**Project on**

**House Rental System**



**Rupesh Dhakal**

**00172917**

**Computing Project**

**Level 5 Diploma in Computing**

**Softwarica college of IT and E-Commerce**

**Kathmandu, Nepal**

**2019-01-04**

# Abstract

# Acknowledgement

I would like to express my humble thanks to my module teacher Mr. Kiran Rana sir for providing me a golden opportunity to make this project happen at the first place. He is a very inspirational teacher I have ever seen. I would also like to thank my colleagues who believed in me and my idea for developing this web application. All the teacher who have gave me necessary knowledge for completing this project.

I would also like to thanks all the teachers who had always supported me since day one I joined this very college. Mr. Sujit sir who taught me how to design good dynamic website and one of the humble people I have seen. I would also like to thank Mr. Sudip Lal Bajimaya sir who taught me better way of developing software. Mr. Achyut Timsina and Pratik Bhushal sir for providing me detail knowledge about MVC pattern that I have used in this very project.

Lastly, I would like to thank my family and friend who always supported me in all the situation I had. I would have stressed out completely with out their support.

**Table of content**

# Chapter 1: Introduction

HajurKo property is the web application which allows all the people to keep direct relationship between owners and buyers. In this application you can find all the listed house and rooms that is for rent or sale along with best deals. This web application is an ingenious solution to the problem for searching property. Any person that uses this application will have the privilege of listing their own property as along with booking the property they desire. Old ways of searching house, rooms or flat will be minimized with the help of this application. In Short it is an unlimited resource for property and intelligent solution to find suitable home, office and any other property for you.

## Background to the system

The main purpose of this project is to provide the facility of buying and selling of houses and rooms to the people.

Tension, frustration evolves in human if they have to search for an apartment/ rooms or a house. It is difficult to find the suitable property in reasonable price and in desired location. People tends to satay with bad owner then finding the new place to stay all because it’s hard to find place to stay until now. This is the reason behind developing a web application that provide the facility of buying and selling of houses and rooms just in one click. You can access the web application anytime, anywhere. People will have free access to all of our services without any noise and stress which can also be said as voice like your own. I tend to see that many people heir personal real estate agent to get property for them which costs more and you might not get property according to your choice and test. Our application eliminated you to heir the external agent to get the property.

This project can minimize the old ways of searching houses and room for work places or resident. Way of visiting every home and asking for room are bit old trends that are time consuming and frustrating this in my notion this project will definitely overcome that problem.

My main moto is to develop the web application that will have variety of houses and rooms that can be choose. Spending relatively less time compared to the old ways if we can get better result then why not adapt the new trends.

## Justification

## Aims

The overall goals that will be fulfilled via this application are:

* **Guidance:**

This web application will provide proper guidance to get the desired property. The main moto behind this is to provide user facility to choose their property without any hustle.

* **Flexibility:**

providing flexible service that will make user feel application butter smooth. Allowing user to use the application freely and without any errors. Allows user to perform multiple function without any difficulty.

* **Security:**

Security is the key to make any application success. HajurKo property provides extra layer of security to user’s data. Also, to the advertisement process.

* **Time and Efficiency:**

Storing any new data to the application is rocket fast, without any lag and flaws. Same goes to retrieving and removing the data. User can invest their expensive time to other work rather than finding suitable property. Frustration regarding the property will be carried by hajurKo property.

## Objectives:

Objectives are the things that helps to meet the aims we expect to be fulfilled. Some individual stage that leads to achieve those overall goal is listed below:

* Basically, hajurko Property is simple to understand hence providing guidance is mandatory nut in case if needed people can contact admin by clicking on support listed in footer area of website.
* By make older process of searching property automated I can save time and make the property searching process more efficient.
* Security and verification are also quite strong regarding it is the first-generation product. You will have to get verified by admin to post any kinds of property information which will make sure that the property is genuine.
* You require login to access to your panel which will restrict other user accessing your information. Plus, point is that your password is hashed which can not be easily hacked by the hackers.
* Look and feel regarding the interface is detailly studied keeping in mind that user may get bored with outdated and ugly design.

## Overview of the project design

The main need of the project is to minimize the human effort, time that goes for searching the property. In the current scenario of the society it’s very hard to find the desired homes and property. Even if the property is found there are lots of negotiation and scarification because of different factors and due to lack of choices.

This web application can minimize this problem to some extent by letting you to choose from wide ranges and varieties of properties. All kinds of properties are listed in this application where you can get information of the owner and contact them Realtime. Keep things simple is the main motto behind developing this application where user can understand the application easily.

I am very fortunate to provide such service to the people which makes their daily life lot easier.

# Chapter 2: Analysis

## Introduction to analysis

Analysis is an examination and evaluation of the information is called analysis. Analysis of any project can be done by breaking down it into various parts which helps us to understand the project more clearly. We can also identify the relationship of different topics and we can also identify how they fit together by analyzing. Without analysis we might not get what is the main purpose of the project. It is also the second stage of SDLC

### Needs for analysis

* It is important to identify the opinion of person that uses the final product hence analysis is needed. Eg: things like design of an application is acceptable by the end-user or not?
* Analysis is necessary to identify the best ways to design and built the frame work for successful product at the end.
* It is also important because it allows us to identify the core problems that to be solved after product is built.
* It is basically the first step of the software development.

### Object oriented analysis

**Object-oriented analysis** is a popular technical approach for analyzing and designing an application, system, or business by applying OOP. (Point, 2019)

By implementing object-oriented analysis, we can minimize the complexity of the product because we divide project according to their modularity. As the system is divided into different models it is easier for us to upgrade from small system to larger in future as project might not get completed at the first phase it also helps during updating the project. Modifying the application in certain module will not affect the other parts as they are modularized according to their functions.

### Pitfalls and merits of the project

## Requirement

Gathering requirement is the crucial phase of any software development as this is the phase where we will know the purpose of the application. How application final output should be and other. If requirement is not clear it might create problem during implementation phase. As I am following waterfall model it’s very important to clarify the requirement as it is hard to reverse the process once it is done in case of waterfall model. In this phase I will also prioritize the requirements that I have identified. I will be creating different Diagrams which have their own purpose of making development work run smoothly.

### Functional and non- functional requirements

**Functional requirement** specifies the behaviors or function. Any requirements that specifies something the system should do is functional requirements. (ReqTest, 2019) For example: user should be able to update their profile without any errors.

Typical example of functional requirement is:

* Business rules
* Transaction corrections
* Administrative Functions
* Authentication
* Certification etc.

**Non-Functional requirement** specifies the criteria that judge the operation of the oystercatcher that the specific behaviors. It specifies how the system should behave. For example: the database value of the user should be updated in just a mater of a second. Here, performance, efficiency matters.

Typical example of non-functional requirement is:

* Performance
* Availability
* Scalability
* Recoverability
* Reliability
* Maintainability etc.

### Requirement prioritization

Requirement prioritization is done to identify which requirement is more precious and should be completed first in order to achieve the minimum goals (requirements) gathered from the user or clients. We should also prioritize the requirement to minimize the risk that may arise during the development i.e. Highly risky requirements can be implemented at first.

I have prioritized my requirements to understand its importance on the project. For prioritizing the task, I have used MoSCoW prioritization.

**M – Must have**

**S – Should have**

**C – Could have**

**W – Won’t have**

**Functional prioritization:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S/N** | **Functional Requirement** | **MoSCoW Prioritization** | **Justification** |
|  | Registration | M | Allows user to register to the system. |
|  | Login | M | Login grants access to the user to use facility of an application. |
|  | Post(add) advertisement | M | Allows user to post advertisement about the property |
|  | Book property | M | Buyer can book the property and contact owner later to buy the property |
|  | Search property | M | Desired property can be searched. Property can be searched according to location and own customization. |
|  | Add to Wishlist | C | Property that is liked by a user can be added to Wishlist so they can check it later. |
|  | View detail | M | Detail of the property can be viewed. |
|  | Filter property | S | Filtering the search of property according to the wish. |
|  | Comment | S | Comment can be provided so owner can get feedback regarding the price and other. |
|  | Update property | M | Added property can be updated. |
|  | Delete property | M | If we are not interested on posting an advertisement then we can delete it. |
|  | View property | M | Basically, we can view the property. |
|  | Edit profile | S | Profile of the user can be updated. |
|  | Chat | C | Buyer can directly chat with the owner if they are online. |
|  | Cost calculation | M | Cost of the product can be calculated by including the tax. |
|  | Online payment | W | This feature might no be available in the final product but it basically helps to pay for property online. |
|  | Bid property | W | Bidding for the price where owner starts the bit from low possible price. |

**Non-Functional prioritization**

|  |  |  |  |
| --- | --- | --- | --- |
| **S/N** | **Non-Functional Requirement** | **MoSCoW Prioritization** | **Justification** |
|  | Scalability | M | Application should be able to run in different environment. |
|  | Efficiency | M | Application should be effective regarding time, cost and other. |
|  | Verification | M | Verification of the user should be done to know their identity (Via login). |
|  | Reliability | M | Application should be trustworthy (should be able to generate correct result ). |
|  | Usability | M | Should be easily useable. |
|  | Interoperability | S | Exchange of the information. |
|  | Maintainability | M | Application can be easily maintainable. |

## Natural Language Analysis (NLA)

**Natural language analysis** helps to get the list of candidate class for any project. Relationships among the classes and their attributes can also be identified.

Before drawing class diagram, it is necessary to identify the possible classes, attributes and relation between different classes. NLA is the analysis process which helps to identify Nouns, Verb and adjective in the form of the descriptive text.

* **Nouns are the candidate class**
* **Verb are the are the potential functions of the class**
* **Adjectives are the potential attributes.**

Steps of constructing class diagram:

* Identify all the possible nouns and verbs
* Filtration is necessary as I am required to identify the genuine classes among all the classes.

For filtration following task was performed:

* Got rid of duplicate
* Complex words were removed
* Removing Irrelevancies candidate class (out of scope)
* Synonymous word was removed example: Meeting and Gathering
* Technical word was removed since they should be mention in the future example: keep a database.

Similar filtration process is also performed to identify the verbs. All the above filtration process was done to identify the suitable class for the project.

## Initial class diagram

It is not a final class diagram but it helps to provide small overview and structure of system in term of classes. Relation between the classes can also be identified (Inheritance, Association etc.). Initial class diagram is shown below.

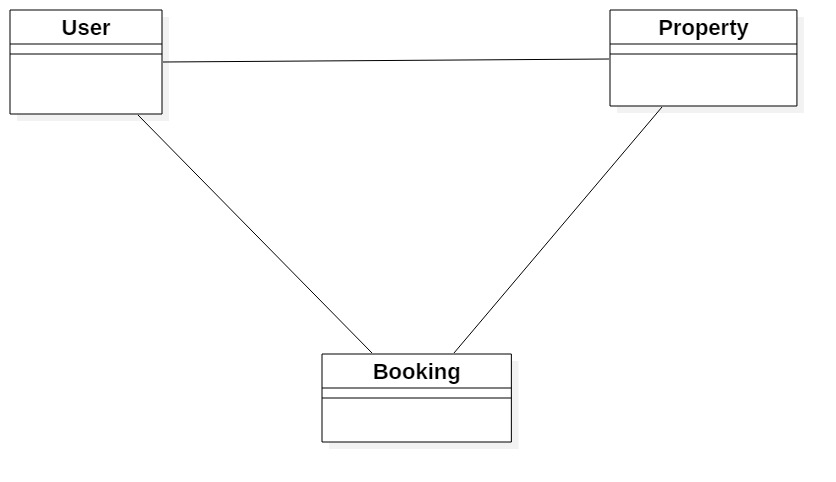


Figure 1 Initial Class Diagram

## Use case Diagram

Use case diagram helps to represent the action that will be performed by different **actors**. Action performed by the actor is shown is **use cases.** Actor can be user, customers etc. it is based on the requirement of the system.

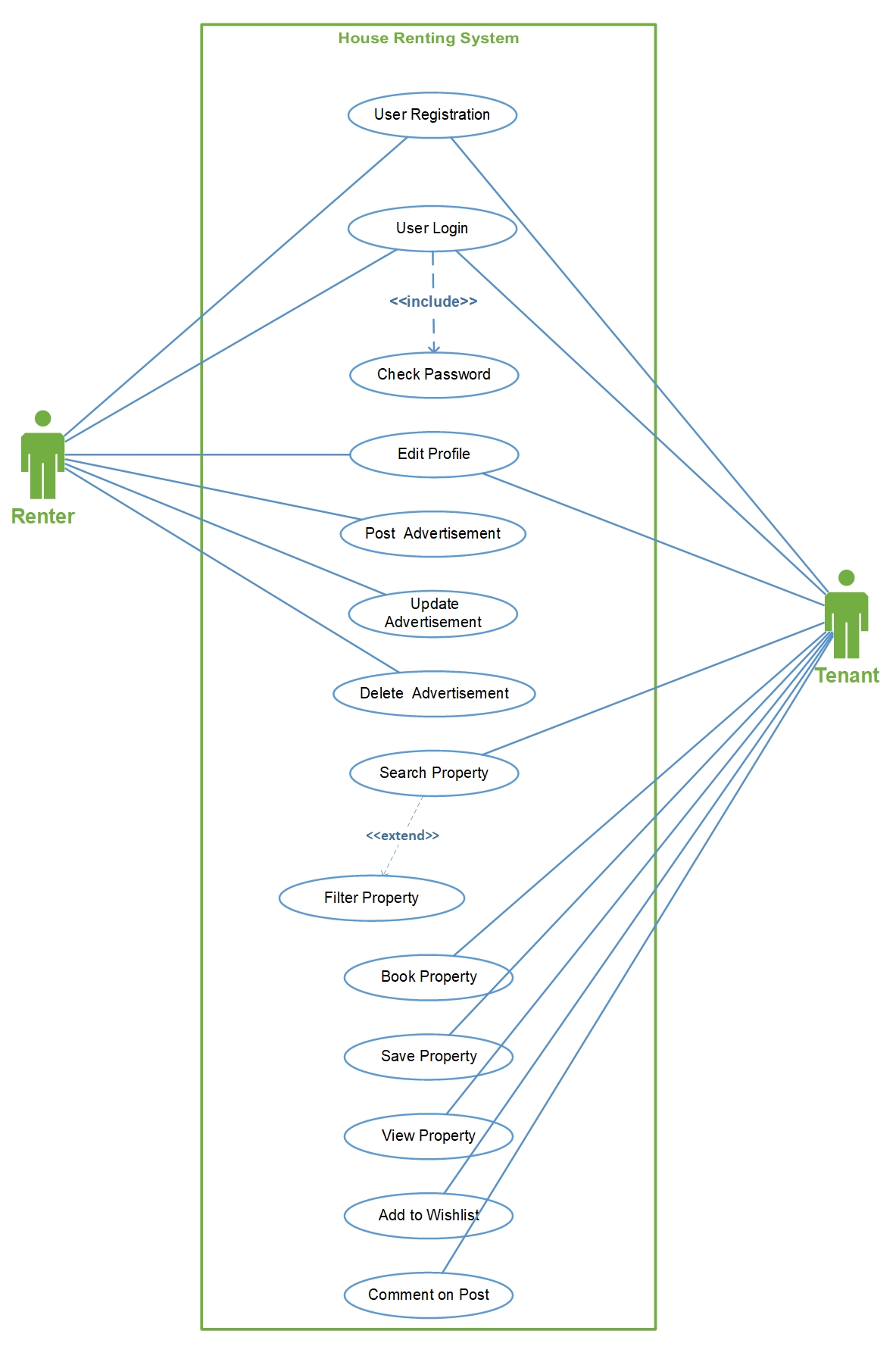


Figure 2 Use case diagram

* **Stickman** Figures represents the **actor** of the system.
* **Oval shape** represents the different **use case** that is performed by the actors.
* **Line** helps to represent the **relationship** of the actor (in above figure relationship is association).
* **Boundary** around the use case helps to represent the system boundary.

Some of the advantage of use case diagram are:

* Helps to understand who is directly related with the system.
* Helps to identify the different tasks that can be performed by the system.
* It also can help to identify the functional requirement of the system.

Use case diagram was created using Microsoft **VISIO.**

## System Architecture

I am going to use 3-tier Architecture. 3-tier architecture is a hierarchical software architecture which is divided into three layers: A presentation layer, An application layer and A data layer. Reasons for using this architecture are:

* **Increase in efficiency:** Work is divided into several system as each tier has their own function which will help to increase efficiency.
* **Increase in security:** Since RDBMS provides single point access and governs who is retrieving the data and how it is updated.
* **Increase in scalability:** Since System can run in different hardware and OS. The technological stack (OS or related utilities) can be updated without impacting other areas of application.

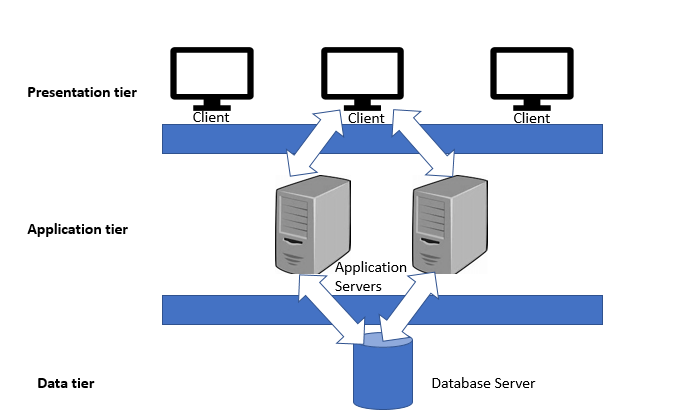


Figure 3 Three tier architecture

**Presentation tier:** Presentation tire is the frontend layer primarily consists of user interface. Interface is commonly available in graphical form from which user interacts with the application.

**Application tier:** Application tier contains the business logics which drives the application which may be written in different programming languages example: C#, PHP, JAVA etc.

**Data tier:** Consists the database/ data storage system Example: MySQL, Oracle I have used **MySQL** database and **XAMPP** 3.2 application to commutate with the database.

# Chapter 3: Design

## Justification

Design just doesn’t represent the UI Design of the application but it also represents methods, functions, objects and overall interaction and structure of the code. Design phase is one of the most important phase of software design because in this phase we decide how we are going to build our project from the very beginning i.e. its framework. Division of the classes also takes place according to what data will a class carry and according to the main function performed by the particular class. Similarly, UI modeling is also an important aspect of the design because it determines how user is going to react to it. Example: software that performs task well but has bad interface might not impress the user but software with good design can flow positive vibes to the user. All the diagram was created using **Star UML** whichis great tool for creating UML diagrams.

## Dynamic modelling

Dynamic modeling is used to define behavior of the system. Dynamic modeling is also used to represent interaction, workflow of the system.

### Activity Diagram

Activity diagram is a dynamic diagram which helps to represent logical process of a system in diagrammatic format. It represents dynamic aspect of a system. Basically, it is a flow chart that shows flow of one activity to other activity. (TutorialPoints, 2019) Activity diagram also help to show detail process of different use cases that is identified in use case diagram. They also show how system work flow are constructed i.e. how they start and end.

These are the notations I have used in my activity diagrams:

* Swim-Lane: All of the classes are divided into different swim-lane to make it easy to understand.
* Initial node: Initial node represents the starting point of the activity.
* Activity: Rectangular boxes that represent the actions.
* Decision: Diamond shape that shows different choices and conditions.
* Flow: they are the arrow which guides towards the end of the diagram.
* Final node: This represent the end of activity.

The activity diagram is shown below:

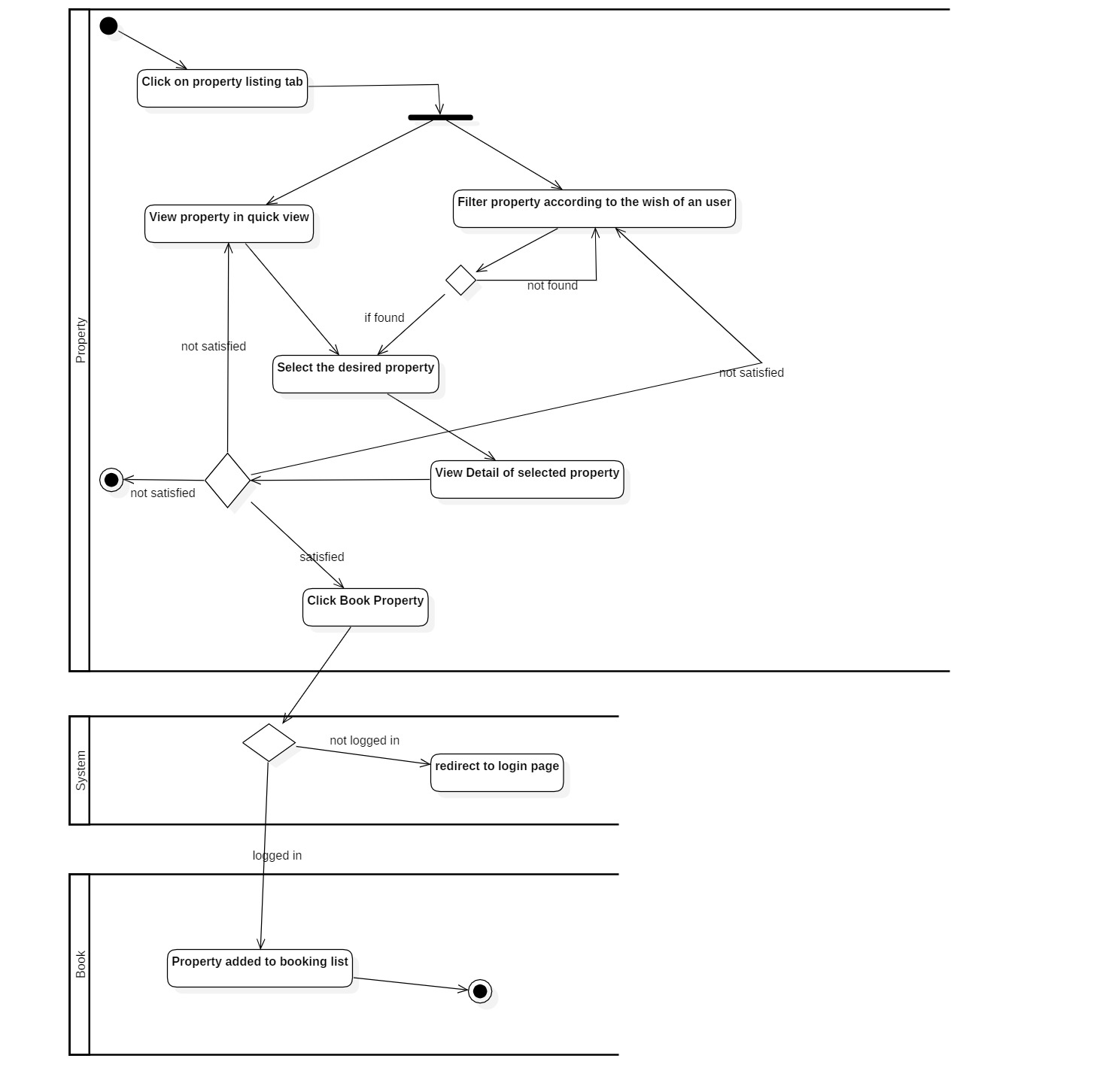


Figure 4 activity diagram showing property booking

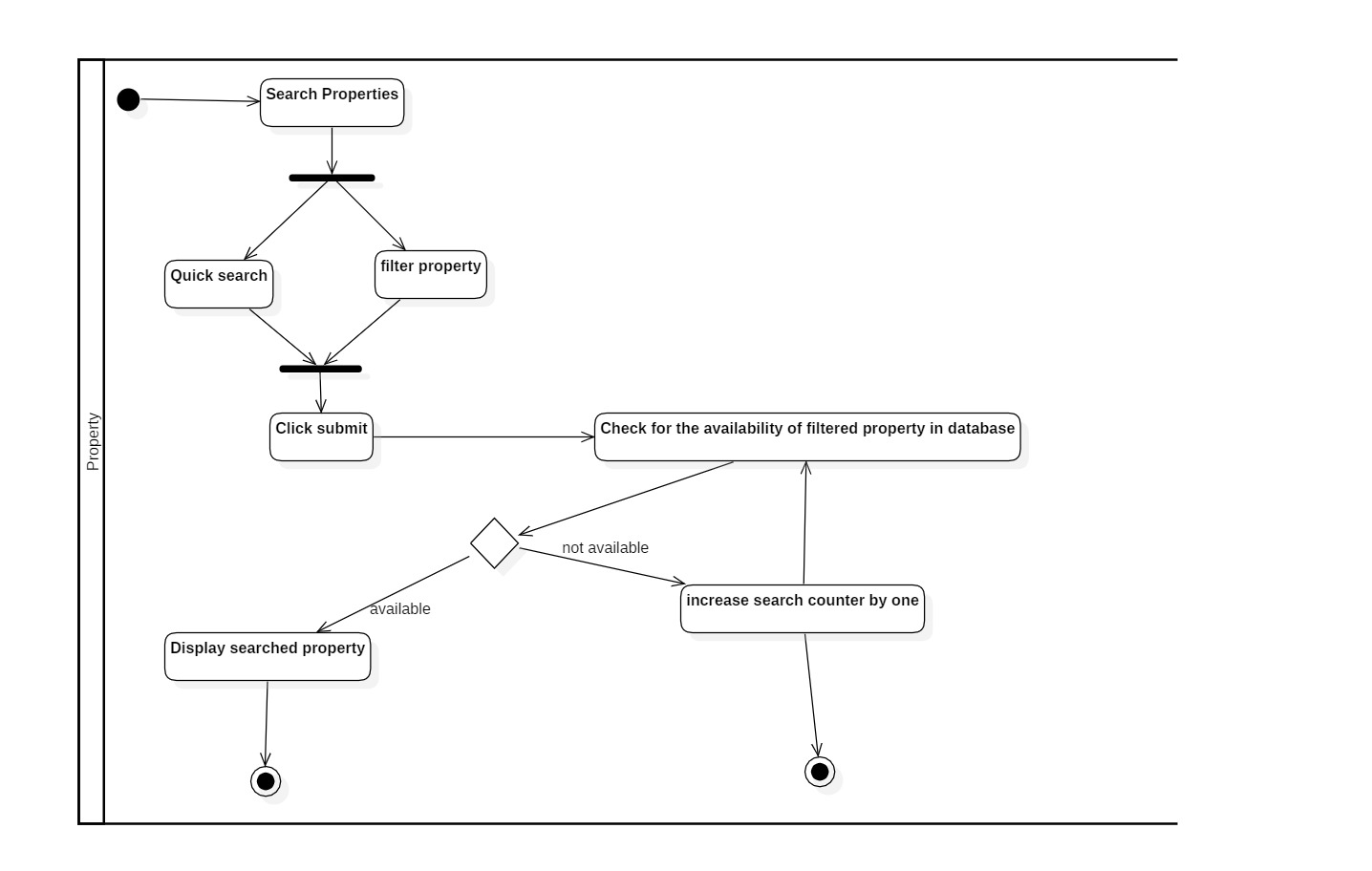


Figure 5 Activity diagram - search property

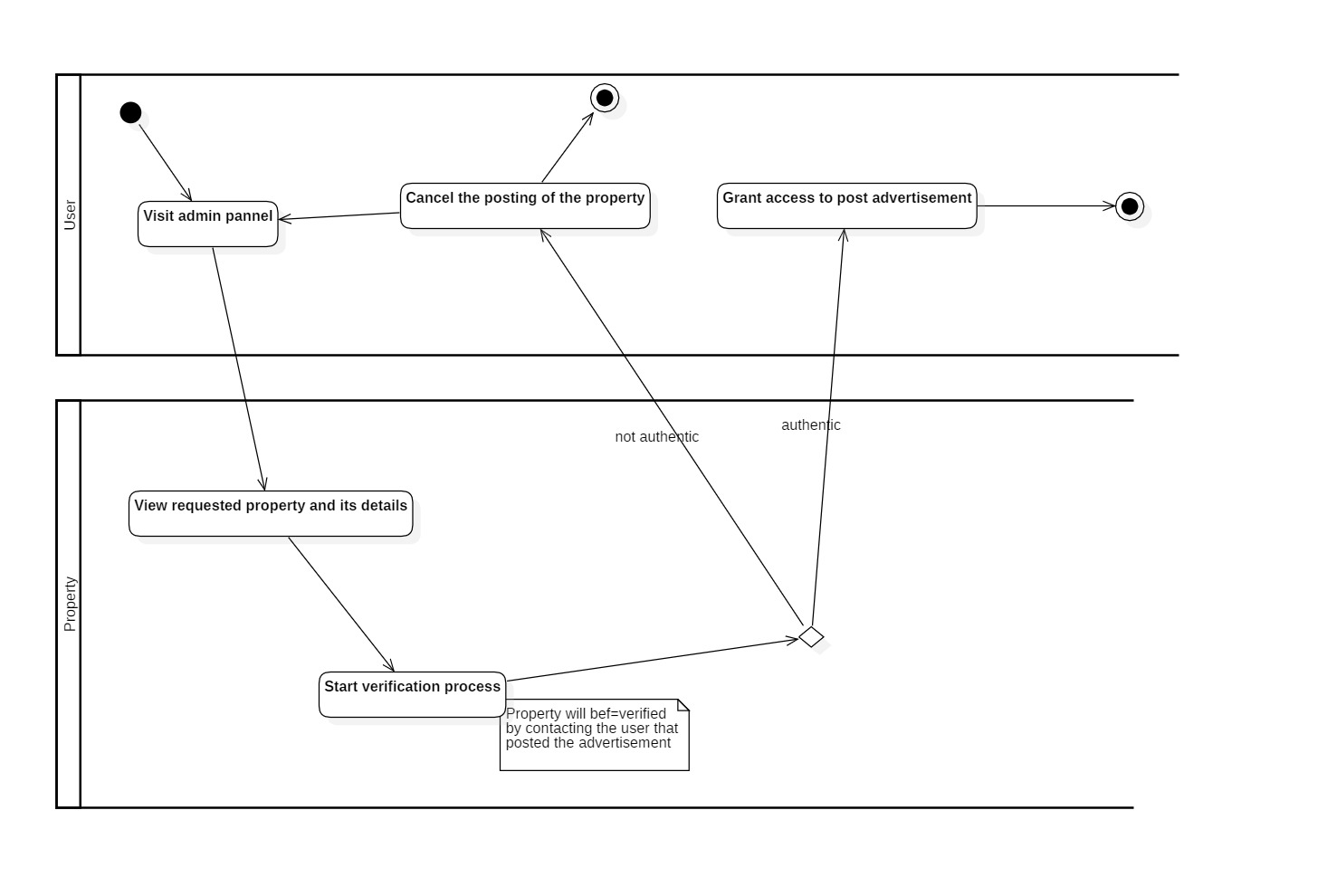


Figure 6 Activity Diagram - Admin verification

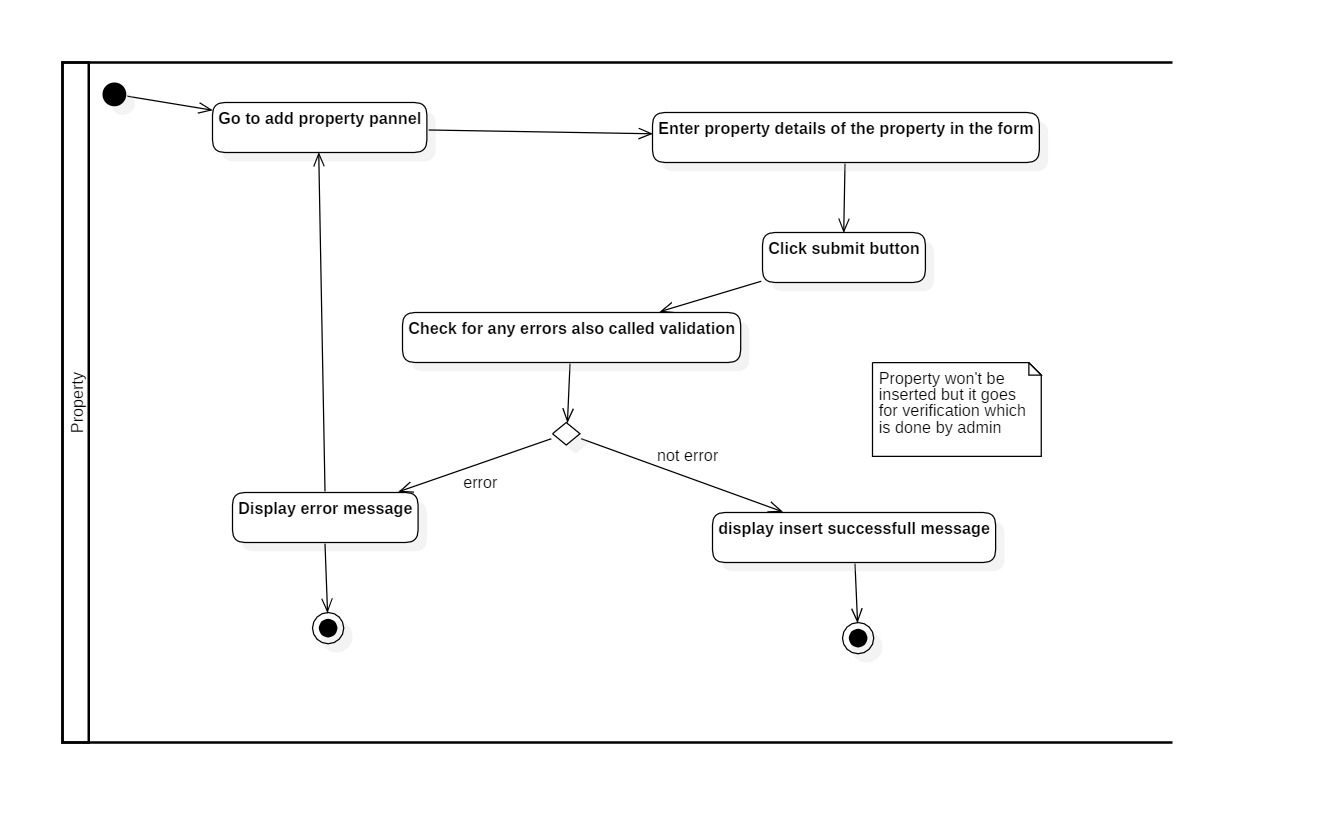


Figure 7 Activity Diagram - Add property

### Sequence Diagram

Sequence diagram is a dynamic diagram which helps to show logical process of the system in sequential way which looks much cleaner as compared it to activity diagram. How the user interact with the system and internal functionality of the system is also shown.

Sequence diagram consists of different notations like:

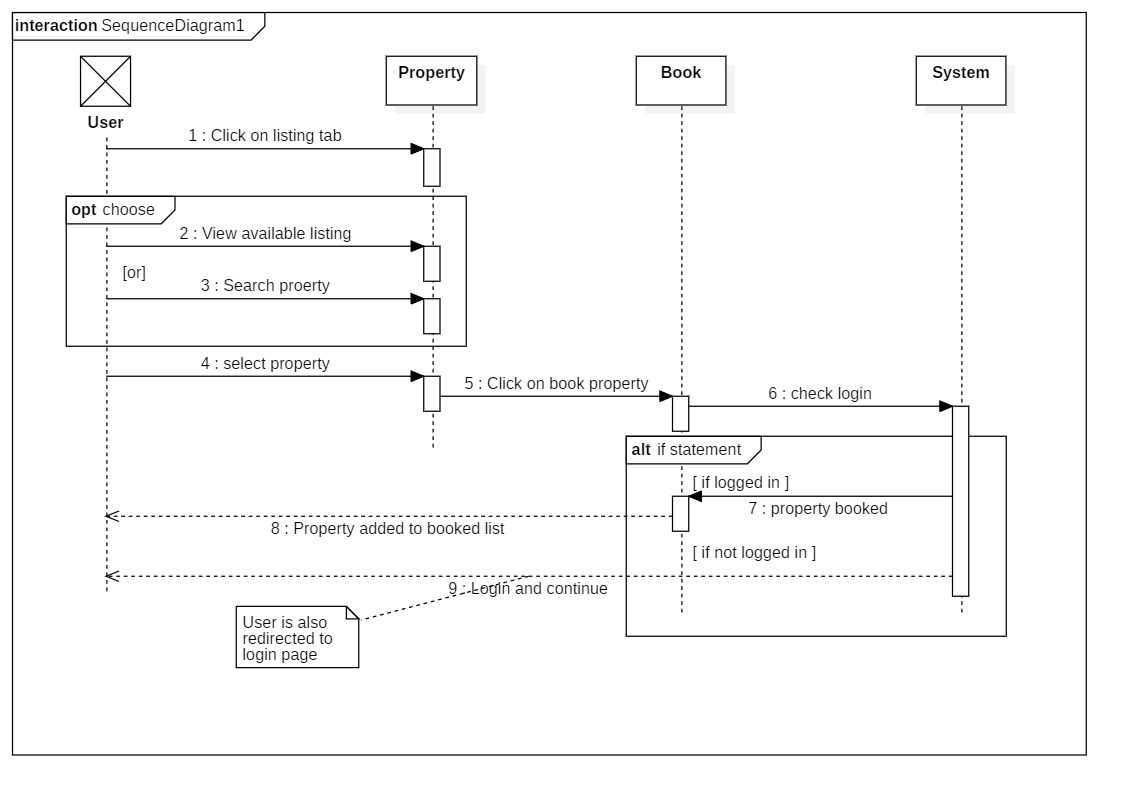


Figure 8 Sequence Diagram - Book property

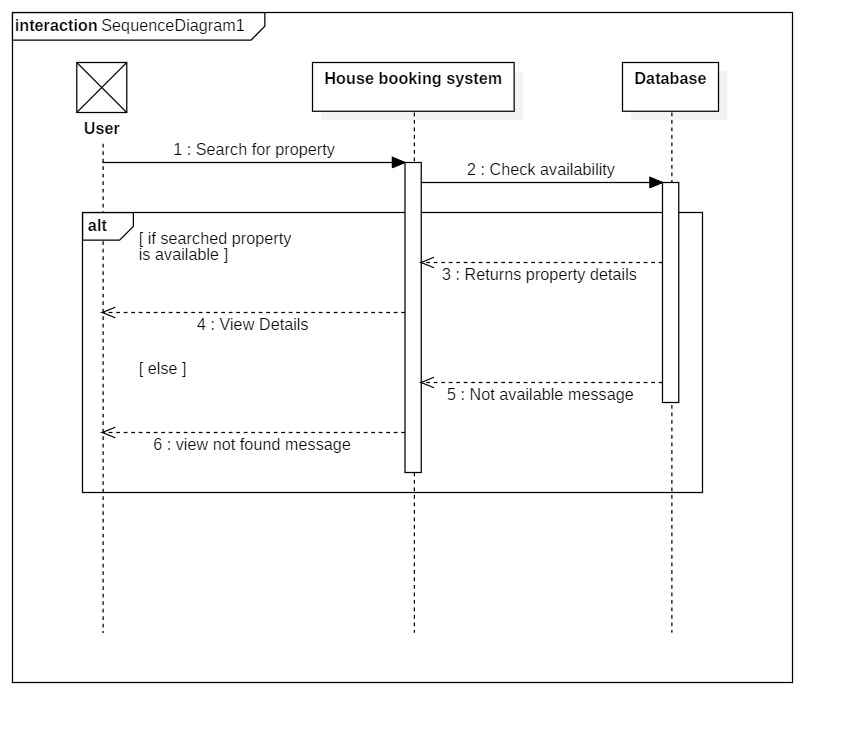


Figure 9 Sequence Diagram - Search

## Chapter 4: Implementation

Implementation is the fourth stage of SDLC where actual coding of the program takes place. I stage like analysis we analyzed the information and in design I created different diagrams and, I have converted those diagrams in to working code in this phase of software development.

### Programing language and framework

Choosing right programing language is crucial. I have chosen **PHP** as a main programming language and I have also used **JavaScript** and bit of a **jQuery** to perform multiple functions and to provide some cool feature and animations to the interface. Since I have built my application in web form this is the main reason for choosing PHP as my main programming language for **framework,** I have used **Laravel** framework because it is simply the best PHP framework available. It supports MVC pattern which is probably the best pattern to write the program on and it also increases performance of the system. MVC eliminates the problem of code management. It has multiple built-in functionality which comes in handy during development of an application. Laravel consists of object-oriented libraries which provides amazing features like security and patches. Laravel Artisan command is used to execute any king of services.

MVC pattern in Laravel looks like this:

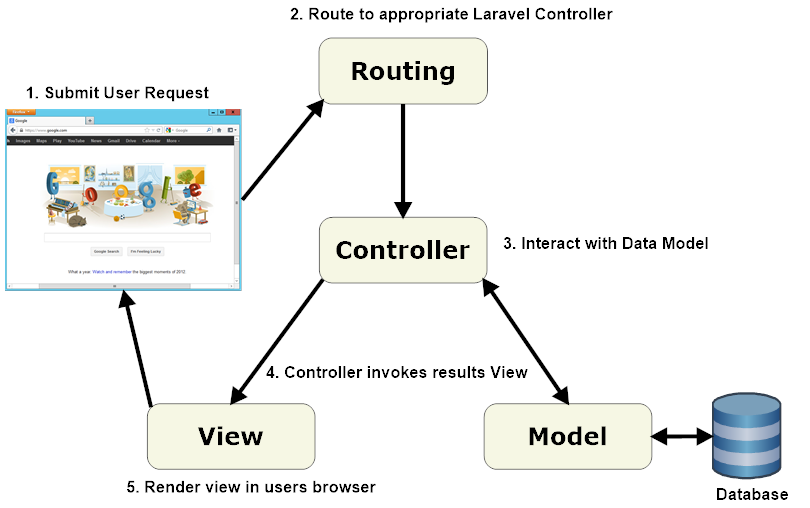


Figure 10 Laravel MVC design pattern

Figure link: (ValueCoders, 2019)

## Chapter 5: Testing

The fifth stage of SDLC is testing where code is tested to make it sure that it is performing as intended. Without any flaws, without any errors. Testing also helps to assure the quality of the code. Basically, it is the process of executing a program in order to find the bugs which will allow us to solve the problem that we have identified. This process of **validating and verifying** helps us to make sure that software/ program is bug free. There are various types of testing methods but in this topic I have particularly focused on unit and black box testing.

**Vendor/bin/phpunit** is used to execute the test and **Vendor/bin/phpunit –filter (function name)** is written in command to execute individual function. Laravel automatically sets configuration for testing in .env file. Laravel also gives us chance to manually configure environmental variable.

**Black box testing:**

Black box testing is a kind of testing where the internal behavior, implementation etc. is unknown. It is also known as behavioral testing. I have tested my GUI using black box testing. Basically, we identify how the system is performing. We can identify following categories of error using Black box testing: (fundamentals, 2019)

* Error in validation
* Error in performance of the system
* GUI error also called as interface error.

Some of the testing along with their test cases is shown below:

Test Case – 1

|  |  |
| --- | --- |
| **Test Suite** | **Black Box Testing 1** |
| Test Number | 1 |
| Purpose of the test | password validation |
| Expected Result | Successful validation |
| Actual result | success |
|  |  |

Test Case – 2

|  |  |
| --- | --- |
| **Test Suite** | **Black Box Testing 2** |
| Test Number | 2 |
| Purpose of the test | Empty box form validation |
| Expected Result | Successful validation |
| Actual result | success |

Test Case – 3

|  |  |
| --- | --- |
| **Test Suite** | **Black Box Testing 3** |
| Test Number | 3 |
| Purpose of the test | email validation |
| Expected Result | Successful validation |
| Actual result | success |

**Unit testing:**

Unit testing are the test that focus on very small and isolated part of the code. In fact, most unit test focus on the single method single or small functionality testing. According to **software testing** unit testing is the level of testing where each individual unit are tested. Individual units may be single function, procedure etc. (fundamentals s. t., 2019)

Some unit testing along with their test cases are shown below:

Test Case – 1

|  |  |
| --- | --- |
| **Test Suite** | **Unit Test 1** |
| Test Number | 1 |
| Purpose of the test | To check whether the user can register or not |
| Test Data | ("GET", "/login/$email/$password") |
| Expected Result | User should be registered successfully. |
| Actual result | User was successfully registered |
| Class involve in test | User |
| Comments |  |

Test Case – 2

|  |  |
| --- | --- |
| **Test Suite** | **Unit Test 2** |
| Test number | 2 |
| Purpose of the test | To check whether the user can login or not |
| Test Data |  |
| Expected Result | User should be able to login successfully |
| Actual result | User can login |
| Class involve in test | User |
| Comments |  |

Test Case – 3

|  |  |
| --- | --- |
| **Test Suite** | **Unit Test 3** |
| Test number | 3 |
| Purpose of the test | User should not be able to go to login page once they are logged-in |
| Test Data |  |
| Expected Result | user should be unable to visit the login page |
| Actual result | Test result was positive |
| Class involve in test | User |
| Comments |  |

Test Case – 4

|  |  |
| --- | --- |
| **Test Suite** | **Unit Test 4** |
| Test number | 4 |
| Purpose of the test | Whether user can update the profile or not |
| Test Data |  |
| Expected Result | User should be able to update the profile |
| Actual result | Test was not passed |
| Class involve in test | User |
| Comments | Asserting the provided data failed due to some reason nut test was passed in black box testing |

Test Case – 5

|  |  |
| --- | --- |
| **Test Suite** | **Unit Test 5** |
| Test number | 5 |
| Purpose of the test | Testing whether the user is admin or not. |
| Test Data |  |
| Expected Result | User should be admin |
| Actual result | Test failed |
| Class involve in test | User |
| Comments | BadMethodCallException (test was passed in black box testing) |

Test Case – 6

|  |  |
| --- | --- |
| **Test Suite** | **Unit Test 6** |
| Test number | 6 |
| Purpose of the test | Whether user can search by location or not |
| Test Data |  |
| Expected Result | User should be able to search via location |
| Actual result | Search was successful |
| Class involve in test | Properties |
| Comments |  |

Test Case – 7

|  |  |
| --- | --- |
| **Test Suite** | **Unit Test 7** |
| Test number | 7 |
| Purpose of the test | Whether property type is empty or not |
| Test Data |  |
| Expected Result | Property type box should not be empty |
| Actual result | Result was positive |
| Class involve in test | Proptypes |
| Comments |  |

Test Case – 8

|  |  |
| --- | --- |
| **Test Suite** | **Unit Test 8** |
| Test number | 8 |
| Purpose of the test | Whether property can be added or not |
| Test Data |  |
| Expected Result | Property should be added |
| Actual result | Test failed |
| Class involve in test | Properties |
| Comments | Integrity constraint violation |

Test Case – 9

|  |  |
| --- | --- |
| **Test Suite** | **Unit Test 9** |
| Test number | 9 |
| Purpose of the test | Whether user can add property if they aren’t logged-in |
| Test Data |  |
| Expected Result | They should not be able to visit the add property page at all |
| Actual result | Test result was positive and user was unable to visit. |
| Class involve in test | Properties, Facilities, Rooms |
| Comments |  |

Test Case – 10

|  |  |
| --- | --- |
| **Test Suite** | **Unit Test 10** |
| Test number | 10 |
| Purpose of the test | Whether user can view property without login or not |
| Test Data |  |
| Expected Result | They should be able to view the property |
| Actual result | User was able to view the property |
| Class involve in test | Properties, Facilities, Rooms, User |
| Comments |  |

Test Case – 11

|  |  |
| --- | --- |
| **Test Suite** | **Unit Test 11** |
| Test number | 11 |
| Purpose of the test | Whether user can book property if they are not logged-in |
| Test Data |  |
| Expected Result | Should not be able to book property if not logged-in |
| Actual result | Test failed |
| Class involve in test | Properties, Bookings |
| Comments | Integrity constraint voilation |

## Chapter 6: Other project issues

### Risk Management

Risk management is the process of identifying, accessing and prioritizing the risk that may occur in the system. Risk is prioritized according to the harmfulness of the risk and the risk that harms the most is either solved or an alternative solution is prepared for it. It is the situation where the project is exposed to danger. **Risk can not be solver completely but it can be minimized to make sure that system is running smoothly.** So, being proactive to these kings of problem will definitely help in minimizing the risk.

In this particular project I have calculated risk by using formula that is:

**Impact = Likelihood \* Consequences**

**Risk Likelihood values are shown as follows**

|  |  |
| --- | --- |
| Likelihood | Value |
| Low | 1 |
| Medium | 2 |
| High | 3 |

**Risk Consequence values are shown below**

|  |  |
| --- | --- |
| Consequence | Value |
| Very low | 1 |
| Low | 2 |
| Medium | 3 |
| High | 4 |
| Very High | 5 |

**Some of the risk that I have identified are:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S.NO | Risks | Likelihood | Consequences | Impact | Solution |
| 1 | Schedule risk | 2 | 4 | 8 | Task should be completed in time from the very beginning to an end |
| 2 | Cost and resources risk | 2 | 3 | 6 | Available resources should be managed properly to use them in future. |
| 3 | Failure to meet requirement | 2 | 5 | 10 | We should not proceed our work until we are clear about the requirement. |
| 4 | System failure | 1 | 5 | 5 | Project should be backed up properly so it can be used even there is system failure. Eg: backup in cloud. |
| 5 | Lack of planning | 2 | 4 | 8 | Better planning should be done to avoid above mentioned risks. |

# Configuration Management

Configuration Management is the process of establishing and maintaining consistency of the product’s performance, functional and physical attributes with its requirements, design and operation information through its life (Rouse, 2019) . In Information Technology it is also called **Software Configuration Management (SCM). SCM** process is looked up by the active person searching for the best solution to handling changes in software projects. Git hub is also one of the ways to manage the modified code where all of the changes are updated regularly.

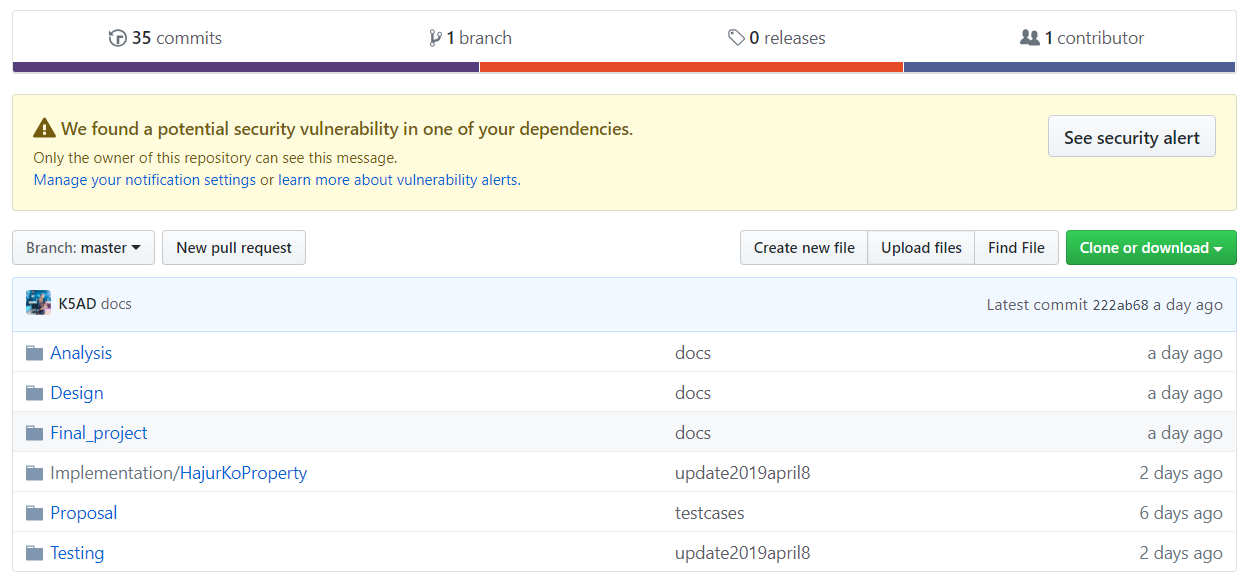


Figure 11 Configuration Management - gilt repo

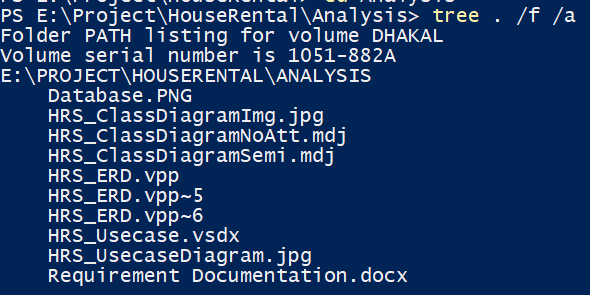


Figure 12 Configuration Management - tree structure – Analysis

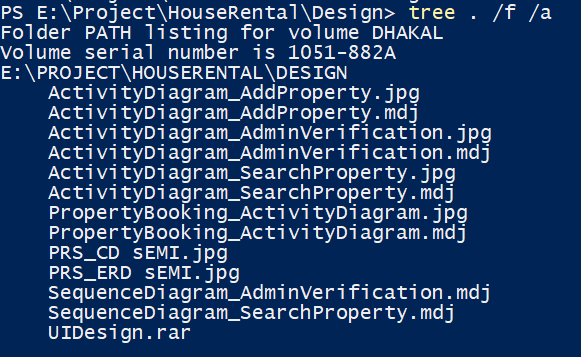


Figure 13 Configuration Management - tree structure - Design

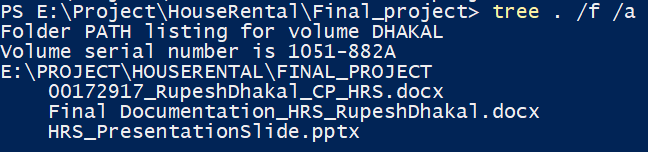


Figure 14 Configuration Management - tree structure - Final project

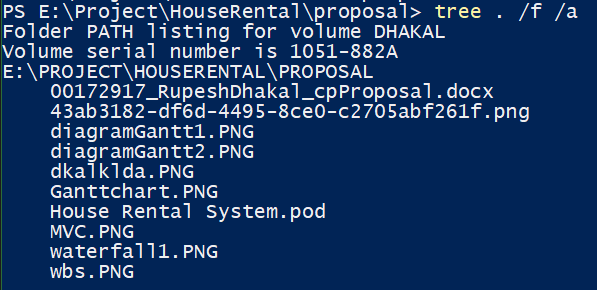


Figure 15 Configuration Management - tree structure – Proposal

### Future work

There was some work that was not completed due to some issues like time management and so on. Those work was holed for future development i.e. in the next update of this application.

Some of the future work are listed below:

* Improvement in design using UI/UX: Improving design will make user feel that there were certain changes made and also make them to stick for longer period of time.
* Realtime chat system between owner and buyer. it can also be said as community forum where one person places their queries and get rid of their confusions.
* Online payment system: This time around user can only book the property which is not very comfy. Enabling online payment will allows them to buy the property wherever they are.
* Strong security mechanism: Currently this web application consists of basic verification which updated to version 2.
* Google map API implementation: it’s costly to apply google map for startup project hence it will also be implemented in future applications update.

### User manual

User manual is necessary for guiding user about using the application. User manual are intended to assist user, providing the detail about the system and inform on how to use a particular system.

**Login System**

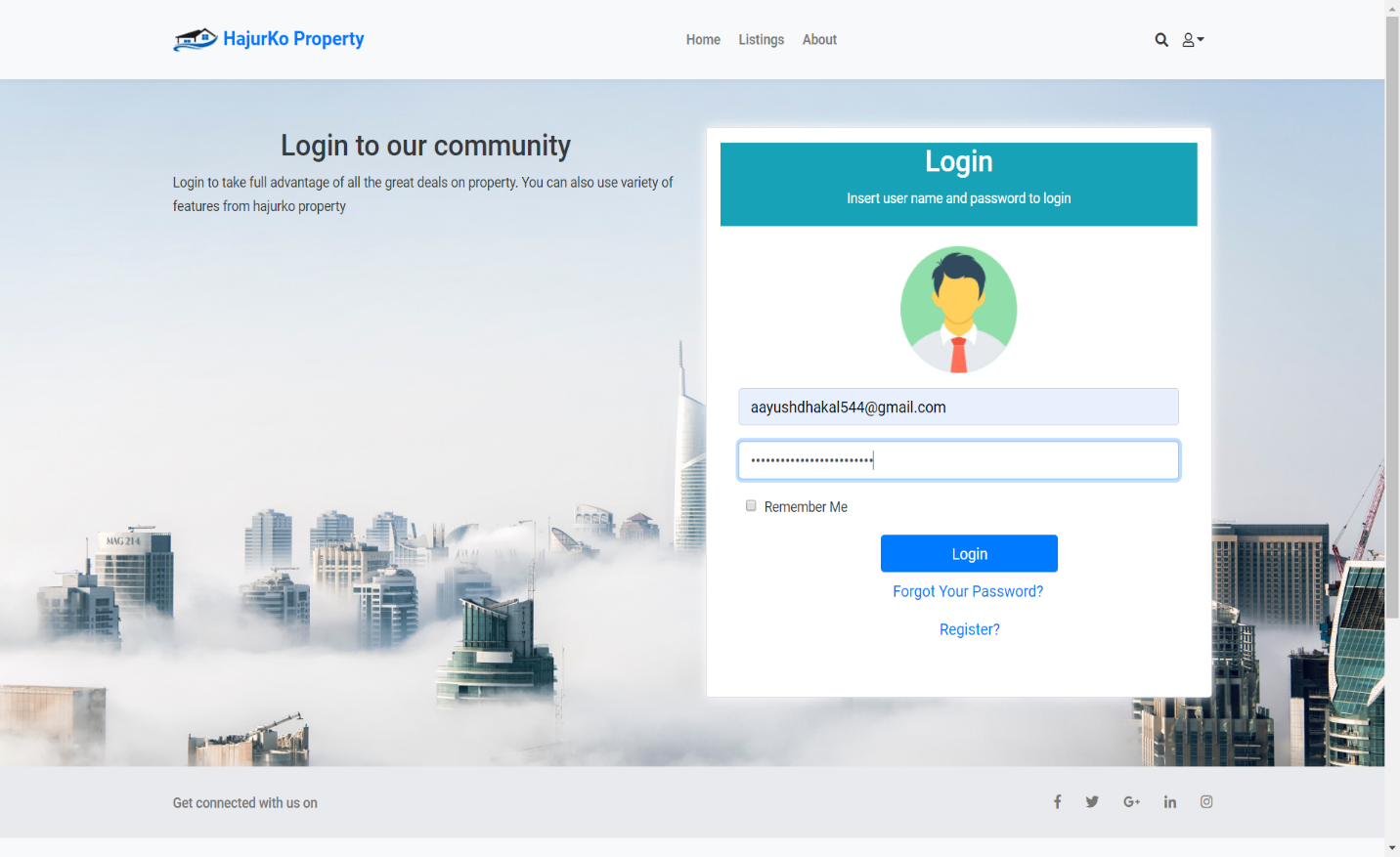


Figure 16 User Manual – login

**Steps to follow:**

* Firstly, enter correct website URL
* Insert valid credentials (Email and Password)
* Click login button
* Then, system will validate information you have given
* If information matched with the database information you will be successfully logged-in.

**Registration System**



Figure 17 User Manual – Registration

**Steps to follow:**

* Firstly, enter correct website URL
* Insert valid credentials **as shown in the above diagram**
* Click register button
* Then, system will validate information you have given like valid email, valid passwords (password length)
* If information matched with the database information you will be successfully logged-in.

**Quick Search Property**

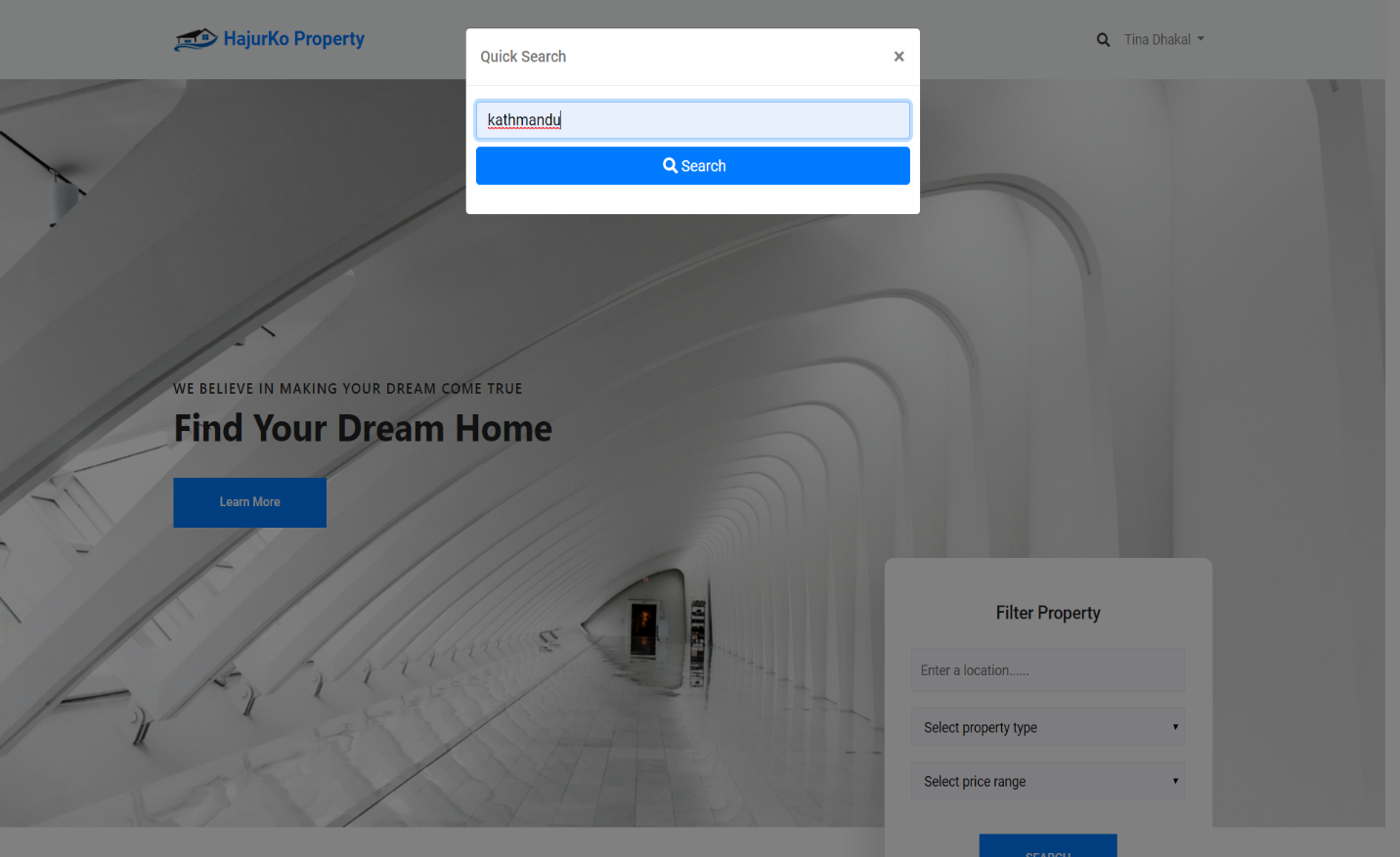


Figure 18 User Manual - Quick Search

**Steps to follow**

* Firstly, enter correct website URL
* Click on search icon that is shown in navigation bar (left hand side of user’s name dropdown)
* Quick search window will appear
* Insert in location of the property as you can only search the property via location here
* Click on search button
* You can view your result if it is available in the database

**Custom property searching**

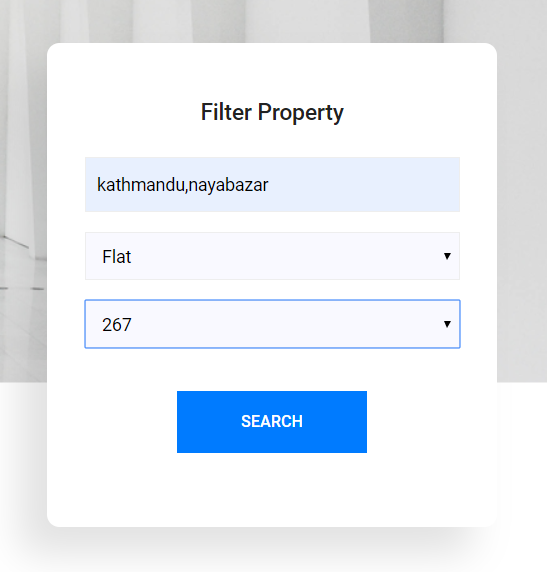


Figure 19 User Manual - Custom Search

**Steps to follow**

* Firstly, enter correct website URL
* Scroll down the home page until you find the box that looks similar as above figure.
* Insert in the desired location, property type, price
* Click on search button
* You can view your result if it is available in the database

**Add property window**

**Steps to follow**

* Firstly, you have to be logged-in to add the property
* Click on user name shown on top right corner of navigation bar as highlighted in figure below

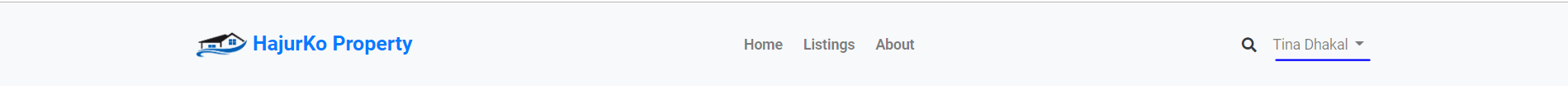
****

Figure 20 User Manual – navbar

* You will be presented with menu like this:

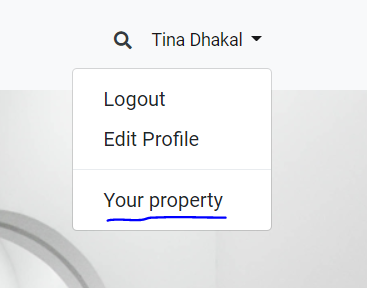
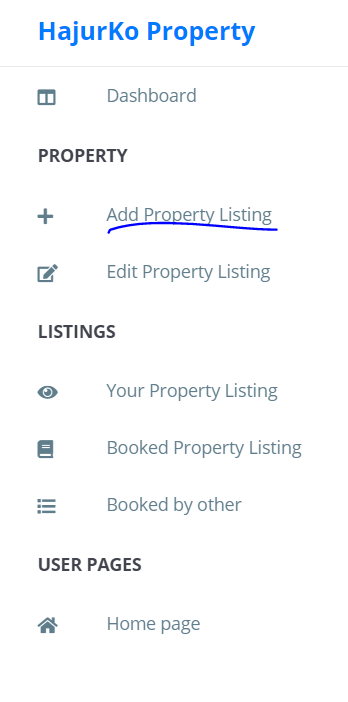


Figure 21 User Manual – dropdown

* Click on your property as highlighted in above figure



* You will get the side navigation as shown in the above figure
* Then, click on add property as highlighted in above figure
* You will be directed to adding property page as shown in figure below

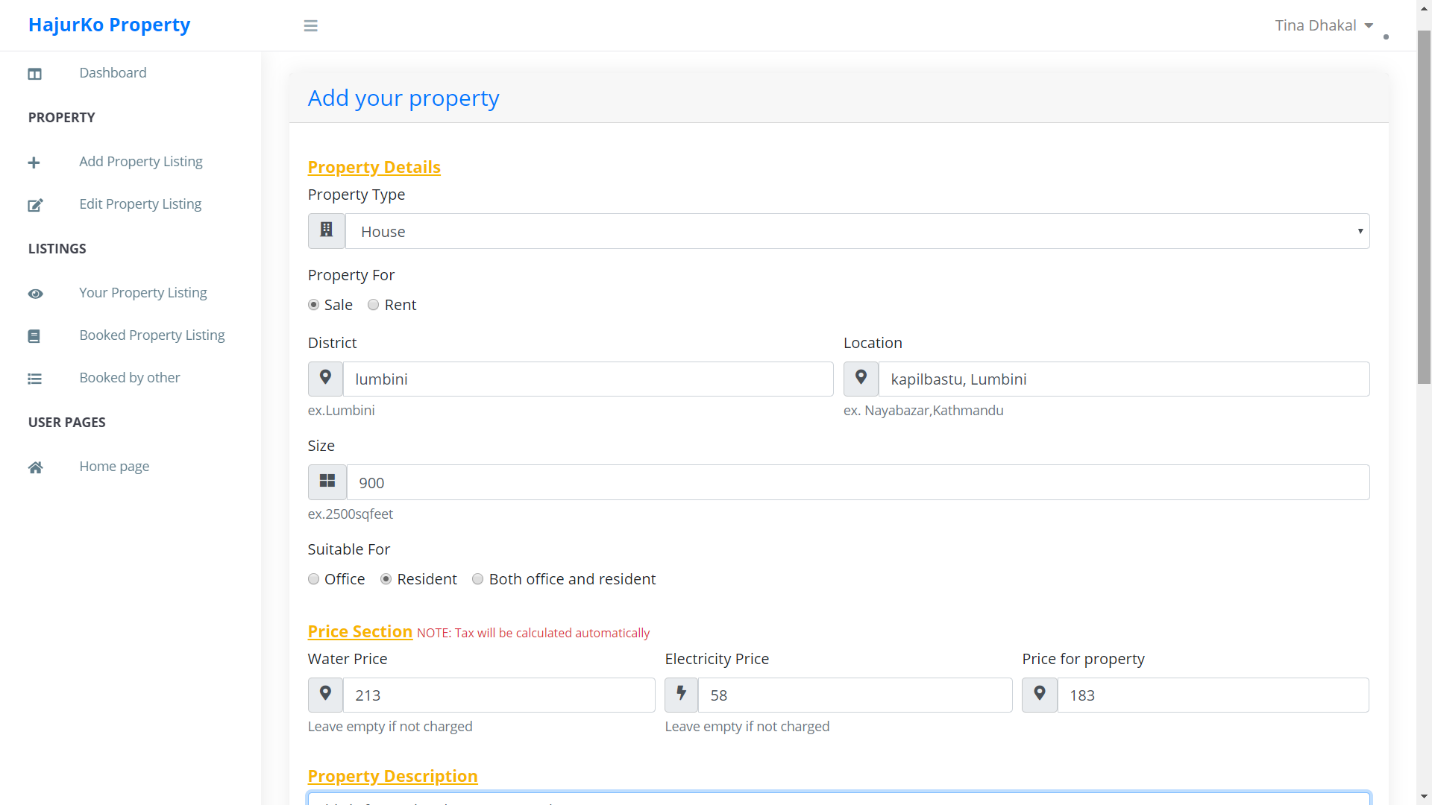


Figure 22 User Manual - Adding property

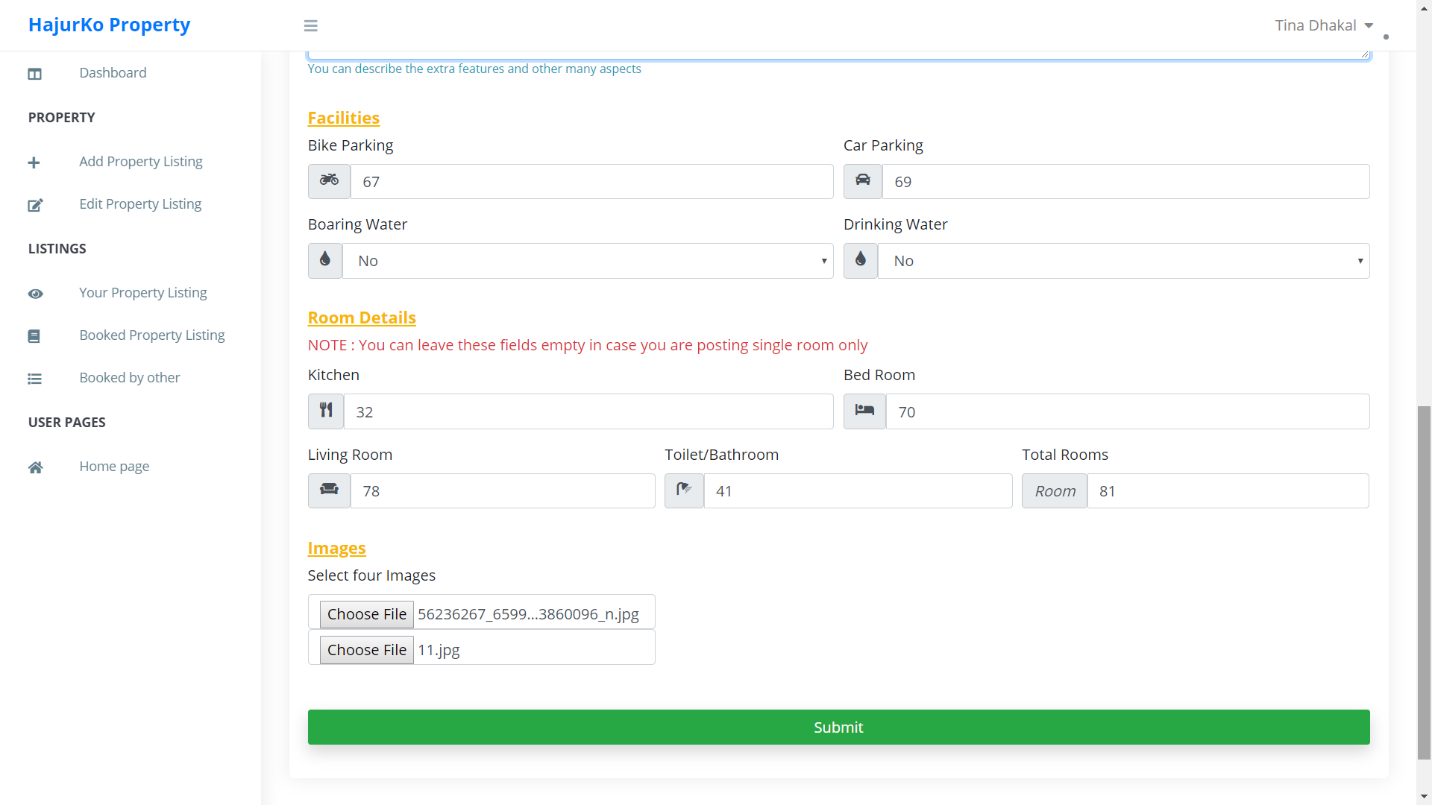


Figure 23 User Manual - Adding Property 2

* Insert all the fields and click on submit button

# Chapter 7: Conclusion

At conclusion I successfully created web application that performs main task which I intended. All the basics requirements were fulfilled. Some of the functionality are missing due to some reason which is listed in future work above. I hope this project will solve the current problem that is faced by people to search property. Firstly, analysis was done to identify the project purpose. Then design was created (different dynamic and structural models). Based on the design implementation was done which created working application. The application is also completely tested in crucial parts to ensure it work as intended. Lastly, other issues regarding the project was mentioned.

# Chapter 8: Reference and bibliography

fundamentals, s. t. (2019, 4 9). *software testing fundamentals*. Retrieved from software testing fundamentals: http://softwaretestingfundamentals.com/unit-testing/

fundamentals, S. t. (2019, 4 9). *Software testing fundamentals*. Retrieved from Software testing fundamentals: http://softwaretestingfundamentals.com/black-box-testing/

Point, T. (2019, 4 8). *Tutorials Point*. Retrieved from Tutorials Point: https://www.tutorialspoint.com/object\_oriented\_analysis\_design/ooad\_object\_oriented\_analysis.htm

ReqTest. (2019, 4 8). *ReqTest*. Retrieved from ReqTest: https://reqtest.com/requirements-blog/understanding-the-difference-between-functional-and-non-functional-requirements/

TutorialPoints. (2019, 02 19). *TutotialPoints*. Retrieved from tutorialPoints: https://www.tutorialspoint.com/uml/uml\_activity\_diagram.htm

ValueCoders. (2019, 4 9). *ValueCoders*. Retrieved from ValueCoders: https://www.valuecoders.com/blog/technology-and-apps/laravel-best-php-framework-2017/

# Chapter 9: Appendix