Software Requirements Specification

For

House Rental Management System

Prepared by

Mir Mohammod Tahasin (MUH2025007M)
Rubya Rashed (MUH2025014M)
Md. Foysal Mahmud (MUH2025025M)
Wakil Ahammad (ASH2025026M)
Sanjida Akter samanta (BFH2025010F)

Fardin Ahosan Shawon(ASH1825019M)

Institute of Information Technology

Noakhali Science and Technology University

20.02.2023

1. Introduction

A House Rental System is a software application designed to assist with the management and organization of properties that are available for rent. This system may be used by property owners, property management companies, and tenants to streamline the process of renting a house or apartment. The system may include features such as: A database of available properties, including information about location, size, price, and amenities.

1.1 Problem Statement

There are several potential problems that a house rental system may aim to solve:

Difficulty in managing multiple rental properties: Landlords or property managers with a large number of rental properties may find it challenging to keep track of all their listings, rental agreements, and other tasks related to property management. A house rental system can help streamline these processes by providing a central location for storing and organizing all relevant information.

Inefficient communication between landlords and tenants: Miscommunication or a lack of transparency can lead to issues such as late rent payments, maintenance requests falling through the cracks, or misunderstandings around lease terms. A house rental system can help facilitate communication between landlords and tenants by providing a platform for sending and receiving messages and other important information.

Time-consuming manual processes: Traditional methods of managing rentals, such as using paper documents and manual record-keeping, can be time-consuming and prone to errors. A house rental system can automate many of these processes, saving time and reducing the potential for mistakes.

Lack of a comprehensive view of rental activity: Without a central system for managing rentals, it can be difficult for landlords or property managers to get a complete picture of their rental business. A house rental system can provide a comprehensive overview of rental activity, including information on current and past tenants, rental income, and maintenance expenses.

1.2 Purpose

A house rental system is a software application that allows individuals or companies to manage the rental of houses or other properties. The system typically includes features for managing rental listings, handling rental applications and contracts, accepting payments, and managing maintenance requests and other tasks related to property management. Some house rental systems may also include additional features such as background check capabilities, communication tools for landlords and tenants, and integrations with other property management software. The specific requirements and functionality of a house rental system will depend on the needs and goals of the user.

1.3 Project Scope

The scope of a house rental system will depend on the specific requirements and goals of the project. However, some common elements that might be included in the scope of a house rental system are:

Rental listings: The system should allow landlords or property managers to create and manage listings for rental properties, including details such as the location, size, features, and rental rate of the property.

Rental applications and contracts: The system should provide a way for landlords to review and accept rental applications, and for tenants to sign rental agreements electronically.

Payment processing: The system should allow landlords to accept rent payments from tenants electronically, and track the payment history of each tenant.

Maintenance requests: The system should provide a way for tenants to submit maintenance requests, and for landlords or property managers to track and manage these requests.

Communication tools: The system should provide a platform for landlords and tenants to communicate with each other, such as through messaging or email.

Background checks: The system may include functionality for performing background checks on rental applicants.

Reporting and analytics: The system should provide landlords or property managers with access to reports and analytics on their rental business, such as information on rental income, occupancy rates, and maintenance expenses.

This is just a general list of possible elements that might be included in the scope of a house rental system. The specific requirements of the project will depend on the needs and goals of the user.

1.4 Glossary

This subsection contains definitions of all the terms and abbreviations used in the document.

i. MB – Megabytes
ii. UI – User Interface
iii. SRS – Software Requirement Specifications
iv. API – Application Program Interface
v. XML – Extensible Markup Language
vi. RESTful – Representational State Transfer
vii. HTML – Hyper Text Markup Language

1.5 References

IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998

1.6 Overview

A house rental system is a system that allows individuals or companies to rent out houses or apartments to tenants. The system typically involves a landlord or property owner who owns the rental unit and a tenant who pays rent to live in the unit. There are several different types of house rental systems, including: Traditional leasing: In this system, a tenant signs a lease agreement with a landlord, agreeing to rent the property. There are several ways that a house rental system can be set up. In some cases, the landlord or property owner may handle the rental process directly, advertising the property for rent and managing all aspects of the rental agreement themselves. In other cases, a third party such as a property management company may be responsible for managing the rental of the property on behalf of the landlord or owner. A house rental system may also involve the use of software or other technology to help manage the rental process. For example, a landlord may use a property management software program to track rent payments, schedule maintenance and repairs, and handle other aspects of the rental process. Overall, the goal of a house rental system is to provide a clear and organized way for landlords and tenants to manage the rental of a house or other residential property. This feature enables landlords to list their properties, including details such as property type, location, price, and amenities. This feature enables landlords to collect rent payments online and track payment history. This feature allows landlords to manage maintenance requests from tenants, schedule repairs, and track the status of maintenance tasks.

2. Stakeholders and Characteristics

The stakeholders in a house rental system might include:

I. **House Owner:** These individuals or companies own the rental properties and are responsible for managing them. They will likely be the primary users of the system, and will have specific needs and goals related to managing their rental business.

II. **Tenants:** Tenants are the individuals or families who rent the properties from the landlords. They will use the system to communicate with their landlords, submit maintenance requests, and make rent payments.

III. **Payment processors:** Payment processors such as credit card companies, banks, and online payment providers are stakeholders as they provide the infrastructure for processing rent payments.

IV. **System developers:** System developers and vendors are stakeholders as they design, develop, and maintain the system, and provide technical support to users.

V. **Bangladesh Police:** Police Officers will get a same version of the application but the authorization will be different. They are the higher authority of the system. They can see and collect the tenant information like NID card number, number of people and their name. They can also get landlords information like there house information, how many people lives in a house. This module minimizes the problem of taking tenants information for a house. So, if they need any people’s information like where they lived, they can get the information easily by the house rental management system.

The characteristics of a house rental system will depend on the specific requirements and goals of the project. However, some common characteristics that a house rental system might have include:

User-friendly interface: The system should be easy for landlords, tenants, and other stakeholders to use, with a clear and intuitive design.

Scalability: The system should be able to handle a large number of rental listings and users, and should be able to support growth as the number of rentals increases.

Security: The system should have robust security measures in place to protect the privacy and sensitive information of landlords, tenants, and other stakeholders.

Integration with other software: The system may need to integrate with other property management software or financial systems.

Customization: The system should be flexible and customizable to meet the specific needs and goals of the landlord or property manager.

3. Design and Implementation Constrains

We have employed design and implementation constraints to ensure the success of this project. It also refers to a tool that allows developers and testers to inspect and interact with the application's user interface (UI) elements.

3.1 JavaScript, JSX and React.js

The visual layout of the components that a user could interact with in a website or technical product is referred to as user interface design, or UI design. In other terms, it is a website's visual design.

3.1.1 Programming Language

JavaScript is an ECMAScript-compliant high-level, frequently just-in-time compiled language. It has first-class functions, dynamic typing, and prototype-based object orientation. Its multi-paradigm, allowing you to program in event-driven, functional, or imperative styles. React is a front-end JavaScript toolkit for creating user interfaces using UI components that is free and open-source. Meta and a community of individual developers and businesses support it. JavaScript XML is abbreviated as JSX. It's just a JavaScript syntactic extension. It allows us to create HTML directly in React (within JavaScript code). It is straightforward to generate a template in React using JSX, but it is not a simple template language; instead, it has all of JavaScript's capability.

It is faster than standard JavaScript because it optimizes when converting to standard JavaScript. Rather than dividing the markup and functionality in different files, React makes use of components.

3.1.2 CSS Framework

Cascading Style Sheets (CSS) is a language for specifying the appearance of a document written in a markup language like HTML. Along with HTML and JavaScript, CSS is a key component of the World Wide Web. Semantic UI is a website using UI component framework. Developers may use Semantic UI to create websites with quick and clear HTML, as well as a fully mobile responsive experience. Semantic UI offers a React-integrated version called Semantic UI React, which includes the following functionalities:

- JQuery Free.
- Declarative API.
- Augmentation.
- Shorthand Props.
- Sub Components.
- Auto Controlled State

3.2 Server-Side Technology

Server-side development refers to the actions that take place behind the scenes when an application is used. It primarily focuses on databases, scripting, website architecture, backend logic, APIs, and Servers.

3.2.1 Python, Django Rest API

Python is a dynamically semantic, interpreted, object-oriented high-level programming language. Its high-level built-in data structures, together with dynamic typing and dynamic binding, make it ideal for Rapid Application Development and as a scripting or glue language for connecting existing components. Python's concise, easy-to-learn syntax prioritizes readability, which lowers software maintenance costs. Software Requirements Specification for To-Let Python is widely used to create an application's back end. It should come as no surprise that there are a variety of Python frameworks available to help with server-side programming. The Django REST Framework (DRF) is a popular, robust, and versatile framework for creating Web APIs. To get data from our server to our application, we'll use the Django Rest API. Representational State Transfer is abbreviated as REST. Application Programming Interface is what API stands for.

3.2.2 Database Server

PostgreSQL is an advanced, enterprise class open source relational database that supports both SQL (relational) and JSON (non-relational) querying. PostgreSQL's speed, security and robustness make it suitable for 99% of applications, so it's a great starting place for our application. It is a dependable, powerful, and stable solution with sophisticated features such as the following:

- User-defined types.
- Table inheritance.
- Sophisticated locking mechanism.
- Foreign key referential integrity.
- Views, rules, subquery.
- Nested transactions (save points)
- Multi-version concurrency control (MVCC)
- Asynchronous replication.

3.2.3 Cloud Storage

Amazon S3 is a type of object storage that allows to store and retrieve any quantity of data from any location. It's a straightforward storage solution with industry-leading durability, availability, performance, security, and scalability.

3.3 Legal requirements

The system will need to comply with relevant laws and regulations, such as those related to data privacy, fair housing, and consumer protection.

3.3.1 Technology constraints

The system will need to be developed using technology that is compatible with the hardware and software infrastructure of the user. It may also need to be responsive and accessible on different devices, such as desktop computers, laptops, tablets, and smartphones.

3.3.2 Integration with other systems

If the house rental system will be integrating with other software or systems, such as a financial management system or a background check service, the design and implementation will need to take into account the interfaces and capabilities of those systems.

3.3.3 Data management

The system will need to securely store and manage large amounts of data, including rental listings, rental agreements, payment records, and maintenance requests.

3.3.4 User needs and preferences

The design and implementation of the system should take into account the needs and preferences of the landlords, tenants, and other stakeholders who will be using it.

3.3.5 Budget and resources

The development of the system will need to be feasible within the constraints of the available budget and resources.

4. Requirement Specification

All the requirements based on elicitation process is described in this section.

4.1 Functional Requirement

Functional requirements are those requirements that are used to illustrate the internal working nature of the system, the description of the system, and explanation of each subsystem. It consists of what task the system should perform, the processes involved, which data should the system holds and the interfaces with the use.

4.1.1 Create account and login to the system

FR-1	Landlords and tenants can create account and login to the system		
Description	The system should allow users to create an account and log in to the system.		
Stakeholders	House Owner & Tenants	Priority	High

4.1.2 Adds a new house to the system

FR-2	Landlords wants to adds a new house to the system		
Description	The system should allow landlords to list their properties for rent, including details such as location, price, number of bedrooms, and any other relevant information.		
Stakeholders	House Owner	**Priority**	High

4.1.3 Search ads by specific criteria

FR-3	Tenants wants to search house ads by specific criteria		
Description	The system should allow tenants to search for properties to rent based on location, price, number of bedrooms, and any other relevant criteria		
Stakeholders	Tenants	**Priority**	High

4.1.4 Search ads by User location

FR-4	Search house ads based on User location		
Description	Tenant's only needs to click on an option "House nearby" and system at first detect the stakeholder's position then based on that location it will list all the available houses near that area if any exists.		
Stakeholders	Tenants, House Owner	**Priority**	High

4.1.5 Add favorite house ads

FR-5	Stakeholders will add favorite ads to remember them		
Description	If tenants found any desired ads he wants to see later, he can bookmark that ads and see the ads in favorite section.		
Stakeholders	Tenants, House Owner	**Priority**	Medium

4.1.6 Notifications for Tenants about new houses

FR-6	Stakeholders wants notification when a house is available for rent nearby his location.		
Description	While stakeholders want to notified when there a house nearby him or on a specific area. For getting notified stakeholders need to enable their notification. And whenever there is a house available for rent stakeholders will be notified.		
Stakeholders	Tenants	**Priority**	Low

4.1.7 Communicate with landlords for renting a house

FR-7	Tenants can communicate with landlords		
Description	After observing house ads, if stakeholders found a vacant house for him, he can contact through SMS, E-Mail, and Phone call with landlords. Phone number, E-Mail account will be placed in ads information.		
Stakeholders	Tenants	**Priority**	High

4.1.8 User can reset password

FR-8	User want to reset password for making it not vulnerable		
Description	The "reset password" function allows users to reset their password if they have forgotten it or if they suspect that their account has been compromised.		
Stakeholders	House Owner and Tenants	**Priority**	Low

4.1.9 Search Houses using google map

FR-9	User can search houses using google map		
Description	By using Google Maps to search for houses, user can easily explore different neighborhoods and get an idea of the proximity of local amenities, such as schools, shopping, and public transportation.		
Stakeholders	Tenants	**Priority**	High

4.1.10 Change existing system language

FR-10	User can change the existing system language		
Description	If house owner / tenants don’t know about our existing English language or they don’t have any idea about it, then they can change the existing language English to Bangla.		
Stakeholders	House Owner and Tenants	**Priority**	Low

4.1.11 Conversation with admin

FR-11	Tenants and house owner wants to communicate with the admin for any information or problems.		
Description	Tenants/House owner may want to communicate with the admin for any system issues or objection about any misbehavior. The admin must have to response and find a solution for the user’s problems.		
Stakeholders	Tenants and Owner	**Priority**	Medium

4.1.12 Add reviews and ratings for houses

FR-12	Tenants wants to add reviews and ratings for houses		
Description	The system should allow tenants to leave reviews and ratings for their rental experience.		
Stakeholders	Tenants	**Priority**	Low

4.1.13 Boost existing ads

FR-13	House Owner wants to boost his ads		
Description	House Owner can boost the ads when they are creating ads. For boosting the ads, they have to choose the specific payment credential and need to pay the exact amount of money. They also can boost existing ads doing the above process.		
Stakeholders	House Owner	**Priority**	High

4.1.14 Remove ads from the system

FR-14	Landlords wants to Remove the ads from the system		
Description	As landlords house get rented so they don’t want to show the ads. For so they have to go to the ads and select it and then remove. When removing if they selected for 30 day to show the ads and removing it at 5th day of showing then they will get money for other 25 day.		
Stakeholders	House Owner	**Priority**	High

4.1.15 Money Transaction

FR-15	Tenants wants to pay for boosted add		
Description	The system should have a payment gateway for for getting the money from the users. The users will be given “Bkash, Nagad, Visa Card/Master Card” for giving the fees for ads boosting.		
Stakeholders	House Owner	**Priority**	High

4.1.16 View about their properties

FR-16	House owner wants to view about their properties		
Description	The system should allow House owner to view and manage their properties		
Stakeholders	House owner	**Priority**	High

4.2 Data Requirement

For our application we have to store many information-like pictures from landlord’s house, tenant’s picture this information needs to be stored in the system. For storing pictures system database needs to be limited.

4.2.1 Storing image data

DR-1	Stakeholders have to provide specific sized picture for the house		
Description	Landlords will upload their ads information, house information and tenant’s Information. Also, images from the houses needs to be within 3 MB.		
Stakeholders	House Owner	**Priority**	High

4.2.2 Property Data

DR-2	House Owner have to provide property data		
Description	This includes information about the rental properties such as property type, address, number of bedrooms, bathrooms, amenities, and rental price.		
Stakeholders	House Owner	**Priority**	High

4.2.3 Tenant Data

DR-3	Tenants have to provide tenants data		
Description	This includes information about tenants such as name, contact details, lease agreement details, payment history, and maintenance requests. Those information have stored.		
Stakeholders	Tenants	**Priority**	High

4.2.4 Financial data

DR-4	Financial data will be stored in the system database.		
Description	This includes information about financial transactions such as rent payments, maintenance expenses, and other related expenses. This will be stored in in the system database.		
Stakeholders	House owner, Tenants	**Priority**	High

4.2.5 Analytics and reporting data

DR-5	Analytical report of ads and boosted ads		
Description	This includes data that is used to generate analytics and reports such as financial reports, post reach, boosted ads, ads reach.		
Stakeholders	House owner, Tenants	**Priority**	High

4.3 Performance Requirement

It is important to maintain performance of the software system. To ensure performance we maintain these steps:

4.3.1 Speed and Latency Requirements

PR-1	Faster searching houses and load ads quickly		
Description	Loading the ads information will be faster and can load it within seconds. And any search result will show up within 1 seconds and images of the house will Load within 5 seconds if the network speed is good.		
Stakeholders	House owner and Tenants	**Priority**	Medium

4.3.2 Precision and Accuracy Requirements

There are no precision and accuracy requirements for our project.

4.3.3 Capacity Requirements

This system can load up to thousands of tenant’s information and thousands of ads information.

4.4 Dependability Requirements

If House Rental Management system can provide availability, reliability, safety, security then this system will be dependable.

4.4.1 Availability and Reliability Requirements

This system will be available for 24 hours. Stakeholders can use the system anytime they wanted and can see ads or rate a house and upload information which is very reliable for stakeholders.

4.4.2 Safety Requirements

This system will not contain any malware and this will not harm any stakeholder's device.

4.5 Maintainability and Supportability

4.5.1 Maintenance Requirements

MR-1	Make the code maintainable.		
Description	Code must be developed so that it can be modified later and will be readable.		
Stakeholders	Developers	**Priority**	High

4.5.2 Supportability Requirements

For accessing information, the system will use some authorization techniques to ensure that correct data is used by the correct user.

4.6 Security Requirements

Securing information is much more important for a system to get users dependability. Here is some of them:

4.6.1 Access Requirements

For accessing information, the system will use some authorization techniques to ensure that correct data is used by correct user.

4.6.2 Integrity Requirements

Integrity requirements refers to a security system which ensures an expectation of data quality. It also ensures that all data of the system would never be exposed to the malicious modification or accidental destruction. For preventing anonymous access to user password, the system will use encryption technique called Hash Function for encrypting user password.

4.6.3 Privacy Requirements

Privacy requirements enhances to protect stakeholder's privacy. In this way, all data or a partial part of data are going to be disclosed according to system's privacy policy. To ensure privacy, the central database should be protected by the anonymous. Users are permitted to get access to those data which are being associated by them which can be ensured by the user log in system.

4.7 Usability and Human Integrity Requirements

This system will provide more user-friendly environment

4.7.1 Ease of Use Requirements

Our system will be easier to use by any type of people and they don't need any training to use the system.

4.7.2 Personalization and Internationalization Requirements

There are no personalization and internationalization requirements in our system.

4.8 Look and Feel Requirements

Look and feel requirements mainly refers how the system will look like.

4.8.1 Appearance Requirements

AR-1	Text color and font		
Description	All texts and description will be at a good font size so that users can understand what is important and mandatory input fields will be kept red colored until user put correct information.		
Stakeholders	Tenants, House owner, Police Officers	**Priority**	High

4.9 Style Requirements

There are no style requirements in our system.

4.10 Operational and Environmental Requirements

Operational and environmental requirement refers to the capabilities, performance measurements, process, measurements of effectiveness, measurements of performance, measures of sustainability, measurements of technical performances etc.

4.10.1 Expected Physical Requirements

There are no expected physical requirements in our system.

4.10.2 Requirements for Interfacing with Adjacent Systems

There are no requirements for interfacing with adjacent system for our project.

4.10.3 Release Requirements

There are no specific release requirements in our system.

4.11 Legal Requirements

Legal requirements normally refer to the terms and conditions or privacy policy of any organizations. The terms and condition of our application is that, no third-party software or person are allowed to engage to use our data for their business purpose.

5. Requirement Engineering Process

Process Requirements Engineering (RE) determines software requirements according to customer requirements or needs. Requirements engineering process includes requirements elicitation, needs modeling, requirements analysis, requirements assurance & validation, and requirements management.

5.1 Requirement Elicitation Techniques

Techniques Requirements elicitation is the practice of researching and finding system requirements for users, customers, and other stakeholders, also referred to as "requirement gathering". Requirement elicitation can be done by contacting participants directly or by doing some research, analysis and testing.

5.1.1 Interviews

We hold discussions that can be held individually or with a small group of participants. They are an effective way to access services without spending a lot of time with participants because we meet with people to discuss only certain important requirements of this program. Negotiations are useful for obtaining individual requirements for members in organizing workshops where those members of the program come together to resolve any issues or conflicts. We mainly perform our interview based on some specific criteria.

- Short description about goals and objectives
- Registration process
- Searching house
- Rent house

5.1.2 Existing System Analysis

Existing system can help to show how systems are currently operating or what they are what I should do. System include written information about current programs, business processes, needs specifications, and competitor research. Review once textual analysis can help Software Requirements Specification for To-let determine which performance should remain and functionality that isn’t in use. After existing document in analysis, we found several problems with the existing system.

- Existing systems cannot perform category-wise price range. There are no shifting process on it.
- No cloud storage system is provided by the existing systems.

5.1.3 System Interface Analysis

The first thing to do is to identify which systems the system-to-be shall communicate with. It could be a server on the Internet, a piece of software on the same host as the system-to-be, some hardware or something completely different.

5.1.4 Questionnaires

The questionnaire is a useful way to investigate styles, changes in attitudes and users' ideas, and user satisfaction with priorities and preferences. Our lists of questions were as short as possible. The respondent may be tired or frustrated. Had a basic reason for all the questions as well as group the topic areas together for the respondent to focus on. The main advantage of this survey responses was that they were collected in the usual way. Information was summarized by a large number of people.

5.2 Requirement Validation

Requirement validation ensures that the requirements are correct and reflect the quality you want from this program. In the beginning, our requirements looked good but when we read them and tried to work with them, they came out having ambiguities and gaps.

5.2.1 Review the Requirements

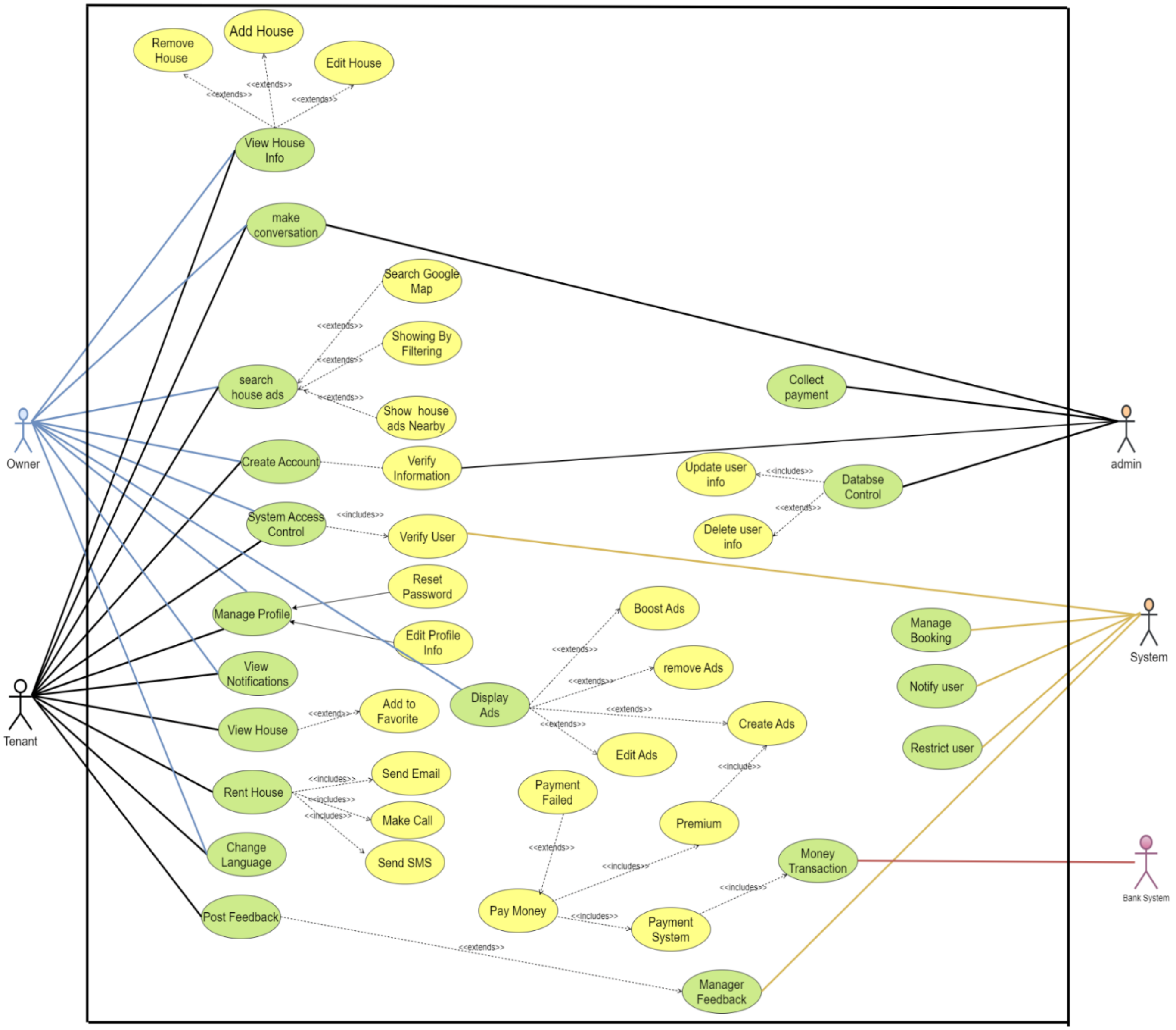
Negative peer review, especially the type of rigorous review called evaluation, is unique among the highest quality software processes available. We had a team of reviewers representing different perspectives and carefully examined written needs, analysis models, and related information on disability.

5.2.2 Test the Requirements

The test creates another view of the requirements. We also performed writing tests regarding assurance of whether the expected performance was found or not. Getting tested by the user needs to document the expected product behavior under specified conditions.

6. Use Case Diagram

Use case diagram comprises actors and use cases, where actors perform several cases or one. This also shows which actors have access to which use case. Here is the use case diagram for house management system.



7. Use Case Description

All use cases from use case diagram are explained here.

Table 1: ***System access control***

Use Case 1	System access control	
Goal	Stakeholders wants to sign in to the system.	
Preconditions	NA	
Success End Condition	Stakeholders can access his old account	
Failed End Condition	Stakeholders don’t have access to account.	
Primary Actors: **Secondary Actors:**	Owners, Tenants System	
Trigger	Access account request	
Main Success Flows	**Step**	**Action**
	1	Users requested to access account.
	2	System will show sign in option
	3	System will verify the user information
	4	User are into the system
Alternative Flows	**Step**	**Branching Action**
	3a	Users forget their password
	3a1	Users click on forget password
	3a2	System will prompt for entering stakeholders’ mobile number and username.
	3a3	System then check the mobile number whether the current device has the given mobile number. If not, it again prompts the user to enter the valid mobile number.
	3a4	If the mobile number is valid system then sends a confirmation code to that mobile number.
	3a5	System then popup for entering the confirmation code.
	3a6	System will check the confirmation code if it’s wrong it again prompt for entering the confirmation code.
	3a7	If the confirmation code is correct system then prompt the user the new password for his username.
	3a8	System now again prompt the user to login with the new username and password.
	4a	If username and password does not match then system will prompt stakeholders to reenter the password.

Quality Requirements	Step	Requirement
		The system must allow user to access within 2 seconds

Table 2: ***Create Account***

Use Case 2	Create Account	
Goal	User will use our system services and be a part to the system.	
Preconditions	User must come to our system	
Success End Condition	Users will be a user of the system. Can use many functionalities based onuser access.	
Failed End Condition	Users won’t be a valid user of the system. Don’t have access to the system functionalities.	
Primary Actors: **Secondary Actors:**	Tenant, Owner.	
Trigger	Sign up request.	
Main Success Flows	**Step**	**Action**
	1	Users requested for sign up
	2	Users will choose user type form like tenant or owner.
	3	Users then need to fill preferred information in the form. For tenants they have to provide their name, mobile number, unique username, password and email. For Owners they have to provide their name, mobile number, unique username, password, house address and email.
	4	After filling all the information correctly system will show the submit button. And Users can click and submit the information to the system.
	5	System will now collect the information of the user.
	6	For confirming the user information, system will send a confirmation code to the user mobile number.
	7	A popup window will appear for entering the confirmation code.
	8	User have to write the code at the popup window and click submit.
	9	System will then check and verify the confirmation code.
	10	System will then show the ads information page as default.
Alternative Flows	**Step**	**Branching Action**
	3a	System will check the username while Users typing on that fieldand show whether the username is available or not.
	7a	Users can resend the confirmation code again to their mobilephone.
	10a	If the confirmation code is not valid then system will again ask for the confirmation code.
Quality Requirements	**Step**	**Requirement**
	7	User have to confirm the code within 2 minutes. After this time the confirmation code will be invalid.

Table 3: **Search for house ads**

Use Case 3	Search for house ads	
Goal	User wants to search house.	
Preconditions	User are signed in.	
Success End Condition	User can see house ads based on search criteria.	
Failed End Condition	No house ads found based on that criteria.	
Primary Actors: **Secondary Actors:**	Tenants	
Trigger	Search house ads request.	
Main Success Flows	**Step**	**Action**
	1	User requested to search ads.
	2	User will type and click on search option.
	3	System will search from the database server based on ads title given by User.
	4	System then shows ads which is matched the ads title at first then related ads will show as related ads section at last in the list.
Alternative Flows	**Step**	**Branching Action**
		No alternative flow.
Quality Requirements	**Step**	**Requirement**
		Search result must be shown within 1 seconds

Table 4: ***Search house ads nearby***

Use Case 4	Search house ads nearby	
Goal	Users want to search house in his area.	
Preconditions	Users are signed in.	
Success End Condition	Users can see house ads near his area.	
Failed End Condition	No house ads found near user's area.	
Primary Actors: **Secondary Actors:**	Tenants	
Trigger	Click house ads nearby	
Main Success Flows	**Step**	**Action**
	1	Users requested to search ads nearby.
	2	System will detect user's location.
	3	System will search from the database server based on location captured from stakeholder's current location.
	4	System then shows ads which is matched the location at first then related ads will show as related ads section at last in the list.
Alternative Flows	**Step**	**Branching Action**
	2a	If system can't get access to device location, system will prompt the stakeholders to turn on the location service and give system permission to access the location.
Quality Requirements	**Step**	Requirement
	1.	The system should be able to handle a high volume of searches and data without experiencing slow loading times or other performance issues.
	1.	The system should be secure enough to store user's location.

Table 5: ***Search house by filtering***

Use Case 5	Search house by filtering	
Goal	Users want to search house by specific criteria.	
Preconditions	Users are signed in.	
Success End Condition	Users can see house ads based on their given criteria.	
Failed End Condition	No house ads found by the given criteria.	
Primary Actors: **Secondary Actors:**	Tenants	
Trigger	Search house by filtering request.	
Main Success Flows	**Step**	**Action**
	1	User requested to search ads by giving criteria.
	2	System will show a filter box which contains filter by ads title, rooms count, rent per month, location and flat size.
	3	Users will give necessary filter information and click on search.
	4	System then shows ads which is matched the given criteria at first then related ads will show as related ads section at last in the list.
Alternative Flows	**Step**	**Branching Action**
	2a	If no filter given system will prompt for filling any criteria.
Quality Requirements	**Step**	**Requirement**
		The system should be able to handle a high volume of user filtering searches without experiencing slow loading times or other performance issues.

Table 6: **Search house by Google Map**

Use Case 25	Search House by Google map	
Goal	To search for houses available for rent in a specific location using a map interface..	
Preconditions	The user is accessing this system with a device that supports the map interface. The user has selected the "Search with google map" option on this system	
Success EndCondition	The user has viewed the available houses for rent in the selected location and can contact the admin.	
Failed End Condition	Owners failed to view house.	
Primary Actors: **Secondary Actors:**	Owners,Tenants	
Trigger	The user wants to search for available houses in a specific location using the map interface.	
Main Success Flows	**Step**	**Action**
	1	The user opens the map interface within the house rental system.
	2	The system displays the user's current location or prompts the user to enter a specific location.
	3	The user can zoom in or out on the map to adjust the search radius.
	4	The system displays the available houses within the search radius as markers on the map.
	5	The user can filter the results by price range, number of bedrooms, or other relevant criteria.
	6	The system updates the map to display only the houses that meet the selected criteria.
	7	The user can click on a marker to view more information about the selected house, such as photos, amenities, and contact information.
	8	The system allows the user to contact the owner or property manager to schedule a viewing or request more information.
Alternative Flows	**Step**	**Branching Action**
	2a	If the user encounters any errors or issues during the process, the system provides appropriate error messages and prompts the user to try again or contact support for assistance.
	5a	If the user wants to refine the search criteria, they can update the filters and repeat the search process.
	7a	If the user wants to view the houses in a list view, they can switch to the list view from the map interface.
Quality Requirements	**Step**	**Requirement**
	1	The search results should accurately reflect the location and attributes of the properties displayed on the map.

Table 7: **View houses**

Use Case 7	View Houses	
Goal	Tenants wants to see houses which is added by owner.	
Preconditions	Tenants are signed in.	
Success End Condition	The system displays houses which is for rent.	
Failed End Condition	No owned house and house information still added.	
Primary Actors: **Secondary Actors:**	Tenants	
Trigger	Click on view house	
Main Success Flows	**Step**	**Action**
	1	The system displays houses which is for rent.
Alternative Flows	**Step**	**Branching Action**
		Not alternative flow.
Quality Requirements	**Step**	**Requirement**
		The user interface should be intuitive and easy to use, with clear instructions and visual cues to guide users through the house information and features

*Table 8: **Rent House***

Use Case 8	Rent House	
Goal	Message, call, email the owner if tenants like a home	
Preconditions	Tenants/Owner are signed in.	
Success End Condition	Conversation With owner and confirm this house if want.	
Failed End Condition	Owner Doesn’t reply messages or receive calls or reply emails.	
Primary Actors: **Secondary Actors:**	Tenants, Owner	
Trigger	Click on Call, SMS, Email, Button	
Main Success Flows	**Step**	**Action**
	1	Users Messaged to owner for rent house.
	2	System will load Owner information from the database server.
Alternative Flows	**Step**	**Branching Action**
		No alternative flow.
Quality Requirements	**Step**	**Requirement**
		The system call, SMS, Email must respond fast

Table 9: **Post Feedback**

Use Case 9	Post Feedback	
Goal	Post feedback for system service	
Preconditions	Tenants/Owner are signed in.	
Success End Condition	Tenants and owner share their thoughts about system service.	
Failed End Condition	For some technical reason post can’t be done perfectly.	
Primary Actors: **Secondary Actors:**	Tenants, Owner System	
Trigger	Share some thought about system	
Main Success Flows	**Step**	**Action**
	1	Tenants and owner share their thoughts about system service.
Alternative Flows	**Step**	**Branching Action**
	1	No alternative flows

Table 10: **View Notifications**

Use Case 10	View Notifications.	
Goal	Tenants will notify when they finally rent their house or for Owner when any tenants want to conversation with owner or payment.	
Preconditions	Tenants/Owner are signed in.	
Success End Condition	The system notify Owner or Tenants.	
Failed End Condition	When Payment process is failed or tenant’s messages aren’t process because of technical system.	
Primary Actors: **Secondary Actors:**	Tenants, Owner System	
Trigger	Notify everything	
Main Success Flows	**Step**	**Action**
	1	The system notify Owner or Tenants.
	2	Users requested to see his notifications
	3	System will display all notification information like message, payment process and so on.
	4	If user clicks on any of the notification, then system will display all information about selected notification.
Alternative Flows	**Step**	**Branching Action**
	4a	If user clicks notification remove button beside each notification.
	4b	System will remove that notification from the notification list.
Quality Requirements	**Step**	**Requirement**
		Notifications should be accurate, relevant, and timely, providing users with the information they need to take appropriate action

Table 11: ***Edit profile info***

Use Case 11	Edit profile info	
Goal	Display for tenants and owner information	
Preconditions	Tenants/Owner are signed in.	
Success End Condition	The system Display or Edit Owner or Tenants information.	
Failed End Condition	For some technical reason edit can't be done perfectly.	
Primary Actors: **Secondary Actors:**	Tenants, Owner System	
Trigger	Display and Edit Information.	
Main Success Flows	**Step**	**Action**
	1	The system display/edit Owner or Tenants information.
	2	System will collect user's information from the database server.
	3	System will show users information like profile picture, users name, mobile number, email, username.
Alternative Flows	**Step**	**Branching Action**
	1a	If user clicks remove button beside from Profile account should be deleted.
Quality Requirements	**Step**	**Requirement**
		The user interface should be clear, intuitive, and easy to navigate, with clear instructions and visual cues to guide users through the process of editing their profile information.

Table 12: ***Change language***

Use Case 24	Change Language	
Goal	To change the language from Bangla and English language.	
Preconditions	The user is a registered user of this system. The application supports Bangla and English languages.	
Success EndCondition	Owners successfully change language.	
Failed End Condition	Owners failed to change language.	
Primary Actors: **Secondary Actors:**	Owners,Tenants	
Trigger	The user wishes to change the language of the application.	
Main Success Flows	**Step**	**Action**
	1	The user opens the application and navigates to the settings or preferences page.
	2	The user selects the "Language" option.
	3	The system presents the user with a list of available languages.
	4	The user selects the desired language from the list.
	5	The system saves the new language setting and updates the application to display in the selected language.
	6	The system presents the user with a confirmation message that the language has been changed.
	7	The user continues using the application in the new language.
Alternative Flows	**Step**	**Branching Action**
	3a	If the desired language is not available, the system can prompt the user to select a different language or offer to add the desired language in a future update.
	6a	If the user encounters any errors or issues during the process, the system provides appropriate error messages and prompts the user to try again or contact support for assistance
Quality Requirements	**Step**	**Requirement**
		The feature should provide localization support, including translating all interface text, messages, and notifications into the selected language.

Table 13: ***Add house***

Use Case 13	Add house	
Goal	Owner wants to add his owned house information in the system.	
Preconditions	Owner are signed in.	
Success End Condition	The system will add house at owned house list.	
Failed End Condition	House will not be added in owned house list.	
Primary Actors: **Secondary Actors:**	Owner	
Trigger	Add house request.	
Main Success Flows	**Step**	**Action**
	1	Owner requested to add house.
	2	Owner needs to fill a form by adding house name, house address, number of flats, and number of units in each flat, house holding number.
	3	System will then validate the house holding number.
	4	System then add the house as Owner owned house and show a notification message that the house information is added successfully.
	5	The house will then show in the houses list.
Alternative Flows	**Step**	**Branching Action**
	3a	System will prompt that invalid house holding number entered and ask Owner to enter a valid house holding number.
Quality Requirements	**Step**	**Requirement**
		The feature should ensure that all information about the house is accurate, including the address, price, number of bedrooms and bathrooms, square footage, and other relevant details.

Table 14: ***Edit house***

Use Case 14	Edit house	
Goal	Owner wants to edit his old house from owned houses list	
Preconditions	Owner are signed in. Owner have owned houses.Owner select a house.	
Success End Condition	The system will update the selected house and its information from owned houses list.	
Failed End Condition	No owned house and house information will be updated.	
Primary Actors: **Secondary Actors:**	Owner	
Trigger	Update owned house request.	
Main Success Flows	**Step**	**Action**
	1	Owner requested to update his owned house.
	2	System will prompt for confirm the update of the house.
	3	Owner confirm the update.
	4	System then update the house information for owner.
Alternative Flows	**Step**	**Branching Action**
	3a	Owner cancel the updating.
Quality Requirements	**Step**	**Requirement**
		Not applicable.

Table 15: ***View houses***

Use Case 15	View Houses	
Goal	Tenants wants to see his houses which is added by owner.	
Preconditions	Tenants are signed in.	
Success End Condition	The system displays houses which is for rent.	
Failed End Condition	No owned house and house information still added.	
Primary Actors: **Secondary Actors:**	Tenants Owner, System	
Trigger	View owned houses request.	
Main Success Flows	**Step**	**Action**
	1	Pin-point the location
Alternative Flows	**Step**	**Branching Action**
		Not alternative flow.
Quality Requirements	**Step**	**Requirement**
	1	The feature should have minimal impact on performance, with quick response times and minimal lag or delay when viewing a house.

Table 16: **Remove House**

Use Case 16	Remove house	
Goal	Owner wants to remove his old house from owned houses list	
Preconditions	Owner are signed in. Owner have owned houses.Owner select a house.	
Success End Condition	The system will remove the selected house and its information from owned houses list.	
Failed End Condition	No owned house and house information will be deleted.	
Primary Actors: **Secondary Actors:**	Owner	
Trigger	Delete owned house request.	
Main Success Flows	**Step**	**Action**
	1	Owner requested to delete his owned house.
	2	System will prompt for confirm the deletion of the house.
	3	Owner confirm the deletion.
	4	System will collect Owner house holding number from the selected house.
	5	System then delete the house information for owner.
Alternative Flows	**Step**	**Branching Action**
	2a	Owner cancel the deletion.
Quality Requirements	**Step**	**Requirement**
		The system must show the user easy way to remove the house

Table 17: Display Ads

Use Case 17	Display ads	
Goal	Display ads by filtering or nearby house to users.	
Preconditions	Owner are signed in.	
Success End Condition	The system display ads to owner.	
Failed End Condition	No Ads created.	
Primary Actors: **Secondary Actors:**	Owner Not applicable	
Trigger	Owner Click on display adds	
Main Success Flows	**Step**	**Action**
	1	The system Show Ads to Owner.
	2	Owner requested to see ads.
	3	System will display all Ads.

	4	If owner clicks on any of the ads, then system will display all ads item about selected ads.
Alternative Flows	**Step**	**Branching Action**
		No alternative flow.
Quality Requirements	**Step**	**Requirement**
		Not applicable.

Table 18: Create ads

Use Case 18	Create ads	
Goal	Owner wants to create his own ads.	
Preconditions	Owners is signed in.	
Success End Condition	Owners successfully create his house ads.	
Failed End Condition	No new ads created.	
Primary Actors: **Secondary Actors:**	owner	
Trigger	Create ads request.	
Main Success Flows	**Step**	**Action**
	1	Owners requested for creating new ads.
	2	System will display a form where Owners have to fill up by giving number of beds, rent per month, gas and water bill, picture of that house, deposit paid, terms and conditions to follow acceptance.
	3	After filling up all the information in the form Owners can submit ads information.
	4	System will request for providing ads duration (how many days ads will show) with a minimum of 3 days and maximum of 2 months.
	5	System will calculate total cost for that ads by multiply per day ads cost with total ads duration and display it below ads duration.
	6	System now store the ads information and show newly created ads in created ads list.
Alternative Flows	**Step**	**Branching Action**
	5a	Owners choose cancel option.
Quality Requirements	**Step**	**Requirement**
		Not applicable.

Table 19: **Boost ads**

Use Case 19	Boost ads	
Goal	Owner wants to boost his created ads.	
Preconditions	Owner is signed in. Owner has created ads. Owner selected an ad.	
Success End Condition	Owner successfully boost his created ads.	
Failed End Condition	Owner failed to boost his created ads.	
Primary Actors: **Secondary Actors:**	Owner	
Trigger	Boost ads request.	
Main Success Flows	**Step**	**Action**
	1	Owner request to boost his created ads
	2	System will prompt Owner that how many days he wants to boost his ads.
	3	Owner will provide boost duration as days.
	4	Then system will calculate total cost for boosting that ads by multiply per day boosting cost with total boosting duration.
	5	System will boost the ad and update the ad as boosted ads.
Alternative Flows	**Step**	**Branching Action**
	4a	Owner selected not to boost the ad and clicked cancel.
Quality Requirements	**Step**	**Requirement**
		The feature should have minimal impact on performance, with quick response times and minimal lag or delay when boosting an ad.

Table 20: ***Edit ads***

Use Case 20	Edit ads	
Goal	Owners wants to edit his created ads.	
Preconditions	Owners is signed in. Owners has created ads. Owners selected an ad.	
Success End Condition	Owners successfully update his created ads.	
Failed End Condition	Owners failed to edit his created ads.	
Primary Actors: **Secondary Actors:**	Owners	
Trigger	Edit ads request.	
Main Success Flows	**Step**	**Action**
	1	Owners requested to edit selected ad.
	2	System will display all information about that ads.
	3	Now Owners can modify any information about that ad which he wants to.
	4	After filling up all the information in the form Owners can submit ads information.
	5	System will check Owners selected ad remaining time duration then ask owner whether he wants to extend the time.
	6	Owners can extend the time duration.
	7	System will calculate total cost for that ads by multiply per day ads cost with total ads duration and display it below ads duration.
	8	System store the changed information and update the ads information list.
Alternative Flows	**Step**	**Branching Action**
	4a	Owners choose cancel to update the information.
	6a	Owners don’t extend the time duration. System then store the changed information.
	2a	If ad is already boosted system will show the remaining boosted time.
	2b	If ads remaining time is zero days then system will prompt Owners to extend the time for the ads.
	2c	If owner won’t extend the time duration then system won’t update the ads information
Quality Requirements	**Step**	**Requirement**
		Not applicable.

Table 21: ***Remove ads***

Use Case 21	Remove ads	
Goal	Owners wants to remove his created ads.	
Preconditions	Owners is signed in. Owners has created ads. Owners selected an ad.	
Success End Condition	Owners successfully remove his created ads.	
Failed End Condition	Owners failed to remove his created ads.	
Primary Actors: **Secondary Actors:**	Owners	
Trigger	Remove ads request.	
Main Success Flows	**Step**	**Action**
	1	Owners requested to remove his selected ad.
	2	System will check the ads time duration is still remaining or not if it is not then system will delete the ads from the created ads list and alsofrom ads information list.
	3	After removing the ads, system will display notification that the ad removed successfully.
Alternative Flows	**Step**	**Branching Action**
	2a	If the ads time duration is still remaining, then the system will show how much money system will pay back to Owners.
	2a1	Owners will select next.
	2a2	See use case **Pay money**.
Quality Requirements	**Step**	**Requirement**
		Not applicable.

Table 22: ***Make Payment***

Use Case 22	Make Payment	
Goal	Owner want to make payment	
Preconditions	Owner are signed in.	
Success End Condition	Payment successful for created ads.	
Failed End Condition	Insufficient money on the selected financial service.	
Primary Actors: **Secondary Actors:**	Owner Bank system	
Trigger	Transfer money for creating ads.	
Main Success Flows	**Step**	**Action**
	1	System will give a request to the specific mobile financial bank selectedby the Owner for getting transaction of an amount.
	2	System will provide the amount of balance and the mobile number.
	3	Mobile bank will validate whether the given account has sufficient balance for transaction.
	4	Mobile bank will transfer that amount of balance from mobile bank tosystem bank.
	5	System then ask the stakeholders to enter the transaction id sent from mobile bank to the mobile number (financial service number).
	6	Getting full amount of money system will show a notification that the amount payment successful. And either add ads or remove ads or boost ads or edit ads.
Alternative Flows	**Step**	**Branching Action**
	3a	Account has insufficient balance.
	Step	**Requirement**
Quality Requirements		1.The feature should be secure, with appropriate measures in place to protect users' personal and financial information from unauthorized access or breaches.

Table 23: ***Add ads to favorite***

Use Case 23	Add ads to favorite	
Goal	Bookmark ads to see them in future.	
Preconditions	Users are signed in. Users at the ad's information page.	
Success End Condition	Ads will be added into favorites list.	
Failed End Condition	Ads won't added in the favorites list.	
Primary Actors: **Secondary Actors:**	Tenants System	
Trigger	Request for add ads in favorites list.	
Main Success Flows	**Step**	**Action**
	1	Users will click add to favorite button beside each ad.
	2	System will add the ads in the favorites list.
	3	System will show a notification that ads is added to favorites list.
Alternative Flows	**Step**	**Branching Action**
	1a	If the ads already added to the system then it will remove the ads from favorites list.
	1a1	It will show notification to the user that ads are removed from favorites list.
Quality Requirements	**Step**	**Requirement**
		1. The feature should allow users to easily add ads to their favorites with a single click or tap, without requiring a lot of user input or interaction.

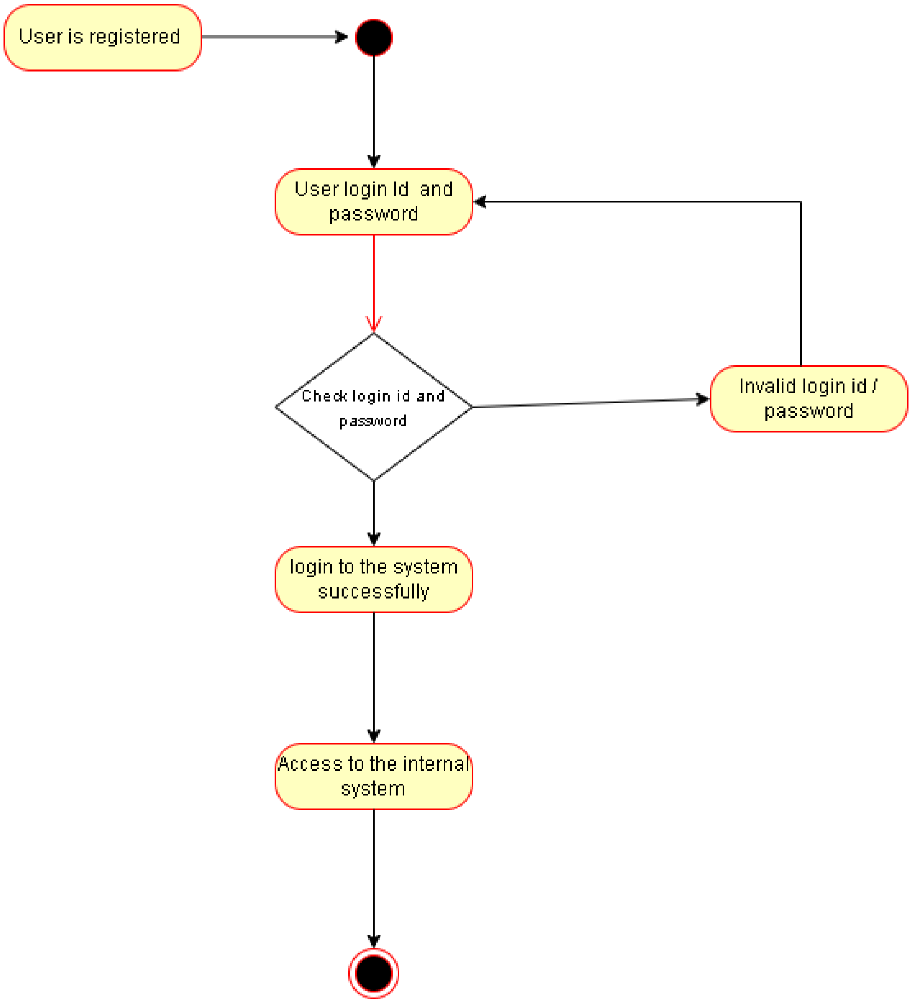
Table 24: ***Make Conversation***

Use Case 24	Make Conversation with Admin	
Goal	To communicate with the admin of the house rental system regarding any issues or queries.	
Preconditions	The User is signed in.	
Success EndCondition	The user has successfully communicated with the admin.	
Failed End Condition	Users failed to communicate.	
Primary Actors: **Secondary Actors:**	Owners,Tenants	
Trigger	The user wants to communicate with the admin.	
Main Success Flows	**Step**	**Action**
	1	The user opens this system and click to the "Send" button.
	2	The system presents the user with chat.
	4	The system connects the user with an available admin.
	5	The user can type a message to the admin, such as a question or a request for assistance.
	6	The admin responds to the user's message and provides the required assistance or information.
	7	The user can continue the conversation with the admin until the issue is resolved.
Alternative Flows	**Step**	**Branching Action**
	1	If the user encounters any errors or issues during the process, the system provides appropriate error messages and prompts the user to try again or contact support for assistance.
	2	If the preferred communication method is not available, the system can prompt the user to select a different method or offer to connect the user with an available representative as soon as possible.
	3	If the user encounters any errors or issues during the process, the system provides appropriate error messages and prompts the user to try again or contact support for assistance.
Quality Requirements	**Step**	**Requirement**
	1	The feature should provide timely and appropriate notifications to users when they receive new messages, with clear information about the sender, the content, and any relevant details

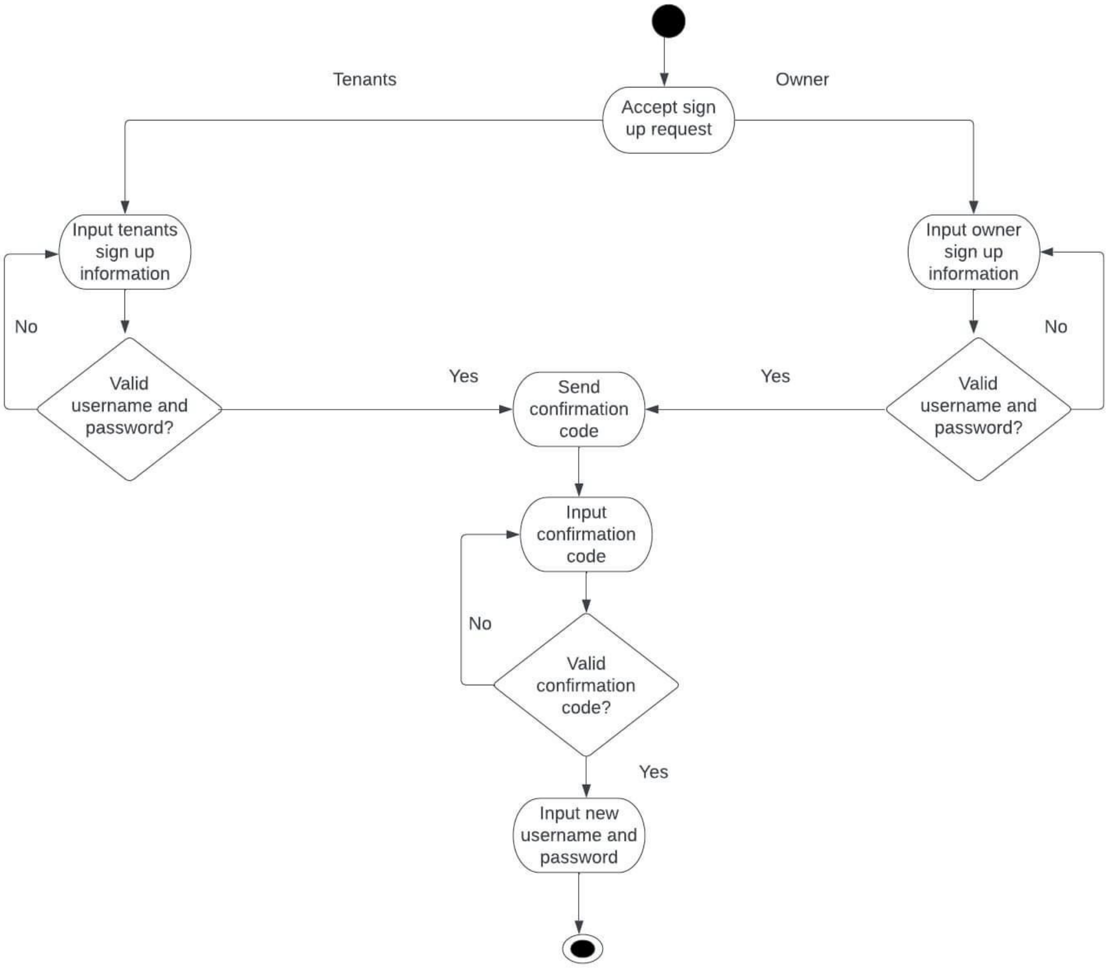
8. Activity Diagram

An activity diagram is a graphical representation of an executed set of procedural system activities and considered a state chart diagram variation. Activity diagrams describe parallel and conditional activities, use cases and system functions at a detailed level. Activity diagram for house rental management system are given below.

8.1 System access control

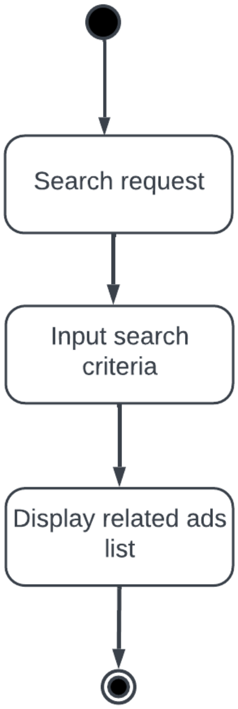

User is registered
User login Id and password
Check login id and password
Invalid login id / password
login to the system successfully
Access to the internal system

8.2 Create Account

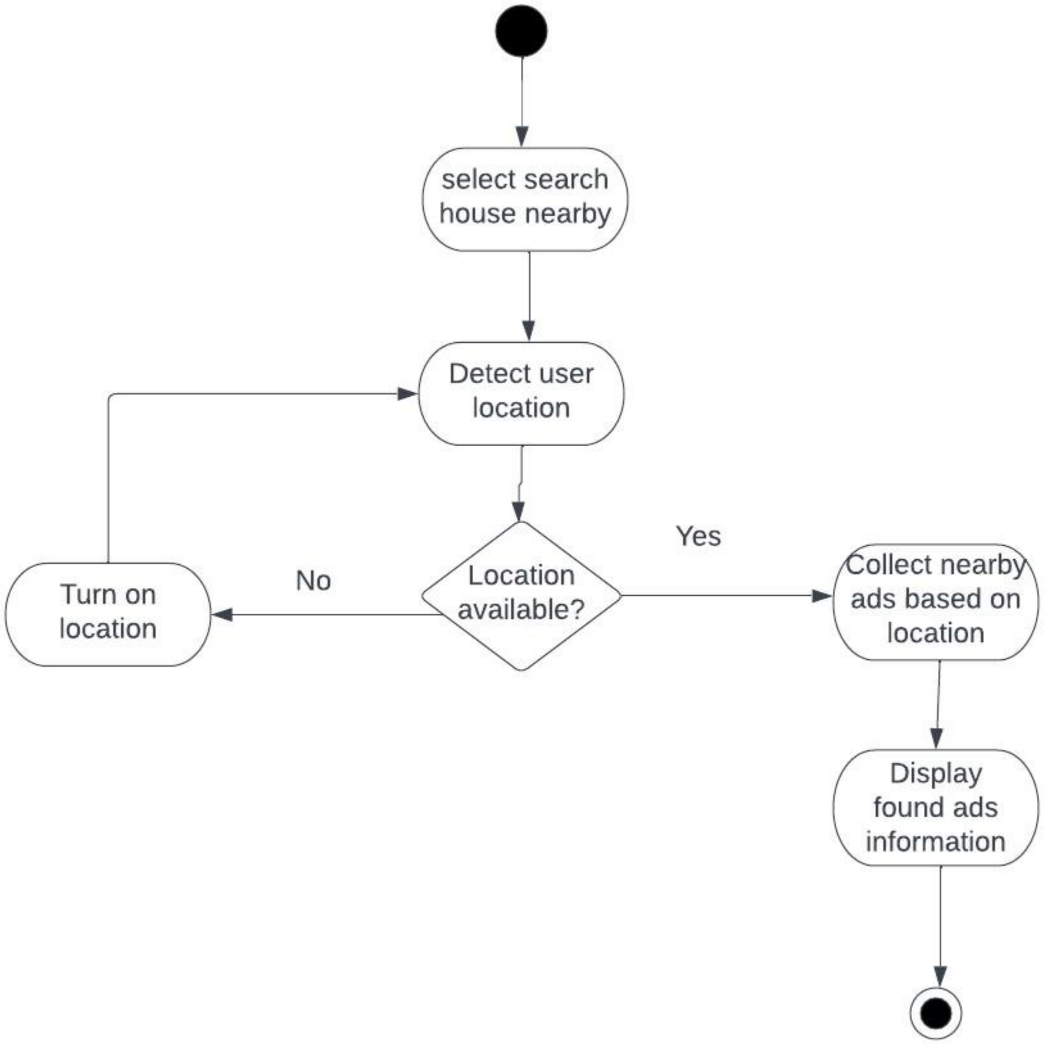


Tenants
Accept sign up request
Owner
Input tenants sign up information
Input owner sign up information
No
No
Valid username and password?
Yes
Send confirmation code
Yes
Valid username and password?
Input confirmation code
No
Valid confirmation code?
Yes
Input new username and password

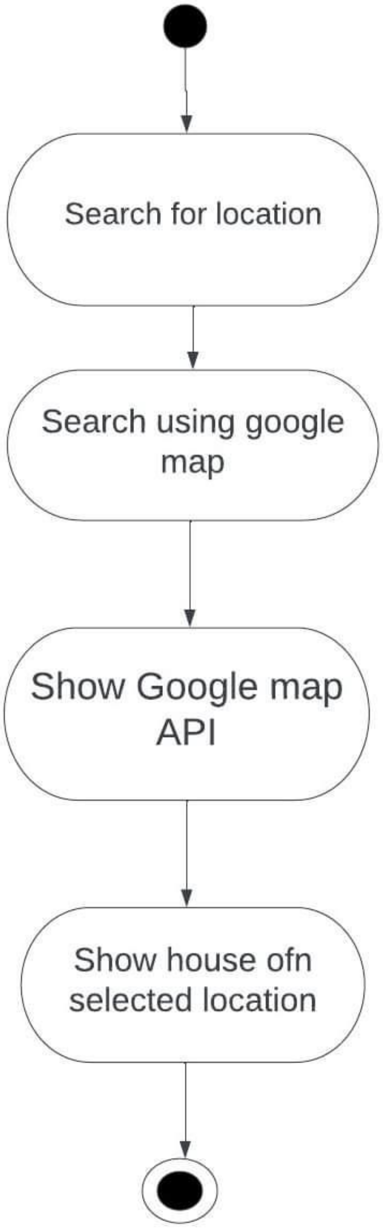
8.3 Search For House Ads



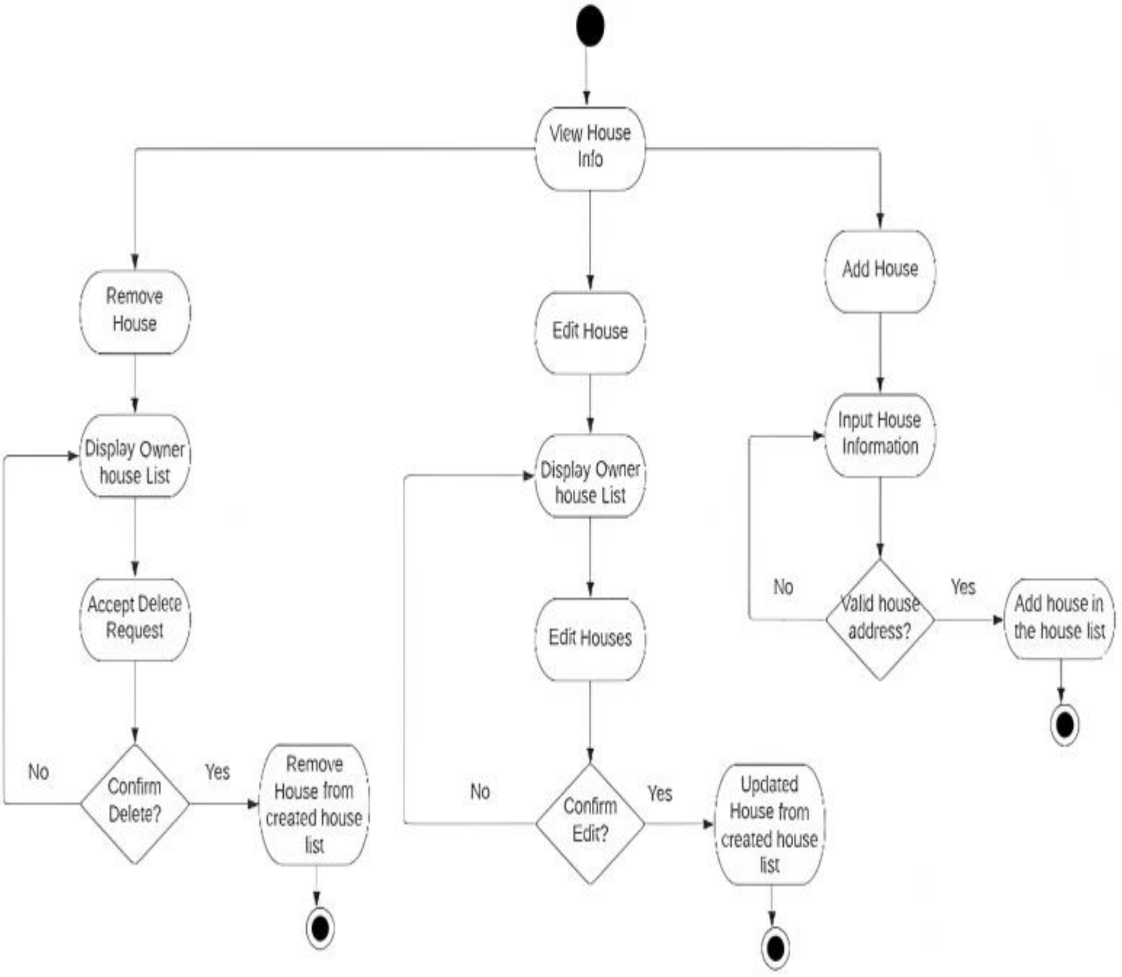
8.4 Search House Ads Nearby


select search house nearby
Detect user location
Location available?
Yes
No
Turn on location
Collect nearby ads based on location
Display found ads information

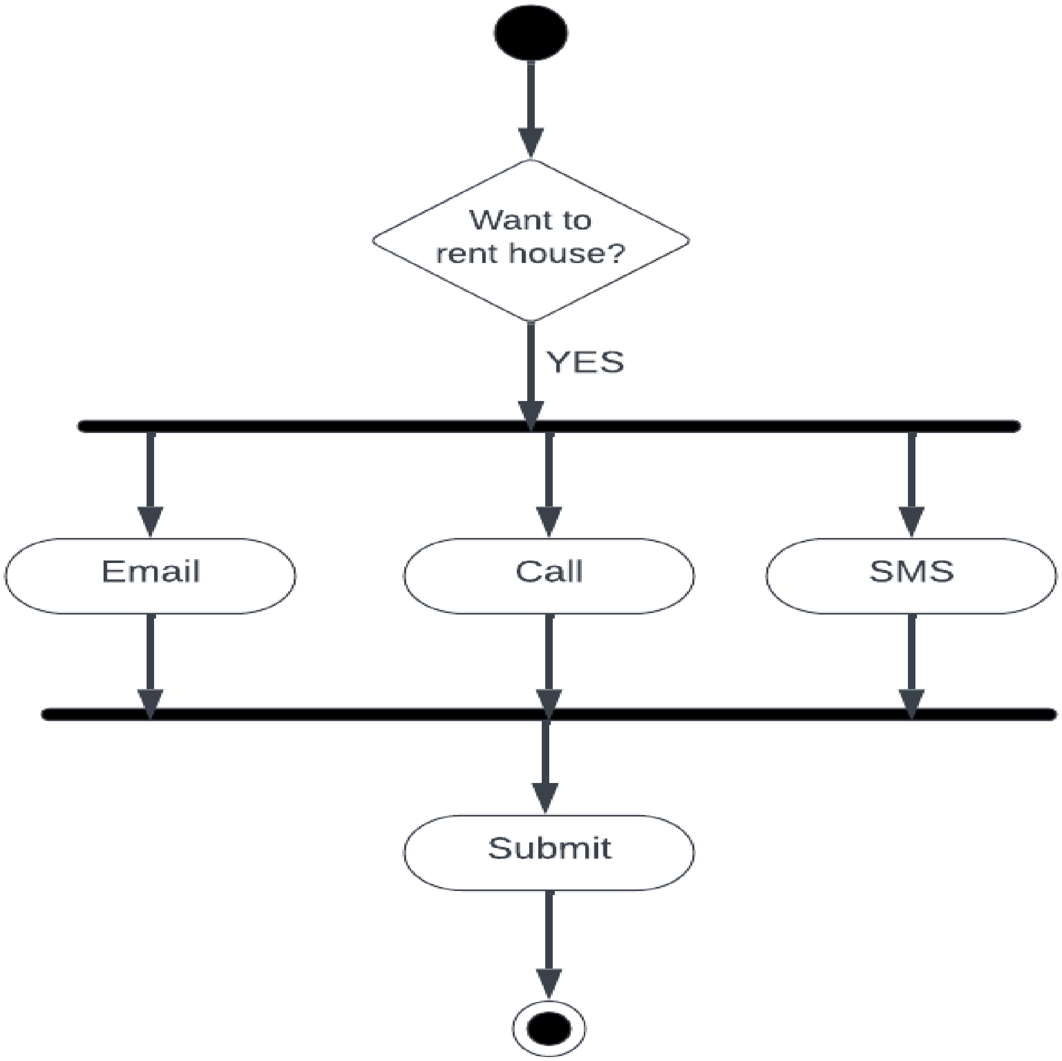
8.5 Search House By Google Map



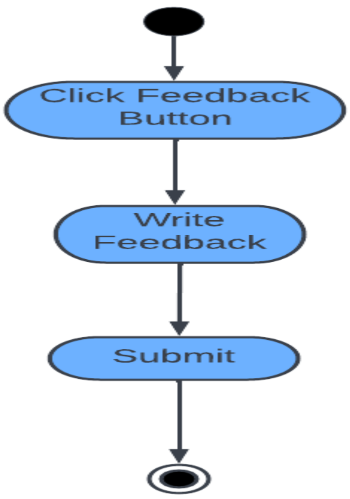
8.6 View Houses



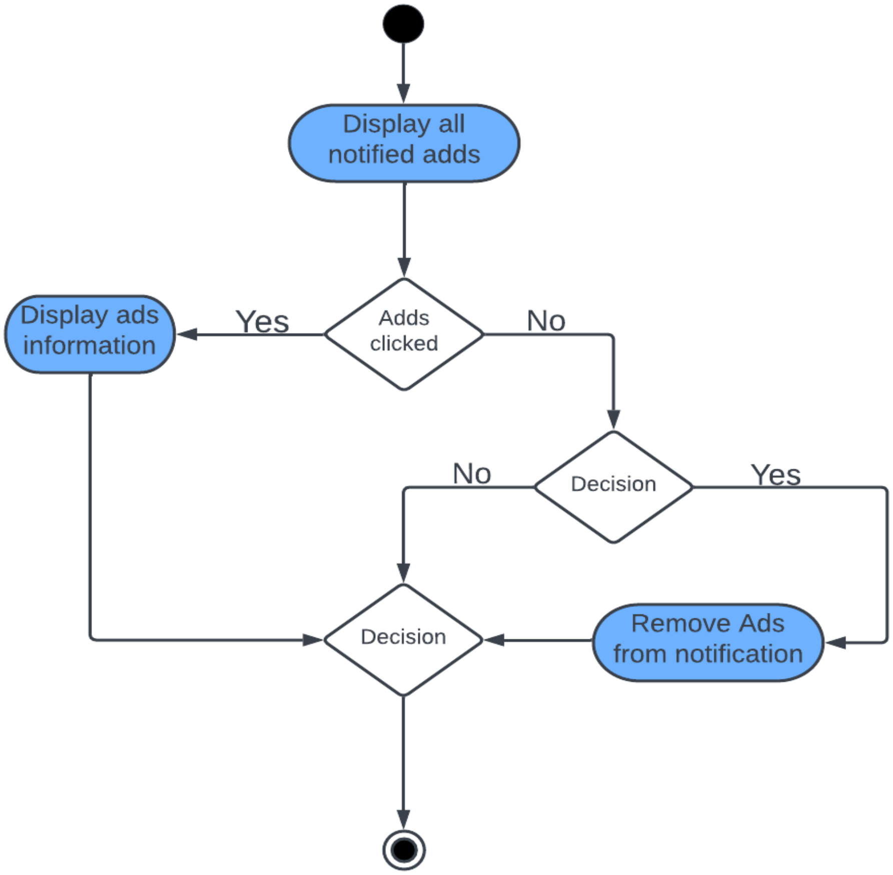
8.7 Rent House



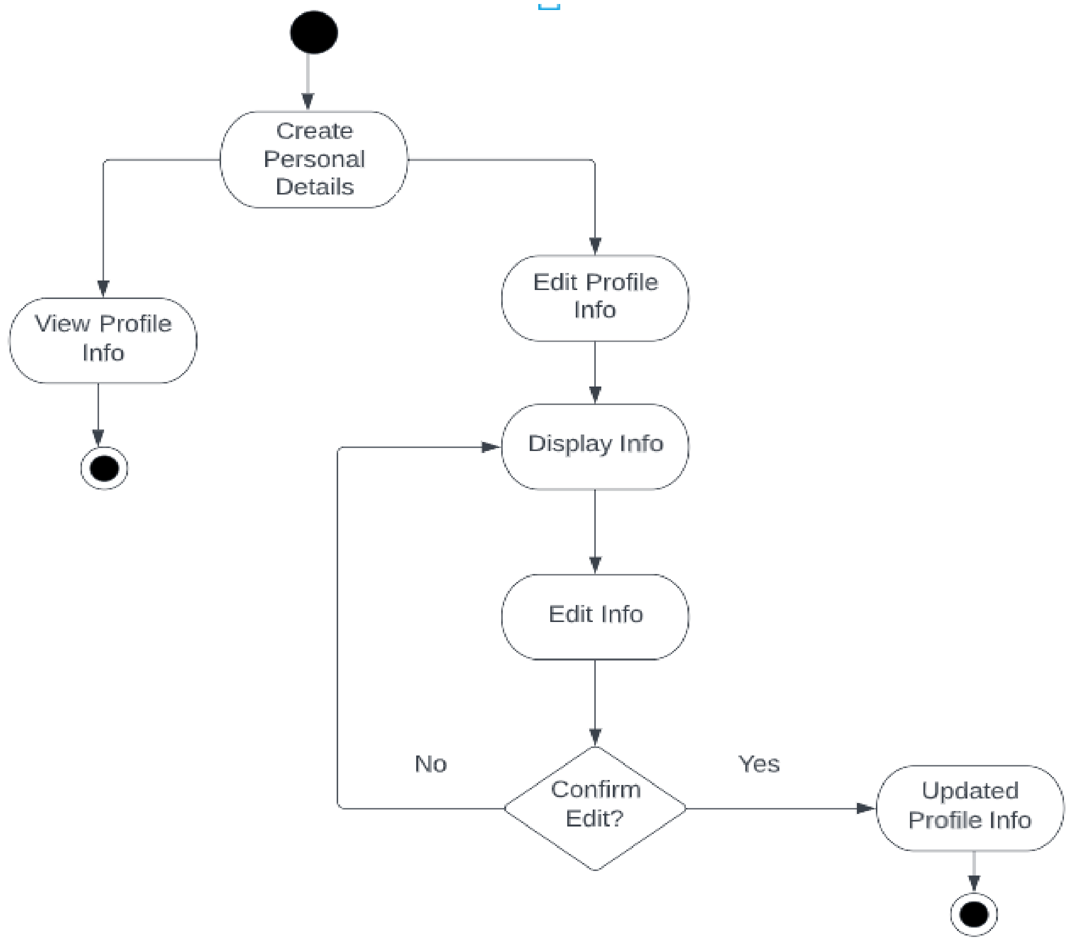
8.8 Post Feedback



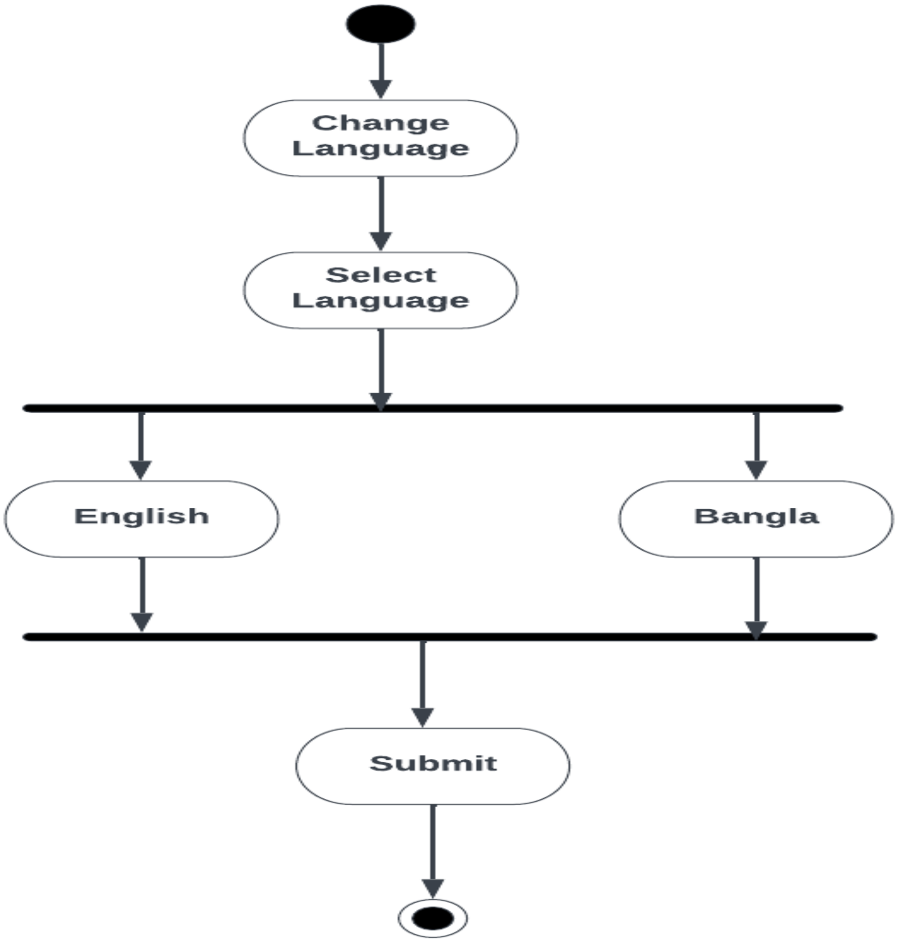
8.9 View Notifications



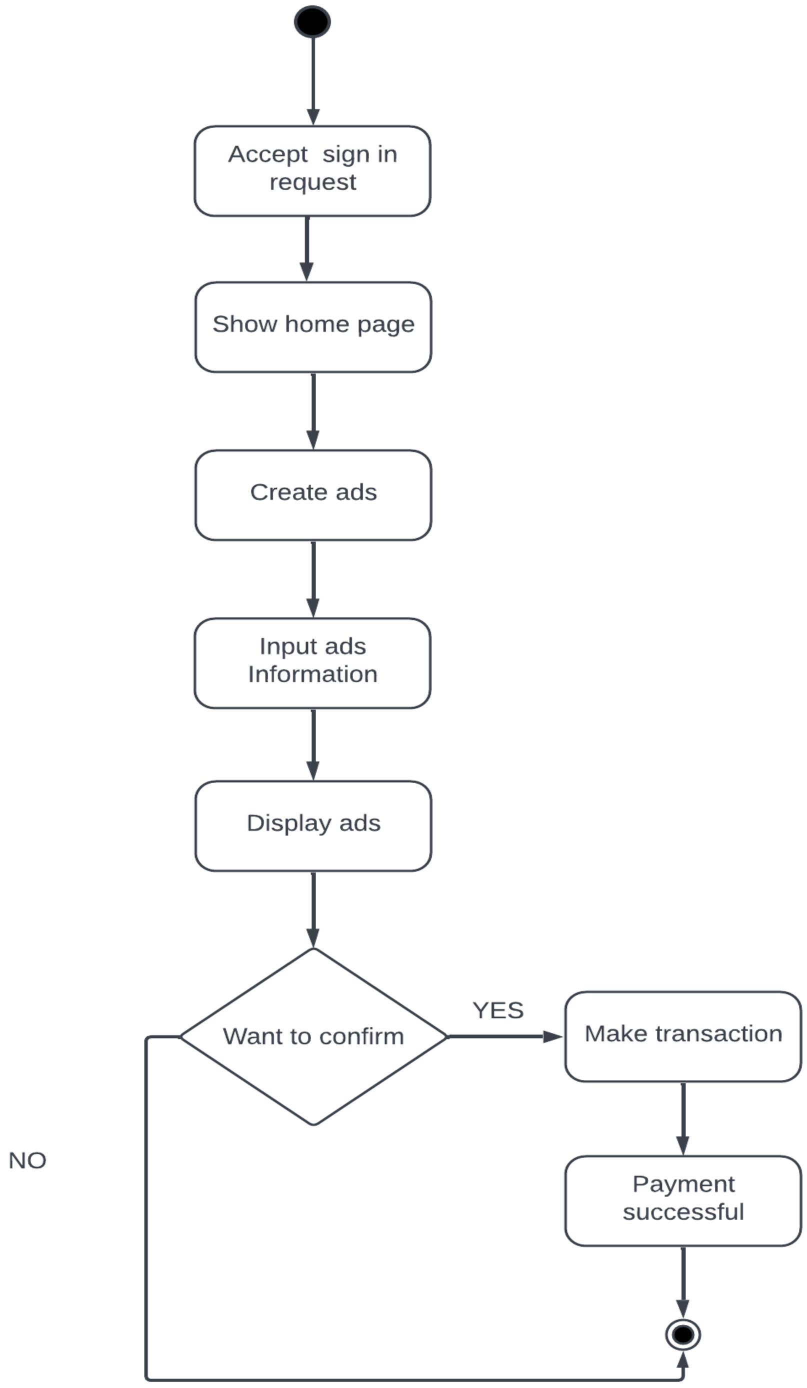
8.10 Edit Profile Info



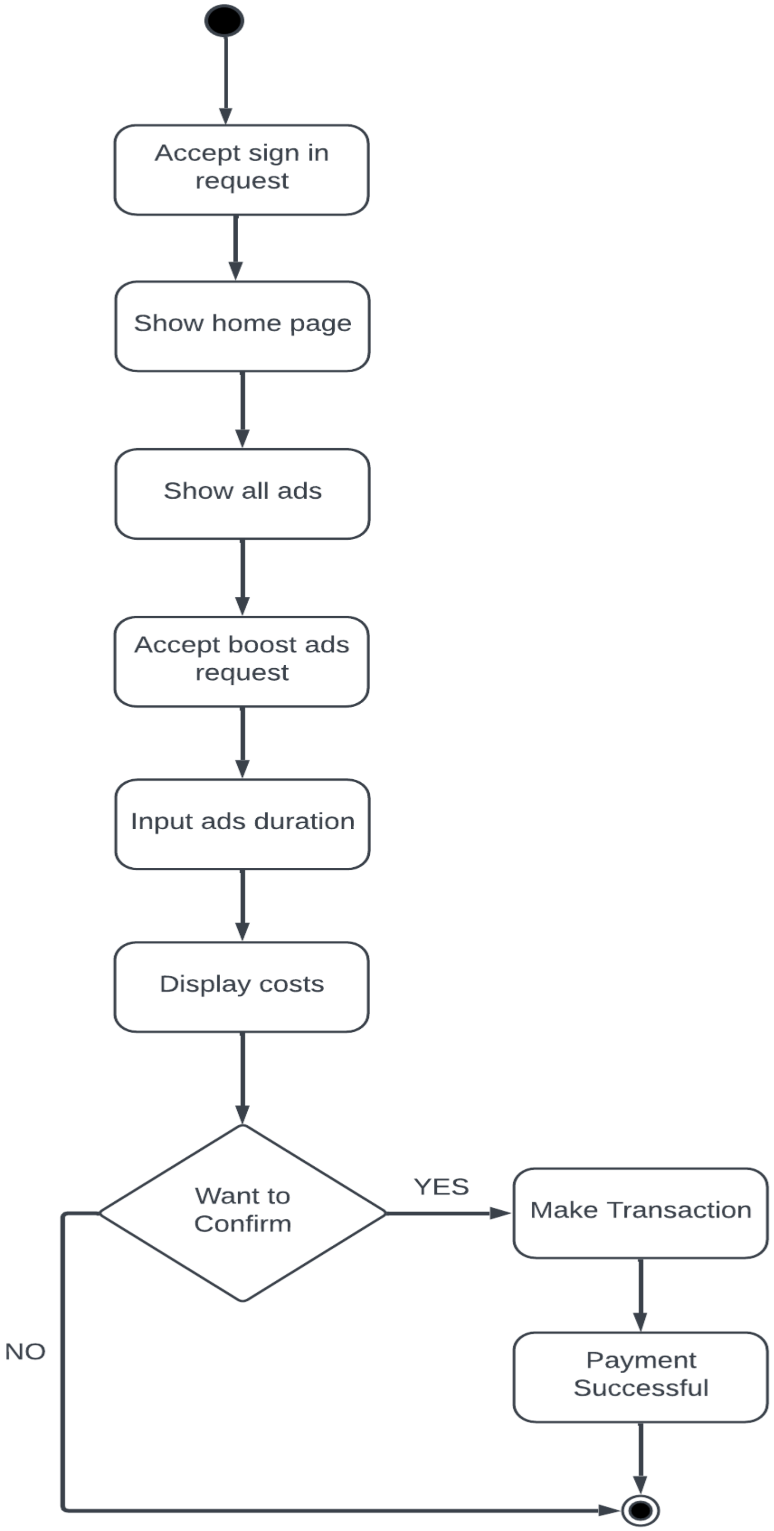
8.11 Change Language



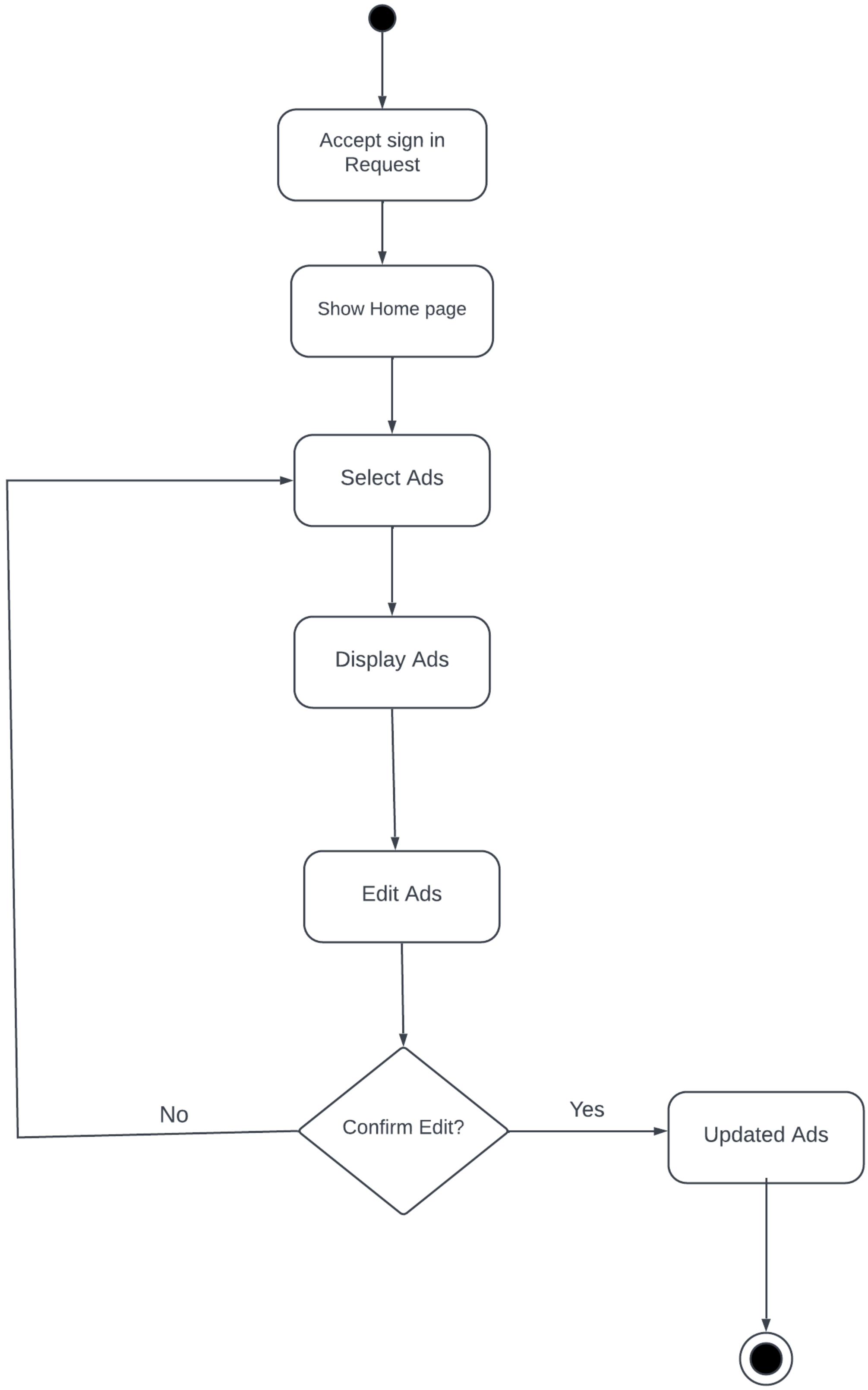
8.12 Create Ads



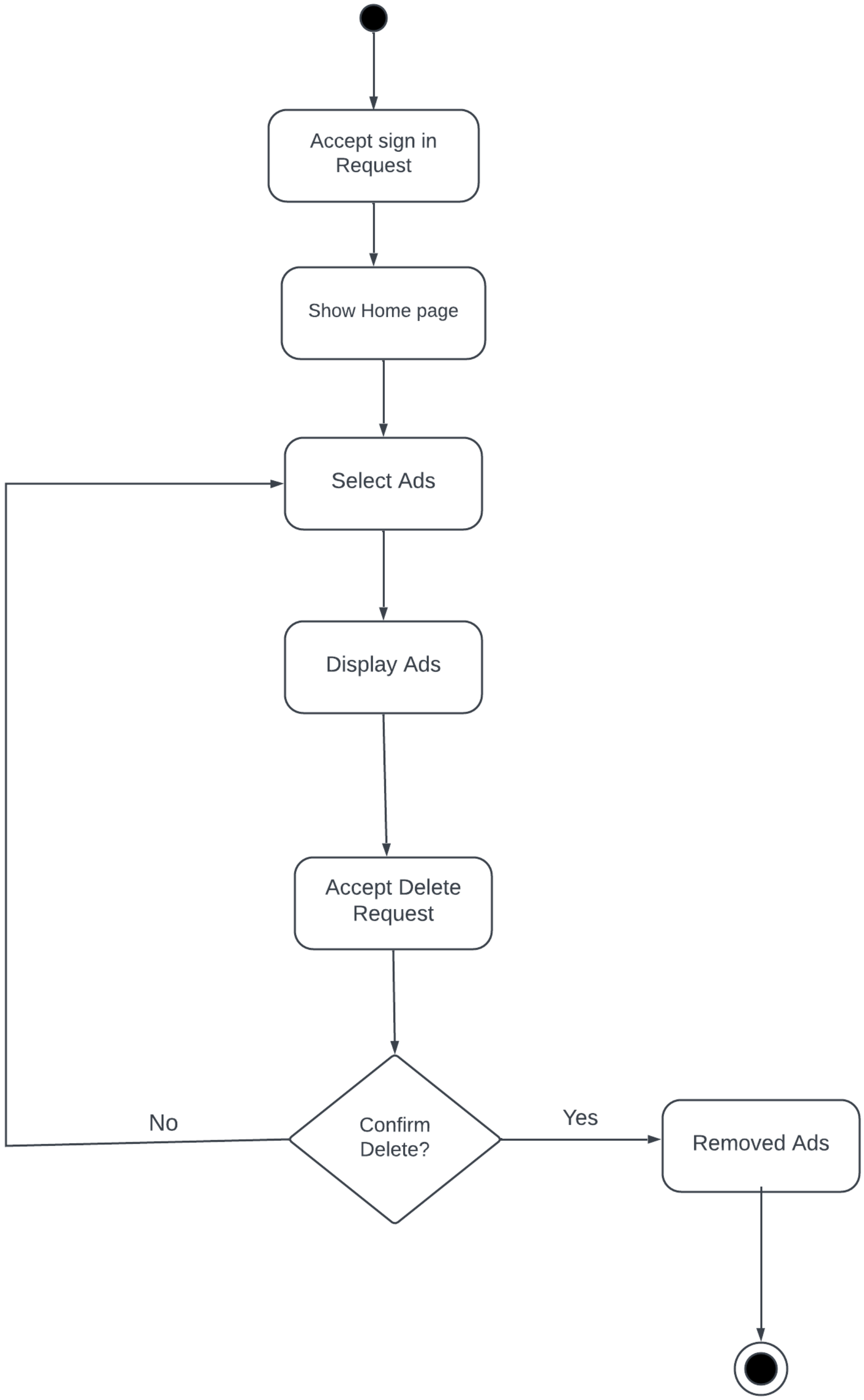
8.13 Boost Ads



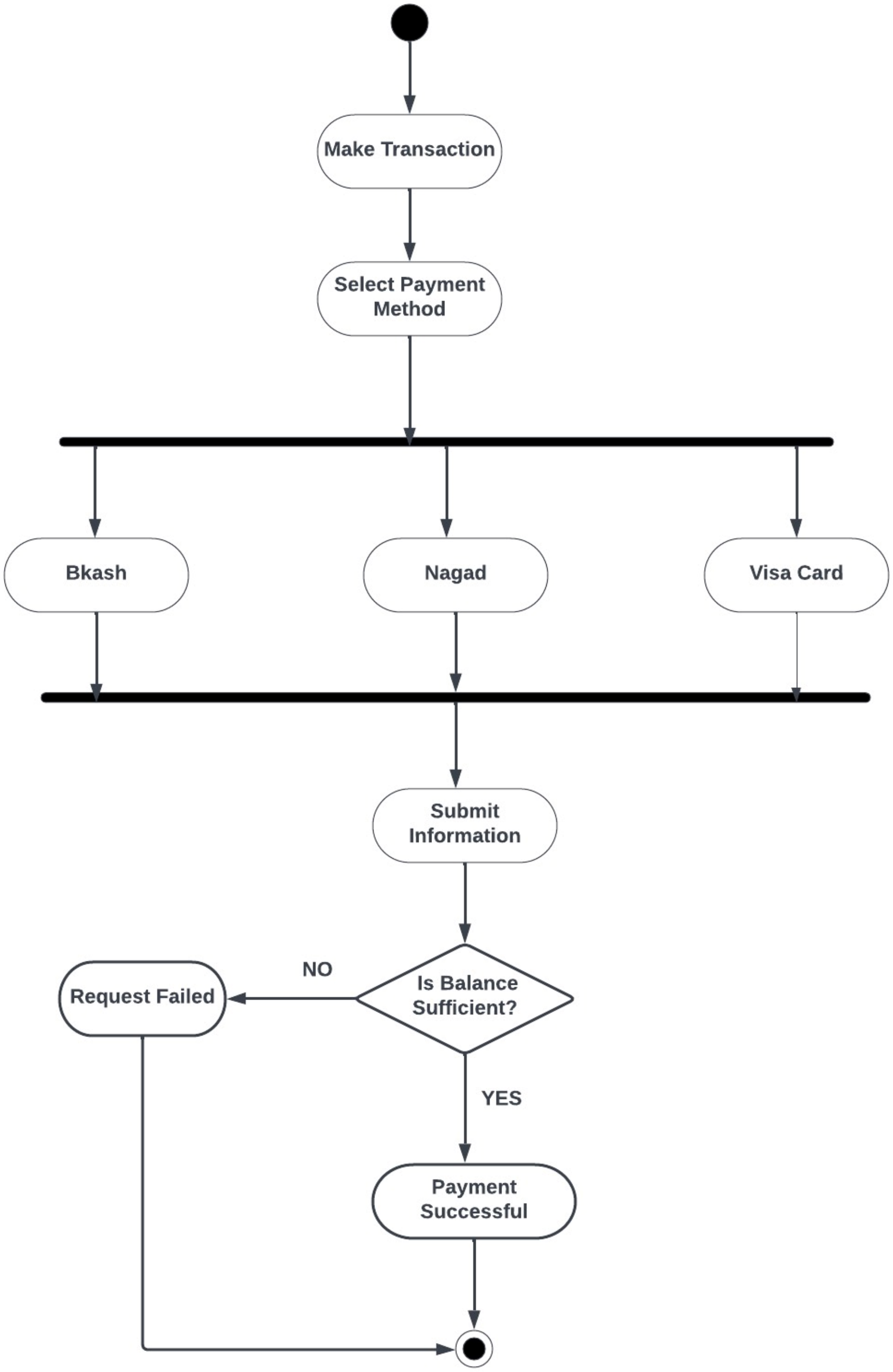
8.14 Edit Ads


Accept sign in Request
Show Home page
Select Ads
Display Ads
Edit Ads
No
Confirm Edit?
Yes
Updated Ads

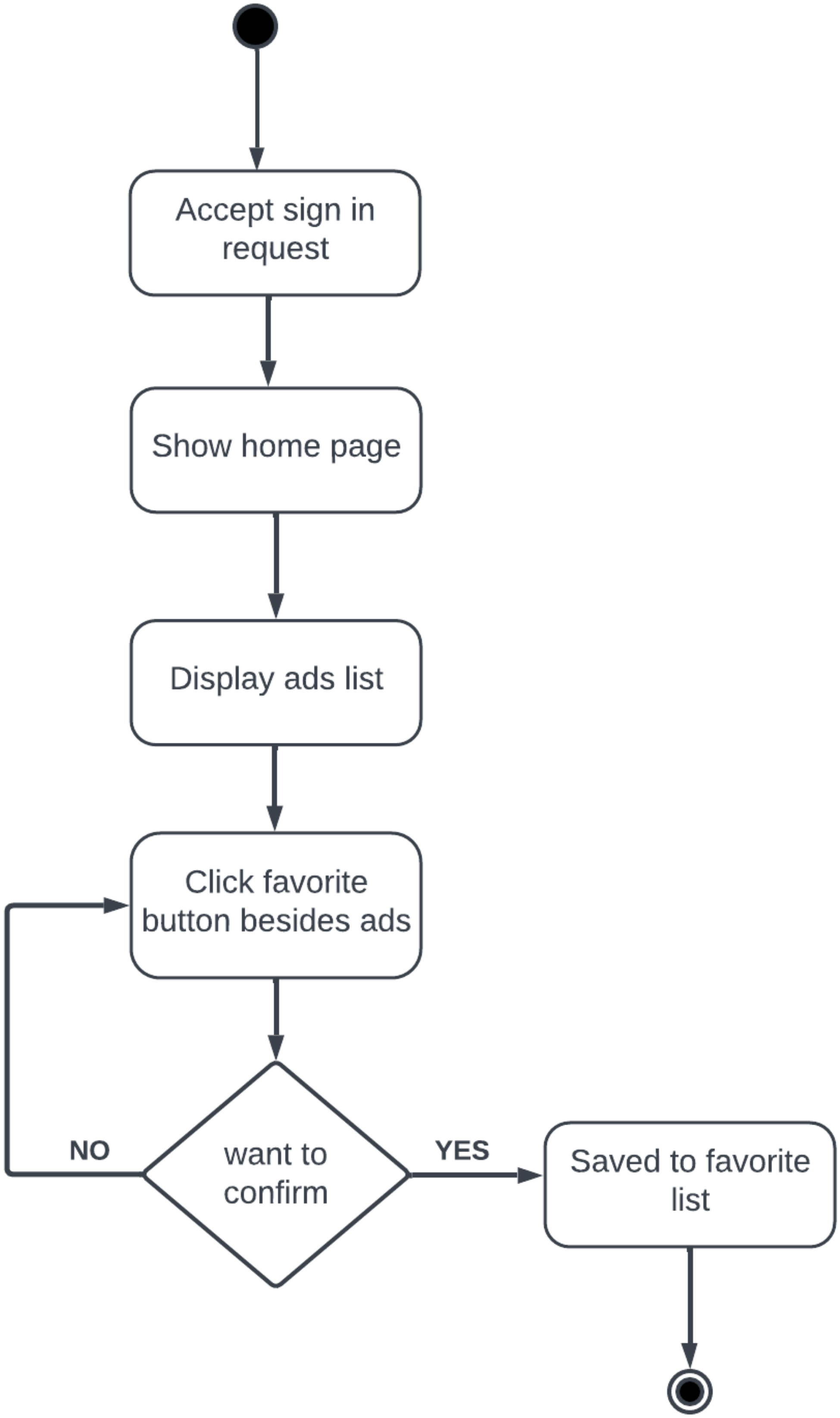
8.15 Remove Ads



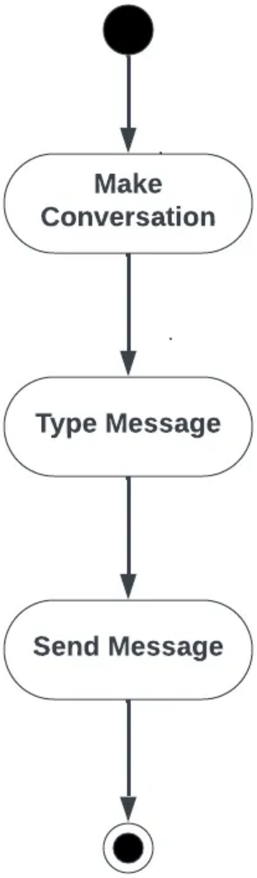
8.16 Make Payment



8.17 Add ads to favorite



8.18 Make Conversation



9. Requirement Traceability Matrix

Traceability Matrix

A Traceability Matrix is a document that co-relates any two-baseline documents that require a many-to-many relationship to check the completeness of the relationship. It is used to track the requirements and to check the current project requirements are met.

The Requirement Traceability Matrix (RTM)

It is a critical document in software development that links and tracks user requirements with corresponding test cases. It serves as a one-stop-shop for capturing all client requirements and their traceability throughout the software development life cycle. The main purpose of RTM is to validate that all requirements have been thoroughly tested and no functionality goes unchecked during the testing phase.

Why RTM is Important?

RTM is important for several reasons. Firstly, it helps the testing team to understand the client's requirements and ensure that the output product is defect-free. By thoroughly understanding the requirements, testers can create positive and negative test cases, which are further split into scenarios and test cases. RTM provides a way to track the testing of each requirement and scenario to ensure that no requirement is left untested.

Furthermore, RTM helps to eliminate any confusion regarding the requirements, their corresponding test cases, and their current status. It also serves as a record of the testing activities done for a specific product, allowing the testing team to monitor their progress and identify any gaps. RTM is a critical tool that ensures the completeness of the relationship between the requirements and test cases. It helps the testing team to deliver a high-quality product that meets the client's requirements and provides a clear record of the testing activities throughout the software development life cycle.

RTM also helps to increase the traceability of changes to the requirements during the development process. It enables the development and testing teams to quickly identify the impact of any changes and take necessary actions to ensure that the requirements continue to be met. This helps to prevent scope creep and ensures that the product is delivered on time and within budget.

Parameters for Requirement Traceability Matrix

- Requirement ID
- Requirement Type and Description
- Test Cases with Status

A requirement traceability matrix can be

- Show the requirement coverage in the number of test cases
- Design status as well as execution status for the specific test case
- If there is any User Acceptance test to be done by the users, then UAT status can also be captured in the same matrix.
- The related defects and the current state can also be mentioned in the same matrix.

9.1 Use Case

Use Case No	Use Case Name
UC1	System access control
UC2	Create Account
UC3	Search for house ads
UC4	Search house ads nearby
UC5	Search house by filtering
UC6	Search house by Google Map
UC7	View houses
UC8	Rent House
UC9	Post Feedback
UC10	View Notifications
UC11	Edit profile info
UC12	Change language
UC13	Add house
UC14	Edit house
UC15	View houses
UC16	Remove House
UC17	Display Ads
UC18	Create ads
UC19	Boost ads
UC20	edit ads
UC21	remove ads
UC22	Make Payment
UC23	add ads to favorite
UC24	make conversation
UC25	Manage feedback

9.2 Test Cases

TC1	Maijdee, Noakhali
TC2	Detect User location using google map
TC3	Rahima akter,Gender-Female,Email-Rahima123@gmail.com
TC4	MA MANJIL, HN-120,RN-10,PICTURE.
TC5	Try to remove ads.
TC6	Try to edit or update existing ads.
TC7	Try to access all ads.
TC8	Try to communicate with admin.
TC9	Try to Boost an existing post.
TC10	Try to search houses using google maps
TC11	Try to change language to bangla
TC12	Try to communicate with admin via message
TC13	Try to update tenant's information.
TC14	Try to add favorite ads.
TC15	Try to give 5 star feedback
TC16	Try to pay money 500 tk for boosting an ad using bkash.
TC17	Stakeholder try to update his/her name.
TC18	Try to reset password.
TC19	Users try to remove his/her profile information.
TC20	Check where system send notify message or not.
TC21	New house added on location sonapur,noakhali

Requirements Traceability Matrix

Project Name	**House Rental Management System**	**Business Area**			**Global**		
Project Manager	**Sanjida Akter Samanta**	**Business Analyst Lead**			**Mir Mohammod Tahasin,Wakil Ahammad**		
QA Lead	**MD Foysal Mahmud, Rubya Rashed**	**Target Implementation Date**					
Category / Functional Activity	**Requirement Description**	**Use Case Reference**	**Design Document Reference**	**Code Module / Reference**	**Test Case Reference**	**User Acceptance Validation**	**Comments**
FR1	Landlords and tenants can create account and login to the system	UC1, UC2			TC3	Verified	
FR2	Landlords wants to adds a new house to the system	UC13			TC4	Pass	
FR3	Tenants wants to search house ads by specific criteria	UC3, UC5			TC1	Pass	
FR4	Search house ads based on User location	UC4			TC2	Found	
FR5	Stakeholders will add favorite ads to remember them	UC23			TC14	Pass	
FR6	Stakeholders wants notification when a house is	UC10			TC21	pass	

	available for rent nearby his location.						
FR7	Tenants can communicate with landlords	UC24			TC8	Pass	
FR8	User want to reset password for making it not vulnerable	UC11			TC18	pass	
FR9	User can search houses using google map	UC6			TC10	Pass	
FR10	User can change the existing system language.	UC12			TC11	Pass	
FR11	Tenants and house owner wants to communicate with the admin for any information or problems.	UC24			TC12	Pass	
FR12	Tenants wants to add reviews and ratings for houses	UC15			TC15	Verified	
FR13	House Owner wants to boost his ads	UC19			TC9	Pass	
FR14	Landlords wants to Remove the ads from the system	UC21			TC5	Pass	
FR15	The system should have a payment gateway for for getting the money from the users. The users will be given “Bkash, Nagad, Visa Card/Master Card” for	UC22			TC16	Pass	

	giving the fees for ads boosting.						
FR16	House owner wants to view about their properties	UC3,UC4			TC7	Pass	

10. Appendix

10.1 Prioritization of requirements

We've prioritized the functional requirements by following Three-level Scale technique.

10.1.1 Three-level Scale

When a Business Analyst categorizes the requirements in any of the ordering or ranking scale, it is subject to the analyst's understanding of the business. Many analysts suggest that this method has some drawbacks and advocate methods that have more than one scale.

The priority of the different sections in the diagram is indicated by the numbering system. The highest priority is given to important items, while the lowest priority is given to less urgent items.

High Priority: These requirements are critical and cannot be ignored. They pertain to compliance or contract obligations and must be addressed in the current release. Failure to implement these requirements can have negative consequences for the business.

Medium Priority: These requirements are important but not as pressing as high priority items. They should be addressed after completing the high priority items. Within this category, items on the right side of the dividing line are of higher priority.

Don't Do These: These items are less important but still urgent. They should be addressed after completing the higher priority medium priority items. The items on the right side of the dividing line within this category have a higher priority.

Low Priority: These items are neither important nor urgent and can be addressed at a later time, after completing the items in the first three categories.

The priority order should be followed by starting with the bottom-right corner of the high-priority section and working up and to the left.

10.1.2 Prioritization of the requirements of House Rental Management System

FR1 – High priority: Its essential requirement for our system. House Owner and tenants can create account and login to the system.

FR2 – High priority: This feature is important because here are landlords to list their properties for rent, including details such as location, price, number of bedrooms, and any other relevant information.

FR3 – High priority: Tenants will find homes by certain criteria, so this needs to be put in place in advance so they can find them easily. So it is essential for our system.

FR4 – High priority: Because our system is based on house location, it is crucial for our system to function properly first.

FR5 – Medium priority: To remember ads, users will save them. It entirely depends on the user.

FR6 –Low priority: This feature is basically less important for the system. Tenants wants notification when a house is available for rent nearby his location.

FR7 – High priority: stakeholders get in touch with landlords via SMS, email, and phone call if they have a vacant home for them after looking at house ads. Therefore, it is crucial to our system.

FR8 – Medium priority: User wants to change password to make it secure, so it’s highly crucial to our system

FR9 – High priority: user can easily explore various neighborhoods and get a sense of the closeness of neighborhood amenities, such as schools, shopping, and public transportation, by using Google Maps to search for homes.

FR10 –Low priority: If necessary, users can change the language of the current system.

FR11 – Medium priority: Tenants and property owners wish to communicate with the administrator regarding any information or problems to our system.

FR12 – Low priority: For house rent, rating is not as important. And looking for house ads is not dependent on the rating on our system.

FR13 – High priority: Boosting existing ads in a house rental management system can increase the visibility of the property to potential renters and improve the chances of finding a suitable tenant.

FR14 – High priority: Landlords can remove any ads that are not important to their system. So that the work of his system can be arranged ads in new way.

FR15 – High Priority: Transaction money is very important and it must be secured enough to gain faith of the owners so that they can pay money without any hesitation.

FR16 – High priority: House owner can view details of their properties, which will help them understand what their system will look like.

DR1 – High priority: The images and files from the stakeholders must be stored in a efficient manner so that it will be secured for future usage. It is our systems high priority to store and manage those data

DR2 – High priority: Our system must store the property data provided by the house owner. It is very important features to attract tenants.

DR3 – High priority: Our system must store the tenant’s data provided by the tenants. It is very important features to attract tenants. Tenants information like name, contact, payment history must be stored the tenants information.

DR4 – High priority: Financial information is so important for our system. Those information can be used in future for various security reasons and technical reasons.

DR5 – High priority: Ads is so much important for our system. Ads is the only source for our system. So our system must store and do analytics of the ads of the house owner. It will help to gain new customer.

PR1 – High priority: Our system must maintain high speed and latency throughout searching time. This is very important for user interaction into the system.

MR1 – High Priority: It’s important to update the database of the newly renting peoples because police may need to get any information of the current tenants of any house.

AR1 – High Priority: To attract the online users to use our system more and more, color and font are so much important part. Just to ensure that users do not get distracted for the color combination or font size and font color