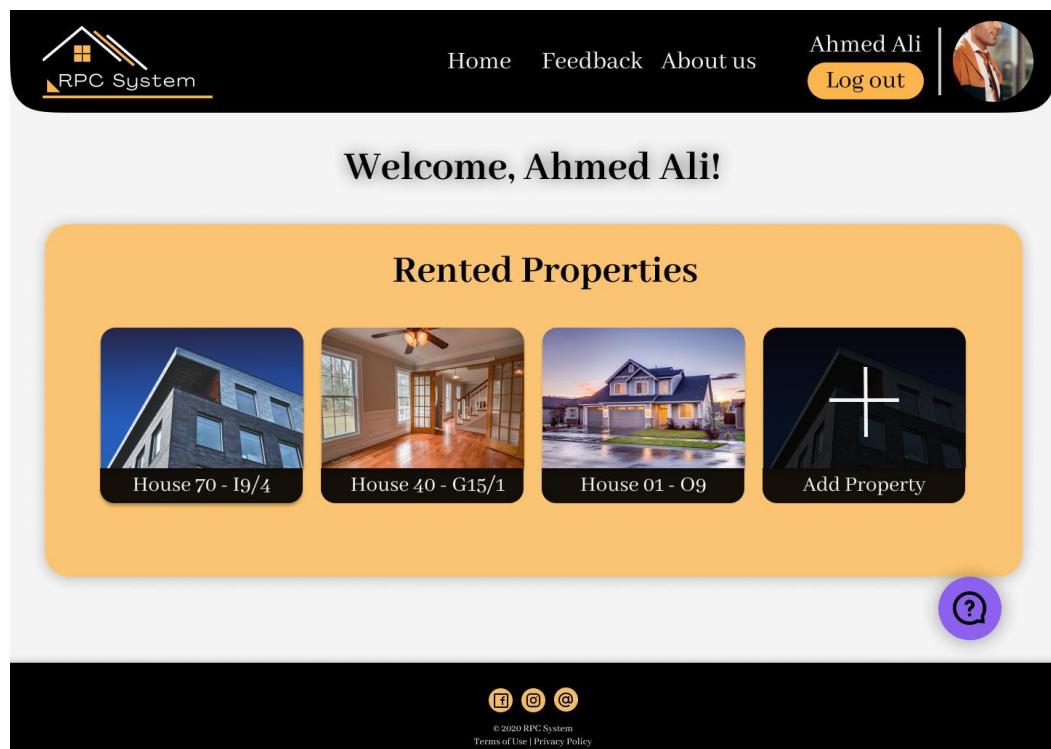


Figure 45: User Interface of Renter Menu

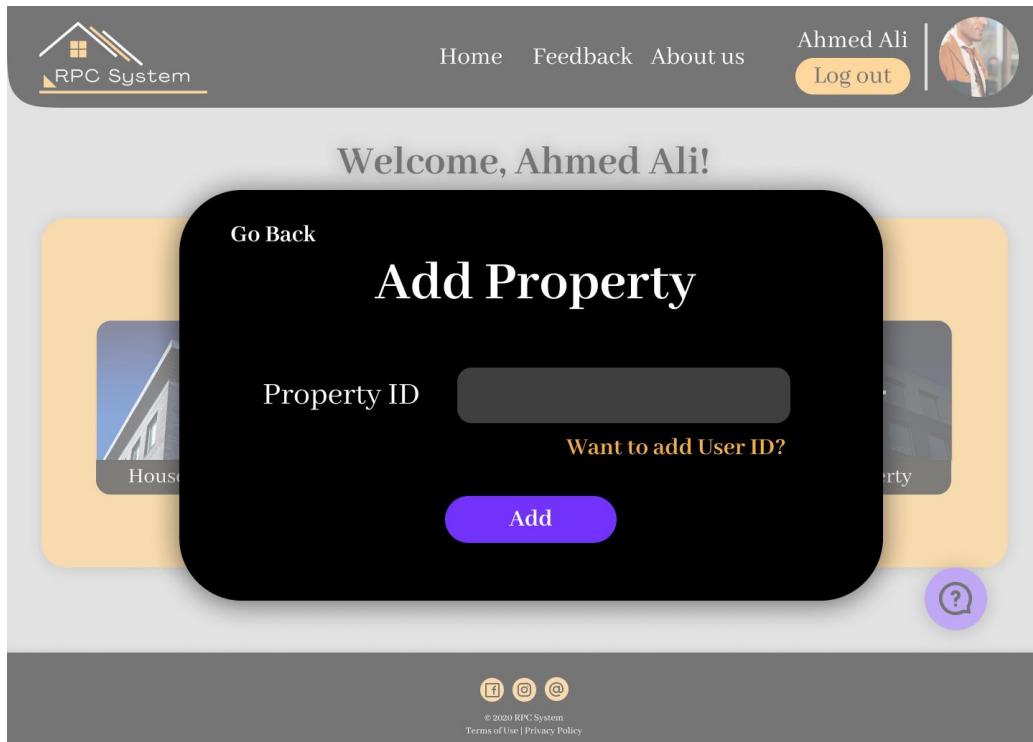


Figure 46: User Interface of Adding Properties through Property Id (Renter)

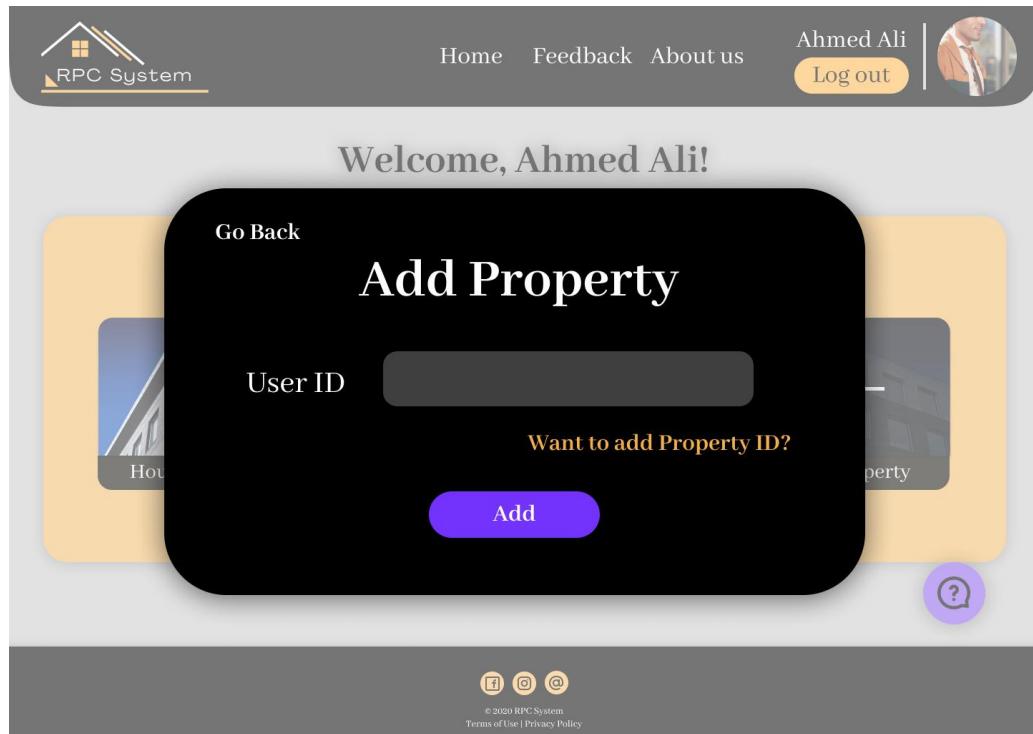


Figure 47: User Interface of Adding Properties through User Id (Renter)

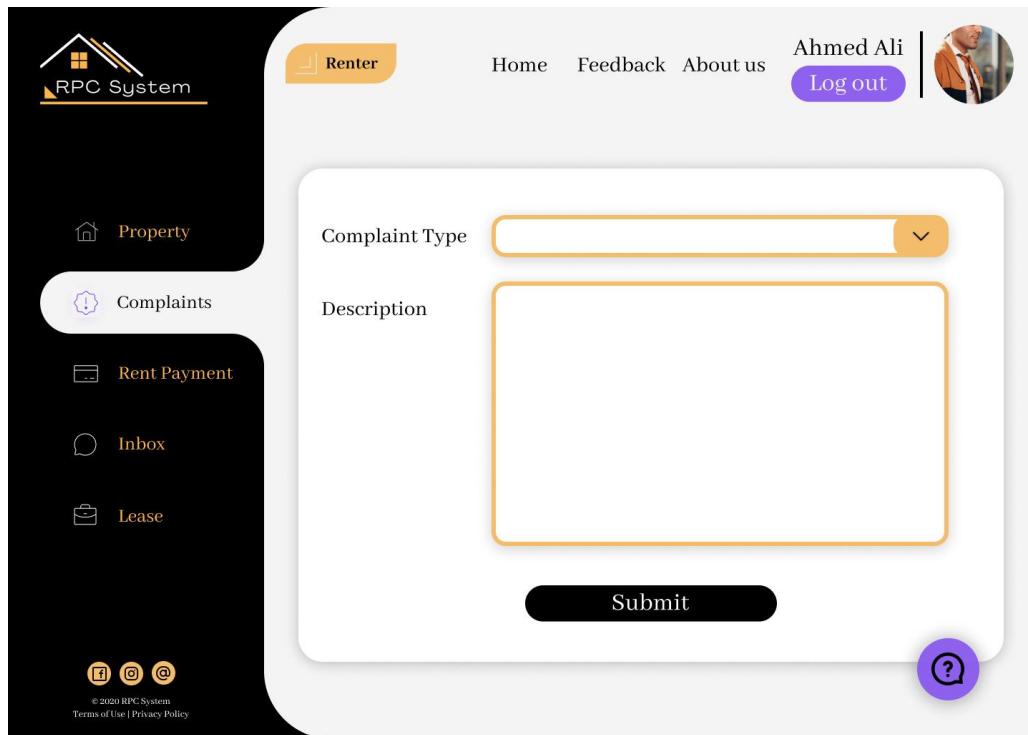


Figure 48: User Interface of Renter Complaint Menu

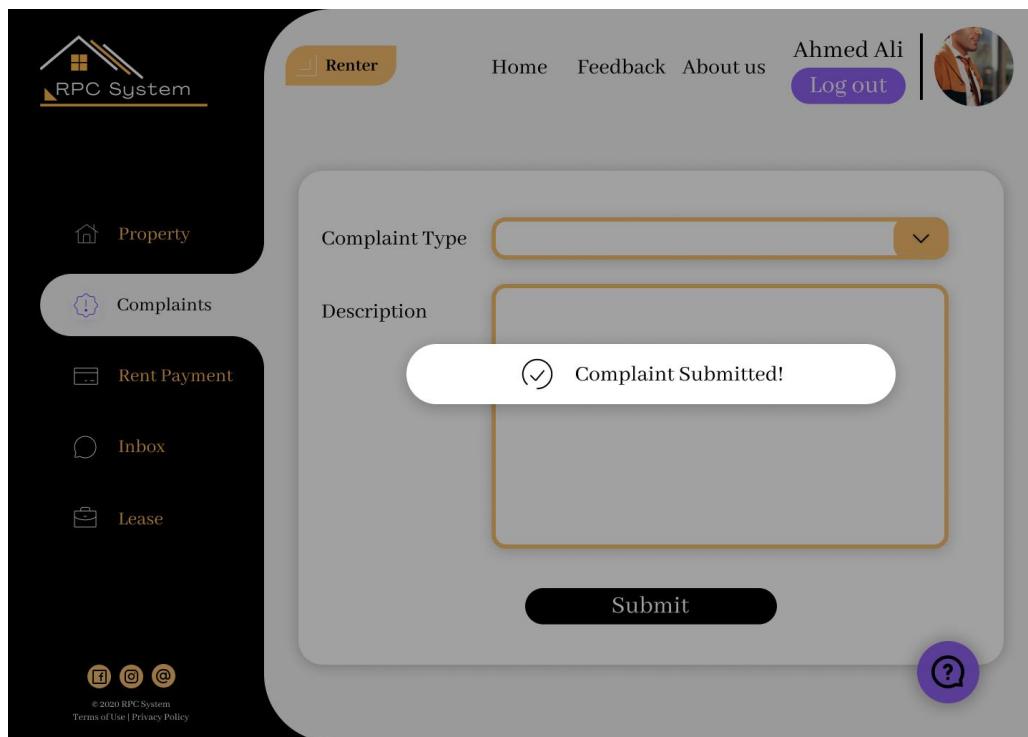


Figure 49: User Interface of Complaint Submission (Renter)

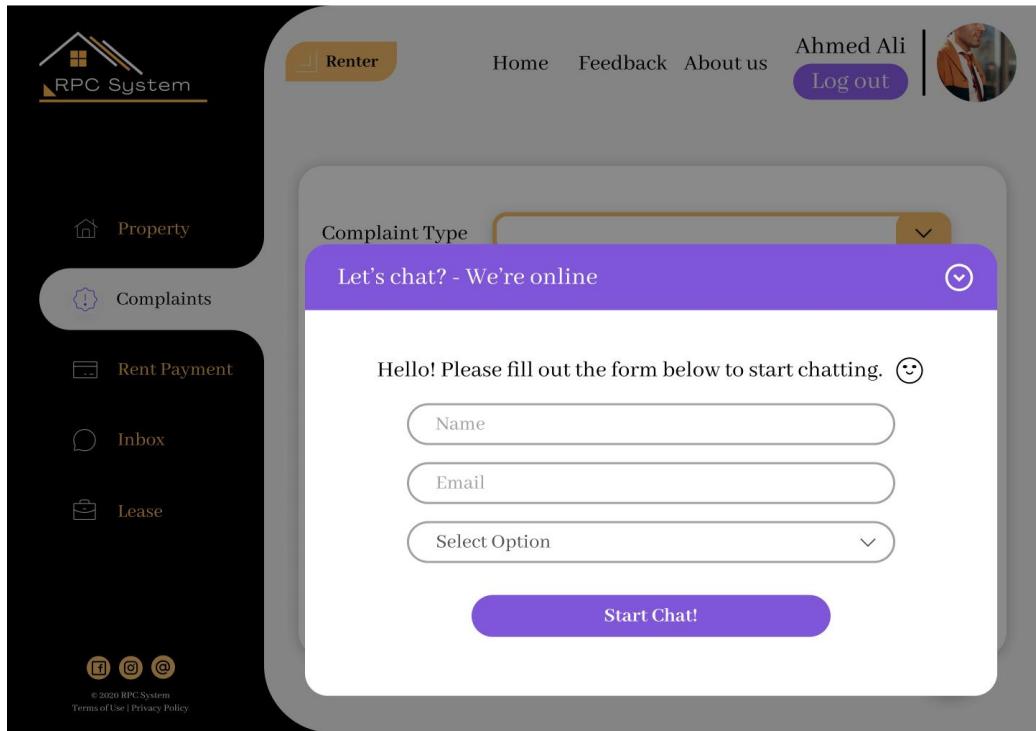


Figure 50: User Interface of FAQ menu

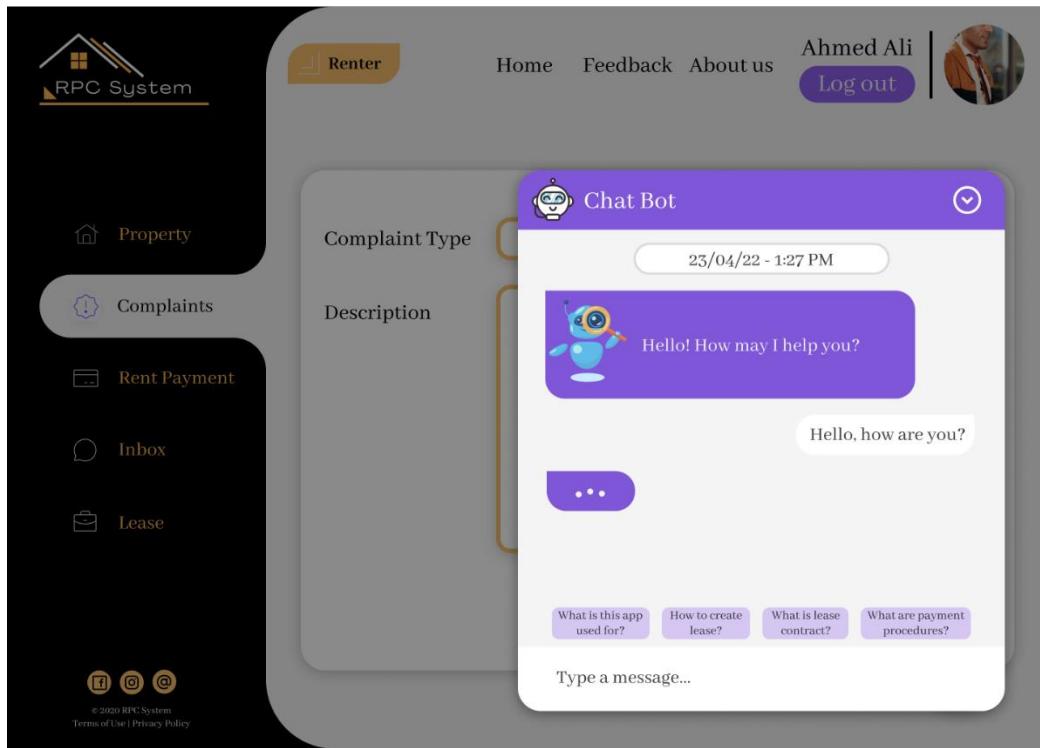


Figure 51: User Interface of Chat with Bot

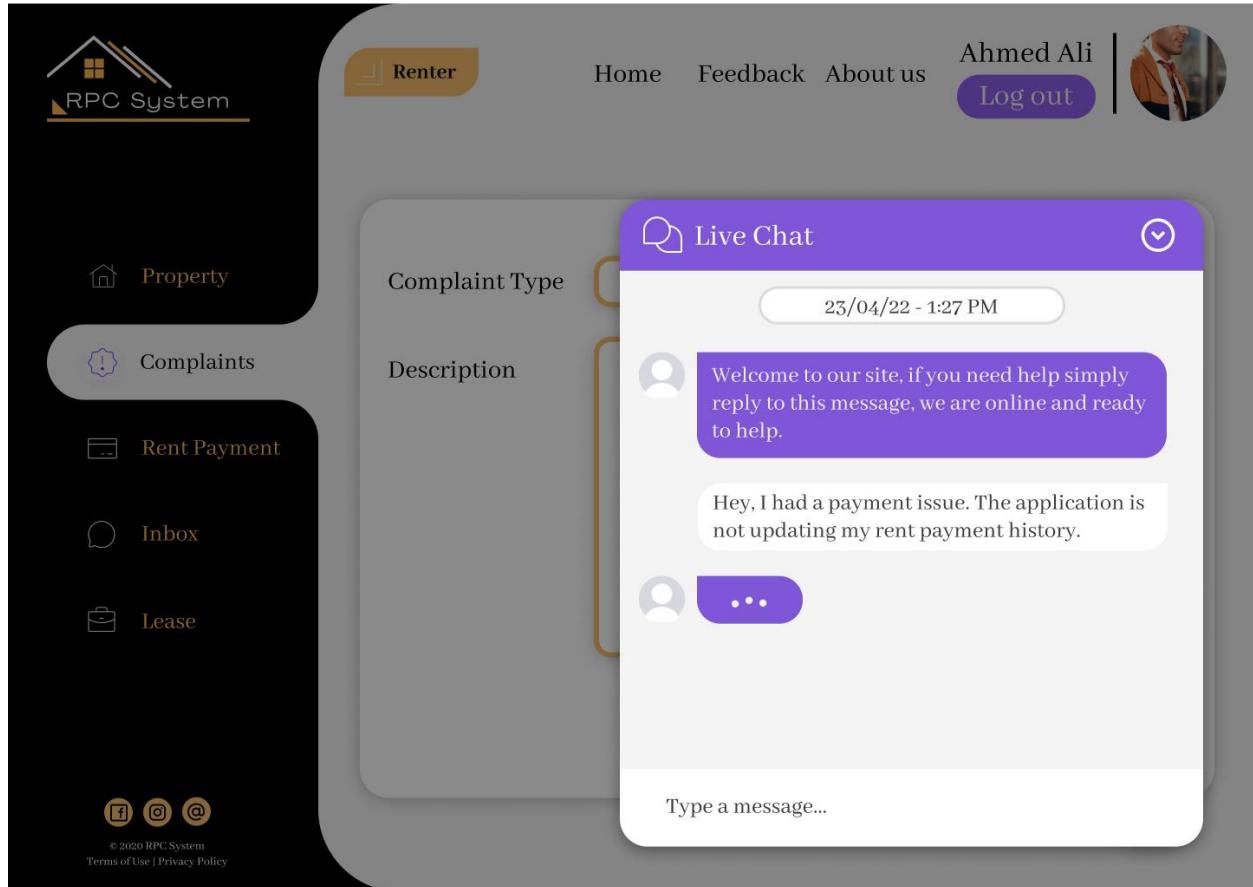


Figure 52: User Interface of Live Chat

6. Testing and Evaluation

This chapter discusses the verification and validation process of the overall working of the implemented system. Using Grey Box testing technique, each project use case is thoroughly tested with carefully curated test cases to ensure the product is producing desired results. These test cases are documented in this section to give an overview of the testing phase along with the contrast between expected and actual results of these test cases. Grey Box testing was opted because of its property of efficient and thorough testing at both internal and external level of product.

6.1. Verification

All the processes cycle of the system are tested thoroughly to make sure that the system is bug free as much as possible. All the prepared documents, design and code are tested to check whether they are according to the requirements or not. All the security measures are evaluated in detail. The data from each user needs to be verified because the app has to deal with it.

6.2. Validation

The key objective of validation is to make sure that whether the product being built is according to the user requirements or not. So, validation of the system is done make sure that all the built product is in accordance with the user requirements.

6.3. Usability Testing

The features and uses of the system are checked whether they are user-friendly or not. The usability will be tested to make sure that the end-user can understand the product easily. As the system will be used by both technical and non-technical users, they should not face any major bugs.

6.4. Module/Unit Testing

It is the most basic level of testing. So, each functional requirement is tested individually by the group members. The functional requirements of each module are tested. Most of the bugs can be found in this phase because each basic functional requirement is checked thoroughly.

6.5. Integration Testing

Integration level testing is done to make sure that the modules and major components of the system are working properly as an entire entity. Testing is done to detect all kinds of major and minor bugs.

6.6. System Testing

In system level testing, the system is tested as a whole after the implementation of each module. Unusual operations are performed to find any bugs or failure of the system. The functionalities of the system are tested to ensure that the system's result satisfy the user requirements.

6.7. Acceptance Testing

The software is deployed to the users or clients, and it will be checked whether the system is accepted by them or not. It is also made sure that the system is fulfilling their requirements.

6.8. Manual Testing

The software is used by testers, clients and customers manually without any access to the source code. Each module and functionality is tested thoroughly to check whether it produces the desired results at the end user level.

6.9. Test Cases

The testing of our system are discussed in detail below:

6.9.1. Unit Testing

- *Module 1: Profile Management*

UC-ID	Use Case Name
UC-1.1	Sign up
UC-1.2	Log in
UC-1.3	Log out
UC-1.4	Forget Password
UC-1.5	View Profile
UC-1.6	Edit Profile
UC-1.7	View Profile Display
UC-1.8	Change Profile Display
UC-1.9	Delete Account
UC-1.10	Disable Account
UC-1.11	Generate User Id
UC-1.12	Generate Captcha Images
UC-1.13	Generate Pop-Up Ads

1. TC-1.1 against UC-1.1

Test Id:	TC-1.1	Test Case Designed by:	Hunia Nadeem
Test Case Title:	Verify Sign Up	Test Case Executed by:	Idrees Ghazi
Module Name:	Profile Management	Test Case Execution Date:	07-06-2022
Test Data:	First Name: RPC Last Name: System Email: RpcSystem@gmail.com , CNIC:61101-2345234-3, New Phone Number: 0322-3234125, Password: Rpc_123	Priority:	High
Precondition:		User should have an authentic and valid email address and contact number.	
Steps /Action		System Response	
<ol style="list-style-type: none"> 1. Load the web URL or Open the application APK. 2. Welcome screen opens 3. Click on Get Started Button. 4. Click on Sign up button at the bottom. 5. Add user credentials by providing credible information. 6. Click on the Signup button at the bottom. 		<ol style="list-style-type: none"> 1. App gets loaded. 2. Welcome screen is displayed 3. Login Screen Opens. 4. The Sign up screen appears. 5. Credentials are entered in the text fields. 6. User Account is created and Sign up is completed. 	
Expected Result:		Account is created.	
Actual Result:		The user account has been created successfully.	
Status:		Pass	

2. TC-1.2 against UC-1.2

Test Id:	TC-1.2	Test Case Designed by:	Hunia Nadeem
Test Case Title:	Verify Log In	Test Case Executed by:	Idrees Ghazi
Module Name:	Profile Management	Test Case Execution Date:	07-06-2022
Test Data:	Email: RpcSystem@gmail.com Password: Rpc_123	Priority:	High
Precondition:		User must have created an account before.	
Steps /Action		System Response	
3. Load the web URL or Open the application apk. 4. The user will click the Get Started Button. 5. The user will enter an authentic email. 6. The user will enter the password used at the time of account creation. 7. Then user will click on the log in button.		1. App gets opened. 2. The login screen will appear. 3. Email is entered in the text box. 4. Password is entered in the text box. 5. The user will log in to the account if the credentials entered are correct.	
Expected Result:		User logs into the account.	
Actual Result:		The user has logged into the account successfully.	
Status:		Pass	

3. TC-1.3 against UC-1.3

Test Id:	TC-1.3	Test Case Designed by:	Hunia Nadeem
Test Case Title:	Verify Log Out	Test Case Executed by:	Idrees Ghazi
Module Name:	Profile Management	Test Case Execution Date:	07-06-2022
Test Data:	Logging out of the user's account	Priority:	High
Precondition:		User must be logged in an account.	
Steps /Action		System Response	
1. The user will click on the three-bar button at the top. 2. The user will click on the log out button.		1. The side pane menu will open. 2. The user will log out of the system.	
Expected Result:		User logs out from the account.	
Actual Result:		The user has logged out from the account successfully.	
Status:		Pass	

4. TC-1.4 against UC-1.4

Test Id:	TC-1.4	Test Case Designed by:	Hunia Nadeem
Test Case Title:	Verifying Forget Password	Test Case Executed by:	Idrees Ahmed Ghazi
Module Name:	Profile Management	Test Case Execution Date:	07-06-2022
Test Data:	Email: RpcSystem@gmail.com New Password: Rpc1_123	Priority:	Medium
Precondition:		User must have signed up for an account.	
Steps /Action		System Response	
1. Load the web URL or Open the application apk. 2. Click on the Get Started Button. 3. Click on forget password button. 4. Add new password 5. Confirm new password. 6. Click Next.		1. The application opens. 2. The login screen appears on the front. 3. The Forget Password menu opens to add a new password. 4. New password is added. 5. New password is confirmed 6. The password has been updated	
Expected Result:		The password has been updated	
Actual Result:		After the execution of this Use Case, the password has been updated successfully.	
Status:		Pass	

5. TC-1.5 against UC-1.5

Test Id:	TC-1.5	Test Case Designed by:	Hunia Nadeem
Test Case Title:	Verifying View Profile	Test Case Executed by:	Idrees Ahmed Ghazi
Module Name:	Profile Management	Test Case Execution Date:	07-06-2022
Test Data:	Email: RpcSystem@gmail.com Password: Rpc_123	Priority:	Medium
Precondition:		User must be logged in to their account.	
Steps /Action		System Response	
1. Load the web URL or Open the application apk. 2. Click on the Get Started Button. 3. The user will log into the account. 4. The user will click on the view profile button or profile picture at the top.		1. The application opens. 2. The login screen appears on the front 3. The user will be logged into the account. 4. The user will be able to view the profile.	
Expected Result:		The user will be able to view profile.	
Actual Result:		After the execution of this Use Case, the profile is viewed successfully with all the required information to be seen.	
Status:		Pass	

6. TC-1.6 against UC-1.6

Test Id:	TC-1.6	Test Case Designed by:	Hunia Nadeem
Test Case Title:	Verifying Edit Profile	Test Case Executed by:	Idrees Ahmed Ghazi
Module Name:	Profile Management	Test Case Execution Date:	07-06-2022
Test Data:	Email: RpcSystem@gmail.com Password: Rpc_123 New Phone Number: 0322-3234125	Priority:	Medium
Precondition:		User must be logged in to their account.	
Steps /Action		System Response	
1. Open the application. 2. Login to the account. 3. Go to view profile. 4. Click on edit profile. 5. OTP will be generated and sent to the user's contact or email.		1. Application is opened 2. User is logged in. 3. User's profile is viewed. 4. Edit profile menu will open. 5. On correct input of OTP, successful data change message will be displayed to the user.	
Expected Result:		The user will be able edit profile.	
Actual Result:		After the execution of this Use Case, the profile can be edited.	
Status:		Pass	

7. TC-1.7 against UC-1.7

Test Id:	TC-1.7	Test Case Designed by:	Hunia Nadeem
Test Case Title:	Verifying View Profile Display	Test Case Executed by:	Idrees Ahmed Ghazi
Module Name:	Profile Management	Test Case Execution Date:	07-06-2022
Test Data:	Email: RpcSystem@gmail.com Password: Rpc_123	Priority:	Medium
Precondition:		User must be logged in to their account.	
Steps /Action		System Response	
1. Open the application. 2. Login to the account. 3. Go to view profile. 4. The user will click on the small picture displayed.		1. Application is opened 2. User is logged in. 3. User's profile is viewed. 4. The user will view the profile picture.	
Expected Result:		The user will be able to view profile picture.	
Actual Result:		After the execution of this Use Case, the user is able to view the profile picture.	
Status:		Pass	

8. TC-1.8 against UC-1.8

Test Id:	TC-1.8	Test Case Designed by:	Hunia Nadeem
Test Case Title:	Verifying Change profile display	Test Case Executed by:	Idrees Ahmed Ghazi
Module Name:	Profile Management	Test Case Execution Date:	07-06-2022
Test Data:	Email: RpcSystem@gmail.com Password: Rpc_123 New Phone Number: 0322-3234125 New Profile Picture file.jpg	Priority:	Medium
Precondition:		User must be logged in to their account.	
Steps /Action		System Response	
1. Open the application. 2. Login to the account. 3. Go to view profile. 4. The user will click on the small picture displayed. 5. The user will click on change picture button. 6. The user will add a new picture. 7. The user will click Ok button.		1. Application is opened 2. User is logged in. 3. User's profile is viewed. 4. The user will view the profile picture. 5. The photos of the user from gallery will open. 6. The user's new profile picture will be loaded. 7. The user's profile picture will be changed.	
Expected Result:		The user will be able to change profile picture.	
Actual Result:		After the execution of this Use Case, the user is able to change the profile picture successfully.	
Status:		Pass	

9. TC-1.9 against UC-1.9

Test Id:	TC-1.9	Test Case Designed by:	Hunia Nadeem
Test Case Title:	Verifying Delete Account	Test Case Executed by:	Idrees Ahmed Ghazi
Module Name:	Profile Management	Test Case Execution Date:	07-06-2022
Test Data:	Email: RpcSystem@gmail.com Password: Rpc_123 CNIC: 61101-2345234-3	Priority:	High
Precondition:		User must have created the account before.	
Steps /Action		System Response	
1. Open the application. 2. Login to the account. 3. Go to view profile. 4. The user will click on the delete account. 5. The user will provide a reason for their choice. 6. The user will enter password and CNIC for their verification. 7. The user will click Ok button.		1. Application is opened 2. User is logged in. 3. User's profile is viewed. 4. The user will be able to view the delete account menu. 5. The user's reason will be added. 6. The user's password and CNIC will be verified. 7. The user's account will be deleted if the credentials entered are correct.	
Expected Result:		The user will be able to delete the account	
Actual Result:		After the execution of this Use Case, the user is able to delete the account successfully.	
Status:		Pass	

10. TC-1.10 against UC-1.10

Test Id:	TC-1.10	Test Case Designed by:	Hunia Nadeem
Test Case Title:	Verifying Disable Account	Test Case Executed by:	Idrees Ahmed Ghazi
Module Name:	Profile Management	Test Case Execution Date:	07-06-2022
Test Data:	Email: RpcSystem@gmail.com Password: Rpc_123 CNIC: 61101-2345234-3	Priority:	High
Precondition:		User must have signed up for an account.	
Steps /Action		System Response	
1. Open the application. 2. Login to the account. 3. Go to view profile. 4. The user will click on the disable account button. 5. The user will provide a reason for their choice. 6. The user will enter password and CNIC for their verification. 7. The user will click Ok button.		1. Application is opened 2. User is logged in. 3. User's profile is viewed. 4. The user will be able to view the disable account menu. 5. The user's reason will be added. 6. The user's password and CNIC will be verified. 7. The user's account will be disabled if the credentials entered are correct.	
Expected Result:		The user will be able to disable the account	
Actual Result:		After the execution of this Use Case, the user is able to disable the account successfully.	
Status:		Pass	

11. TC-1.11 against UC-1.11

Test Id:	TC-1.11	Test Case Designed by:	Hunia Nadeem
Test Case Title:	Verifying Generate User Id	Test Case Executed by:	Idrees Ahmed Ghazi
Module Name:	Profile Management	Test Case Execution Date:	07-06-2022
Test Data:	First Name: RPC Last Name: System Email: RpcSystem@gmail.com CNIC: 61101-2345234-3 New Phone Number: 0322-3234125 Password: Rpc_123	Priority:	High
Precondition:		Account creation process is performed successfully.	
Steps /Action		System Response	
1. Click on the application. 2. Welcome screen opens 3. Click on Get Started Button. 4. Click on Sign up button at the bottom. 5. Add user credentials by providing credible information. 6. Click on the Signup button at the bottom.		1. App gets opened 2. Welcome screen is displayed 3. Login Screen Opens. 4. The Sign up screen appears. 5. Credentials are entered in the text fields. 6. User Account is created and Sign up is completed and user id for the user is generated	
Expected Result:		The user id for the account will be generated.	
Actual Result:		After the execution of this Use Case, the user Id for the account is generated.	
Status:		Pass	

12. TC-1.12 against UC-1.12

Test Id:	TC-1.12	Test Case Designed by:	Hunia Nadeem
Test Case Title:	Verifying Generate Captcha Images	Test Case Executed by:	Idrees Ahmed Ghazi
Module Name:	Profile Management	Test Case Execution Date:	07-06-2022
Test Data:	Email: RpcSystem@gmail.com Password: Rpc_123	Priority:	High
Precondition:		User wants to log-in to the application	
Steps /Action		System Response	
1. The user will open the app. 2. The user will click the Get Started Button. 3. The user will enter an authentic email. 4. The user will enter the password used at the time of account creation. 5. The user will click on the Not a Robot Button. 6. The user will select the required images.		1. App gets opened. 2. The login screen will appear. 3. Email is entered in the text box. 4. Password is entered in the text box. 5. The Captcha images will be generated, 6. The user will be able to submit the images and if they are correct then the user is verified.	
Expected Result:		The Captcha Images for user verification will be generated.	
Actual Result:		After the execution of this Use Case, The Captcha Images for user verification are generated successfully.	
Status:		Pass	

13. TC-1.13 against UC-1.13

Test Id:	TC-1.13	Test Case Designed by:	Hunia Nadeem
Test Case Title:	Verifying Generate Pop-Up Ads	Test Case Executed by:	Idrees Ahmed Ghazi
Module Name:	Profile Management	Test Case Execution Date:	07-06-2022
Test Data:	Email: RpcSystem@gmail.com Password: Rpc_123	Priority:	High
Precondition:		User wants to log-in to the application	
Steps /Action		System Response	
1. The user will open the app. 2. The user will click the Get Started Button. 3. The user will enter an authentic email. 4. The user will enter the password used at the time of account creation. 5. Then user will click on the log in button.		1. App gets opened. 2. The login screen will appear. 3. Email is entered in the text box. 4. Password is entered in the text box. 5. The user will log in to the account if the credentials entered are correct and pop up ads will be generated.	
Expected Result:		The Pop-up ads for monetization of the app will be generated.	
Actual Result:		After the execution of this Use Case, the Pop-up ads for monetization of the app are generated successfully.	
Status:		Pass	

➤ **Module 2: User Interaction Chat Box**

UC-ID	Use Case Name
UC-2.1	View Message
UC-2.2	Send Message
UC-2.3	Mute Message
UC-2.4	Unmute Message
UC-2.5	Pin Message
UC-2.6	Make Voice Call
UC-2.7	Redirect to Whatsapp
UC-2.8	Search Text Message
UC-2.9	Search User
UC-2.10	View Chat Media
UC-2.11	Clear Chat
UC-2.12	Unpin Message

1. TC-2.1 against UC-2.1

Test Id:	TC-2.1	Test Case Designed by:	Idrees Ahmed Ghazi
Test Case Title:	Check Message Visibility	Test Case Executed by:	Hunia Nadeem
Module Name:	User Interaction Chat Box	Test Case Execution Date:	13-06-2022
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123	Priority:	High
Steps /Action	System Response		
1. Load the web URL or click on mobile application icon. 2. Select Get Started. 3. Enter registered email to log in. 4. Enter registered password to login. 5. Press Login Button. 6. Select Renter/Proprietor as your Role 7. Select the property that initiated the conversation. 8. Select Message from the displayed menu	1. Main application homepage will open. 2. Log in screen will open. 3. Email is accepted. 4. Password is accepted 5. Account is logged in. Role Selection Page is displayed. 6. Renter Portal is opened. 7. Detailed property portal will open. 8. A window will open list of conversations that took place between the renters and proprietor of that property.		
Expected Result:	Messages to be viewed successfully.		
Actual Result:	Message viewed successfully.		
Status:	Pass		

2. TC-2.2 against UC-2.2

Test Id:	TC-2.2	Test Case Designed by:	Idrees Ahmed Ghazi
Test Case Title:	Verify Message Transmission	Test Case Executed by:	Hunia Nadeem
Module Name:	User Interaction Chat Box	Test Case Execution Date:	13-06-2022
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123	Priority:	High
Steps /Action		System Response	
1. Load the web URL or click on mobile application icon. 2. Select Get Started. 3. Enter registered email to log in. 4. Enter registered password to login. 5. Press Login Button. 6. Select Renter/Proprietor as your Role 7. Select the property that initiated the conversation. 8. Select Message from the displayed menu 9. Select the chat in which you want to send the message. 10. Type message in the message bar. 11. Press Send button.		1. Main application homepage will open. 2. Log in screen will open. 3. Email is accepted. 4. Password is accepted 5. Account is logged in. Role Selection Page is displayed. 6. Renter Portal is opened. 7. Detailed property portal will open. 8. A window will open list of conversations that took place between the renters and proprietor of that property. 9. Chat opens. 10. Message is displayed in the message bar. 11. Message is sent to the receiver.	
Expected Result:		Messages to be sent & received at the receiving end successfully.	
Actual Result:		Message sent & received successfully.	
Status:		Pass	

3. TC-2.3 against UC-2.3

Test Id:	TC-2.3	Test Case Designed by:	Idrees Ahmed Ghazi
Test Case Title:	Verify Message is Mutable	Test Case Executed by:	Hunia Nadeem
Module Name:	User Interaction Chat Box	Test Case Execution Date:	13-06-2022
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123	Priority:	Low
Steps /Action		System Response	
1. Load the web URL or click on mobile application icon. 2. Select Get Started. 3. Enter registered email to log in. 4. Enter registered password to login. 5. Press Login Button. 6. Select Renter/Proprietor as your Role 7. Select the property that initiated the conversation. 8. Select Message from the displayed menu 9. Right click or click the 3 dot button of the chat you want to mute. 10. Select Mute Option		1. Main application homepage will open. 2. Log in screen will open. 3. Email is accepted. 4. Password is accepted 5. Account is logged in. Role Selection Page is displayed. 6. Renter Portal is opened. 7. Detailed property portal will open. 8. A window will open list of conversations that took place between the renters and proprietor of that property. 9. Popup appears. 10. Chat is muted.	
Expected Result:		When message is received, no notification will pop up to indicate the message came in.	
Actual Result:		No notification generated on message receival of the muted chat successfully.	
Status:		Pass	

4. TC-2.4 against UC-2.4

Test Id:	TC-2.4	Test Case Designed by:	Idrees Ahmed Ghazi
Test Case Title:	Verify Message is Unmutable	Test Case Executed by:	Hunia Nadeem
Module Name:	User Interaction Chat Box	Test Case Execution Date:	13-06-2022
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123	Priority:	Low
Steps /Action		System Response	
1. Load the web URL or click on mobile application icon. 2. Select Get Started. 3. Enter registered email to log in. 4. Enter registered password to login. 5. Press Login Button. 6. Select Renter/Proprietor as your Role 7. Select the property that initiated the conversation. 8. Select Message from the displayed menu 9. Right click or click the 3 dot button of the chat you want to unmute. 10. Select Unmute Option.		1. Main application homepage will open. 2. Log in screen will open. 3. Email is accepted. 4. Password is accepted 5. Account is logged in. Role Selection Page is displayed. 6. Renter Portal is opened. 7. Detailed property portal will open. 8. A window will open list of conversations that took place between the renters and proprietor of that property. 9. Popup appears. 10. Chat is unmuted.	
Expected Result:		When message is received, now notification will pop up to indicate the message came in.	
Actual Result:		Now notification generated on message receival of the unmuted chat successfully.	
Status:		Pass	

5. TC-2.5 against UC-2.5

Test Id:	TC-2.5	Test Case Designed by:	Idrees Ahmed Ghazi
Test Case Title:	Check Message is Pinnable	Test Case Executed by:	Hunia Nadeem
Module Name:	User Interaction Chat Box	Test Case Execution Date:	13-06-2022
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123	Priority:	Low
Steps /Action		System Response	
1. Load the web URL or click on mobile application icon. 2. Select Get Started. 3. Enter registered email to log in. 4. Enter registered password to login. 5. Press Login Button. 6. Select Renter/Proprietor as your Role 7. Select the property that initiated the conversation. 8. Select Message from the displayed menu 9. Right click or click the 3 dot button of the chat you want to pin. 10. Select Pin Option.		1. Main application homepage will open. 2. Log in screen will open. 3. Email is accepted. 4. Password is accepted 5. Account is logged in. Role Selection Page is displayed. 6. Renter Portal is opened. 7. Detailed property portal will open. 8. A window will open list of conversations that took place between the renters and proprietor of that property. 9. Popup appears. 10. Chat is pinned.	
Expected Result:		When message is pinned, it is displayed top on list.	
Actual Result:		Message is successfully pinned.	
Status:		Pass	

6. TC-2.6 against UC-2.6

Test Id:	TC-2.6	Test Case Designed by:	Idrees Ahmed Ghazi
Test Case Title:	Verify Voice Calling	Test Case Executed by:	Hunia Nadeem
Module Name:	User Interaction Chat Box	Test Case Execution Date:	13-06-2022
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123	Priority:	Medium
Steps /Action		System Response	
1. Click on mobile application icon. 2. Select Get Started. 3. Enter registered email to log in. 4. Enter registered password to login. 5. Press Login Button. 6. Select Renter/Proprietor as your Role 7. Select the property that initiated the conversation. 8. Select Message from the displayed menu 9. Select the chat through which you want to make a call. 10. Click on Call icon button.		1. Main application homepage will open. 2. Log in screen will open. 3. Email is accepted. 4. Password is accepted 5. Account is logged in. Role Selection Page is displayed. 6. Renter Portal is opened. 7. Detailed property portal will open. 8. A window will open list of conversations that took place between the renters and proprietor of that property. 9. Chat opens. 10. You will be redirected to mobile calling app.	
Expected Result:		User will be redirected to mobile calling app with the callee's number on the number area.	
Actual Result:		Call is placed successfully.	
Status:		Pass	

7. TC-2.7 against UC-2.7

Test Id:	TC-2.7	Test Case Designed by:	Idrees Ahmed Ghazi
Test Case Title:	Verify Redirection to Whatsapp	Test Case Executed by:	Hunia Nadeem
Module Name:	User Interaction Chat Box	Test Case Execution Date:	13-06-2022
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123	Priority:	Medium
Steps /Action		System Response	
1. Load the web URL or Click on mobile application icon. 2. Select Get Started. 3. Enter registered email to log in. 4. Enter registered password to login. 5. Press Login Button. 6. Select Renter/Proprietor as your Role 7. Select the property that initiated the conversation. 8. Select Message from the displayed menu 9. Select the chat through which you want to redirect to Whatsapp. 10. Click on Whatsapp icon button.		1. Main application homepage will open. 2. Log in screen will open. 3. Email is accepted. 4. Password is accepted 5. Account is logged in. Role Selection Page is displayed. 6. Renter Portal is opened. 7. Detailed property portal will open. 8. A window will open list of conversations that took place between the renters and proprietor of that property. 9. Chat opens. 10. You will be redirected to WhatsApp application.	
Expected Result:		User will be redirected to Whatsapp.	
Actual Result:		Successfully redirected to Whatsapp.	
Status:		Pass	

8. TC-2.8 against UC-2.8

Test Id:	TC-2.8	Test Case Designed by:	Idrees Ahmed Ghazi
Test Case Title:	Verify Text Message is Searchable.	Test Case Executed by:	Hunia Nadeem
Module Name:	User Interaction Chat Box	Test Case Execution Date:	13-06-2022
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123, Test message: "hello, testing!"	Priority:	Low
Steps /Action		System Response	
1. Load the web URL or Click on mobile application icon. 2. Select Get Started. 3. Enter registered email to log in. 4. Enter registered password to login. 5. Press Login Button. 6. Select Renter/Proprietor as your Role 7. Select the property that initiated the conversation. 8. Select Message from the displayed menu 9. Select the chat in which you want to search the message. 10. Send the test message "hello, testing!" 11. Click on Search icon button. 12. Type "hello, testing!"		1. Main application homepage will open. 2. Log in screen will open. 3. Email is accepted. 4. Password is accepted 5. Account is logged in. Role Selection Page is displayed. 6. Renter Portal is opened. 7. Detailed property portal will open. 8. A window will open list of conversations that took place between the renters and proprietor of that property. 9. Chat opens. 10. Test message is sent. 11. Text area appears where you can type the message for searching. 12. Message is displayed on screen if searching is successful.	
Expected Result:		Searched message will be filtered and displayed in chat box.	
Actual Result:		Message is successfully searched and displayed.	
Status:		Pass	

9. TC-2.9 against UC-2.9

Test Id:	TC-2.9	Test Case Designed by:	Idrees Ahmed Ghazi
Test Case Title:	Verify User is Searchable	Test Case Executed by:	Hunia Nadeem
Module Name:	User Interaction Chat Box	Test Case Execution Date:	13-06-2022
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123	Priority:	Low
Steps /Action		System Response	
1. Load the web URL or click on mobile application icon. 2. Select Get Started. 3. Enter registered email to log in. 4. Enter registered password to login. 5. Press Login Button. 6. Select Renter/Proprietor as your Role 7. Select the property that initiated the conversation. 8. Select Message from the displayed menu 9. Click on Search icon button. 10. Type user name/user ID in the search bar.		1. Main application homepage will open. 2. Log in screen will open. 3. Email is accepted. 4. Password is accepted 5. Account is logged in. Role Selection Page is displayed. 6. Renter Portal is opened. 7. Detailed property portal will open. 8. A window will open list of conversations that took place between the renters and proprietor of that property. 9. Text area appears where you can type the user name/ID for searching. 10. If user exists, their chat will be displayed.	
Expected Result:		Searched user will be filtered and displayed in list.	
Actual Result:		User is successfully searched.	
Status:		Pass	

10. TC-2.10 against UC-2.10

Test Id:	TC-2.10	Test Case Designed by:	Idrees Ahmed Ghazi
Test Case Title:	Verify Chat Media is Viewable.	Test Case Executed by:	Hunia Nadeem
Module Name:	User Interaction Chat Box	Test Case Execution Date:	13-06-2022
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123	Priority:	Low
Steps /Action		System Response	
1. Load the web URL or Click on mobile application icon. 2. Select Get Started. 3. Enter registered email to log in. 4. Enter registered password to login. 5. Press Login Button. 6. Select Renter/Proprietor as your Role 7. Select the property that initiated the conversation. 8. Select Message from the displayed menu 9. Select the chat you want to clear. 10. Click on 3 dot button. 11. Choose view chat media.		1. Main application homepage will open. 2. Log in screen will open. 3. Email is accepted. 4. Password is accepted 5. Account is logged in. Role Selection Page is displayed. 6. Renter Portal is opened. 7. Detailed property portal will open. 8. A window will open list of conversations that took place between the renters and proprietor of that property. 9. Chat opens. 10. Popup appears. 11. chat media pop up appears displaying all the chat media present in the chat.	
Expected Result:		Chat media is filtered out and displayed.	
Actual Result:		Chat media is successfully displayed.	
Status:		Pass	

11. TC-2.11 against UC-2.11

Test Id:	TC-2.11	Test Case Designed by:	Idrees Ahmed Ghazi
Test Case Title:	Verify Chat is clearable	Test Case Executed by:	Hunia Nadeem
Module Name:	User Interaction Chat Box	Test Case Execution Date:	13-06-2022
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123	Priority:	Low
Steps /Action		System Response	
12. Load the web URL or Click on mobile application icon. 13. Select Get Started. 14. Enter registered email to log in. 15. Enter registered password to login. 16. Press Login Button. 17. Select Renter/Proprietor as your Role 18. Select the property that initiated the conversation. 19. Select Message from the displayed menu 20. Select the chat you want to clear. 21. Click on 3 dot button. 22. Choose clear chat.		12. Main application homepage will open. 13. Log in screen will open. 14. Email is accepted. 15. Password is accepted 16. Account is logged in. Role Selection Page is displayed. 17. Renter Portal is opened. 18. Detailed property portal will open. 19. A window will open list of conversations that took place between the renters and proprietor of that property. 20. Chat opens. 21. Popup appears. 22. Chat messages are deleted on deleter's end.	
Expected Result:		Chat box is cleared on pressing the option.	
Actual Result:		Chat box got cleared.	
Status:		Pass	

12. TC-2.14 against UC-2.14

Test Id:	TC-2.14	Test Case Designed by:	Idrees Ahmed Ghazi
Test Case Title:	Check Message is Unpinnable	Test Case Executed by:	Hunia Nadeem
Module Name:	User Interaction Chat Box	Test Case Execution Date:	13-06-2022
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123	Priority:	Low
Steps /Action		System Response	
1. Load the web URL or click on mobile application icon. 2. Select Get Started. 3. Enter registered email to log in. 4. Enter registered password to login. 5. Press Login Button. 6. Select Renter/Proprietor as your Role 7. Select the property that initiated the conversation. 8. Select Message from the displayed menu 9. Right click or click the 3 dot button of the chat you want to pin. 10. Select Unpin Option.		1. Main application homepage will open. 2. Log in screen will open. 3. Email is accepted. 4. Password is accepted 5. Account is logged in. Role Selection Page is displayed. 6. Renter Portal is opened. 7. Detailed property portal will open. 8. A window will open list of conversations that took place between the renters and proprietor of that property. 9. Popup appears. 10. Chat is unpinned.	
Expected Result:		When message is unpinned, it is displayed not displayed on top of the list.	
Actual Result:		Message is successfully unpinned.	
Status:		Pass	

➤ **Module 3: Rent Payment Portal**

UC-ID	Use Case Name
UC-3.1	Pay Rent
UC-3.2	Generate Invoice
UC-3.3	Print Invoice
UC-3.4	View Payment History
UC-3.5	Save Card Details
UC-3.6	Request Payment Deadline Extension
UC-3.7	Respond to Payment Deadline Extension Request
UC-3.8	View Payment Status
UC-3.9	Edit Payment Details
UC-3.10	Buy Premium Account
UC-3.11	Open Payment Details
UC-3.12	Hide Pop-up Ads

1. TC-3.1 against UC-3.1

Test Id:	TC-3.1	Test Case Designed by:	Hunia Nadeem
Test Case Title:	Verify rent payment	Test Case Executed by:	Idrees Ghazi
Module Name:	Rent Payment Portal	Test Case Execution Date:	13-06-2022
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123 Acc. No: 3243 5367 5435 6432 CVC : 344 Expiry Date: 4/26 Address: House 33, DHA	Priority:	High
Steps /Action	System Response		
1. Load the web URL or click on mobile application icon. 2. Select Get Started. 3. Enter registered email to log in. 4. Enter registered password to login. 5. Press Login Button. 6. Select your Role. 7. Open payment menu. 8. Provide payment details and pay rent.	1. Main application homepage will open. 2. Log in screen will open. 3. Email is accepted. 4. Password is accepted 5. Account is logged in. Role Selection Page is displayed. 6. Role selected Portal is opened. 7. Payment menu is opened. 8. Rent is paid.		
Expected Result:	Rent is to be paid successfully.		
Actual Result:	Rent is paid successfully.		
Status:	Pass		

2. TC-3.2 against UC-3.2

Test Id:	TC-3.2	Test Case Designed by:	Hunia Nadeem
Test Case Title:	Verifying invoice generation	Test Case Executed by:	Idrees Ghazi
Module Name:	Rent Payment Portal	Test Case Execution Date:	13-06-2022
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123 Acc. No: 3243 5367 5435 6432 CVC : 344 Expiry Date: 4/26 Address: House 33, DHA	Priority:	Medium
Steps /Action		System Response	
1. Load the web URL or click on mobile application icon. 2. Select Get Started. 3. Enter registered email to log in. 4. Enter registered password to login. 5. Press Login Button. 6. Select your Role. 7. Open payment menu. 8. Provide payment details and pay rent. 9. Click on generate invoice.		1. Main application homepage will open. 2. Log in screen will open. 3. Email is accepted. 4. Password is accepted 5. Account is logged in. Role Selection Page is displayed. 6. Role selected Portal is opened. 7. Payment menu is opened. 8. Rent is paid. 9. Invoice is generated.	
Expected Result:		Payment invoice to be generated successfully.	
Actual Result:		Payment invoice is generated successfully.	
Status:		Pass	

4. TC-3.3 against UC-3.3

Test Id:	TC-3.3	Test Case Designed by:	Hunia Nadeem
Test Case Title:	Verifying printing invoice.	Test Case Executed by:	Idrees Ghazi
Module Name:	Rent Payment Portal	Test Case Execution Date:	13-06-2022
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123 Acc. No: 3243 5367 5435 6432 CVC : 344 Expiry Date: 4/26 Address: House 33, DHA	Priority:	Low
Steps /Action	System Response		
1. Load the web URL or click on mobile application icon. 2. Select Get Started. 3. Enter registered email to log in. 4. Enter registered password to login. 5. Press Login Button. 6. Select your Role. 7. Open payment menu. 8. Provide payment details and pay rent. 9. Click on generate invoice. 10. Click on Print Invoice	1. Main application homepage will open. 2. Log in screen will open. 3. Email is accepted. 4. Password is accepted 5. Account is logged in. Role Selection Page is displayed. 6. Role selected Portal is opened. 7. Payment menu is opened. 8. Rent is paid. 9. Invoice is generated. 10. Invoice is printed.		
Expected Result:	Payment invoice to be generated successfully.		
Actual Result:	Payment invoice is generated successfully.		
Status:	Pass		

5. TC-3.4 against UC-3.4

Test Id:	TC-3.4	Test Case Designed by:	Hunia Nadeem
Test Case Title:	Verifying the display of payments history.	Test Case Executed by:	Idrees Ghazi
Module Name:	Rent Payment Portal	Test Case Execution Date:	14-06-2022
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123	Priority:	Medium
Steps /Action		System Response	
1. Load the web URL or click on mobile application icon. 2. Select Get Started. 3. Enter registered email to log in. 4. Enter registered password to login. 5. Press Login Button. 6. Select your Role. 7. Open payment menu.		1. Main application homepage will open. 2. Log in screen will open. 3. Email is accepted. 4. Password is accepted 5. Account is logged in. Role Selection Page is displayed. 6. Role selected Portal is opened. 7. Payment menu is opened and the payment history is displayed in the form of table.	
Expected Result:		Payment history to be displayed successfully.	
Actual Result:		Payment history is displayed successfully.	
Status:		Pass	

6. TC-3.5 against UC-3.5

Test Id:	TC-3.5	Test Case Designed by:	Hunia Nadeem
Test Case Title:	Verifying the record of card details	Test Case Executed by:	Idrees Ghazi
Module Name:	Rent Payment Portal	Test Case Execution Date:	14-06-2022
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123 Acc. No: 3243 5367 5435 6432 CVC : 344 Expiry Date: 4/26 Address: House 33, DHA	Priority:	Low
Steps /Action		System Response	
1. Load the web URL or click on mobile application icon. 2. Select Get Started. 3. Enter registered email to log in. 4. Enter registered password to login. 5. Press Login Button. 6. Select your Role. 7. Open payment menu. 8. Pay rent 9. Click on save card details		1. Main application homepage will open. 2. Log in screen will open. 3. Email is accepted. 4. Password is accepted 5. Account is logged in. Role Selection Page is displayed. 6. Role selected Portal is opened. 7. Payment menu is opened, and the payment history is displayed in the form of table. 8. Rent is paid. 9. Card details are saved	
Expected Result:		Card details are to be saved successfully.	
Actual Result:		Card details are saved successfully	
Status:		Pass	

7. TC-3.6 against UC-3.6

Test Id:	TC-3.6	Test Case Designed by:	Hunia Nadeem
Test Case Title:	Verifying the request for payment deadline extension.	Test Case Executed by:	Idrees Ghazi
Module Name:	Rent Payment Portal	Test Case Execution Date:	14-06-22
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123	Priority:	Medium
Steps /Action		System Response	
1. Load the web URL or click on mobile application icon. 2. Select Get Started. 3. Enter registered email to log in. 4. Enter registered password to login. 5. Press Login Button. 6. Select Renter as your Role. 7. Open payment menu. 8. Click on requesting payment deadline extension		1. Main application homepage will open. 2. Log in screen will open. 3. Email is accepted. 4. Password is accepted 5. Account is logged in. Role Selection Page is displayed. 6. Renter Portal is opened. 7. Payment menu is opened, and the payment history is displayed in the form of table. 8. Request for payment deadline extension is sent.	
Expected Result:		Request for payment deadline extension to be sent successfully.	
Actual Result:		Request for payment deadline extension is sent successfully.	
Status:		Pass	

8. TC-3.7 against UC-3.7

Test Id:	TC-3.7	Test Case Designed by:	Hunia Nadeem
Test Case Title:	Verifying Responding to Payment deadline extension.	Test Case Executed by:	Idrees Ghazi
Module Name:	Rent Payment Portal	Test Case Execution Date:	14-06-22
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123	Priority:	Medium
Steps /Action		System Response	
1. Load the web URL or click on mobile application icon. 2. Select Get Started. 3. Enter registered email to log in. 4. Enter registered password to login. 5. Press Login Button. 6. Select Proprietor as your Role. 7. Open payment menu. 8. Click on respond to payment deadline extension		1. Main application homepage will open. 2. Log in screen will open. 3. Email is accepted. 4. Password is accepted 5. Account is logged in. Role Selection Page is displayed. 6. Proprietor Portal is opened. 7. Payment menu is opened. 8. Responding to payment deadline extension is done.	
Expected Result:		Responding to payment deadline extension to be done successfully.	
Actual Result:		Responding to payment deadline extension is done successfully.	
Status:		Pass	

9. TC-3.8 against UC-3.8

Test Id:	TC-3.8	Test Case Designed by:	Hunia Nadeem
Test Case Title:	Verify viewing payment status.	Test Case Executed by:	Idrees Ghazi
Module Name:	Rent Payment Portal	Test Case Execution Date:	14-06-22
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123	Priority:	Medium
Steps /Action		System Response	
1. Load the web URL or click on mobile application icon. 2. Select Get Started. 3. Enter registered email to log in. 4. Enter registered password to login. 5. Press Login Button. 6. Select your Role. 7. Open payment menu.		1. Main application homepage will open. 2. Log in screen will open. 3. Email is accepted. 4. Password is accepted 5. Account is logged in. Role Selection Page is displayed. 6. Selected Portal is opened. 7. Payment menu is opened, and payment status is shown.	
Expected Result:		Payment status to be shown in the payment menu.	
Actual Result:		Payment status is shown successfully.	
Status:		Pass	

10. TC-3.9 against UC-3.9

Test Id:	TC-3.9	Test Case Designed by:	Hunia Nadeem
Test Case Title:	Verifying editing payment details	Test Case Executed by:	Idrees Ghazi
Module Name:	Rent Payment Portal	Test Case Execution Date:	14-06-22
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123	Priority:	Medium
Steps /Action		System Response	
1. Load the web URL or click on mobile application icon. 2. Select Get Started. 3. Enter registered email to log in. 4. Enter registered password to login. 5. Press Login Button. 6. Select Proprietor as your Role. 7. Open payment menu. 8. Select edit payment details.		1. Main application homepage will open. 2. Log in screen will open. 3. Email is accepted. 4. Password is accepted 5. Account is logged in. Role Selection Page is displayed. 6. Proprietor Portal is opened. 7. Payment menu is opened, and payment status is shown. 8. Payment details can be edited	
Expected Result:		Payment details to be edited in the payment menu.	
Actual Result:		Payment details are edited successfully.	
Status:		Pass	

11. TC-3.10 against UC-3.10

Test Id:	TC-3.10	Test Case Designed by:	Hunia Nadeem
Test Case Title:	Verify activating premium account.	Test Case Executed by:	Idrees Ghazi
Module Name:	Rent Payment Portal	Test Case Execution Date:	14-06-22
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123 Acc. No: 3243 5367 5435 6432 CVC : 344 Expiry Date: 4/26 Address: House 33, DHA	Priority:	Medium
Steps /Action		System Response	
1. Load the web URL or click on mobile application icon. 2. Select Get Started. 3. Enter registered email to log in. 4. Enter registered password to login. 5. Press Login Button. 6. Select your Role. 7. Open payment menu. 8. Click on Buy premium.		1. Main application homepage will open. 2. Log in screen will open. 3. Email is accepted. 4. Password is accepted 5. Account is logged in. Role Selection Page is displayed. 6. Selected Role Portal is opened. 7. Payment Menu is opened 8. Premium Account is bought and activated.	
Expected Result:		Premium account to be activated successfully.	
Actual Result:		Premium account is activated successfully after buying premium account.	
Status:		Pass	

12. TC-3.11 against UC-3.11

Test Id:	TC-3.11	Test Case Designed by:	Hunia Nadeem
Test Case Title:	Verify opening payment details.	Test Case Executed by:	Idrees Ghazi
Module Name:	Rent Payment Portal	Test Case Execution Date:	14-06-22
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123	Priority:	Medium
Steps /Action		System Response	
1. Load the web URL or click on mobile application icon. 2. Select Get Started. 3. Enter registered email to log in. 4. Enter registered password to login. 5. Press Login Button. 6. Select your Role. 7. Open payment menu. 8. Click on individual payment transactions done before.		1. Main application homepage will open. 2. Log in screen will open. 3. Email is accepted. 4. Password is accepted 5. Account is logged in. Role Selection Page is displayed. 6. Selected Role Portal is opened. 7. Payment Menu is opened 8. Individual comprehensive payments details are shown.	
Expected Result:		Payment details to be viewed	
Actual Result:		Payment details are successfully viewed.	
Status:		Pass	

13. TC-3.12 against UC-3.12

Test Id:	TC-3.12	Test Case Designed by:	Hunia Nadeem
Test Case Title:	Verifying hiding pop-up ads	Test Case Executed by:	Idrees Ghazi
Module Name:	Rent Payment Portal	Test Case Execution Date:	14-06-22
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123	Priority:	Medium
Steps /Action		System Response	
1. Load the web URL or click on mobile application icon. 2. Select Get Started. 3. Enter registered email to log in of a premium account. 4. Enter registered password to login of a premium account. 5. Press Login Button. 6. Click on hide Ads.		1. Main application homepage will open. 2. Log in screen will open. 3. Email is accepted. 4. Password is accepted 5. Account is logged in. Role Selection Page is displayed. 6. Ads are hidden	
Expected Result:		Hiding monetized Ads.	
Actual Result:		Ads are hidden successfully.	
Status:		Pass	

➤ **Module 4: Complaint Management Portal**

UC-ID	Use Case Name
UC-4.1	Create Complaint
UC-4.2	View Complaint Status
UC-4.3	Delete Complaint
UC-4.4	Respond to Complaint
UC-4.5	Edit Compliant Description

1. TC-4.1 against UC-4.1

Test Id:	TC-4.1	Test Case Designed by:	Idrees Ahmed Ghazi
Test Case Title:	Verify Complaint Creation	Test Case Executed by:	Hunia Nadeem
Module Name:	Complaint Management Portal	Test Case Execution Date:	07-06-2022
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123, Complaint Description: "Test Complaint!", Type: "Security"	Priority:	High
Steps /Action		System Response	
1. Load the web URL or click on mobile application icon. 2. Select Get Started. 3. Enter registered email to log in. 4. Enter registered password to login. 5. Press Login Button. 6. Select Renter as your Role 7. Select the property of which complaint is to be created. 8. Select Complaint from the displayed menu 9. Select create complaint. 10. Select the complaint type 11. Provide description of the complaint. 12. Press Submit button.		1. Main application homepage will open. 2. Log in screen will open. 3. Email is accepted. 4. Password is accepted 5. Account is logged in. Role Selection Page is displayed. 6. Renter Portal is opened. 7. Detailed property portal will open. 8. A window will open which will ask to view complaints or create complaints. 9. Window asking for complaint details will be displayed. 10. Complaint Type is accepted. 11. Complaint description is accepted. 12. Complaint Submitted.	
Expected Result:		Complaint will be submitted to the relevant proprietor successfully.	
Actual Result:		Complaint submitted successfully.	
Status:		Pass	

2. TC-4.2 against UC-4.2

Test Id:	TC-4.2	Test Case Designed by:	Idrees Ahmed Ghazi
Test Case Title:	Confirm Complaint Status Visibility	Test Case Executed by:	Hunia Nadeem
Module Name:	Complaint Management Portal	Test Case Execution Date:	07-06-2022
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123	Priority:	High
Steps /Action		System Response	
1. Load the web URL or click on mobile application icon. 2. Select Get Started. 3. Enter registered email to log in. 4. Enter registered password to login. 5. Press Login Button. 6. Select Renter as your Role 7. Select the property of which complaint is to be viewed. 8. Select Complaint from the displayed menu 9. Select view complaints.		1. Main application homepage will open. 2. Log in screen will open. 3. Email is accepted. 4. Password is accepted 5. Account is logged in. Role Selection Page is displayed. 6. Renter Portal is opened. 7. Detailed property portal will open. 8. A window will open which will ask to view complaints or create complaints. 9. Window will appear displaying all the complaints with their status.	
Expected Result:		Complaint Status will be viewed successfully.	
Actual Result:		Complaint viewed successfully.	
Status:		Pass	

3. TC-4.3 against UC-4.3

Test Id:	TC-4.3	Test Case Designed by:	Idrees Ahmed Ghazi
Test Case Title:	Verify Complaint Deletion	Test Case Executed by:	Hunia Nadeem
Module Name:	Complaint Management Portal	Test Case Execution Date:	07-06-2022
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123	Priority:	Medium
Steps /Action		System Response	
1. Load the web URL or click on mobile application icon. 2. Select Get Started. 3. Enter registered email to log in. 4. Enter registered password to login. 5. Press Login Button. 6. Select Renter as your Role 7. Select the property of which complaint is to be deleted. 8. Select Complaint from the displayed menu 9. Select view complaints. 10. Select the complaint to be deleted. 11. Select Delete Complaint 12. Select “Yes” to delete complaint successfully.		1. Main application homepage will open. 2. Log in screen will open. 3. Email is accepted. 4. Password is accepted 5. Account is logged in. Role Selection Page is displayed. 6. Renter Portal is opened. 7. Detailed property portal will open. 8. A window will open which will ask to view complaints or create complaints. 9. Window will appear displaying all the complaints with their status. 10. Drop down menu appears. 11. Confirmation message displayed. 12. Complaint Deleted from both renter and proprietor’s portal.	
Expected Result:		Complaint will be deleted from both renter and proprietor portals successfully.	
Actual Result:		Complaint deleted successfully.	
Status:		Pass	

4. TC-4.4 against UC-4.4

Test Id:	TC-4.4	Test Case Designed by:	Idrees Ahmed Ghazi
Test Case Title:	Confirm Reply to Complaint	Test Case Executed by:	Hunia Nadeem
Module Name:	Complaint Management Portal	Test Case Execution Date:	07-06-2022
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123	Priority:	High
Steps /Action		System Response	
1. Load the web URL or click on mobile application icon. 2. Select Get Started. 3. Enter registered email to log in. 4. Enter registered password to login. 5. Press Login Button. 6. Select Proprietor as your Role 7. Select the property whose complaint is to be replied to. 8. Select Complaint from the displayed menu 9. Select the 3 dot button of complaint to be replied. 10. Select “Mark as Resolved”		1. Main application homepage will open. 2. Log in screen will open. 3. Email is accepted. 4. Password is accepted 5. Account is logged in. Role Selection Page is displayed. 6. Proprietor Portal is opened. 7. Detailed property portal will open. 8. A window will open which will display all the complaints. 9. Drop down menu appears. 10. Complaint Status changes to Resolved.	
Expected Result:		Complaint's status is updated successfully.	
Actual Result:		Complaint's status is updated	
Status:		Pass	

5. TC-4.5 against UC-4.5

Test Id:	TC-4.5	Test Case Designed by:	Idrees Ahmed Ghazi
Test Case Title:	Confirm Complaint is Editable	Test Case Executed by:	Hunia Nadeem
Module Name:	Complaint Management Portal	Test Case Execution Date:	07-06-2022
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123	Priority:	Medium
Steps /Action		System Response	
1. Load the web URL or click on mobile application icon. 2. Select Get Started. 3. Enter registered email to log in. 4. Enter registered password to login. 5. Press Login Button. 6. Select Renter as your Role 7. Select the property of which complaint is to be edited. 8. Select Complaint from the displayed menu 9. Select view complaints. 10. Select the complaint to be edited. 11. Select Edit Complaint 12. Edit the complaint. 13. Select Submit Changes.		1. Main application homepage will open. 2. Log in screen will open. 3. Email is accepted. 4. Password is accepted 5. Account is logged in. Role Selection Page is displayed. 6. Renter Portal is opened. 7. Detailed property portal will open. 8. A window will open which will ask to view complaints or create complaints. 9. Window will appear displaying all the complaints with their status. 10. Drop down menu appears. 11. Complaint Creation Window opens where you can change only complaint description. 12. Complaint description is accepted. 13. Complaint changes are submitted to both renter and proprietor's portal.	
Expected Result:		Complaint description will be edited to the relevant proprietor successfully.	
Actual Result:		Complaint edited successfully.	
Status:		Pass	

➤ **Module 5: Tenant Management Portal**

UC-ID	Use Case Name
UC-5.1	Add Property
UC-5.2	View Property
UC-5.3	Delete Property
UC-5.4	Generate Property Id
UC-5.5	View Renters
UC-5.6	View Renter Details
UC-5.7	Add Renter
UC-5.8	Remove Renter
UC-5.9	View Lease
UC-5.10	Print Lease
UC-5.11	Renew Lease
UC-5.12	Update Rent
UC-5.13	Edit Property Details
UC-5.14	Go to Payment Portal
UC-5.15	Respond to Renter Connectivity Request

1. TC-5.1 against UC-5.1

Test Id:	TC-5.1	Test Case Designed by:	Idrees Ahmed Ghazi
Test Case Title:	Verify Property Addition	Test Case Executed by:	Hunia Nadeem
Module Name:	Tenant Management Portal	Test Case Execution Date:	13-06-2022
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123	Priority:	High
Steps /Action		System Response	
1. Load the web URL or click on mobile application icon. 2. Select Get Started. 3. Enter registered email to log in. 4. Enter registered password to login. 5. Press Login Button. 6. Select Proprietor as your Role 7. Select the property button with “+” sign. 8. Enter property details. 9. Click on Add Property Button.		1. Main application homepage will open. 2. Log in screen will open. 3. Email is accepted. 4. Password is accepted 5. Account is logged in. Role Selection Page is displayed. 6. Proprietor Portal is opened. 7. Add property pop up/screen appears. 8. Details accepted. 9. Property added.	
Expected Result:		Property added to the list of properties.	
Actual Result:		Property added successfully.	
Status:		Pass	

2. TC-5.2 against UC-5.2

Test Id:	TC-5.2	Test Case Designed by:	Idrees Ahmed Ghazi
Test Case Title:	Check Property Details Visibility	Test Case Executed by:	Hunia Nadeem
Module Name:	Tenant Management Portal	Test Case Execution Date:	13-06-2022
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123	Priority:	High
Steps /Action		System Response	
1. Load the web URL or click on mobile application icon. 2. Select Get Started. 3. Enter registered email to log in. 4. Enter registered password to login. 5. Press Login Button. 6. Select Proprietor as your Role 7. Select the property you want to view.		1. Main application homepage will open. 2. Log in screen will open. 3. Email is accepted. 4. Password is accepted 5. Account is logged in. Role Selection Page is displayed. 6. Proprietor Portal is opened. 7. Property details opened from the Property menu.	
Expected Result:		Property details viewed on opening the property.	
Actual Result:		Property viewed successfully.	
Status:		Pass	

3. TC-5.3 against UC-5.3

Test Id:	TC-5.3	Test Case Designed by:	Idrees Ahmed Ghazi
Test Case Title:	Verify Property Deletion	Test Case Executed by:	Hunia Nadeem
Module Name:	Tenant Management Portal	Test Case Execution Date:	13-06-2022
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123	Priority:	Medium
Steps /Action		System Response	
1. Load the web URL or click on mobile application icon. 2. Select Get Started. 3. Enter registered email to log in. 4. Enter registered password to login. 5. Press Login Button. 6. Select Proprietor as your Role 7. Select the property you want to delete. 8. Click on Delete Property button. 9. Press yes.		1. Main application homepage will open. 2. Log in screen will open. 3. Email is accepted. 4. Password is accepted 5. Account is logged in. Role Selection Page is displayed. 6. Proprietor Portal is opened. 7. Property details opened from the Property menu. 8. Property deletion confirmation is asked. 9. Property deleted.	
Expected Result:		Property to be deleted successfully.	
Actual Result:		Property deleted successfully.	
Status:		Pass	

4. TC-5.5 against UC-5.5

Test Id:	TC-5.5	Test Case Designed by:	Idrees Ahmed Ghazi
Test Case Title:	Verify Renter Visibility	Test Case Executed by:	Hunia Nadeem
Module Name:	Tenant Management Portal	Test Case Execution Date:	13-06-2022
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123	Priority:	Low
Steps /Action		System Response	
1. Load the web URL or click on mobile application icon. 2. Select Get Started. 3. Enter registered email to log in. 4. Enter registered password to login. 5. Press Login Button. 6. Select Proprietor as your Role 7. Select the property whose renters you want to view. 8. Click on View Renters button.		1. Main application homepage will open. 2. Log in screen will open. 3. Email is accepted. 4. Password is accepted 5. Account is logged in. Role Selection Page is displayed. 6. Proprietor Portal is opened. 7. Property details opened from the Property menu. 8. Renters' table is displayed with Renters' details.	
Expected Result:		Renter details to be viewed successfully.	
Actual Result:		Renter details viewed successfully.	
Status:		Pass	

5. TC-5.6 against UC-5.6

Test Id:	TC-5.6	Test Case Designed by:	Idrees Ahmed Ghazi
Test Case Title:	Verify Renter Addition	Test Case Executed by:	Hunia Nadeem
Module Name:	Tenant Management Portal	Test Case Execution Date:	13-06-2022
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123	Priority:	High
Steps /Action		System Response	
1. Load the web URL or click on mobile application icon. 2. Select Get Started. 3. Enter registered email to log in. 4. Enter registered password to login. 5. Press Login Button. 6. Select Proprietor as your Role 7. Select the property in which you want to add renters. 8. Click on View Renters button. 9. Click on Add Renter button. 10. Enter renter details. 11. Click on Add button.		1. Main application homepage will open. 2. Log in screen will open. 3. Email is accepted. 4. Password is accepted 5. Account is logged in. Role Selection Page is displayed. 6. Proprietor Portal is opened. 7. Property details opened from the Property menu. 8. Renters' table is displayed with Renters' details. 9. Pop up appears that ask for renter details. 10. Renter details accepted. 11. Renter added.	
Expected Result:		Renter to be added successfully in the property.	
Actual Result:		Renter added successfully.	
Status:		Pass	

6. TC-5.7 against UC-5.7

Test Id:	TC-5.7	Test Case Designed by:	Idrees Ahmed Ghazi
Test Case Title:	Verify Renter Deletion	Test Case Executed by:	Hunia Nadeem
Module Name:	Tenant Management Portal	Test Case Execution Date:	13-06-2022
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123	Priority:	Medium
Steps /Action		System Response	
1. Load the web URL or click on mobile application icon. 2. Select Get Started. 3. Enter registered email to log in. 4. Enter registered password to login. 5. Press Login Button. 6. Select Proprietor as your Role 7. Select the property whose renters you want to delete. 8. Click on View Renters button. 9. Select the renter you want to delete and click on its 3 dot button. 10. Click on Delete button.		1. Main application homepage will open. 2. Log in screen will open. 3. Email is accepted. 4. Password is accepted 5. Account is logged in. Role Selection Page is displayed. 6. Proprietor Portal is opened. 7. Property details opened from the Property menu. 8. Renters' table is displayed with Renters' details. 9. Pop up appears. 10. Renter deleted from the property.	
Expected Result:		Renter to be deleted successfully from the property.	
Actual Result:		Renter deleted successfully.	
Status:		Pass	

7. TC-5.8 against UC-5.8

Test Id:	TC-5.8	Test Case Designed by:	Idrees Ahmed Ghazi
Test Case Title:	Verify Lease Visibility	Test Case Executed by:	Hunia Nadeem
Module Name:	Tenant Management Portal	Test Case Execution Date:	13-06-2022
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123	Priority:	High
Steps /Action		System Response	
1. Load the web URL or click on mobile application icon. 2. Select Get Started. 3. Enter registered email to log in. 4. Enter registered password to login. 5. Press Login Button. 6. Select Proprietor as your Role 7. Select the property whose lease is to be viewed. 8. Click on Lease from the side menu.		1. Main application homepage will open. 2. Log in screen will open. 3. Email is accepted. 4. Password is accepted 5. Account is logged in. Role Selection Page is displayed. 6. Proprietor Portal is opened. 7. Property details opened from the Property menu. 8. Lease details opened.	
Expected Result:		Property Lease to be viewed successfully.	
Actual Result:		Lease viewed successfully.	
Status:		Pass	

8. TC-5.9 against UC-5.9

Test Id:	TC-5.9	Test Case Designed by:	Idrees Ahmed Ghazi
Test Case Title:	Verify Lease Printability	Test Case Executed by:	Hunia Nadeem
Module Name:	Tenant Management Portal	Test Case Execution Date:	13-06-2022
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123	Priority:	Low
Steps /Action		System Response	
1. Load the web URL or click on mobile application icon. 2. Select Get Started. 3. Enter registered email to log in. 4. Enter registered password to login. 5. Press Login Button. 6. Select Proprietor as your Role 7. Select the property whose lease is to be printed. 8. Click on Lease from the side menu. 9. Select the lease to be printed.		1. Main application homepage will open. 2. Log in screen will open. 3. Email is accepted. 4. Password is accepted 5. Account is logged in. Role Selection Page is displayed. 6. Proprietor Portal is opened. 7. Property details opened from the Property menu. 8. List of lease for each renter is displayed. 9. Printing pop screen appears that prints the document.	
Expected Result:		Property Lease to be printed successfully.	
Actual Result:		Lease printed successfully.	
Status:		Pass	

9. TC-5.10 against UC-5.10

Test Id:	TC-5.10	Test Case Designed by:	Idrees Ahmed Ghazi
Test Case Title:	Verify Lease Renewability	Test Case Executed by:	Hunia Nadeem
Module Name:	Tenant Management Portal	Test Case Execution Date:	13-06-2022
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123	Priority:	Low
Steps /Action		System Response	
1. Load the web URL or click on mobile application icon. 2. Select Get Started. 3. Enter registered email to log in. 4. Enter registered password to login. 5. Press Login Button. 6. Select Proprietor as your Role 7. Select the property whose lease is to be renewed. 8. Click on Lease from the side menu. 9. Select the lease to be renewed. 10. Click on Submit Changes.		1. Main application homepage will open. 2. Log in screen will open. 3. Email is accepted. 4. Password is accepted 5. Account is logged in. Role Selection Page is displayed. 6. Proprietor Portal is opened. 7. Property details opened from the Property menu. 8. List of lease for each renter is displayed. 9. Lease in editable form along with the reason for change will be asked. 10. Lease renewal request will be sent to the respective renter.	
Expected Result:		Property Lease renewal request sent to the renter successfully.	
Actual Result:		Lease renewal requested successfully.	
Status:		Pass	

10. TC-5.11 against UC-5.11

Test Id:	TC-5.11	Test Case Designed by:	Idrees Ahmed Ghazi
Test Case Title:	Verify Renter Connectivity Request is Respondable	Test Case Executed by:	Hunia Nadeem
Module Name:	Tenant Management Portal	Test Case Execution Date:	13-06-2022
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123	Priority:	Medium
Steps /Action		System Response	
1. Load the web URL or click on mobile application icon. 2. Select Get Started. 3. Enter registered email to log in. 4. Enter registered password to login. 5. Press Login Button. 6. Select Proprietor as your Role 7. Select the property in which renter requests are to be responded. 8. Click on View Requests Buttons.		1. Main application homepage will open. 2. Log in screen will open. 3. Email is accepted. 4. Password is accepted 5. Account is logged in. Role Selection Page is displayed. 6. Proprietor Portal is opened. 7. Property details opened from the Property menu. 8. Renter connectivity requests are listed which can be approved or declined.	
Expected Result:		Renter connectivity requests are received and can be responded to.	
Actual Result:		Requests received and successfully responded.	
Status:		Pass	

➤ **Module 6: Property Management Portal**

UC-ID	Use Case Name
UC-6.1	View Lease
UC-6.2	View Property Details
UC-6.3	View Landlord's Details
UC-6.4	Check Rent Tenure
UC-6.5	Request Lease Renewal
UC-6.6	Print Lease
UC-6.7	Respond to Lease
UC-6.8	Connect With Landlord
UC-6.9	Go to Payment Portal

1. TC-6.1 against UC-6.1

Test Id:	TC-6.1	Test Case Designed by:	Hunia Nadeem
Test Case Title:	Verify Lease Visibility for Renter	Test Case Executed by:	Idrees Ghazi
Module Name:	Property Management Portal	Test Case Execution Date:	13-06-2022
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123	Priority:	Medium
Steps /Action		System Response	
1. Load the web URL or click on mobile application icon. 2. Select Get Started. 3. Enter registered email to log in. 4. Enter registered password to login. 5. Press Login Button. 6. Select Renter as your Role 7. Select the property whose lease is to be viewed. 8. Click on Lease from the side menu.		1. Main application homepage will open. 2. Log in screen will open. 3. Email is accepted. 4. Password is accepted 5. Account is logged in. Role Selection Page is displayed. 6. Renter Portal is opened. 7. Property details opened from the Property menu. 8. Lease details opened.	
Expected Result:		Property Lease to be viewed successfully.	
Actual Result:		Lease viewed successfully.	
Status:		Passed	

2. TC-6.2 against UC-6.2

Test Id:	TC-6.2	Test Case Designed by:	Hunia Nadeem
Test Case Title:	Check Property Details Visibility	Test Case Executed by:	Idrees Ghazi
Module Name:	Property Management Portal	Test Case Execution Date:	13-06-2022
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123	Priority:	Medium
Steps /Action		System Response	
1. Load the web URL or click on mobile application icon. 2. Select Get Started. 3. Enter registered email to log in. 4. Enter registered password to login. 5. Press Login Button. 6. Select Renter as your Role 7. Select the property you want to view.		1. Main application homepage will open. 2. Log in screen will open. 3. Email is accepted. 4. Password is accepted 5. Account is logged in. Role Selection Page is displayed. 6. Renter Portal is opened. 7. Property details opened from the Property menu.	
Expected Result:		Property details viewed on opening the property.	
Actual Result:		Property viewed successfully.	
Status:		Pass	

3. TC-6.3 against UC-6.3

Test Id:	TC-6.3	Test Case Designed by:	Hunia Nadeem
Test Case Title:	Verify Landlord's Details Visibility	Test Case Executed by:	Idrees Ghazi
Module Name:	Property Management Portal	Test Case Execution Date:	13-06-2022
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123	Priority:	Medium
Steps /Action		System Response	
1. Load the web URL or click on mobile application icon. 2. Select Get Started. 3. Enter registered email to log in. 4. Enter registered password to login. 5. Press Login Button. 6. Select Renter as your Role 7. Select the property whose renters you want to view. 8. Click on View Proprietors button.		1. Main application homepage will open. 2. Log in screen will open. 3. Email is accepted. 4. Password is accepted 5. Account is logged in. Role Selection Page is displayed. 6. Renter Portal is opened. 7. Property details opened from the Property menu. 8. Proprietor details are displayed.	
Expected Result:		Landlord details to be viewed successfully.	
Actual Result:		Landlord details viewed successfully.	
Status:		Pass	

4. TC-6.4 against UC-6.4

Test Id:	TC-6.4	Test Case Designed by:	Hunia Nadeem
Test Case Title:	Verify Checking Rent's Tenure	Test Case Executed by:	Idrees Ghazi
Module Name:	Property Management Portal	Test Case Execution Date:	13-06-2022
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123	Priority:	Medium
Steps /Action		System Response	
1. Load the web URL or click on mobile application icon. 2. Select Get Started. 3. Enter registered email to log in. 4. Enter registered password to login. 5. Press Login Button. 6. Select Renter as your Role 7. Select the property whose rent's tenure you want to check. 8. Click on Check rent tenure button.		1. Main application homepage will open. 2. Log in screen will open. 3. Email is accepted. 4. Password is accepted 5. Account is logged in. Role Selection Page is displayed. 6. Renter Portal is opened. 7. Property details opened from the Property menu. 8. Tenure details are displayed.	
Expected Result:		Checking Rent Tenure successfully.	
Actual Result:		Rent tenure checked successfully.	
Status:		Pass	

5. TC-6.5 against UC-6.5

Test Id:	TC-6.5	Test Case Designed by:	Hunia Nadeem
Test Case Title:	Verify sending request for lease renewal.	Test Case Executed by:	Idrees Ghazi
Module Name:	Property Management Portal	Test Case Execution Date:	13-06-2022
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123 Lease Description: Please renew lease contract.	Priority:	Medium
Steps /Action		System Response	
1. Load the web URL or click on mobile application icon. 2. Select Get Started. 3. Enter registered email to log in. 4. Enter registered password to login. 5. Press Login Button. 6. Select Renter as your Role 7. Select the property whose lease is to be renewed 8. Click on Lease button. 9. Now click on renew lease. 10. Provide details and send request.		1. Main application homepage will open. 2. Log in screen will open. 3. Email is accepted. 4. Password is accepted 5. Account is logged in. Role Selection Page is displayed. 6. Renter Portal is opened. 7. Property details opened from the Property menu. 8. Lease details are displayed. 9. Renew Lease menu is opened. 10. Request is sent to the Proprietor successfully.	
Expected Result:		Request for lease removal should be sent successfully	
Actual Result:		Request for lease renewal is sent successfully.	
Status:		Pass	

6. TC-6.6 against UC-6.6

Test Id:	TC-6.6	Test Case Designed by:	Hunia Nadeem
Test Case Title:	Verify Lease printability	Test Case Executed by:	Idrees Ghazi
Module Name:	Property Management Portal	Test Case Execution Date:	13-06-2022
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123	Priority:	Low
Steps /Action		System Response	
1. Load the web URL or click on mobile application icon. 2. Select Get Started. 3. Enter registered email to log in. 4. Enter registered password to login. 5. Press Login Button. 6. Select Renter as your Role 7. Select the property whose lease is to be renewed 8. Click on Lease button. 9. Now click on printing lease.		1. Main application homepage will open. 2. Log in screen will open. 3. Email is accepted. 4. Password is accepted 5. Account is logged in. Role Selection Page is displayed. 6. Renter Portal is opened. 7. Property details opened from the Property menu. 8. Lease details are displayed. 9. Lease is printed	
Expected Result:		Lease to be printed successfully.	
Actual Result:		Lease is printed successfully.	
Status:		Pass	

7. TC-6.7 against UC-6.7

Test Id:	TC-6.7	Test Case Designed by:	Hunia Nadeem
Test Case Title:	Verifying responding to lease.	Test Case Executed by:	Idrees Ghazi
Module Name:	Property Management Portal	Test Case Execution Date:	13-06-2022
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123	Priority:	High
Steps /Action		System Response	
1. Load the web URL or click on mobile application icon. 2. Select Get Started. 3. Enter registered email to log in. 4. Enter registered password to login. 5. Press Login Button. 6. Select Renter as your Role 7. Open notification button 8. A request from proprietor would be there to be responded.		1. Main application homepage will open. 2. Log in screen will open. 3. Email is accepted. 4. Password is accepted 5. Account is logged in. Role Selection Page is displayed. 6. Renter Portal is opened. 7. Property details opened from the Property menu. 8. Notification menu is displayed 9. Request to lease is responded.	
Expected Result:		Request for Lease to be responded successfully.	
Actual Result:		Request for Lease is responded successfully.	
Status:		Pass	

8. TC-6.8 against UC-6.8

Test Id:	TC-6.8	Test Case Designed by:	Hunia Nadeem
Test Case Title:	Verifying Connecting to landlord	Test Case Executed by:	Idrees Ghazi
Module Name:	Property Management Portal	Test Case Execution Date:	13-06-2022
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123	Priority:	High
Steps /Action		System Response	
1. Load the web URL or click on mobile application icon. 2. Select Get Started. 3. Enter registered email to log in. 4. Enter registered password to login. 5. Press Login Button. 6. Select Renter as your Role 7. Click Add property. 8. Click on Connect landlord.		1. Main application homepage will open. 2. Log in screen will open. 3. Email is accepted. 4. Password is accepted 5. Account is logged in. Role Selection Page is displayed. 6. Renter Portal is opened. 7. Property is added. 8. Landlord is connected.	
Expected Result:		Landlord to be connected with successfully.	
Actual Result:		Landlord is connected successfully.	
Status:		Pass	

➤ **Module 7: Push Notifications**

UC-ID	Use Case Name
UC-7.1	Turn On Rent Reminders
UC-7.2	Turn Off Rent Reminders
UC-7.3	Turn On Chat Notifications
UC-7.4	Turn Off Chat Notifications
UC-7.5	Set Ringtone
UC-7.6	Set Vibration

1. TC-7.1 against UC-7.1

Test Id:	TC-7.1	Test Case Designed by:	Hunia Nadeem
Test Case Title:	Verifying turning on rent reminders	Test Case Executed by:	Idrees Ghazi
Module Name:	Push Notifications	Test Case Execution Date:	13-06-2022
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123	Priority:	Medium
Steps /Action		System Response	
1. Load the web URL or click on mobile application icon. 2. Select Get Started. 3. Enter registered email to log in. 4. Enter registered password to login. 5. Press Login Button. 6. Open side pane. 7. Click on the settings option. 8. Switch on rent reminders.		1. Main application homepage will open. 2. Log in screen will open. 3. Email is accepted. 4. Password is accepted 5. Account is logged in. Role Selection Page is displayed. 6. Side pane is opened. 7. Setting menu is displayed. 8. Rent reminders are switched on.	
Expected Result:		Rent reminders to be turned on successfully.	
Actual Result:		Rent reminders are turned on successfully	
Status:		Pass	

2. TC-7.2 against UC-7.2

Test Id:	TC-7.2	Test Case Designed by:	Hunia Nadeem
Test Case Title:	Verifying turning off rent reminders	Test Case Executed by:	Idrees Ghazi
Module Name:	Push Notifications	Test Case Execution Date:	13-06-2022
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123	Priority:	Medium
Steps /Action		System Response	
1. Load the web URL or click on mobile application icon. 2. Select Get Started. 3. Enter registered email to log in. 4. Enter registered password to login. 5. Press Login Button. 6. Open side pane. 7. Click on the settings option. 8. Switch Off rent reminders.		1. Main application homepage will open. 2. Log in screen will open. 3. Email is accepted. 4. Password is accepted 5. Account is logged in. Role Selection Page is displayed. 6. Side pane is opened. 7. Setting menu is displayed. 8. Rent reminders are switched off.	
Expected Result:		Rent reminders to be turned off successfully.	
Actual Result:		Rent reminders are turned off successfully	
Status:		Pass	

3. TC-7.3 against UC-7.3

Test Id:	TC-7.3	Test Case Designed by:	Hunia Nadeem
Test Case Title:	Verifying turning on Chat notifications	Test Case Executed by:	Idrees Ghazi
Module Name:	Push Notifications	Test Case Execution Date:	13-06-2022
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123	Priority:	Medium
Steps /Action		System Response	
1. Load the web URL or click on mobile application icon. 2. Select Get Started. 3. Enter registered email to log in. 4. Enter registered password to login. 5. Press Login Button. 6. Open side pane. 7. Click on the settings option. 8. Switch on Chat notifications.		1. Main application homepage will open. 2. Log in screen will open. 3. Email is accepted. 4. Password is accepted 5. Account is logged in. Role Selection Page is displayed. 6. Side pane is opened. 7. Setting menu is displayed. 8. Chat notifications are switched on.	
Expected Result:		Chat notifications to be turned off successfully.	
Actual Result:		Chat notifications are turned off successfully	
Status:		Pass	

4. TC-7.4 against UC-7.4

Test Id:	TC-7.4	Test Case Designed by:	Hunia Nadeem
Test Case Title:	Verifying turning off Chat notifications.	Test Case Executed by:	Idrees Ghazi
Module Name:	Push Notifications	Test Case Execution Date:	13-06-2022
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123	Priority:	Medium
Steps /Action		System Response	
1. Load the web URL or click on mobile application icon. 2. Select Get Started. 3. Enter registered email to log in. 4. Enter registered password to login. 5. Press Login Button. 6. Open side pane. 7. Click on the settings option. 8. Switch Off Chat notifications.		1. Main application homepage will open. 2. Log in screen will open. 3. Email is accepted. 4. Password is accepted 5. Account is logged in. Role Selection Page is displayed. 6. Side pane is opened. 7. Setting menu is displayed. 8. Chat notifications are switched off.	
Expected Result:		Chat notifications to be turned off successfully.	
Actual Result:		Chat notifications are turned off successfully	
Status:		Pass	

5. TC-7.5 against UC-7.5

Test Id:	TC-7.5	Test Case Designed by:	Hunia Nadeem
Test Case Title:	Verifying setting up ringtone	Test Case Executed by:	Idrees Ghazi
Module Name:	Push Notifications	Test Case Execution Date:	13-06-2022
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123	Priority:	Low
Steps /Action		System Response	
1. Load the web URL or click on mobile application icon. 2. Select Get Started. 3. Enter registered email to log in. 4. Enter registered password to login. 5. Press Login Button. 6. Open side pane. 7. Click on the settings option. 8. Set a ringtone		1. Main application homepage will open. 2. Log in screen will open. 3. Email is accepted. 4. Password is accepted 5. Account is logged in. Role Selection Page is displayed. 6. Side pane is opened. 7. Setting menu is displayed. 8. Ringtone is set	
Expected Result:		Setting ringtone successfully.	
Actual Result:		Ringtone is set successfully	
Status:		Pass	

6. TC-7.6 against UC-7.6

Test Id:	TC-7.6	Test Case Designed by:	Hunia Nadeem
Test Case Title:	Verifying setting up vibration.	Test Case Executed by:	Idrees Ghazi
Module Name:	Push Notifications	Test Case Execution Date:	13-06-2022
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123	Priority:	Low
Steps /Action		System Response	
1. Load the web URL or click on mobile application icon. 2. Select Get Started. 3. Enter registered email to log in. 4. Enter registered password to login. 5. Press Login Button. 6. Open side pane. 7. Click on the settings option. 8. Set the vibration level.		1. Main application homepage will open. 2. Log in screen will open. 3. Email is accepted. 4. Password is accepted 5. Account is logged in. Role Selection Page is displayed. 6. Side pane is opened. 7. Setting menu is displayed. 8. Vibration level is set	
Expected Result:		Setting vibration successfully.	
Actual Result:		Vibration is set successfully	
Status:		Pass	

➤ **Module 8: Help and Support**

UC-ID	Use Case Name
UC-8.1	Chat With Bot
UC-8.2	Do Live Chat
UC-8.3	Contact Support Team
UC-8.4	Send Feedback
UC-8.5	Change Bot's Language
UC-8.6	View Feedback History

1. TC-8.1 against UC-8.1

Test Id:	TC-8.1	Test Case Designed by:	Idrees Ahmed Ghazi		
Test Case Title:	Verify Chat Bot Interaction	Test Case Executed by:	Hunia Nadeem		
Module Name:	Help and Support	Test Case Execution Date:	07-06-2022		
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123 Question 1: "Hello" Question 2: "What payment methods you use?" Question 3: "Thank you!"	Priority:	Medium		
Steps /Action		System Response			
1. Load the web URL 1.1. Click on the Chat Bot icon on bottom right of web page. 1.2. Send test question 1 "Hello" 1.3. Send test question 2 "What payment methods you use?" 1.4. Send test question 3 "Thank you!" 2. Click on mobile application icon. 2.1. Select Get Started. 2.2. Enter registered email to log in. 2.3. Enter registered password to login. 2.4. Press Login Button. 2.5. Open side-pane menu 2.6. Select FAQ option 2.7. Test Questions are asked like steps 1.1-1.4		1. Main web homepage will open. 1.1. Chatbot pop will open. 1.2. Bot replies "Hello! How may I assist you?" 1.3. Bot replies "You can pay your rent using MasterVisa, UnionPay, JazzCash, EasyPaisa, and PayPal." 1.4. Bot replies "It was great helping you!" 2. Main application homepage will open. 2.1. Log in screen will open. 2.2. Email is accepted. 2.3. Password is accepted 2.4. Account is logged in. Role Selection Page is displayed. 2.5. Side slider menu appears. 2.6. Chatbot screen opens 2.7. Replies are received like steps 1.1-1.4			
Expected Result:	Successful interaction with Chatbot occurs.				
Actual Result:	Chatbot interaction successful				
Status:	Pass				

2. TC-8.2 against UC-8.2

Test Id:	TC-8.2	Test Case Designed by:	Idrees Ahmed Ghazi
Test Case Title:	Verify Live Chat Interaction	Test Case Executed by:	Hunia Nadeem
Module Name:	Help and Support	Test Case Execution Date:	07-06-2022
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123 Question 1: "Live Chat"	Priority:	Medium
Steps /Action		System Response	
1. Load the web URL 1.1. Click on the Chat Bot icon on bottom right of web page. 1.2. Send test question 1 "Live Chat" 1.3. If there is any available support person, chat is switched to them. 2. Click on mobile application icon. 2.1. Select Get Started. 2.2. Enter registered email to log in. 2.3. Enter registered password to login. 2.4. Press Login Button. 2.5. Open side-pane menu 2.6. Select FAQ option 2.7. Test Questions are asked like steps 1.1-1.3		1. Main web homepage will open. 1.1. Chatbot pop will open. 1.2. Bot finds available support person. 1.3. Live chat occurs. 2. Main application homepage will open. 2.1. Log in screen will open. 2.2. Email is accepted. 2.3. Password is accepted 2.4. Account is logged in. Role Selection Page is displayed. 2.5. Side slider menu appears. 2.6. Chatbot screen opens 2.7. Replies are received like steps 1.1-1.3	
Expected Result:		Successful Live chat through Chatbot occurs.	
Actual Result:		Successful Live chat through Chatbot occurred.	
Status:		Pass	

3. TC-8.3 against UC-8.3

Test Id:	TC-8.3	Test Case Designed by:	Idrees Ahmed Ghazi
Test Case Title:	Confirm Support Team's Contact	Test Case Executed by:	Hunia Nadeem
Module Name:	Help and Support	Test Case Execution Date:	07-06-2022
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123	Priority:	Low
Steps /Action		System Response	
1. Load the web URL 1.1. Click on the About us option top webpage header 1.2. Click on the provided email 2. Click on mobile application icon. 2.1. Select Get Started. 2.2. Enter registered email to log in. 2.3. Enter registered password to login. 2.4. Press Login Button. 2.5. Open side-pane menu 2.6. Select About us option 2.7. Click on provided email		1. Main web homepage will open. 1.1. About us window appears where Support Team's details are displayed 1.2. Page redirected to standard email, i.e. Gmail, where you can send email directly to the support team. 2. Main application homepage will open. 2.1. Log in screen will open. 2.2. Email is accepted. 2.3. Password is accepted 2.4. Account is logged in. Role Selection Page is displayed. 2.5. Side slider menu appears. 2.6. About us screen opens where Support Team's details are displayed 2.7. You will be redirected to email app installed on your device where you can send email directly to the support team.	
Expected Result:		Successfully redirected to the support team's email.	
Actual Result:		Successful redirection occurred	
Status:		Pass	

4. TC-8.4 against UC-8.4

Test Id:	TC-8.4	Test Case Designed by:	Idrees Ahmed Ghazi
Test Case Title:	Verify Feedback Uploadation	Test Case Executed by:	Hunia Nadeem
Module Name:	Help and Support	Test Case Execution Date:	07-06-2022
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123, Rating: "3", Feedback: "Test Feedback"	Priority:	Medium
Steps /Action		System Response	
1. Load the web URL or click on mobile application icon. 2. Select Get Started. 3. Enter registered email to log in. 4. Enter registered password to login. 5. Press Login Button. 6. Open Feedback option from webpage header or application's side-pane menu. 7. Enter Rating 8. Enter Feedback 9. Click Submit button		1. Main application homepage will open. 2. Log in screen will open. 3. Email is accepted. 4. Password is accepted 5. Account is logged in. Role Selection Page is displayed. 6. Feedback page opens. 7. Rating accepted 8. Feedback accepted 9. Feedback submitted successfully	
Expected Result:		Successfully system feedback is submitted.	
Actual Result:		Feedback submission successful	
Status:		Pass	

5. TC-8.5 against UC-8.5

Test Id:	TC-8.5	Test Case Designed by:	Idrees Ahmed Ghazi
Test Case Title:	Verify Change in Bot's Language	Test Case Executed by:	Hunia Nadeem
Module Name:	Help and Support	Test Case Execution Date:	07-06-2022
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123	Priority:	Low
Steps /Action		System Response	
1. Load the web URL 1.1. Click on the Chat Bot icon on bottom right of web page. 1.2. Select Change Language 1.3. Select Language 2. Click on mobile application icon. 2.1. Select Get Started. 2.2. Enter registered email to log in. 2.3. Enter registered password to login. 2.4. Press Login Button. 2.5. Open side-pane menu 2.6. Select FAQ option 2.7. Test Questions are asked like steps 1.1-1.3		1. Main web homepage will open. 1.1. Chatbot pop will open. 1.2. Available Languages appear 1.3. Language changed. 2. Main application homepage will open. 2.1. Log in screen will open. 2.2. Email is accepted. 2.3. Password is accepted 2.4. Account is logged in. Role Selection Page is displayed. 2.5. Side slider menu appears. 2.6. Chatbot screen opens 2.7. Replies are received like steps 1.1-1.3	
Expected Result:		Successful Chatbot language changes.	
Actual Result:		Chatbot language changed successfully	
Status:		Pass	

7. TC-8.6 against UC-8.6

Test Id:	TC-8.6	Test Case Designed by:	Idrees Ahmed Ghazi
Test Case Title:	Check Feedback History	Test Case Executed by:	Hunia Nadeem
Module Name:	Help and Support	Test Case Execution Date:	07-06-2022
Test Data:	Email: rpcSystem@gmail.com , Password: rpc_123	Priority:	Low
Steps /Action		System Response	
1. Load the web URL or click on mobile application icon. 2. Select Get Started. 3. Enter registered email to log in. 4. Enter registered password to login. 5. Press Login Button. 6. Open Feedback option from webpage header or application's side-pane menu. 7. Select View Submitted Feedbacks option		1 Main application homepage will open. 2 Log in screen will open. 3 Email is accepted. 4 Password is accepted 5 Account is logged in. Role Selection Page is displayed. 6 Feedback page opens. 7 List of feedbacks submitted will appear	
Expected Result:		Feedback history is to be viewed successfully	
Actual Result:		Feedback history is viewed successfully	
Status:		Pass	

6.10. Environmental Needs

Following are the environmental needs for carrying out our test plan:

- Stable internet connection for both mobile and web applications
- Devices with Android 6 and above, or iOS 11 and above.
- Web browsers that support HTML 5.
- Authentic CNIC that will be matched with Citizen Portal database of CNICs.
- Authentic bank account of EasyPaisa, JazzCash, PayPal and MasterVisa must exist.

7. Conclusion and Future Work

This is the concluding chapter of the document that concludes the purpose of our product. Any improvements or future work that can be done to improve the working of our product are discussed. Alongside, the work contributions of each group member throughout the project have been shared.

7.1. Conclusion

The Renters/Proprietors Coordination Application's ultimate objective is to resolve the prime problems that surface after acquiring or renting a property. By automating the post-property renting process, the issues revolving around data insecurity, payment frauds, time conflicts, communication delays, and delayed complaint resolutions will be promptly handled. Machine Learning and Artificial intelligence will help secure and smooth the user experience. Nonetheless, a plethora of post-property renting management options will be a click away from the renters and proprietors.

7.2. Future Work

- Artificial Intelligence can be introduced to login using fingerprint detection, face or voice recognition.
- The application availability can be made global by introducing more language and usability support.
- Using better data structures and coding techniques, we can improve the system performance and reduce the maximum response time from 5 seconds to 2 seconds.
- Through better machine learning techniques, security and fraud detection can be improved and the system can be made more secure.

7.3. Lessons Learnt

- **Hunia Nadeem (FA20-BCS-024)**
 1. Leadership and communication skills were improved by working in a team and collaborating over a project.
 2. I learnt to use Figma for UI designing that helped in improving my front-end designing.
 3. I learnt JavaScript while implementing the web modules of the project.
 4. I learnt the best ways for document formatting and designing.
 5. Learnt how to train a Chatbot and how to integrate it into the website.
- **Idrees Ahmed Ghazi (FA20-BCS-025)**
 1. My Presentation Document preparation skills were enhanced.
 2. Database management and integration skills got much strengthened.
 3. Learnt the use of the Flutter dart language from scratch.
 4. Polished my team working skills in handling bigger projects like RPC System.

5. Time management skills and discipline got much improved by completing all the assignments when the submission deadline was critical and following the rules of the class.

7.4. Work Division

- Scope Document

Hunia Nadeem (FA20-BCS-024)	Idrees Ahmed Ghazi (FA20-BCS-025)
1. Abstract 2. Introduction 3. Problem Statement 4. Problem Solution for the proposed system and Objectives 5. Vision Statement 6. Modules 7. System Limitations/Constraints 8. WBS and Gantt Chart 9. Conclusion 10. Final Format Preparation of Scope Document (Word .docx file)	1. Related System Analysis/Literature Review 2. Advantages/Benefits of Proposed System 3. Project Scope 4. Software Process and Design Methodology 5. Tools and Technologies 6. Project Stakeholders and Roles 7. Work Division 8. Concepts 9. Mockups 10. Presentation (PowerPoint .ppt file)

- SRS Document

Hunia Nadeem (FA20-BCS-024)	Idrees Ahmed Ghazi (FA20-BCS-025)
1. Overall Description 2. Requirement Identifying Technique (Module 1, 3, 6, 7) 3. Functional Requirements (Module 1, 3, 6, 7) 4. External Interface Requirements (Software & Communication interfaces) 5. Final Format Preparation of SRS Document (Word .docx file)	1. Introduction & Conclusion 2. Requirement Identifying Technique (Module 2, 4, 5, 8) 3. Functional Requirements (Module 2, 4, 5, 8) 4. External Interface Requirements (User interfaces & Mockups) 5. Presentation (PowerPoint .ppt file)

- SDS Document

Hunia Nadeem (FA20-BCS-024)	Idrees Ahmed Ghazi (FA20-BCS-025)
<ol style="list-style-type: none"> SDS Document (Word .docx file) Activity Diagram (Module 1, 2, 3, 4) Sequence Diagram(Module 1, 5, 6, 8) JSON Schema Algorithm and Implementation Class Diagram (collaborated) ERD (collaborated) 	<ol style="list-style-type: none"> SDS Presentation (PowerPoint .ppt file) Activity Diagram (Module 2, 3, 4, 7) Sequence Diagram Data Dictionary User Interface Design Class Diagram (collaborated) ERD (collaborated)

- Project Implementation

Hunia Nadeem (FA20-BCS-024)	Idrees Ahmed Ghazi (FA20-BCS-025)
<ol style="list-style-type: none"> Web Implementation Modules Implemented. <ol style="list-style-type: none"> Profile Management (Minor) Complaint Management (Minor) AI Chatbot (Major) Technologies Used: <ol style="list-style-type: none"> HTML,CSS, JavaScript (Front-End) Tools Used: <ol style="list-style-type: none"> Visual Studio Code 2019 IBM Watson Chatbot 	<ol style="list-style-type: none"> Mobile Application Implementation Modules Implemented. <ol style="list-style-type: none"> Profile Management using firebase (Minor) Complaint Management (Minor) AI Chatbot (Major) Technologies Used: <ol style="list-style-type: none"> Flutter ii) Dart iii) Firebase Tools Used: <ol style="list-style-type: none"> Android Studio IBM Watson Chatbot

- STP Document

Hunia Nadeem (FA20-BCS-024)	Idrees Ahmed Ghazi (FA20-BCS-025)
<ol style="list-style-type: none"> Introduction (Heading 1) Test Deliverables (Heading 5) Conclusion (Heading 6)) Test Cases: Module (2, 4, 5, 8) 	<ol style="list-style-type: none"> STP Presentation (PowerPoint .ppt file) Executive Summary (Heading 2) Testing and Evaluation (Heading 3) Test Cases: Module(1, 3, 6, 7)

- Final Document

Hunia Nadeem (FA20-BCS-024)	Idrees Ahmed Ghazi (FA20-BCS-025)
1. Introduction (Heading 1) 2. Implementation (Heading 5) 3. Testing and Evaluation (Heading 6.8- 6.11) 4. Conclusion (Heading 7) + Lesson Learnt (Heading 7.3)	1. Problem Definition (Heading 2) 2. Requirement Analysis (Heading 3) 3. Architecture & Design (Heading 4) 4. Testing and Evaluation (Heading 6.1- 6.7) 5. Lesson Learnt (Heading 7.3)

8. References

- *Renting problems.* (2020, September 11). Housing Advice NI. <https://www.housingadviceni.org/advice-young-people/renting-problems>
- *Complete this 10-item checklist before Tenant move in.* (2013, October 22). The Balance Small Business. <https://www.thebalancesmb.com/landlord-actions-before-move-in-2125027>
- *What is a lease?* (n.d.). Investopedia. <https://www.investopedia.com/terms/l/lease.asp>
- Wiegers, K. E., & Beatty, J. (2013). *Software requirements.*

9. Appendix

9.1. Figma UI

- https://bit.ly/FigmaUI_Web
- https://bit.ly/FigmaUI_Mobile

9.2. Figma Prototypes

- https://bit.ly/FigmaPrototype_Web
- https://bit.ly/FigmaPrototype_Mobile

9.3. Github Repositories

- https://github.com/HuniaNad/RPC_System.Website
- <https://github.com/idreesghazi/RPC-System-Mobile->

9.4. Youtube

- https://bit.ly/RPCSsystem_YT

9.5. Guidelines

• Website Guide:

1. Download the project files and folders from Github repository.
2. Open “homepage.html” file. You will be directed to the homepage of our website.
3. Firstly, you will have to login in order to access the Renter or Proprietor Portals.
[Credentials are not mandatory to be added]
4. You can either login through **Sign up** button or by clicking on **Get Started**. You will be directed to **Role Selection** page.
5. After logging in, you can choose your role as **Renter** or **Proprietor** by clicking on the relevant button.
6. After choosing your role, you will be directed to the respective role’s homepage where all the rented (Renter’s case) or owned (Proprietor’s case) properties will be displayed.
7. By selecting any property, you will be directed to that property’s menu where the **Complaint Management Module**’s interface has been implemented.
8. You can submit a complaint from Renter’s portal and view complaints from Proprietor’s Portal. (Everything is hardcoded)
9. Chatbot is implemented on all screens.
10. You can logout or go to the respective homepage whenever you want.

- **Mobile Guide:**

1. Download the project code from GitHub repository.
2. Open the project through Android Studio.
3. Now open the android emulator for the app.
4. Make sure that the mobile emulator is Google Pixel 4 Q Version as it works better on it.
5. Now Run the code, the designed flutter App will open.
6. The first menu will be the **Get Started** Screen.
7. Now click on the **Get Started** button.
8. The login screen will appear.
9. If the user has not created an account then click on Sign Up button to create a new account.
10. The user information will be stored to firebase google.
11. Now Login to the account and choose your role.
12. The user can also open the side menu and open the FAQ menu to chat with bot
13. After opening the FAQ, provide your credentials and select whether you want to use Live Chat feature or to chat with bot.
14. Now open the side menu and **sign out**.

10. Plagiarism Report

6/16/22, 8:03 PM

Turnitin Originality Report



report by Idrees Ahmed Ghazi
From Junaid Zaidi Library, CUI, Islamabad.
(2022) (Junaid Zaidi Library, CUI,
Islamabad)

Similarity Index	Similarity by Source
10%	Internet Sources: 5% Publications: 1% Student Papers: 9%

Processed on 16-Jun-2022 7:24 AM

PDT

ID: 1857950424

Word Count: 47510

sources:

- 1** 1% match (student papers from 06-Nov-2018)
[Submitted to Higher Education Commission Pakistan on 2018-11-06](#)
- 2** 1% match (student papers from 10-May-2019)
[Submitted to Higher Education Commission Pakistan on 2019-05-10](#)
- 3** 1% match (student papers from 04-Nov-2018)
[Submitted to Higher Education Commission Pakistan on 2018-11-04](#)
- 4** 1% match (student papers from 26-May-2022)
[Submitted to University of Essex on 2022-05-26](#)
- 5** < 1% match (student papers from 14-Dec-2018)
[Submitted to Higher Education Commission Pakistan on 2018-12-14](#)