A major project final report on

RoomFinder.com

Submitted in Partial Fulfillment of the Requirements for

The Degree of **Software Engineering**

Under Pokhara University

Submitted by:

Akhil Chhetri, 15768

Barsha Thapa, 15752

Himsikha Rai, 15756

Sagar Thapa Shrestha, 15759

Date:

20th Nov 2019

Department of Software Engineering

NEPAL COLLEGE OF
INFORMATION TECHNOLOGY

Balkumari, Lalitpur, Nepal

Acknowledgement

The whole team of the project RoomFinder is extremely thankful for the valuable guidelines and feedbacks from the respected Teachers and supervisor. We are glad and feeling privileged to be able to present our process of transforming idea to a project that simulates the real-world problem-solving product.

We are extremely thankful and deep gratitude towards our project coordinator respected Dr. Roshan Chitrakar for constantly guiding us towards the best path possible we could implement while doing this project.

We wish to record a deep sense of respect and gratitude to respected Er. Bhushan, project supervisor for providing us every kind of support feasible in the journey of idea to a minimal viable product. We would also like to thank him for his valuable guidance and constant supervision as well as for providing necessary information regarding the project and also for his support in completing the project.

This project has a great memory within us in teammates from when we ideate and become more curious towards the implementation. The whole project is the output of efforts from each team members, supervisor, project coordinator and valuable feedbacks in each of the stage of the project presentation.

The journey from the idea to implantation analyzing the real scenario and majority of the issue and trying to draft a simple module to impact greatly was the greatest learning from whole team members. We followed the most efficient and standard approach of engineering the product from very beginning to the end. This project makes us the practically stronger to implement what we have learned from the whole engineering period. We learned things which could not be learned from the theoretical classes and lectures, but we got the chance to implement practically the critical knowledge in real field with aiming to solve the real-world problem.

We would like to thank Nepal college of Information Technology and Pokhara University for such a wonderful design of the syllabus which is impactful to learn and research the real scenario.

Finally, we are too much privileged to learn, implement and improve ourselves in each stage with the help of teachers, colleagues, classmates and friends have been directly or indirectly linked with the project and accomplishing this project.

Abstract

The problem of getting the desired property for rent in Kathmandu or overall Nepal is huge as we mostly faced. In most of the cases, people are wasting huge money and time to get desired property to pay for middle man who search and give information about vacant rooms and flats. As this is the problem which could get a robust solution through technology and Software services as some of the groups and pages using the general platform for social network also provides kind of easier solution to the needy people. So as a Major project for eight semester under Pokhara University, we intended to solve gap between the property owner and property seekers.

The purpose of this project is to create a web application which would help people to easily find rooms and tenants which are available for rentals. Furthermore, they can search nearby rooms/flats and their interest related property through our website's feature. The project is intended to provide a Software as a Service to both vendors and renters with the common web platform.

The first thing a user has to do is to list an empty room for free and get verified by adding your social media (OAuth) and getting background checked. Nepal has one of the most active houses renting systems in the world, i.e. we have literally thousands of rental houses to choose from. There are three main types of people who look for shared accommodation.

RoomFinder is the Web based application which acts as the platform to connect the renters and owners to make finding the rental property. Renter could view their suggested property around as per the price, facility and locations. Our system will use Collaborative filtering to collect the user’s data and sentiments through it which relates the most viewed and their preferences to view display the property as per desired. Along with it, pre booking feature will also available for renters.

The main idea behind this is to improve the finding rental room services, reliability as well as ensuring the trust to people and also to provide timely response to the people when needed and giving people what they prefer and their findings easier. RoomFinder will suggest the related rooms and flats to the renters as per their preferences and location they want. We will use collaborative filtering and graph theory to suggest the nearest location rental. Our entire system will fetch the data from our API which is used to integrate in front end using Angular Js.

Keyword: Rental service, GPS, REST API, Recommendation System, Chat, Booking, PIP, NPM, Django etc.

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Background

This system provides a suitable platform to the visitors by providing all the information about the rental rooms and services required in the real time. Also, this system shall be system with essential properties such as clean web design, detailed users descriptions, quick and accurate search functions, and customer focused content etc. An Online Room Finder will provide the Information about Rooms/Flats/Houses which is available for Rent. RoomFinder web application will make easy to find the location of Rooms/Flats/Houses, select no. of rooms and other facts by the renter. It will make easy to upload the location, contact No., expected rent, No. of rooms, Facilities and other information by landlord/room-master. This application will be able to show the rooms/flats/houses with in a particular area selected by user. A web-based application has a major advantage that it can be accessed from anywhere; software doesn't have to be installed other local computer for its use. Just an inbuilt internet browser is enough to run the application. Our software shall be able to manage the renting and interactions between the renters and the person who makes their room on rents of and other general functions in a web based computerized way. The software shall involve the Google map for easy and exact location of the room. With the help of this system one can easily see the publicly available features such as browse list of rooms which has been posted by the user, views details of room owner, features and prices, and view other static contents of the site. Registered users can view all publicly available features and in addition they can purchase products by adding them into shopping cart. Admins can manage all the contents of products from the backend. Room Finder is a simple yet powerful tool where people can rent and sell their room for rent from their own comfy homes rather than visiting the homes to ask for available rooms for rent or a physical agents/brokers.

1.2 Problem Statement

In the present context most of the people are still relying on traditional agents or brokers either through physical connection or virtual through many social media and common platforms. Many people still prefer to visit the homes to search for rooms which are available for rent. They prefer going to physical agents themselves to rent because many rental rooms finding website are failing to provide the trust and quick and accurate services and also there are very few of these kinds of websites. Rental room management system has become important factor in modern society hence the need to have a rental house management application. So, the need of technology in the field of renting system is essential. The limitations of existing system are:

- Lack of security.
- No guarantee for getting the rooms/ flat.
- Time Consuming.

- Quick and accurate search functions, and customer focused content are hardly acquired by the existing system.
- More manpower to perform real-time jobs.
- It is difficult to find house for the renter in a specific area and exact location i.e Google map or any virtual maps.
- Some landlords/ house owner who are dependent on their finance from renting room/ flats, may face loss due to not getting the renters of their choice.

1.3 Project Objective

The main Aim this project is to provide a guaranteed and reliable platform for both renters and owners for minimizing their problems. We aim to collect the data from both the renters and owners regarding the problems and interests and preference which will be used by our system to provide a robust recommendation/ suggestion to the concerned domain. Below are the listed main objectives we are focusing to solve along with the mission to explore more problems and solutions to regarding issue within a system in further expansion.

- Study of currently available web application developed in foreign countries for rental service. Analysis of the web applications and try to modify them according to Nepali environment and needs of people.
- Try to develop an efficient platform that will be helpful for not only renters but also landlords.
- Provide a guarantee to both renters and owners that our system could work and eliminate their most frequent problem in case of finding the property/ renters.
- Build an advanced chat and recommendation system to recommend the renter and owners their choice of preferences, saving the huge money and time of both renters and owners.

1.4 Project Scope and Limitations

Although we aim to solve the most frequent existing problem between the renters and owners and making the renting process easier, it may have some of the limitations which system may not be involved or responsible for the actions involved during the process of execution. It may also not meet all the expectations of the users however we try to be best solution in the existing market with enhanced services and features as per the feedbacks of the systems user. Below are the system’s scope and limitations.

1.4.1 Scope

As a scope of the system, our system is aimed to solve the various problems in between the renters and owners.

- The project will consist of creating a room finder web application.
- Any visitor can easily go through our website and look for what they would like to rent.
- Modules of this application will include a update for a product and deletion of products and way to motivate costumers to continue to rent.
- Can get the rooms or property as per the preferable location, price, and facilities.
- Can search for roommates and communicate with them.
- Pre-book the room/ flat directly from the system.

1.4.2 Limitations

- The software doesn't deal with payment for the current version.
- Doesn’t guarantee the property quality and its facilities within it.
- Our system tries to minimize the SCAMS and wrong intended persons in the domain like fraud renters and owners, but system is not responsible if any fraud activity happens or any renter does wrong activity within the property of the owner.
- Renters are suggested to book to only the verified property, the system is not involved in deciding the price of the property, hence system will not be able to negotiate the price of the property.
- Our system is a solely a medium, our system doesn’t own any property and customers directly. Failing to get more users like renters and owners to get engaged in the system will fail to provide the efficient services as per intended.

1.5 Significance of Study

The findings of this study redound to benefit the society considering that web applications play important role in today's era for renting the rooms/houses/office. The significance of this web Application will be saved both time and resources. It provides rooms Locations by Google Map service with help of GPS. This will increase the efficiency of current renal system. It will provide all nearby locations to renter and multiple renters to the landlord. It supports direct communication between renter and landlord. For the researcher (researchers if it is a group study), the study will help them uncover critical areas in the educational process that many researchers were not able to explore. The major thing that we are concerned with the study of system and platforms implementation is impact. As this problem is a very general problem where every 80% of people are concerned or affected by this issue. To solve this issue, many ways and system exists already but the impact is studied very minimal which only targets and involves in huge bigger case of finance involvement only. Some of the platforms like most Facebook groups and Hamrobazar rental service include more than 90% agents who are not effective and helpful for finding the rooms to renters but the commission is too high which is a next big issue in addition for a middle class and unemployed renters like students. As per the study and analysis for the system, and lack of availability of the promising platform for connecting the direct owners and renters in a single platform, we are motivated enough to make this happen and implement our idea to solve the real issue.

2 Literature Review

This section includes the literature review of web-based room finding and connecting application between the renters and owners.

2.1 Review

Property is like room, flat or any apartments are very crucial for shelter of people living in city area as not all of them could be the owner of the property Hence it is very important to get the service to provide the room and flat rental service in the city with both time and cost flexibility to renters and provide the customers to owners who want to give the room and flat for rent. For the context of Nepal and major cities, there is huge population density and more people shifting to city area seeking the employment, better education and better modern services than in remote areas. From the case study from Kathmandu, among 1 million population, more than 80% are in rented property and it is too hard to get the rental services, room and flat within this city as easier.

Among 1 million people in Kathmandu, almost 70% people are aware of the technology use by certain level and can use internet for services. It is our aim to make people aware to use technology for the easier and convenient services. A automated service like automatic suggestion system, easier interface and visual map along with the video and picture of the room and flat makes renter to confirm the property even without visiting the real place. For conversation and booking, it is better to provide booking as well as the chat feature which will make user to stick for single platform for whole process of finding and booking the property. Finding a proper property for renting is the greatest challenge for people now, as well as finding the room mate that suits preferences and renters for owner is also the greatest challenge which makes the process vulnerable.

Hence, our project RoomFinder.com is looking forward for providing the better solution and filling the gap between the consumers and owners.

2.2 Google Map API.

Google map or Open street map is the mapping technology that are being used widely by many of the applications as well as the users for better interface and virtual mapping the locations. It is very convenient application from which we could get every detail of most of the locality with the geo location as well as provides the accurate navigation and recommendation for nearest business location which could be very helpful for users for referencing. We are using the paid version of the google map API which is faster in terms of the performance and provides better features then in free and testing API.

Mapping technology is the best used and applied technology every day in every corner of the world. It has very good accuracy and very flexible for multi-language and supports the major languages like Nepali as well.

2.3 Real Time chat

Chatting applications or feature within a integrated system is the most used feature in the enterprise level for providing better, faster and almost free way of communicating among the peoples. Messaging applications are widely used for text message, images, and many multimedia. This will reduce the cost and time for communicating between the owners and renters. Chat feature is almost within the service providing global brands which has a greater impact than we see when it comes to applicability. Most people may prefer WhatsApp, we chat, Facebook messenger or viber for communicating but we choose to provide the chat service within for security issues making data more secure of the system's users.

2.4 Recommendation system.

A recommendation system is the feature under the information filtering intended to provide the information as per preferences or the rating provided by the user. User provide the rating to data items, the best rating or most viewed will be suggested to the targeted user using the commercial user. The main aim and objective of recommendation system is to provide the exact content or information to user among the millions of collections of information.

There are many systems using recommendation system like YouTube, Facebook, dating recommendation, restaurants recommendation system and even mostly used in Ecommerce domain for recommending the products to user. The main crucial thing for recommending the information for any user is to get the preferences of the user, what kind of information does user likes or interested on. This is completely interest-based system which will calculate closeness in the interests of user towards any products and recommend the nearest or close related categories of products.

2.4.1 Approaches for recommendation systems.

2.4.1.1 Collaborative filtering

Collaborative filtering is the technique used in recommender system/engine which provides the prediction or filtering about the interests of user by collecting preferences of many user and classifying the interests based on the user. The underlying assumption of the collaborative filtering approach is that if a person A has the same opinion as a person B on an issue, A is more likely to have B's opinion on a different issue than that of a randomly chosen person. For example, a collaborative filtering recommendation system for television tastes could make predictions about which television show a user should like given a partial list of that user's tastes (likes or dislikes).[3] Note that these predictions are specific to the user, but use information gleaned from many users. This differs from the simpler approach of giving an average (non-specific) score for each item of interest, for example based on its number of votes.

2.4.1.2 Content-based filtering

Content-based filtering, also referred to as cognitive filtering, recommends items based on a comparison between the content of the items and a user profile. The content of each item is represented as a set of descriptors or terms, typically the words that occur in a document. The user profile is represented with the same terms and built up by analyzing the content of items which have been seen by the user.

Several issues have to be considered when implementing a content-based filtering system. First, terms can either be assigned automatically or manually. When terms are assigned automatically a method has to be chosen that can extract these terms from items. Second, the terms have to be represented such that both the user profile and the items can be compared in a meaningful way. Third, a learning algorithm has to be chosen that is able to learn the user profile based on seen items and can make recommendations based on this user profile.
The information source that content-based filtering systems are mostly used with are text documents. A standard approach for term parsing selects single words from documents. The vector space model and latent semantic indexing are two methods that use these terms to represent documents as vectors in a multi dimensional space.
Relevance feedback, genetic algorithms, neural networks, and the Bayesian classifier are among the learning techniques for learning a user profile. The vector space model and latent semantic indexing can both be used by these learning methods to represent documents. Some of the learning methods also represent the user profile as one or more vectors in the same multi dimensional space which makes it easy to compare documents and profiles. Other learning methods such as the Bayesian classifier and neural networks do not use this space but represent the user profile in their own way.

The efficiency of a learning method does play an important role in the decision of which method to choose. The most important aspect of efficiency is the computational complexity of the algorithm, although storage requirements can also become an issue as many user profiles have to be maintained. Neural networks and genetic algorithms are usually much slower compared to other learning methods as several iterations are needed to determine whether or not a document is relevant. Instance based methods slow down as more training examples become available because every example has to be compared to all the unseen documents. Among the best performers in terms of speed are the Bayesian classifier and relevance feedback.

The ability of a learning method to adapt to changes in the user’s preferences also plays an important role. The learning method has to be able to evaluate the training data as instances do not last forever but become obsolete as the user’s interests change. Another criteria is the number of training instances needed. A learning method that requires many training instances before it is able to make accurate predictions is only useful when the user's interests remain constant for a long period of time. The Bayesian classifier does not do well here. There are many training instances needed before the probabilities will become accurate enough to base a prediction on. Conversely, a relevance feedback method and a nearest neighbor method that uses a notion of distance can start making suggestions with only one training instance.

Learning methods also differ in their ability to modulate the training data as instances age. In the nearest neighbor method and in a genetic algorithm old training instances will have to be removed entirely. The user models employed by relevance feedback methods and neural networks can be adjusted more smoothly by reducing weights of corresponding terms or nodes.

The learning methods applied to content-based filtering try to find the most relevant documents based on the user's behavior in the past. Such approach however restricts the user to documents similar to those already seen. This is known as the over-specialization problem. As stated before the interests of a user are rarely static but change over time. Instead of adapting to the user's interests after the system has received feedback one could try to predict a user's interests in the future and recommend documents that contain information that is entirely new to the user.

A recommender system has to decide between two types of information delivery when providing the user with recommendations:

- Exploitation. The system chooses documents similar to those for which the user has already expressed a preference.
- Exploration. The system chooses documents where the user profile does not provide evidence to predict the user's reaction.

There are lots of approaches for using in recommendation system like multi-criteria recommender systems, risk aware recommender system, graph based recommender system, mobile recommender system and hybrid recommender system. However, without having the much user experience within our system, we implemented the content based recommender system which will recommend the similar information that user profile contents the preferences about.

2.5 Existing system

We found various similar products that have already been developed in the market. Unlike all those products, this is a webpage where house owners, clients, customers can exchange information effectively and inexpensively. It provides user-friendly interface, satisfying the needs of the consumers.

2.5.1 GharBeti :

Gharbheti is a search engine where you will find listed properties for rent, lease and on sale in Nepal. You can also can list down your properties and view applied users. It facilitates all to keep direct relation between the tenants and landlords, between buyer and seller of properties.

2.5.2 GharJaggaBazar.com:

It is a web application where you can have a fair deal with real estate. It is a reliable way to buy a house and share your land. You can search for rental rooms from rent category but it doesn't include the room locations by the help of GPS through Google maps.

2.5.3 Room finder Kathmandu public group:

It is a Facebook webpage which consists of various members .This group page posts available rooms for rental within Kathmandu valley and user can contact them through contact details or with the help of direct message.

2.5.4 Global market brands

2.5.4.1 Zillow.com

It is a complete robust application/ platform renters, owners and real state agents who wants to do business from the platform. It provides advance search system of the listed property and along with from various attributes. It provides search by name as well as the search by map and provides good map UI to demonstrate the location of the property. Our product is intended to implement the same features along with advanced chat and recommendation feature that will be helpful for the renters owners.

2.5.4.2 Roomster

It provides a platform for finding the perfect and preferable roommate for renters. However, the limitations of the system is it is only dedicated for provide the service of finding the roommates for any room, flat or apartment. Roomfinder will have the feature to suggest the preferable roommate.

2.6 Comparison with Existing Systems

On comparing our project with other similar applications, we came to conclude with the following:

Apps Features	Gharbeti	GharJaggaBazar.com	Roomfinder
Ease of Use	Medium	Low	High
Detailed product description	Yes	Yes	Yes
Clean web design	yes	No	Yes
User-friendly	Yes	No	yes
Use of GPS to locate rooms	Yes	No	Yes
Focus Group	No	No	Yes
Password Protected	Public and Business	Real estates and Business	Public and business
Accurate search function	Yes	No	Yes

Table 1 :Comparison Table

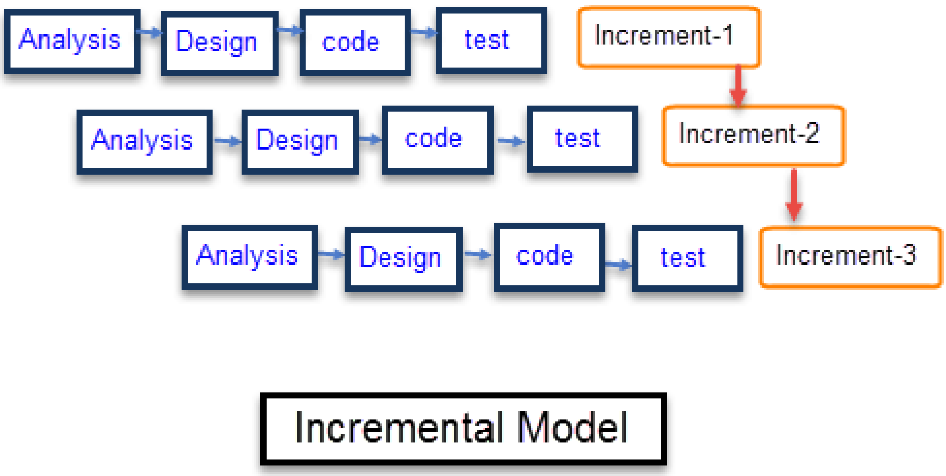
3 Team members and Task Divided

Name	Roles	Responsibilities
Akhil Chhetri	• System designer and Developer. (Full stack developer) • Security and Database administrator.	• System analysis and Design of the feasible system. • Develop required modules and integrate as per in design specification. • Design the various entity modeling and relationship between the database. • Design and develop the recommendation system. • End user documentation.
Barsha Thapa	• Frontend Developer • Documentation • Testing and Q/A expert.	• System analysis and Design. • Frontend Development. • Q/A and testing. • End user documentation.
Himsikha Rai	• Project Leader • System designer and Developer. • End user Documentation.	• Review and approve all project deliverables (Initiation Plan, Detailed Plan, Testing etc.) • System analysis and Development. • User interface and Map API. • End user Documentation.
Sagar Thapa Shrestha	• System designer and developer. • End user Documentation.	• Designing different modules as per the need and develop the modules. • Chat System and booking system. • End user Documentation.

Table 2 Team members and task divided.

4 Methodology

4.1 Software Development Life Cycle:



(REFERENCE: www.guru99.com)

Figure 1: Incremental Model

4.1.1 Development Model:

We are planning to use the incremental model for software development life cycle. In this model the product is first designed and tested incrementally (a little more is added each time) until the product is finished. It involves both development and maintenance. The product is defined as finished when it satisfies all its requirements.

4.1.2 Analysis Phase:

In this phase, analysis should be done in order to find out the requirements of the system. The outcome of this phase is a SRS which is an acronym for “System Requirement Specifications”.

4.1.3 Design Phase:

In this phase the SRS should be translated into the system’s design. Context Diagram, DFD, ER – Diagram, Use Case Diagram and Class Diagram were developed.

4.1.4 Coding Phase:

In this phase coding should be done according to the design and a working system was developed by the end of this process.

4.1.5 Testing Phase:

In this phase, the system shall be tested. With each testing a list of changes to the system developed, is suggested and the changes can be applied to the software and the software can be delivered as a successive increment until a satisfying system is achieved.

4.2 Technologies Used

Following are the technologies used in our project.

- SQlite database for storing the relational and transactional data.
- HTML5, CSS3, JavaScript, Angular JS for creating User interface.
- Collaborative filtering and recommendation algorithm.
- Django REST framework for rest API, Node js and Graphql for recommendation system API.
- Python and JavaScript as Main programming language.
- JSON web token (JWT) for authorization.
- Socket.io for chat based and Realtime connection.
- MongoDB for storing the big data and processing for analysis and research.

4.3 Tools used

Tools required	Purpose
1. Visual studio code	Code editor and text editor
2. Photoshop and ink space.	Creating vector graphics and logo
3. MS-WORD and PowerPoint.	For documentation and presentation slide.
4. Rational rose and Draw.io	For creating the UML diagrams.
5. Browser	Running localhost server and testing the system.
6. MS-Excel	For creating the analysis, visualization and gantt chart.
7. Unit testing and Mocha.js	Testing for python and JavaScript programs.
8. PyCharm	IDE for python
9. PIP and NPM	Python and node package manager.
10. Git and GitHub	For version control, and team collaboration.
11. Trello	For creating task and tracking the sprints performance.

Table 3 :Tools used

5 Requirement

Requirement analysis, in software engineering encompasses those tasks that go into determining

the need and conditions to meet for a new or altered product, taking account of possibly conflicting requirements of the various stakeholders, such as beneficiaries and users. It is the early stage activity of requirement engineering which encompasses all activities concerned with eliciting, analyzing, documenting, validating and managing system requirements.

5.1 Functional Requirements

- Provide renter and owner platform for interaction.
- Provide platform for adding the rental property and book through it.
- Allow owners to view the list of their listed property.
- View the Map location of the property and listed renters in search of roommates.
- Store the user interests and preferences for recommending the related property lists in future.
- Create database to store and extract the list of owners, renters and property.

- ➢ Data Entry Method
  - o Direct Entry using keyboard

    The user information (student, reviewer and editor) as well as the project documents are entered through keyboard.

- ➢ Backup facility
  - o may use backup of the records for security purpose. This may be through hard copy or soft copy output of the report.

- ➢ Output to User
  - o Soft Output Requirements
    Methods of displaying records (Browser)

- ➢ Interface required
  - o Login and register Window
    Users (Student, Reviewer and Editor) can use login window if they already have an account. If they don't, register window will be used to let user sign up for their respective account.
  - o Information viewing windows
    This allows the student to view the list of documents uploaded, feedback and queries

given to them. It allows the reviewer to view the list of documents and review the assigned project document and view the answer of their queries by the student team. Likewise, it allows the editor to view the information if the students and the reviewer and assign the project document to the reviewer.

- User profile window
  It is used to display the profile of the user containing only the information which user has shared with the system.

5.2 Input Requirements:

- Data Requirements:

  The data to be input are:

  - User Details:
    This contains the information about the users (renters, and owners) of the system.
  - Property Details:
    This contains the property information. It contains the name, price, facilities, location and renter rating and review as well.
  - Review and Rating.
    This gives the rating given by the renter to the property of respective owners.
  - Booking request:
    This gives the information of the booking any property through the system either booking through the system using chat feature.
  - Chat Message
    This contains the messages sent from renters of owners.

5.3 Output Requirements

The operator requires following output from the system:

1. Detail information of the property, map location, renter details and the owner details

2. Details of the usability of the system by the users.

The renters require the following output from the system:

1. List of the properties, and locations added by the respective owners.

2. Rating and reviews of the renters to that property and list of renters who are in search of the roommates.

The owner requires the following output from the system:

1. List of the added property by the respective owners.

2. Request of booking the property from the renters.

3. Feedbacks and review by the renters to the property.

4. Message send by the renter to owner.

5.4 Security Requirements:

The system should be protected from unauthorized users and unauthorized access. Hence, for providing security, passwords are encrypted and stored in database using DES Encryption and Decryption protocol. Adoption of this protocol to encrypt and decrypt passwords keeps the passwords safe from the intruders.

- DES Encryption and Decryption:
  DES Encryption and Decryption protocol use same key to encrypt and decrypt the text which is known as private key. Hence the sender and receiver uses same private key that is only shared between sender and receiver to encrypt and decrypt the text. The Data Encryption Standard is a block cipher, meaning a cryptographic key and algorithm are applied to a block of data simultaneously rather than one bit at a time. To encrypt a plaintext message, DES groups it into 64-bit blocks. Each block is enciphered using the secret key into a 64-bit cipher text by means of permutation and substitution. The process involves 16 rounds and can run in four different modes, encrypting blocks individually or making each cipher block dependent on all the previous blocks. Decryption is simply the inverse of encryption, following the same steps but reversing the order in which the keys are applied. For any cipher, the most basic method of attack is brute force, which involves trying each key until you find the right one. The length of the key determines the number of possible keys -- and hence the feasibility of this type of attack. DES uses a 64-bit key, but eight of those bits are used for parity checks, effectively limiting the key to 56-bits. Hence, it would take a maximum of 2^56, or 72,057,594,037,927,936, attempts to find the correct key.

6 System Design

Room Finder is a webpage where house owners, clients, customers can exchange information effectively and inexpensively. It provides user-friendly interface, satisfying the needs of the consumers. It employs a new strategy that facilitates easy management of rental houses. The system after careful analysis has been identified to be presented with the following modules: Online Room Finder deals with the following modules:

Registration (User/Vendor):

The Vendor/User fills the registration form by giving the personal information and successfully registers with the website.

Product (Description/Images/Status):

After entering into vendor's homepage vendor will add his products by filling add product form, by providing sufficient details about product such as product id, available dates, rent etc. and upload the image of that product. It consists of following sub modules:

1. Update Product: Vendor can update the existing product details such as rent, available date etc. by entering the product id of that product.

2. Delete product: Vendor can delete his products by entering the product id.

Client: This module is operated by a client who is an admin to this application who will look after users registering with the application (owners and customers) and active and deactivate them.

Customer: This module is operated by a customer who needs to register with the application. A customer is a person who is looking for rent. Customer can search based on specifications and get results about rent details.

6.1 Use Case Diagram

Developing clear and concise use cases when you design or redesign your ecommerce website, can improve user experience and satisfaction, potentially leading to more sales and more profits. Use cases have their beginnings in software engineering, where they were used to visually or textually describe a system's behavior as it responds to queries, input, or other actions from external systems or "actors. "We have listed here various actors along with their uses cases and their associations.

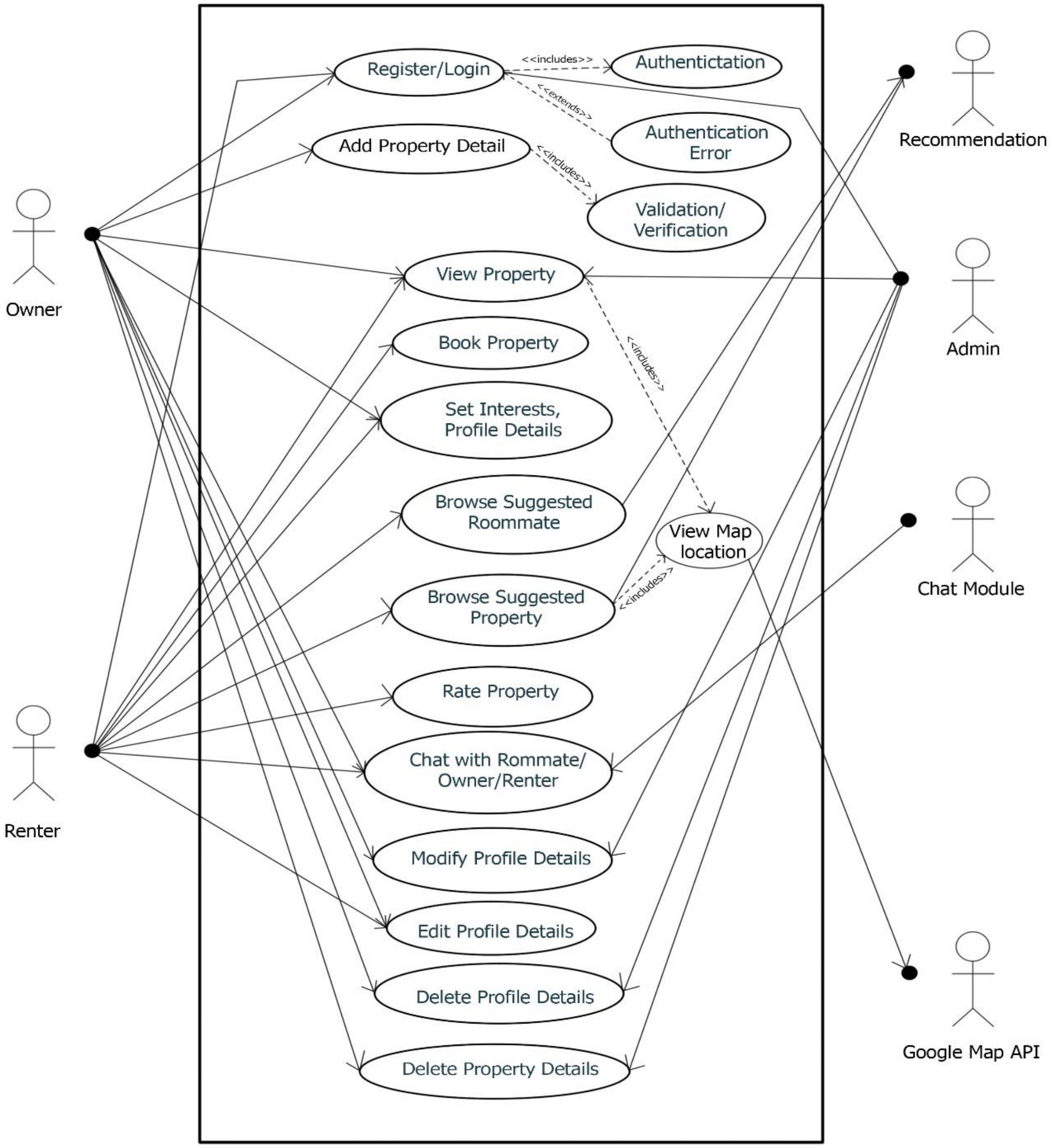


Figure 2 :Use case diagram

6.2 Operation Contract

6.2.1 Use Case UC1: Register

Primary Actor: Renter, Owner

Secondary Actor:

Stakeholders:

User: wants to register themselves.

Preconditions: Users must have a unique password and name to register.

Post conditions: Username and password is registered.

Basic flow:

- User registers themselves using their password and name.
- User selects appropriate option from the menu.
- If a Renter or Owner is a new user, he can request to register page. A register page opens and asks total information about him/her and also asks them to choose login (email address) and password.

Alternate flow:

At any time, user enters a password and name then system notifies.

6.2.2 Use Case UC2: Login

Primary Actor: Renter, Owner

Stakeholders:

Renter: Wants to login.

Owner: Wants to login

Preconditions: Users (Renter, Owner) must have login Id.

Post conditions: The system verifies the name and password matches. If not matches, error messages show to the users if it matches user is logged in and granted access.

Basic flow:

- Users (Renter, Owner) login to system using password and name.

- User selects corresponding appropriate option from the menu. .

Alternate flow:

- At any time, user enters a password and name then system notifies.The system verifies the name and password matches. If not matches, error messages shows to the Users (Renter, Owner).

6.2.3 Use Case UC3: View the Property

Primary Actor: Renter, Owner

Secondary Actor: Admin

Stakeholders:

Renter: Wants to view the property.

Owner: Wants to view the property.

Preconditions:

Post conditions: The selected properties according to the selected criteria are shown to the users.

Basic flow:

- User selects corresponding appropriate option from the menu.
- System shows the available properties according to the area of interest of a user.

Alternate flow:

- At any time, system fails, system logs the error.
- Error message is displayed.

6.2.4 Use Case UC4: Book Property

Primary Actor: Renter

Stakeholders:

Renter: Wants to book the available property of his interest.

Preconditions: User (Renter) must have an loginId and is logged in.

Post conditions: The system verifies the name and password matches. If it doesn't match, error messages are shown to the users, if it matches, user is logged in and they are granted access and booking can be done.

Basic flow:

- User (Renter) login to system using password and name.
- User selects corresponding appropriate option from the menu and proceeds booking.

Alternate flow:

- At any time, user enters a password and name then system notifies. The system verifies the name and password matches. If not matches, error messages shows to the User (Renter) .

6.2.5 Use Case UC5: Set Interests

Primary Actor: Renter, Owner

Stakeholders:

Renter: Wants to set interest to simplify their searching as per their needs.

Preconditions: Users (Renters, Owner) must be granted access to set interest for searching of the properties they are seeking for.

Post conditions: Renters, Owners get the result as per their area of interest on property.

Basic flow:

- Users (Renters, Owner) set areas of interest from the given options in the system.
- User selects corresponding appropriate option from the menu.

Alternate flow:

- The system verifies the name and password matches. If not matches, error messages show to the User (Renter).

6.2.6 Use Case UC6: Set Profile Details

Primary Actor: Renter, Owner

Stakeholders:

Renter: Wants to set interest to simplify their searching as per their needs.

Preconditions: Users must be granted access to set interest for searching of the properties they are seeking for.

Post conditions: Renters, Owner gets the result as per their area of interest on property.

Basic flow and Alternate flow:

- Users set areas of interest from the given options in the system.
- User selects corresponding appropriate option from the menu.

6.2.7 Use Case UC7: Browse suggested Roommates

Primary Actor: Renter

Stakeholders:

Renter: Wants to find tenants of their interests.

Preconditions: Users must have a login ID.

Post conditions: Renters gets the result as per their area of interest on tenants.

Basic flow and alternate flow:

- Users (Renters) set an area of interest from the given options in the system.
- User selects corresponding appropriate option from the menu.

6.2.8 Use Case UC8: Browse suggested Property

Primary Actor: Renter

Stakeholders:

Renter: Wants to find properties of their interests.

Preconditions:

. **Post conditions:** Renters gets the result as per their area of interest on poperties.

Basic flow and alternate flow:

- Users (Renters) set an area of interest from the given options in the system.
- User selects corresponding appropriate option from the menu.

6.2.9 Use Case UC9: Rate Property

Primary Actor: Renter

Stakeholders:

Renter: Wants to rate the properties of their interests.

Preconditions: User must have an login ID.

.**Post conditions:** According to the Renter's experience the properties are rated.

Basic flow:

- User selects corresponding appropriate option from the menu to rate the properties that they have an experience with.
- User then rates the properties.

Alternate flow:

- At any time

6.2.10 Use Case UC10: Chat with Roommate/ Owner/Renter

Primary Actor: Renter, Owner

Secondary Actor: Chat Module

Stakeholders:

Renter/Roommate/Owner: Wants to have conversations.

Preconditions: Users (Renters, Owner, User) must have a Login ID.

.**Post conditions:** Users (Renters, Owner, User) chatting is applicable and on process.

Basic flow and alternate flow:

- User selects corresponding appropriate option from the menu of chat system.
- User then messages are sent vice versa between the users of their own choice.

6.2.11 Use Case UC11: Modify Property Details

Primary Actor: Owner

Secondary Actor: Admin

Stakeholders:

Admin, Owner: Wants to modify properties of their interests.

Preconditions: Users (Admin, Owner) must have a Login ID.

Post conditions: The properties are modified and changed.

Basic flow and alternate flow:

- Users selects the properties to be modified or changed.
- User selects corresponding appropriate option from the menu for modification.

6.2.12 Use Case UC12: Edit Profile Details

Primary Actor: Renter, Owner

Secondary Actor: Admin

Stakeholders:

Renter, Owner, Admin: Wants to edit the profile details.

Preconditions: Users (Admin, Owner, Renter) must have a login ID.

Post conditions: Users profiles have been edited.

Basic flow and alternate flow:

- Users select an appropriate given option from the system for editing.
- User then edits according to their needs.

6.2.13 Use Case UC13: Delete profile details

Primary Actor: Renter, Owner

Secondary Actor: Admin

Stakeholders:

Renter, Owner, Admin: Wants to delete the profile details.

Preconditions: Users (Admin, Owner, Renter) must have a login ID.

Post conditions: Users profiles have been deleted.

Basic flow and alternate flow:

- Users select an appropriate given option from the system for deleting.
- User then deletes according to their profiles.

6.2.14 Use Case UC14: Delete Property Details

Primary Actor: Renter, Owner

Secondary Actor: Admin

Stakeholders:

Renter, Owner, Admin: Wants to delete the property details.

Preconditions: Users (Admin, Owner, Renter) must have a login ID.

Post conditions: The properties details posted by users have been deleted.

Basic flow and alternate flow:

- Users select an appropriate given option from the system for deleting details of properties.
- User then deletes according to their properties details that needs to be eliminated.

6.3 Entity Relationship Diagram

The Entity relationship diagram showing the interrelated things of interest of our system is shown below. It basically composed of entity types and relationships between them.

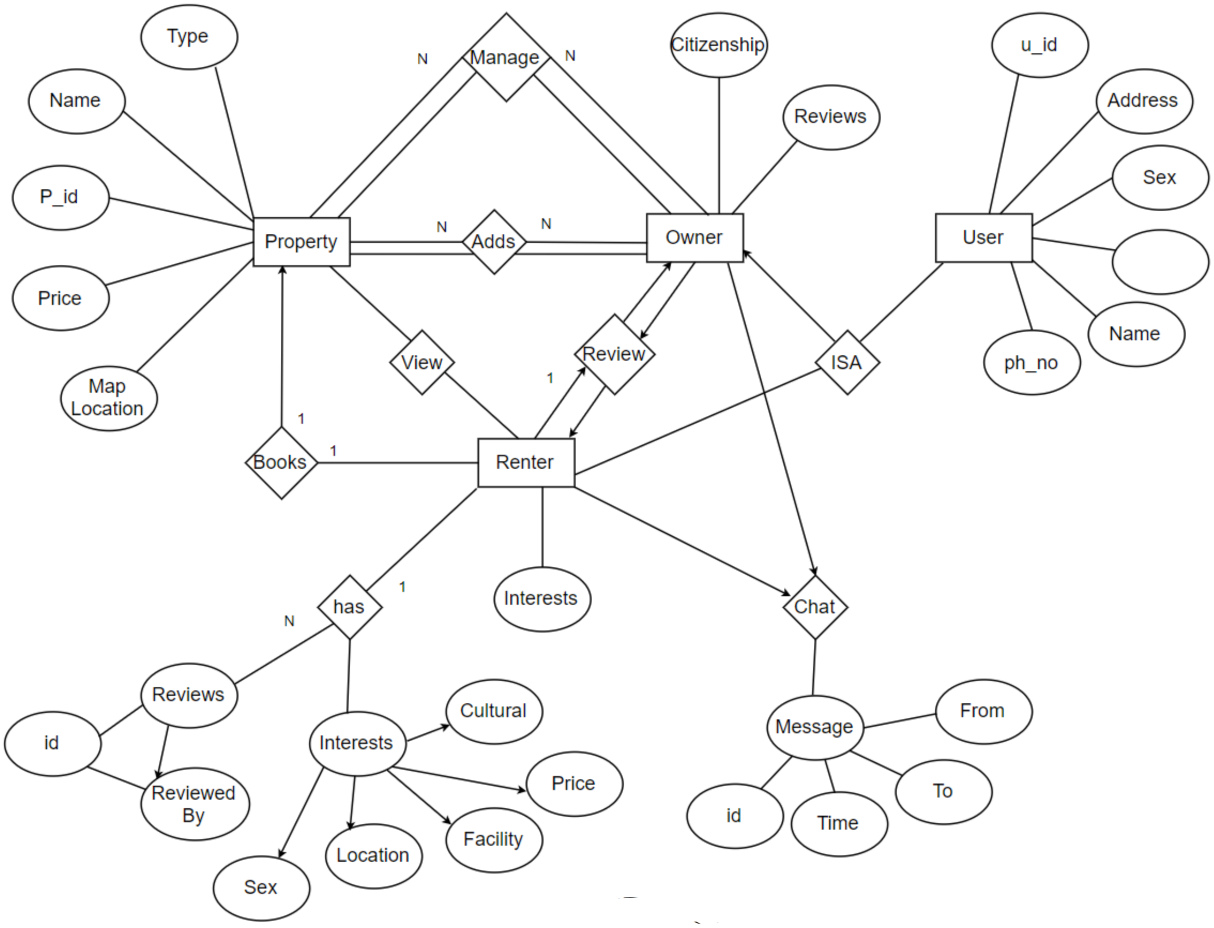


Figure 3: ER diagram

6.4 Sequence Diagram

The sequence diagrams showing our various objects interactions arranged in time sequence is shown below:

6.4.1 Sequence diagram for Login/Signup

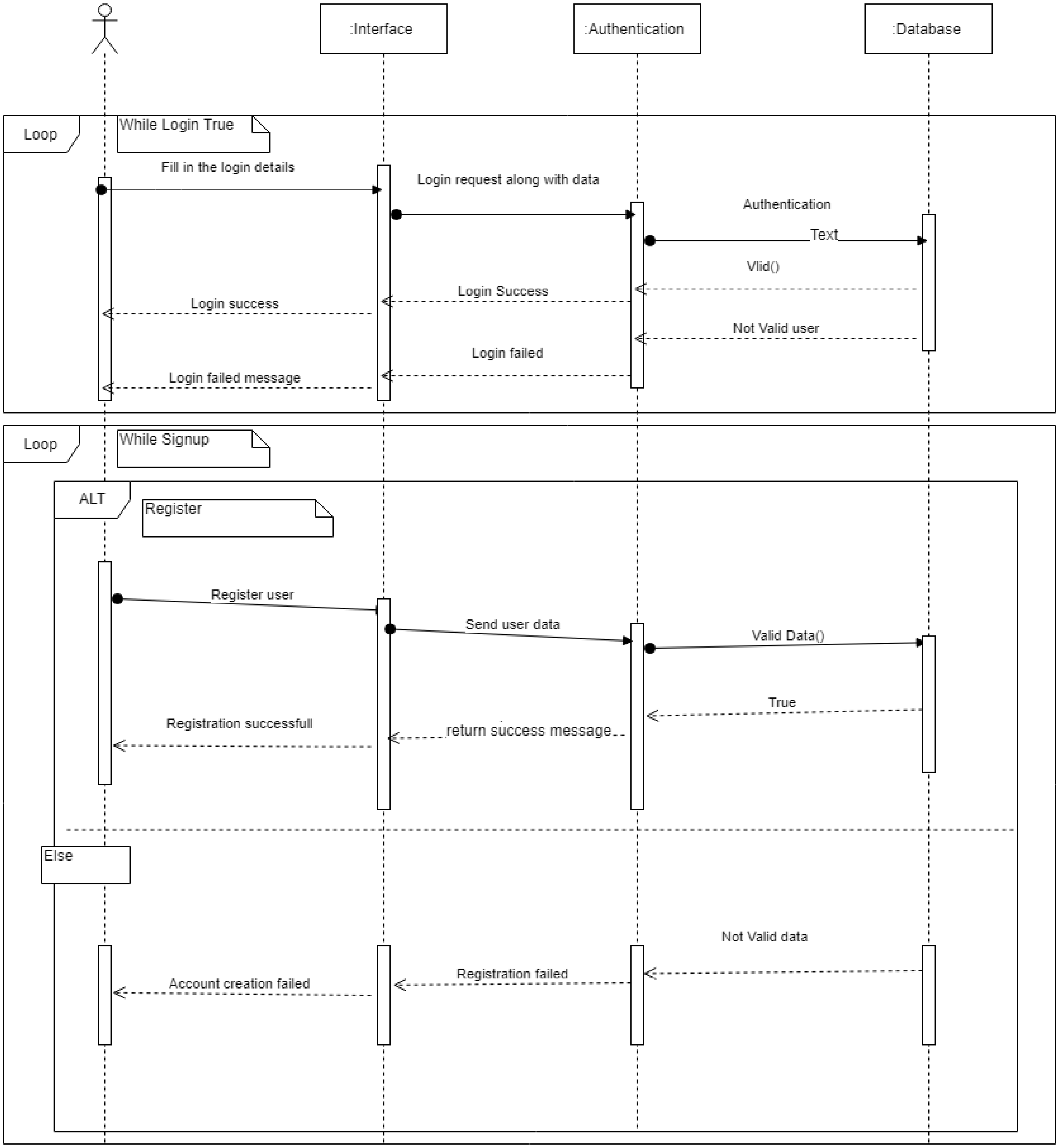


Figure 4: Sequence diagram for User login and signup

6.4.2 Sequence Diagram for creating property and modifying property

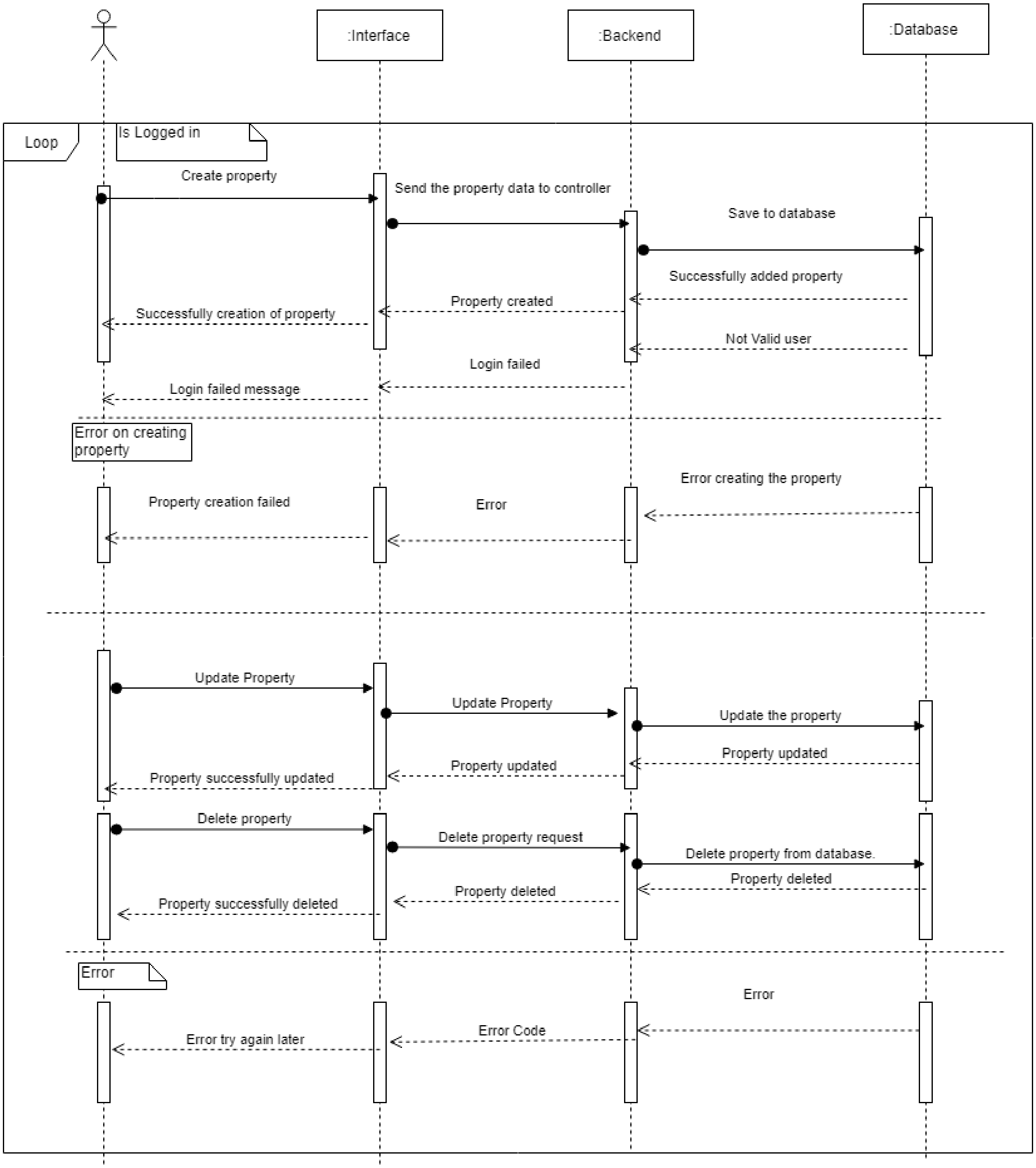


Figure 5: Sequence diagram for creating and updating the property

6.4.3 Sequence Diagram viewing, rating, and get suggested properties.

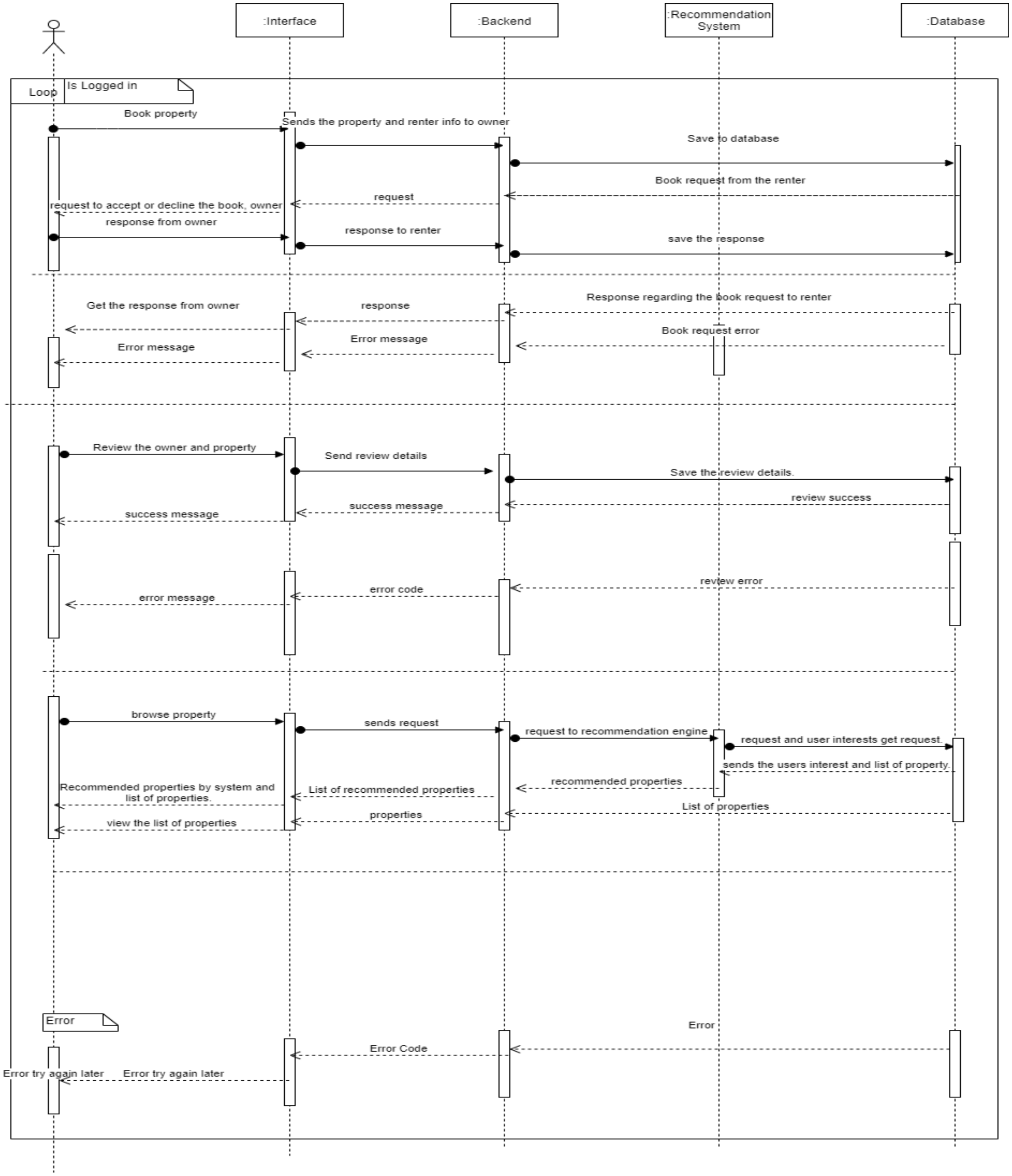


Figure 6 Sequence Diagram for viewing properties, booking and rating the properties.

6.5 Design Class Diagram

The class diagram describing the structure of our system by showing the system's their attributes, operations and their relationships among objects are shown below:

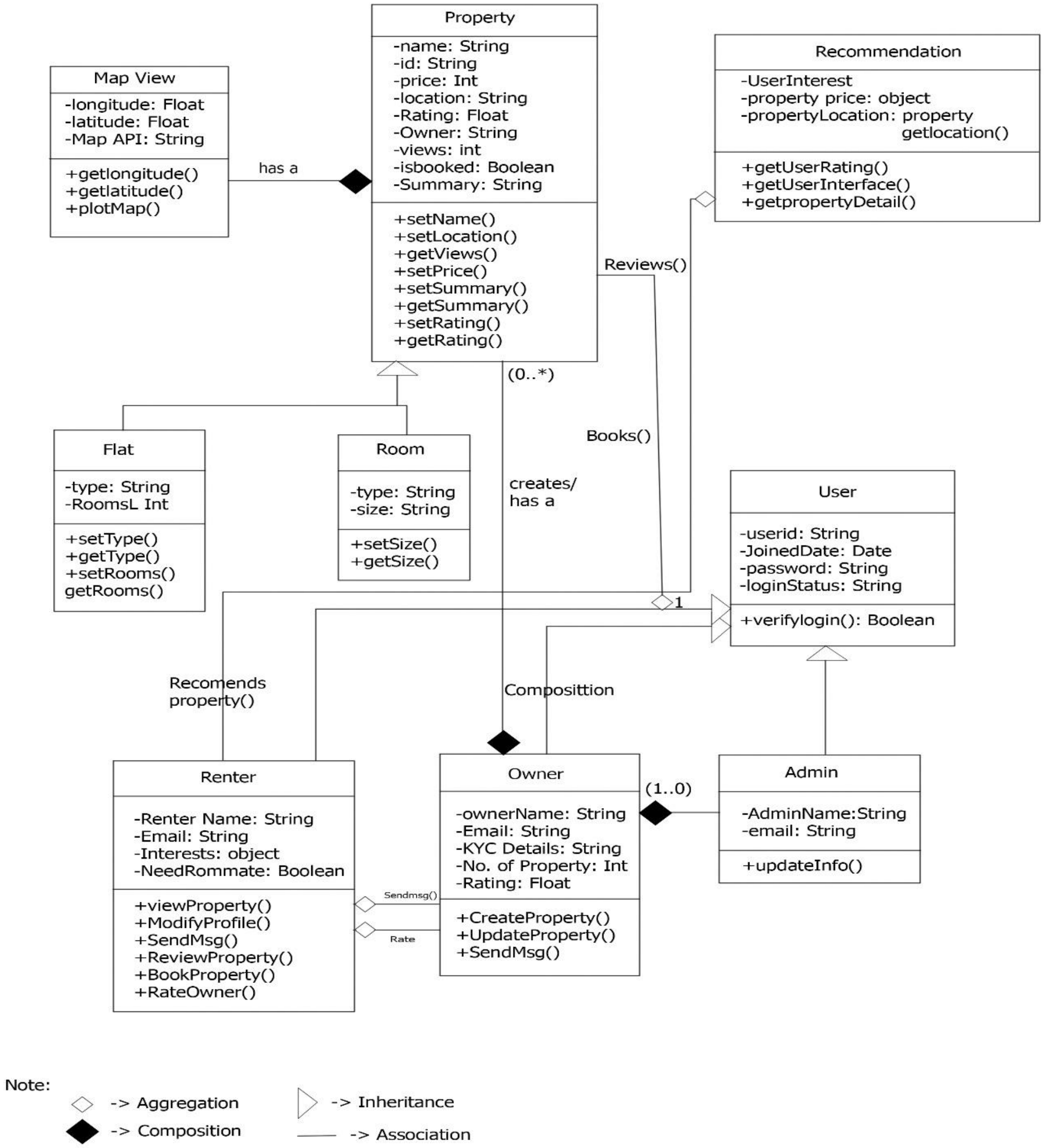


Figure 7: Design Class Diagram

6.6 Data Flow Diagram

The data flow diagram of our system is shown below. It represents the flow of a data of a process

6.6.1 0 level Data flow diagram

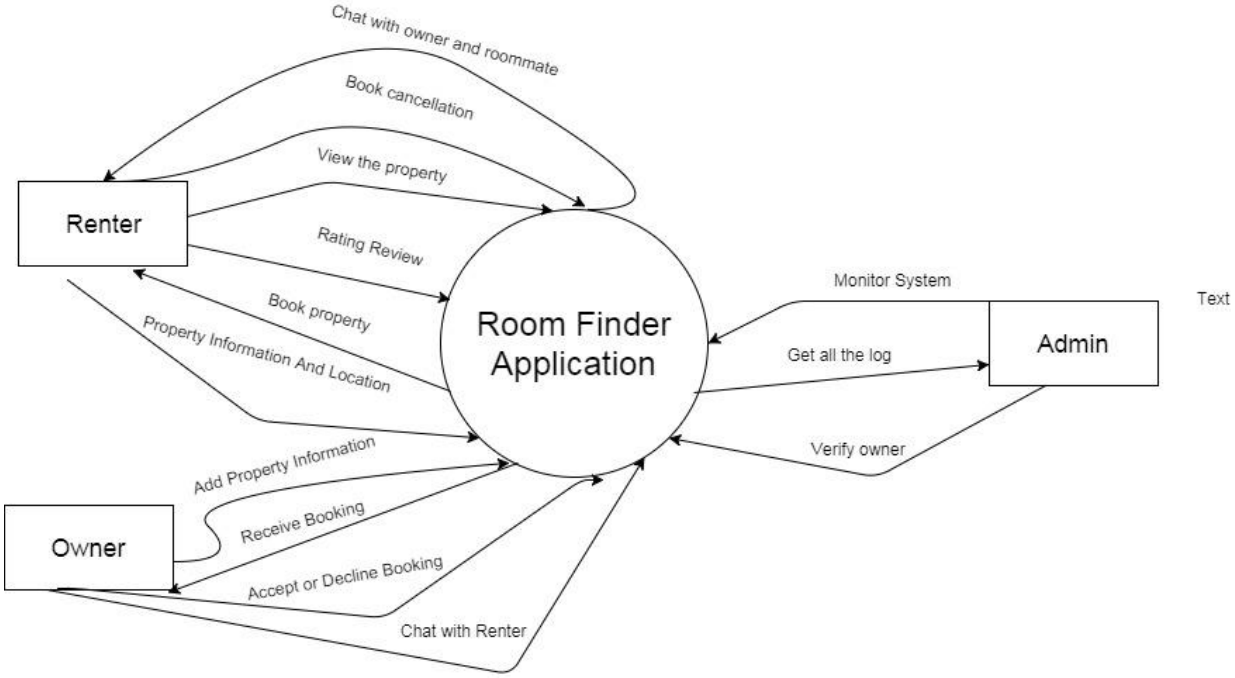


Figure 8: 0 level data flow diagram

6.6.2 1 Level DFD diagram

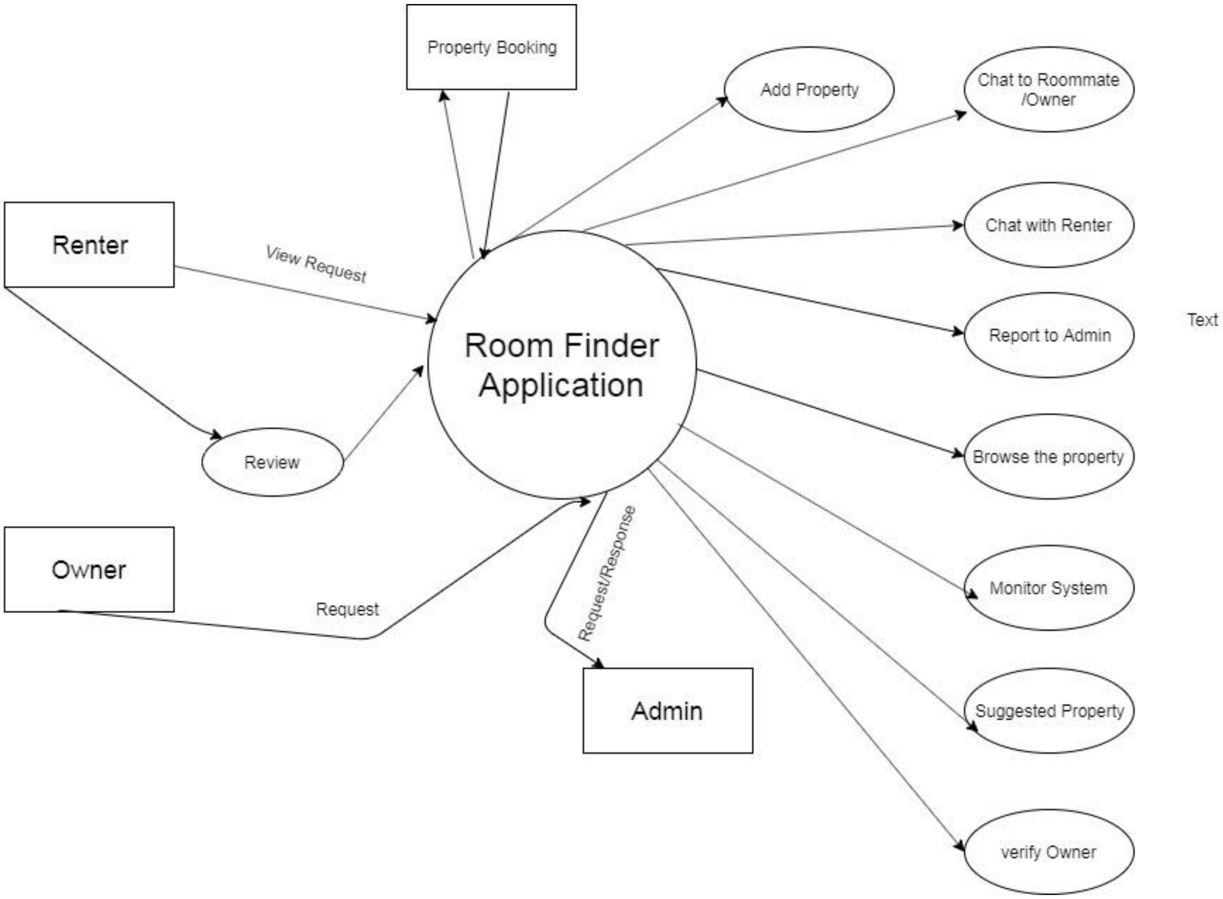


Figure 9: 1 level data flow diagram

6.6.3 2 Level DFD diagram

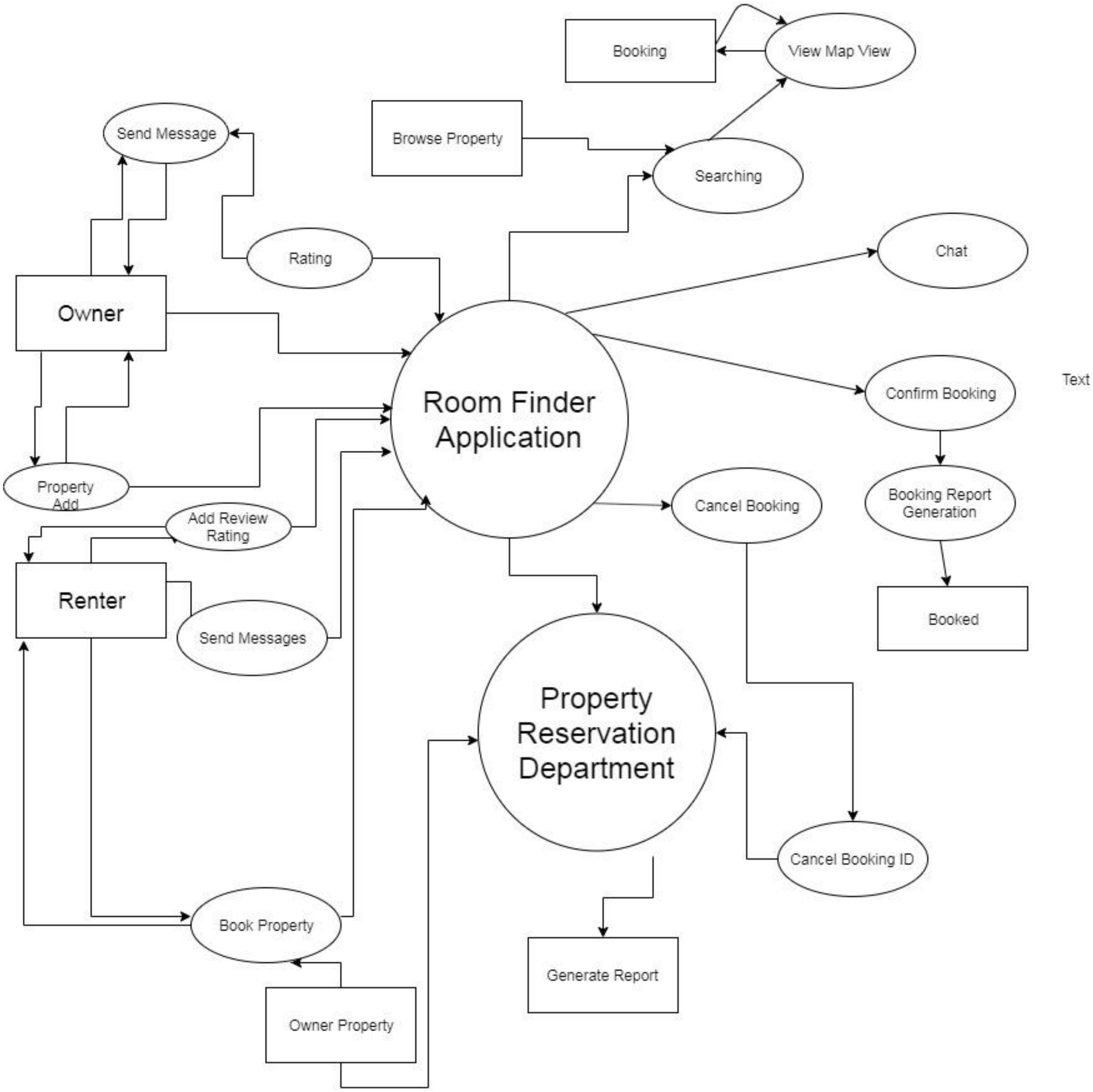


Figure 10: 2 level DFD diagram

7 Features of the project

This application features:

- Login address (registered customers)
- Users access (admin, customers)
- Detailed product description
- Clean web design
- Accurate search function
- Attractive and user-friendly
- Categorized rooms
- Authorization through social medias
- Our web application comprises of chat system
- Searching of room partner
- Recommendation system
- Chatting and booking feature.
- It shall involve the Google map for easy and exact location of the room.
- With the help of this system one can easily see the publicly available features such as browse list of rooms which has been posted by the user, views details of room owner, features and prices, and view other static contents of the site.

8 Testing

Testing is important phase to ensure that the system meets the requirements that guided its design and development, responds correctly to all kinds of inputs and achieves the general result its stakeholders desire. We evaluated (RoomFinder) to make sure that all the developed modules and packages worked properly and functions without any loss in preciseness. For this test plan of our work was created, in which elements such as validation, reliability and user acceptance were tested. The system was tested for normal condition, primarily. Testing was performed on each unit. We used both unit testing and integration testing to assure the system is developed in right track as per the requirements analyzed and the product intended to work for the real people problems.

Test No.	Form	Test	Expected Result	Actual Result	Evidence	Result
1.	Login	Incorrect username/password	Username/Password doesn't match	Username/Password do not match	Test 1.1	Failed
2.	Login	Correct Username/password	Login Successful, Welcome	Login Successful, Welcome	Test 1.2	Passed

3.	Social Login	Callback not responding with token	Login with social media should return the user id from.	Data not returning.	Test 2.1	Failed
4.	Social Login	Login success	Successfully logged in.	Successfully logged in.	Test 2.2	Passed `
5.	Signup	Correct Information	Successfully created account	Successfully created account	Test 3	Passed
6.	Review the property	Successfully submitted the review	Review sent and stored in database.	Review sent and stored and can view.	Test 4	Passed
7.	Create the property.	Submit the property details and map locations.	Successfully created the property.	Successfully created the property.	Test 5.	Passed
8.	Book the property.	Booking property.	Booking request submitted.	Successfully submitted booking request to owner.	Test 6.	Passed
9.	Send message	Message cannot be empty	Message should contain the characters and symbols.	Message is empty.	Test 7.	Failed
10.	Send Message	Successfully sent message.	Message sent.	Message sent successfully.	Test 8.	Passed
11.	Property Map view	The location of the property unknown.	Location is unknown or not supported.	Location unknown.	Test 9	Failed
12.	Property Map View	Successfully loaded the map view of the property.	Map location loaded.	Map location loaded successfully.	Test 10.	Passed
13.	Search property by map	Couldn’t load the map location.	Map location of that name is not known.	Map location is not known.	Test 11.	Failed

Table 4: Testing

9 Budget

9.1 Function Point

Function points are basic data from which productivity metrics could be computed. Function point metric is used to collect direct measure of software engineering not only according to size but also according to functionality.

Information Domain Values	Count	Weighted Value	Total Count (Weight * Count)
No of user inputs	42	4	168
No of user Outputs	9	5	45
No of User Inquires	5	4	20
No of Logical Files	12	11	132
No of External interfaces	2	7	14
Total Count			379

Table 5: Function point and count calculation.

The value of Complexity Multiplier ranges from 0.65 to 1.35. Since, our project is average, the value of the Complexity Multiplier used is average. i.e. We have assumed an average value as 1.17.

Function Point (FP) = Total Count * Complexity Multiplier

= 379 * [0.65 + 0.01 *$\sum_{i=1}^{14} fi$

= 379 * 1.17

= 443.43

= 444

Average productivity = 10 FP / pm

Labor Rate = Rs 8000per month

Effort = Function Point (FP) / Average productivity

=444/10

=44.4

= 44.5

Total Project Cost = FP * (Labor Rate/ Average Productivity)

= 444 * 8000/10

= Rs 3,55,200

- Number of user inputs:

Each user input that provides distinct application-oriented data to the software is counted.

- Number of user outputs:

Each user output that provides application-oriented information to the user is counted. In this context "output" refers to reports, screens, error messages, etc. Individual data items within a report are not counted separately.

- Number of user inquiries:

An inquiry is defined as an on-line input that results in the generation of some immediate software response in the form of an on-line output. Each distinct inquiry is counted.

- Number of files:

Each logical master file is counted.

- Number of external interfaces:

All machine-readable interfaces that are used to transmit information to another system are counted.

9.2 Line of Code

LOC (Lines of Code) is a simple and straight forward way of counting the productivity of a programmer in a given time period. Using Lines of Code metric, the project size is estimated by counting the number of source instructions in the developed program.

Estimated LOC = 4200

Average Productivity = 100 LOC/pm

Labor Rate = Rs 8000 per month

Now,

Estimated Project Cost = Estimated LOC * Cost per LOC

= 4200 * (Labor Rate / Average Productivity)

= 4200 * 8000 / 100

= Rs 3,36,000

10 Proposed deliverables:

The project delivers the software on web-based application viewing rental rooms that are available for renting. The delivered product will be a web application will provide the Information about Rooms/Flats/Houses which is available for Rent. RoomFinder web application will make easy to find the location of Rooms/Flats/Houses, select no. of rooms and other facts by the renter. The software shall involve the Google map for easy and exact location of the room.

11 Task and Time schedule

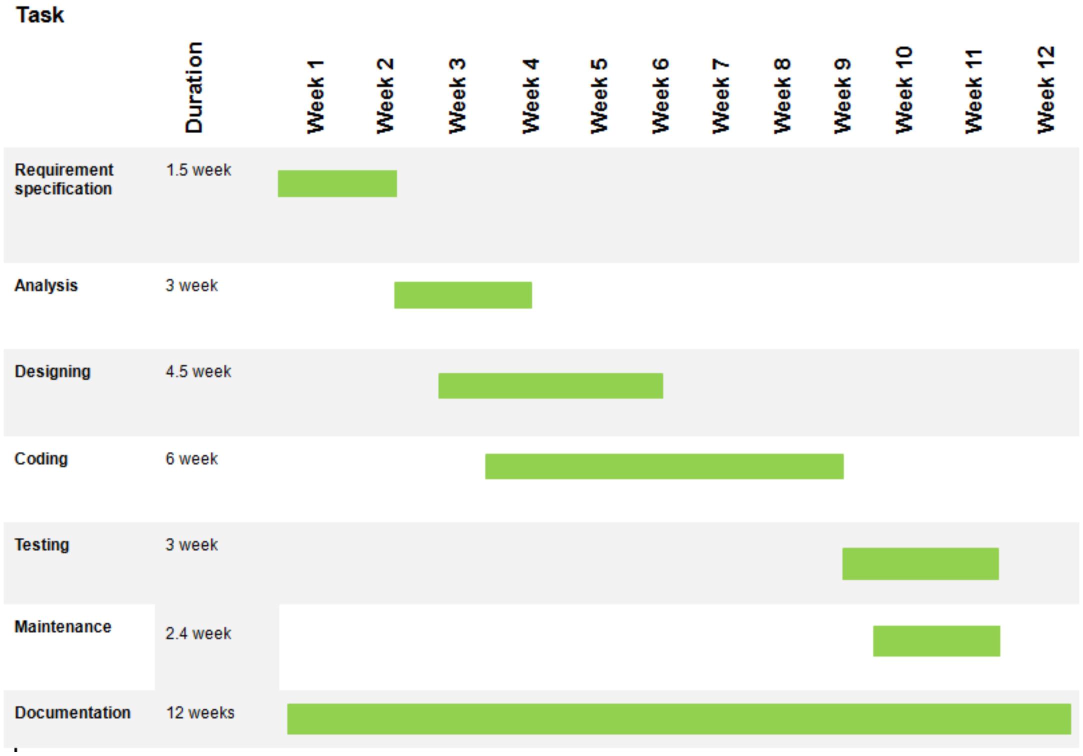


Figure 11 : Gantt chart (weekly)

We have planned to complete the project by 12 weeks. The required number of days required for different phases are:

• Requirement specification	- 2 weeks
• Analysis	- 3 weeks
• Designing	- 6 weeks
• Coding	- 8 weeks
• Testing and Maintenance	- 2.4 weeks
• Documentation	- 12 week

12 Conclusion and Future Extension

RoomFinder is web application that provides a platform for interaction between the property lenders and renters. Overall this project is intended to facilitate renters and owners for ease of providing the medium to book the property and find the required type of services on the property. As a conclusion from this project, we are successful to implement the theoretical knowledge practically for this project from the idea level. As there are lots of the things to improve to solve the wider range of problems along with the preferences of users. Hence our main purpose is to develop a web application which would make renting easy in Nepal. This application will be able to show the rooms/flats/houses with in a particular area selected by user. A web based application has a major advantage that it can be accessed from anywhere; software doesn’t have to be installed other local computer for its use. Just an inbuilt internet browser is enough to run the application.

With the intention to provide a solution for wider ranges, we are looking forwards to implement the better version of the following feature in the system more precisely.

- A separate repository for renting and selling option for owner and renters.
- A more precise recommendation system once we have the practical user data from the system.
- Push notification to the user when the renter add the property as per renters preference and location he is been seeking for.

13 Bibliography

[1]2019. [Online]. Available: https://www.roomster.com/ [Accessed: 14- Oct- 2019].

- With this website we learned what should be the components within our website should look like and behave.
- It gave us idea about how our page shall be made.

[2]2019. [Online]. Available: https://www.nepalhomesearch.com/page-rental-nepal-home-for-rent-44.html [Accessed: 20- Oct- 2019].

- We learned that this website is used for buying, selling and renting properties in Nepal.
- From this we get the basic ideas about what type of website we are building and what are the changes to be made to make it's unique.

[3] 2019. [Online]. Available: https://www.smartdraw.com/use-case-diagram/ [Accessed: 25- Oct- 2019].

- It gave us the knowledge on making proper use case diagrams.
- It taught us that developing clear and concise use cases when designing our ecommerce website, can improve user experience and satisfaction.

[4] Larman, C. (2008). Applying UML and Patterns. Pearson Education, Inc. [Book] [Accessed: 28-Oct-2019]

- We learned about the application of uml patterns and how to implement those theories using draw.io

[5] 2019 [Online] https://ieeexplore.ieee.org/document/8350801/ [accessed: 2 oct 2019]

- Learned how to implement the recommendation feature in the system which includes to learn the pattern from the user searches and behavior to provide the suggestion regarding the related property.

[6] 2019 [Online] https://docs.djangoproject.com/en/2.2/ [accessed: 15 Sep 2019]

- Django rest framework and Django web framework.
- Learned how to use ORM with Django.
- Django channels for chat application.

[7] 2019 [Online] https://restfulapi.net/ [Accessed: 14 Oct 2019]

- Restful API guides and standards.