**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**

# Introduction

House/Room rental system is an 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. Best deals can be viewed at top of the applications. This application is not limited up to the list you can also search for specific rooms, house and flats that you prefer. Old ways of searching house, rooms or flat will be removed with the help of this application. You can also list down your property or view applied properties by other users. In Short it is an intelligent solution to find suitable home, office for you.

## Justification for project

### Background of the project

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

## Problem Statement

Problem statement determines the problem that the project can solve after the project is ready. 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.

## Description of project

### Features

The features of the project are listed below:

* **User Signup and Login**

User can create their account in case they haven’t and can login to the application which will allow them to access rest of the feature.

* **Quick view of best value**

User can view the best value of property pinned at top of the application. It will primarily focus for least price products.

* **Can chat with owner of the property**

User will have a facility to chat with the owner of the property which will allow them to negotiate the price and other things

* **Feature of booking the property**

Desired property can be booked to make secure the purchase.

* **Can search property according to location**

User can search the property to the desired location where they find comfortable to commit their daily task

* **User friendly**

There are lots of application that may have similar moto as this very project but this project will surely give better user experience.

This project also consists other minor features which will make using the interface easy and convenient.

# Project Scope

## Scope and limitation of project

This project is user friendly and allows user to register and login. User can also add their advertisement of their own property (House/ Room). They will have a facility to buy the Rooms and houses they desire. The best tool is the chat that will allow user to chat with the owners and can communicate about the property itself.

Everything has its limitation and this project has one. This project is not ready for online payment. If online payment was a thing in this project then user would have a privilege of secure payment user would also have a facility to buy the property on the spot. In future it is possible to eliminate this limitation.

## Aims and Objectives

**Aims:** The aims that I want to achieve are listed below:

* Make available of the facility of posting their property information.
* User should be able to buy the desired property.
* To make user friendly environment
* Make application run smoothly along with performing functionality of the application without any error.

**Objectives:** Action that I will take to achieve my aim are:

* User will have the facility to sign-up and login which will allow them to use the facility of posting the advertisement and also buy the property they desired (Rooms and Houses).
* User can provide the detail along with uploading the picture once they are logged in.
* To make application run user friendly I will understand the requirement clearly first and make sure that all the requirement is well documented to understand what actually project should do.
* To make program free from error I would perform different testing like: Black box testing, Integration testing, Unit testing etc

# Development Methodology

## Methodology used

I have used waterfall methodology to develop the application. Waterfall is the sequential process of developing application i.e. step by stem process so by implementing this methodology I can track my progress pretty smoothly. Other reasons for choosing waterfall methodology are:

* Waterfall model is simple and easy to understand
* Since my project is small and requirements are well understood so using this model will be suitable.
* Project will be easy to manage because the model is rigid itself.

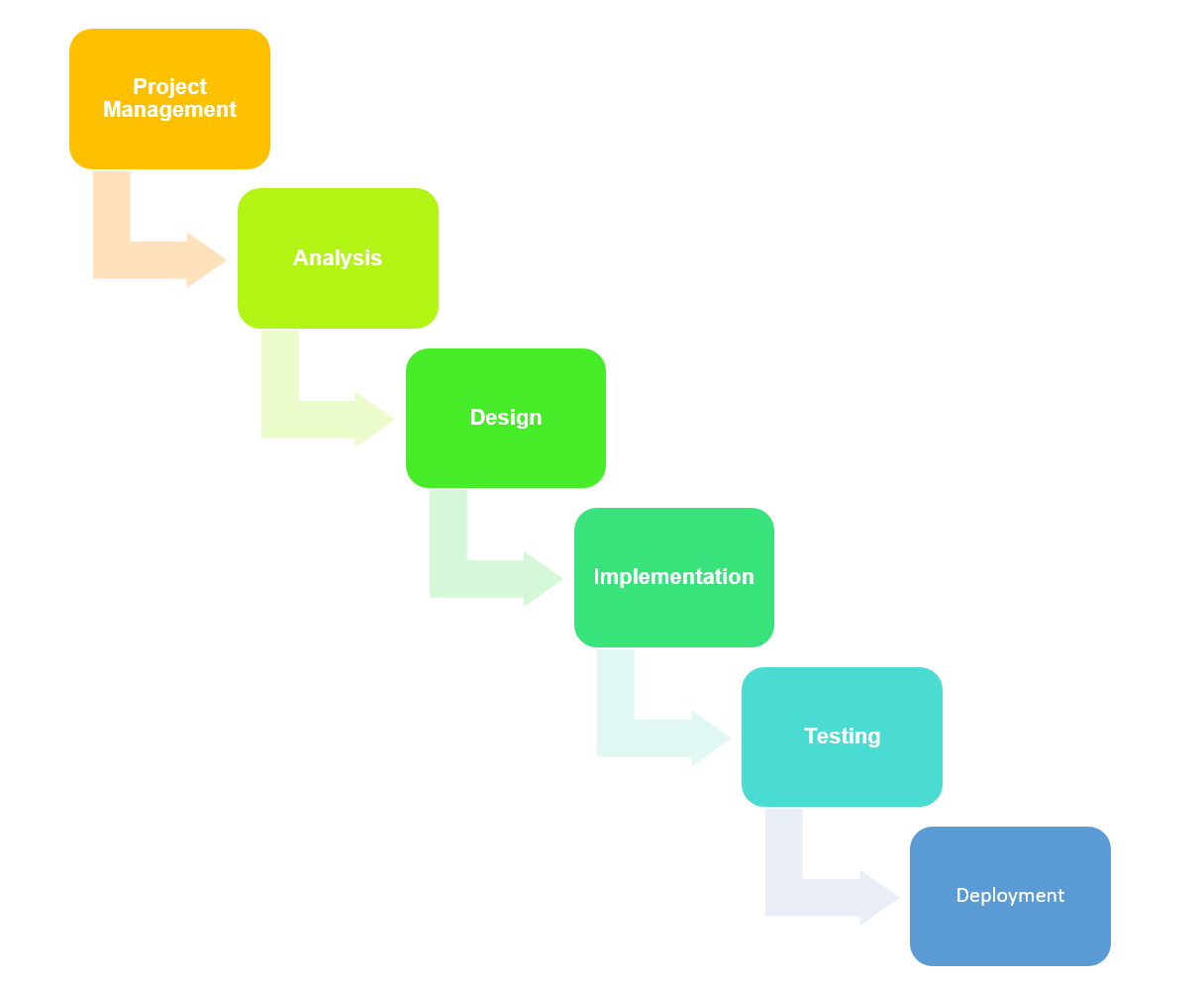


Figure 1 Phases of waterfall model

## Design Pattern

I have used MVC design pattern. Here M - Model, V - View and C – Controller. The reasons for using MVC pattern are: (Solution, 2019)

* Since Model deals with the logical implementation, View deals with the interface (UI) and controller handles user interaction MVC helps to categorize the responsibility of each fields.
* Modification: If we are requiring to make any sort of modification in the code then it will be easier as code are well managed. If modification is done in one class then it will not affect other classes since functionality are divided into individual classes.
* High Cohesion and Low coupling

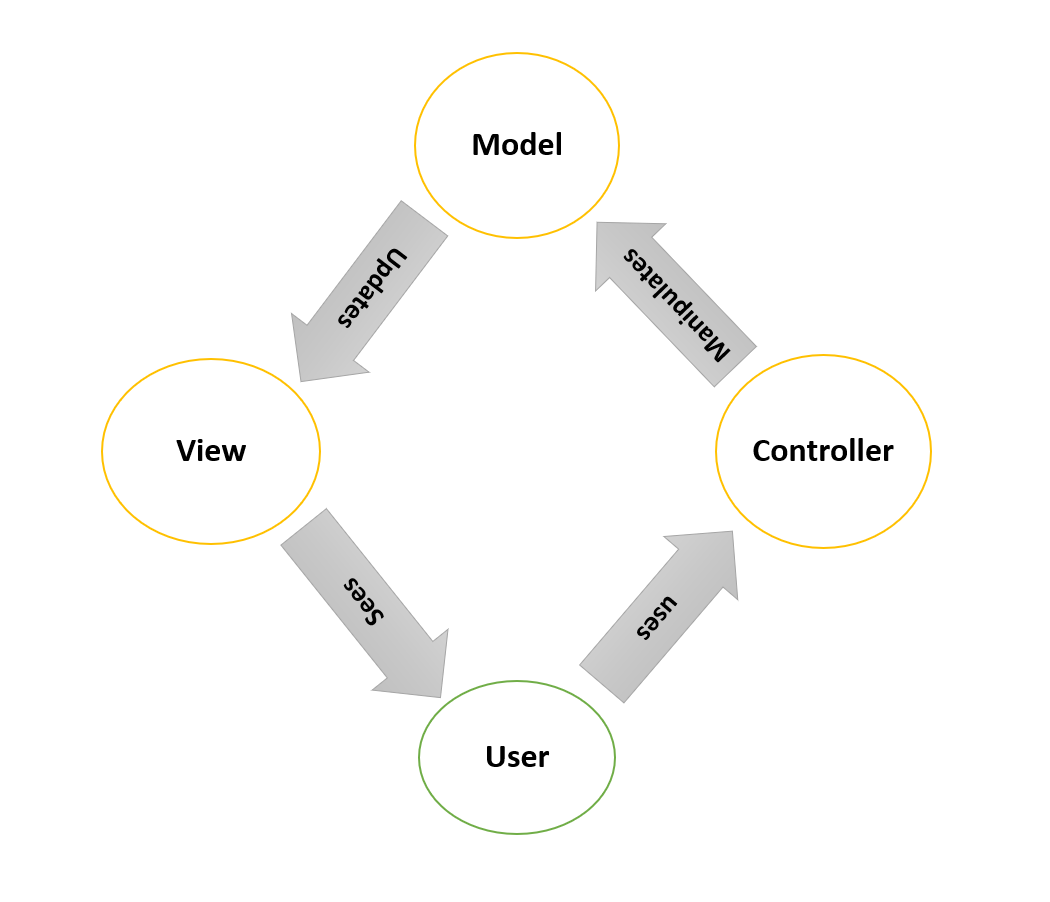


Figure 2 MVC Architecture

## 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

# Work Breakdown Structure (WBS) / Scheduling

## Work Breakdown Structure

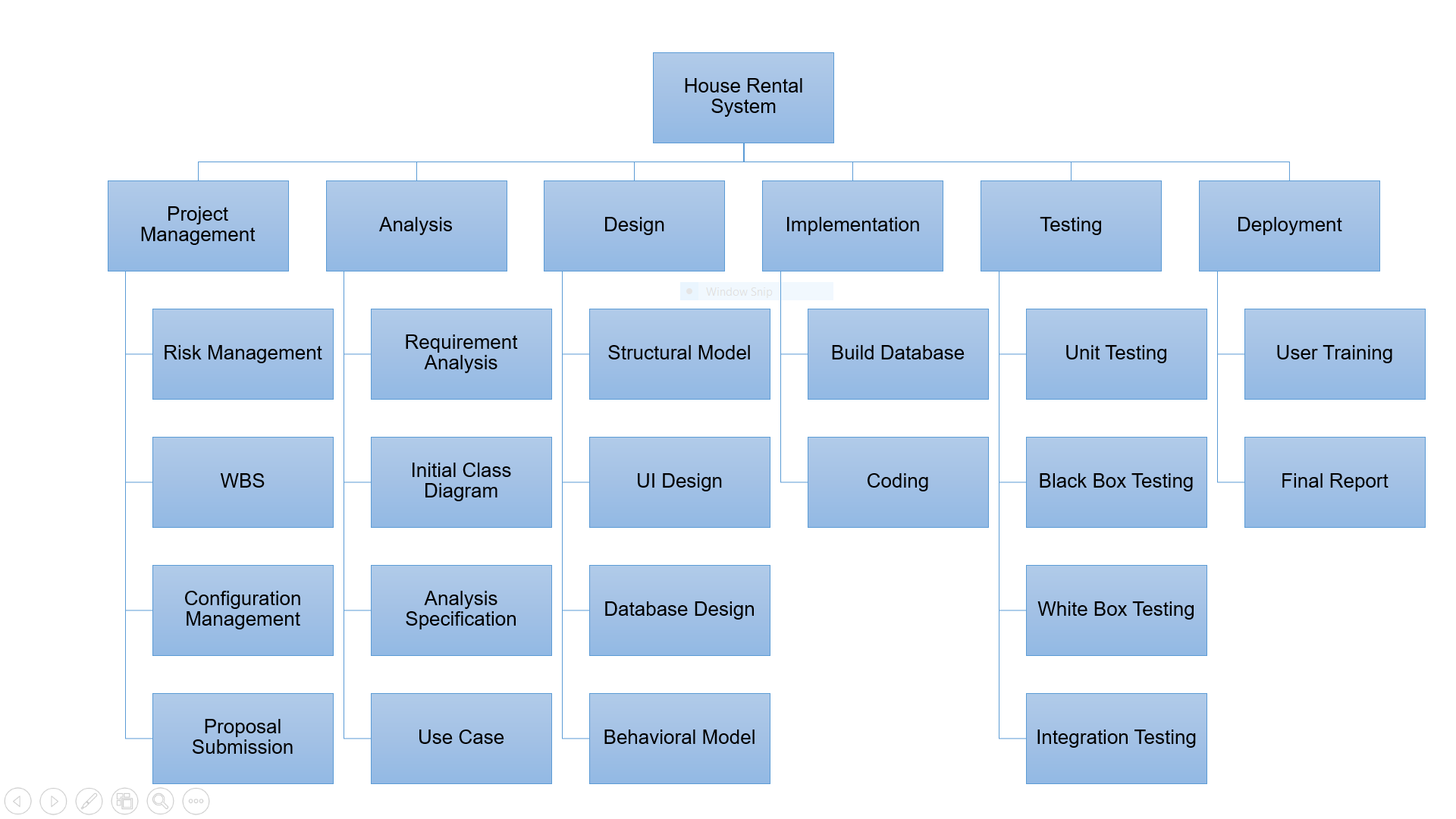


Figure 4 Work Breakdown Structure (WBS)

**Work Breakdown Structure (WBS)**

Process of dividing complex project to simpler and manageable form is called Work Breakdown structure. Larger tasks that is hard for us to understand are broken down into small chunks which is more manageable and can be understand easily. By dividing large project into smaller chunk, we can also keep track of our work and allocate the time as per the importance of the tasks. This will definitely help to manage time and remove frustration which may cause due to project. (Target, 2019)

## Milestones

|  |  |
| --- | --- |
| **Milestones** | **Date** |
| **Project Management**  Risk Management  Work Breakdown Structure  Configuration Management  Proposal Submission | **12/21/2018 -1/1/2019**  12/21/2018 - 12/23/2018  12/24/2018 - 12/26/2018  12/27/2018 - 12/30/2018  12/31/2018 - 1/1/2019 |
| **Analysis**  Requirement Analysis  Use Case  Architecture (Initial Class Diagram)  Analysis Specification | **1/2/2019 - 1/25/2019**  1/2/2019 - 1/8/2019  1/9/2019 - 1/13/2019  1/14/2019 - 1/18/2019  1/19/2019 - 1/25/2019 |
| **Design**  Structural Diagram  Behavioral Diagram  UI Design  Database Design (ER, Data Dictionary) | **1/26/2019 - 2/24/2019**  1/26/2019 - 2/4/2019  2/5/2019 - 2/14/2019  2/15/2019 - 2/19/2019  2/20/2019 - 2/24/2019 |
| **Implementation**  Building Database  Coding | **2/25/2019 - 3/28/2019**  2/25/2019 - 3/1/2019  3/2/2019 - 3/28/2019 |
| **Testing**  Unit Testing  Integration Testing  Blackbox Testing  Whitebox Testing | **3/29/2019 - 4/8/2019**  3/29/2019 - 3/31/2019  4/1/2019 - 4/3/2019  4/4/2019 - 4/6/2019  4/7/2019 - 4/8/2019 |
| **Deployment**  User Training  Final Report | 4/9/2019 - 4/18/2019  4/9/2019 - 4/149\*-/2019  4/15/2019 - 4/18/2019 |

**Description of Milestones:**

* **Project Management (12 days)**
  + - Risk Management (3 days)
    - Work Breakdown Structure (3 days)
    - Configuration Management (4 days)
    - Proposal Submission (2 days)
* **Analysis (24 days)**
  + - Requirement Analysis (7 days)
    - Use Case (5 days)
    - Architecture (Initial Class Diagram) (5 days)
    - Analysis Specification (7 days)
* **Design (30 days)**
  + - Structural Diagram (10 days)
    - Behavioral Diagram (10 days)
    - UI Design (5 days)
    - Database Design (ER, Data Dictionary) (5 days)
* **Implementation (32 days)**
  + - Building Database (5 days)
    - Coding (27 days)
* **Testing (11 days)**
  + - Unit Testing (3 days)
    - Integration Testing (3 days)
    - Blackbox Testing (3 days)
    - Whitebox Testing (2 days)
* **Deployment (10 days)**
  + - User Training (6 days)
    - Final Report (4 days)

## Scheduling / Gantt Chart

Scheduling is a process of arranging the daily tasks that is to be performed. I computing we schedule to keep track of our work and move forward. To see the complete plan of the project (When is it going to start and end) scheduling is done. There are various means of scheduling the project one of them is Gantt chart. I have addressed milestone using Gantt chart shown below.

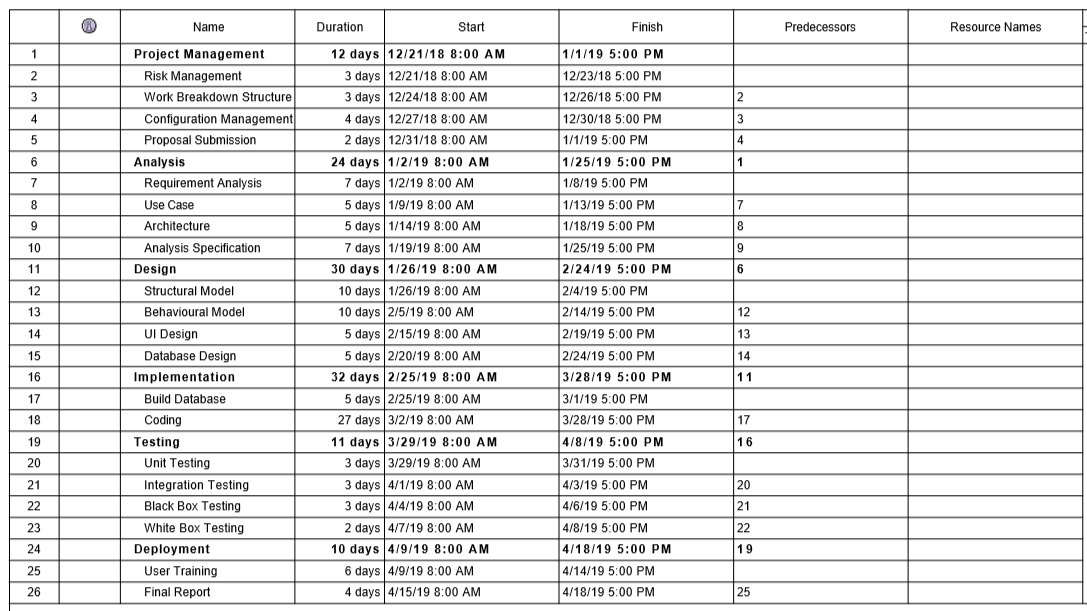


Figure 5 Date and time assigned for tasks

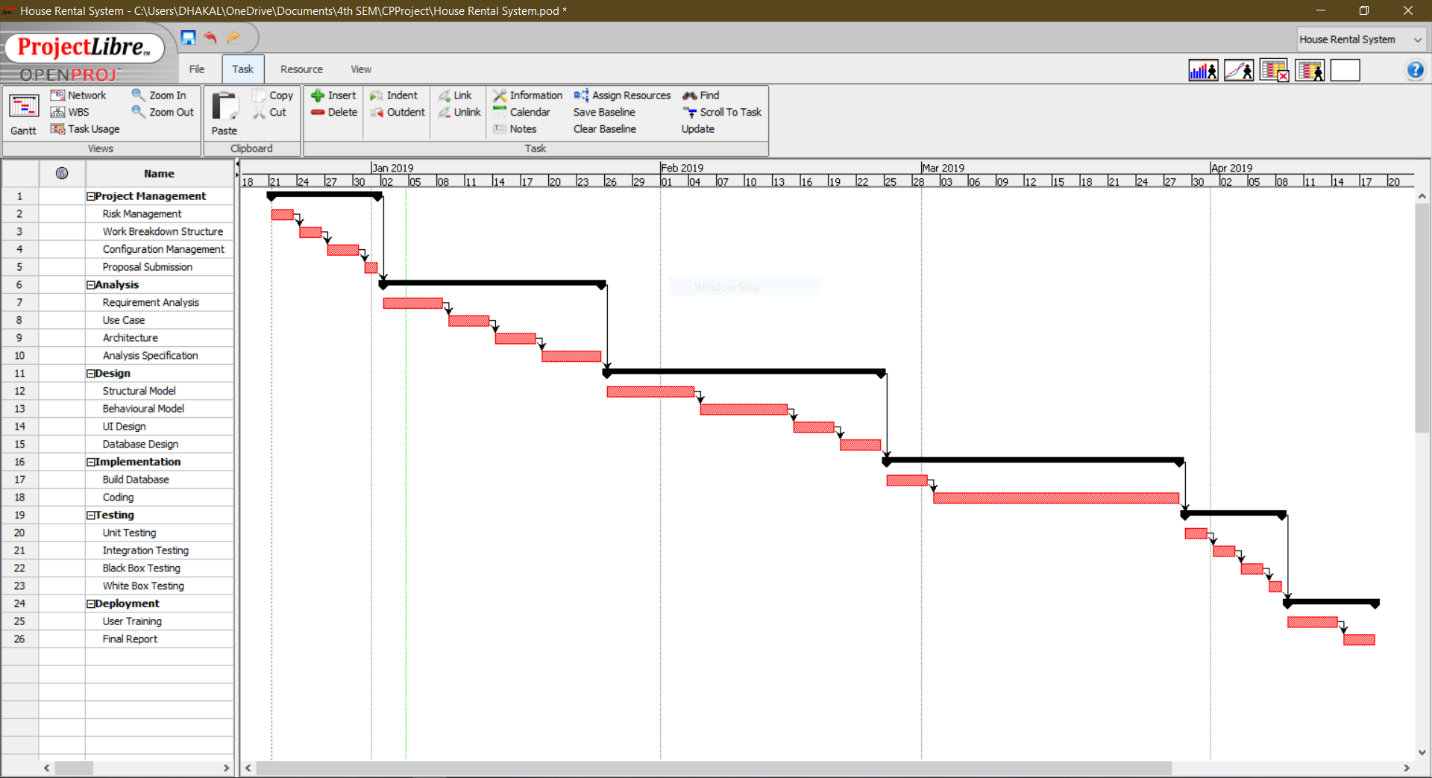


Figure 6 Gantt Chart

# 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.

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.

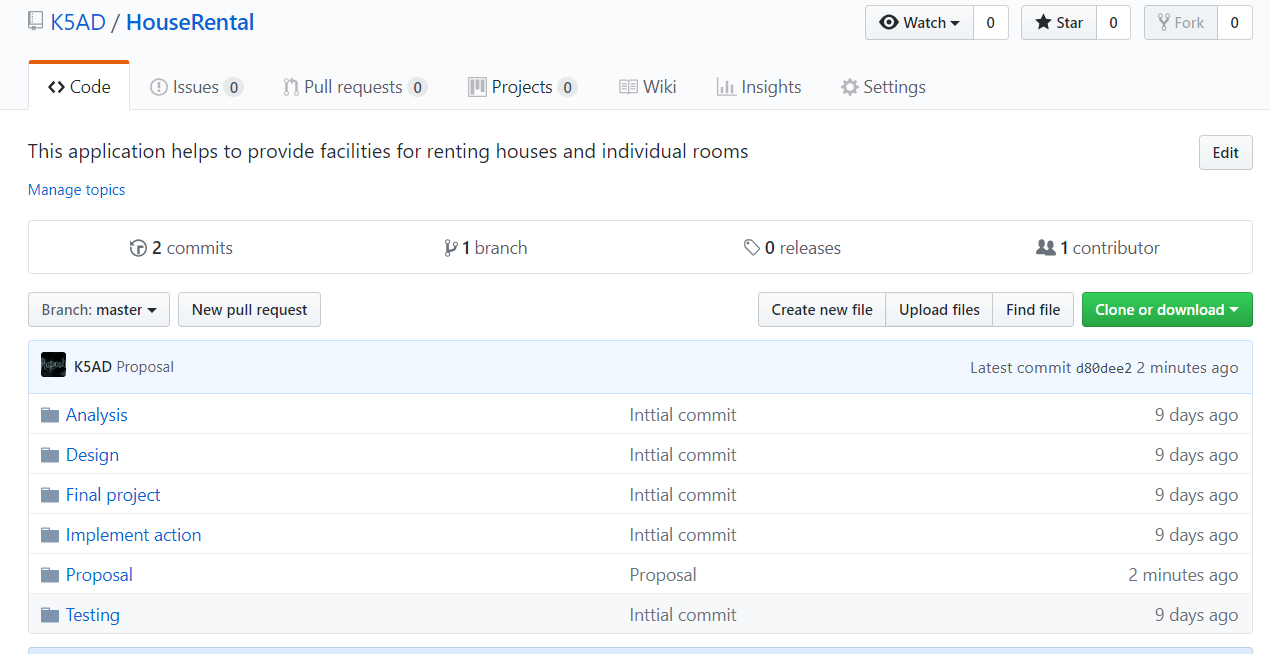


Figure 7 Git hub Repository

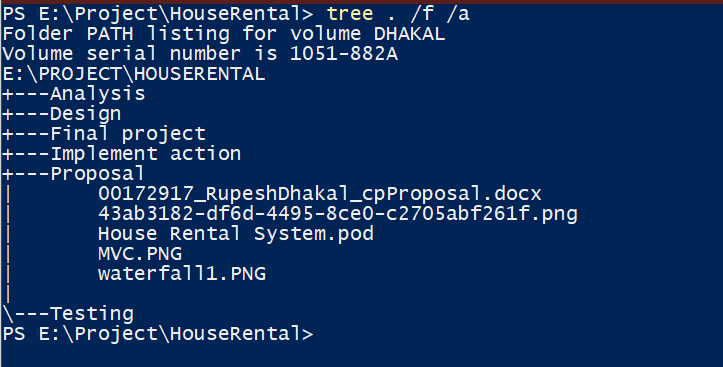


Figure 8 Folder path listin

# 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.

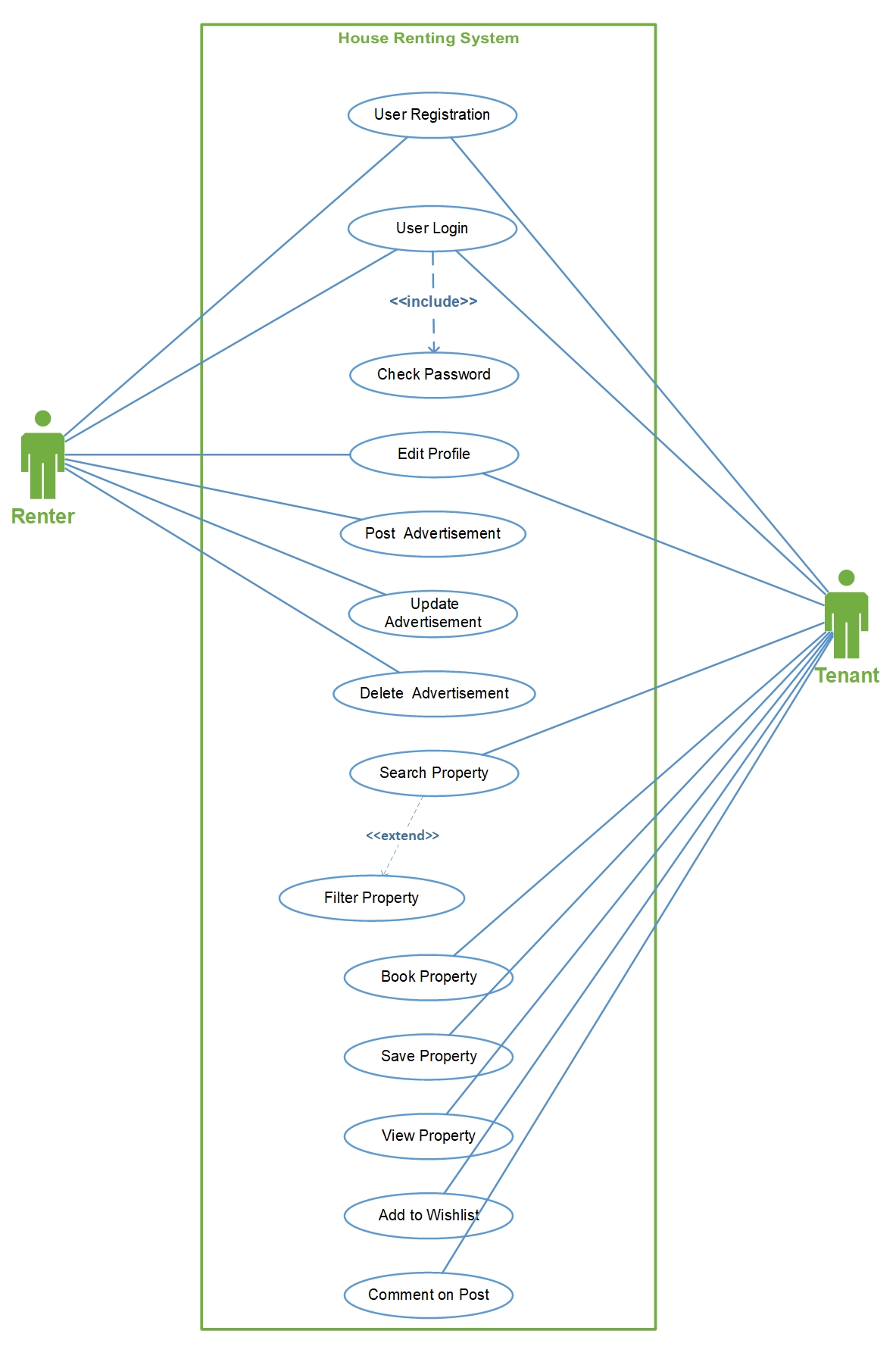


Figure 1 HRS\_Use\_Case\_Diagram

By creating use case diagram in my project, it has made clearer about actors that are involved in this project. The main advantage is that it has helped to identified the roles of every actors. Basically, it shows the functionality of the system divided to different people.

# Requirements

## Functional Requirement

Functional requirement specifies the behaviors or function. Any requirements that specifies something the system should do is functional requirements.

Typical example of functional requirement is:

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

## Non-functional Requirement

Non-Functional requirement specifies the criteria that judge the operation of the oystercatcher that the specific behaviors. It specifies how the system should behave.

Typical example of non-functional requirement is:

* Performance
* Availability
* Scalability
* Recoverability etc.

# Prioritization

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. |

# Architecture

## Initial class diagram

It is not a final class diagram but it helps to provide jist 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.

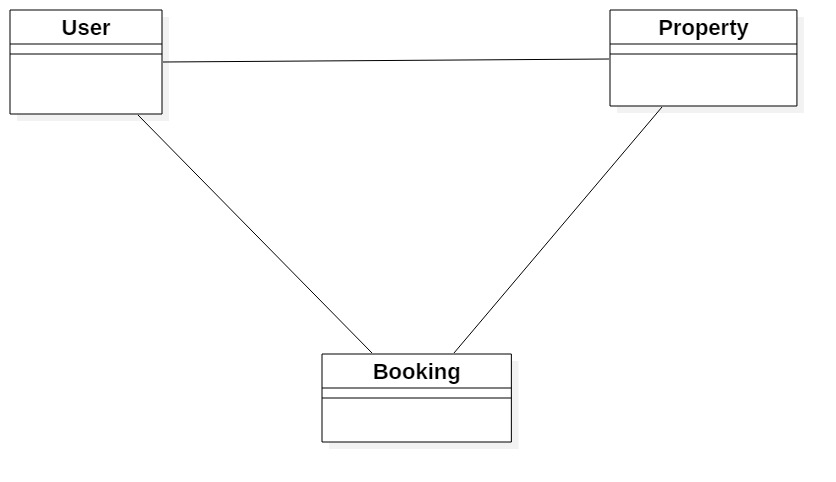


Figure 2 Initial class diagram

## System architecture (ER-Diagram)

Entity relationship diagram (ERD) shows the relation of the entities. It is created by normalizing the existing table. It is a structural diagram used in database design. It contains different notation which helps to identify the relation. I have used crows-feet notation in my ER-Diagram.

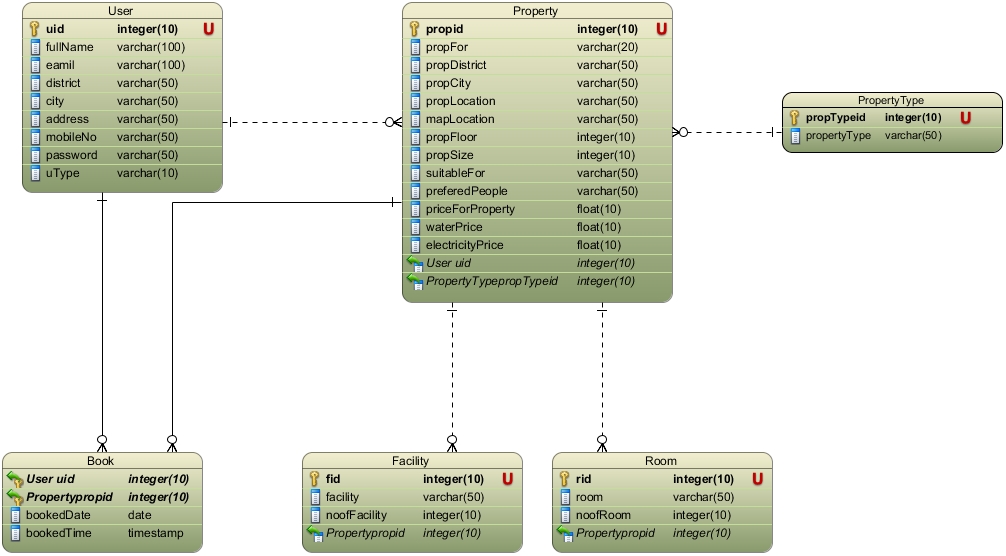


Figure 3 ER-Diagram

It have made clear about the relationship of the entity how they are related

## Class diagram showing MVC architecture

**Class Diagram** is a static structure diagram that describes the structure of the system by showing different classes, their attributes, their operations. Relationship among the object of the class is also shown. (Paradigm, 2019)

I have created the class diagram by showing MVC architecture of the diagram also. There are various ways of identifying classes for class diagram among the NLA is one.

**Natural Language Analysis (NLA):**

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.

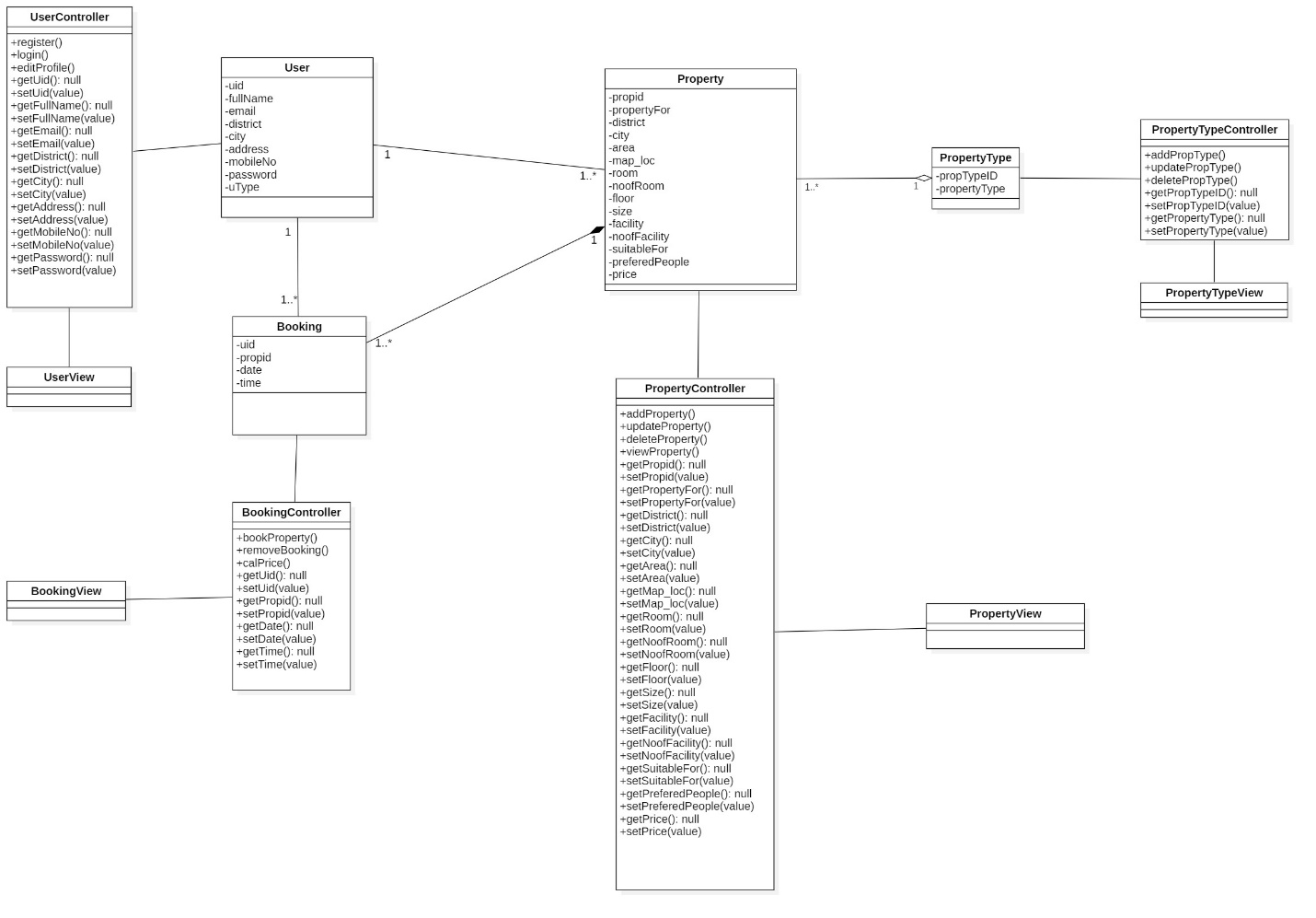


Figure 4 class diagram showing MVC architecture

Class diagram is bit different from other diagram like: sequence and activity diagram. I am clear about the static view of the system and the responsibility every class holds. Using MVC pattern to draw the class diagram has made even clear about the positioning of the attributes, methods and UI-Design.

# 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.

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:

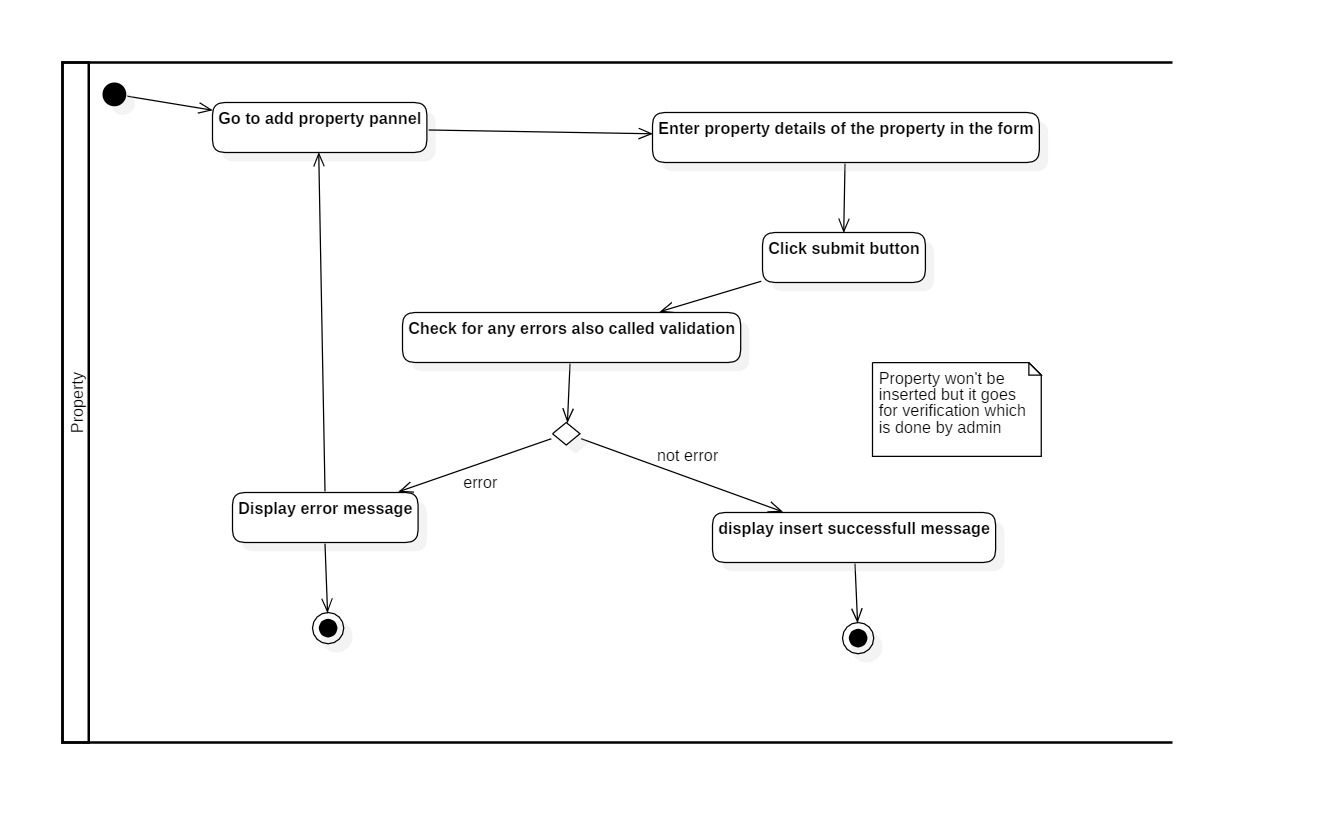


Figure 5 Activity diagram to add property

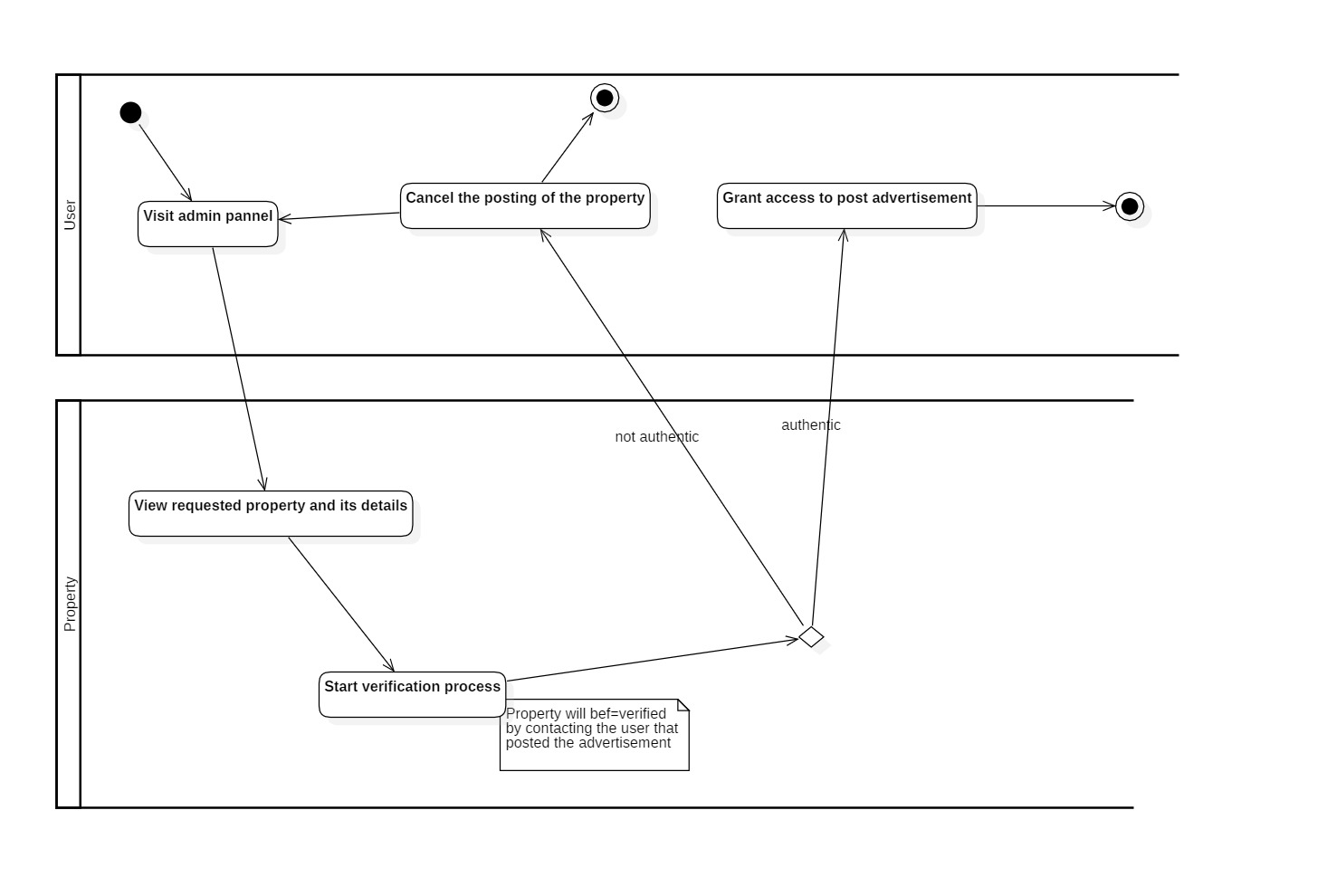


Figure 6 Activity diagram for admin verification

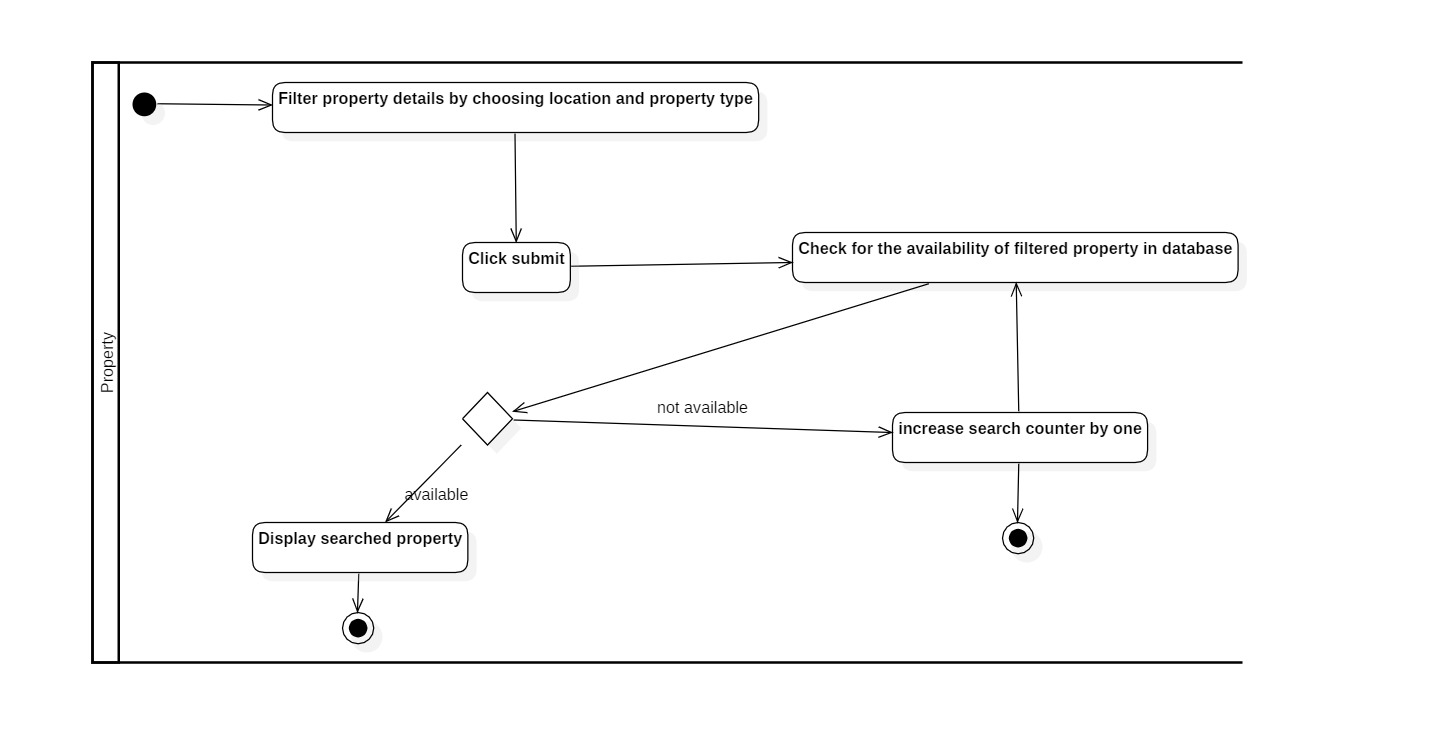


Figure 7 Activity diagram to search property

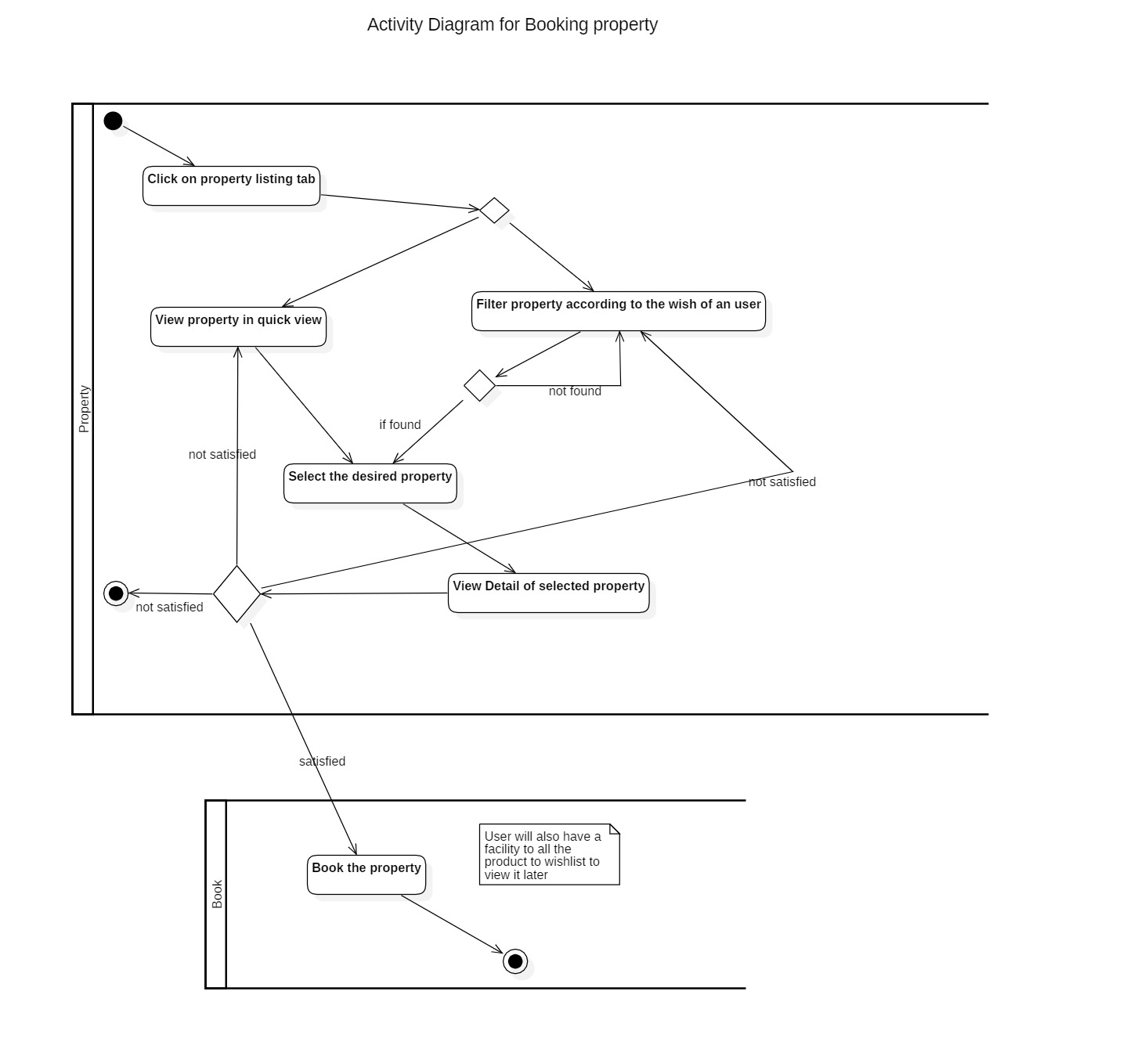


Figure 8 Activity diagram to book property

Mainly activity diagram is used for two purpose:

* **For representing algorithm formally:** Each activity that are shown in the diagram have certain functionality in the real project.
* **For creating list of high-level activity:** This point will be shown in the diagram itself

Particularly by making activity diagram for the project it has helped me to understand my project even better as it allows to defragment the harsh scenario of any project.

At last, all the necessary requirements are successfully identified and prioritized accordingly to know their priorities. Both functional and Non-Functional requirements was identified which helped to clarify the requirement further more. Use case diagram was created to know the responsibility of different actors involved in an application. Initial class diagram was created to know the structure of the application regarding classes.ER-Diagram helped to identified the relationship between different entities.

# Conclusion of the project

At conclusion I could successful built my project if above mention things are done properly, if the requirement of the project is clear. Methodology used for developing the project is described along with the architecture and design pattern used in the project. WBS of the project is given to minimize the complexity of the project. Milestone tracking and Scheduling for the project is done to ensure that project completes in time. Finally, Risk control management is done to identify and solve the risk that will arrive in the project.