System Design Document

Project Title: Humble Abode

- ***have a comfortable home***

1. Introduction:

The system "Humble Abode" is an efficient solution for people wanting to buy, give rent or sell their house. In this system, they are also provided with moving service, cleaning service, interior designers and lists of shops of furniture and accessories. In this detailed design document, the system design of "Humble Abode" will be described. System design is the process of designing the components, module, interfaces and data for a system to satisfy specified requirements. It is the process of creating abstract model of the system.

This report contains 8 sections, this introduction part being the first. The second section holds three context models- Environmental block diagram, Use case diagram and Activity diagram. The third section contains the System architecture style, fourth section includes Conceptual class diagram, fifth section holds Sequence diagram. The sixth section includes State Machine diagram, seventh section contains Detailed class diagram and in the eighth section, the report is concluded.

2. Context Models:

A context model defines how context data are structured and maintained. It shows the boundary between the system and its environment, showing the entities that interact with it. The context models of this system are as follows:

2.1. Environmental block Diagram

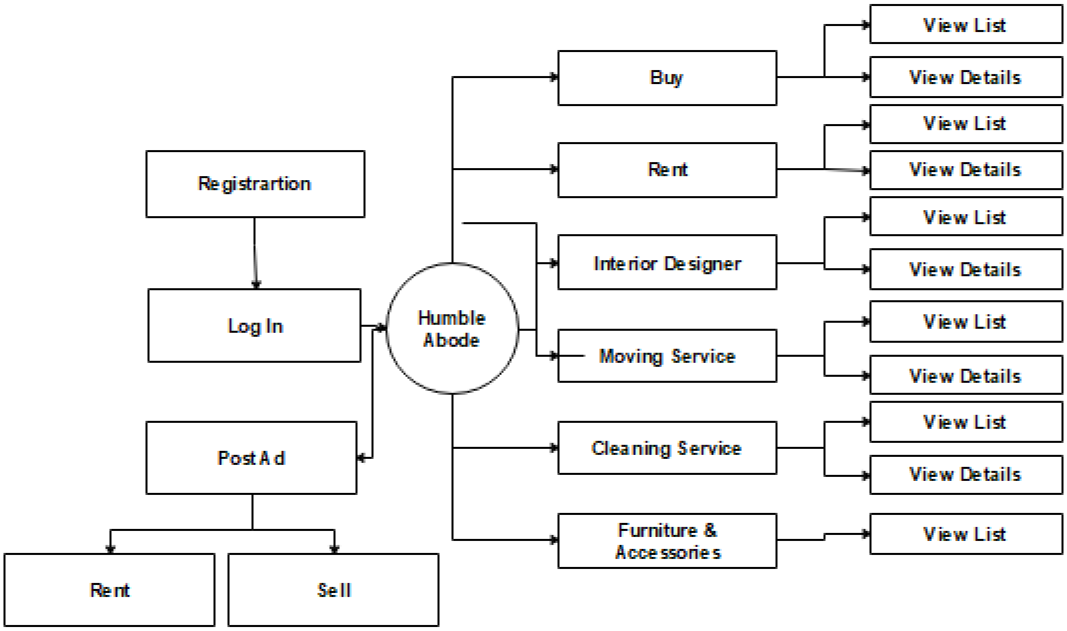

Figure 01: Environmental Block Diagram.

2.2. Use Case Diagram:

A use case diagram is used to analyze various systems. It helps to visualize the different types of roles in a system and how those roles interact with the system. The use case diagram of the system “Humble Abode” is given in figure 2:

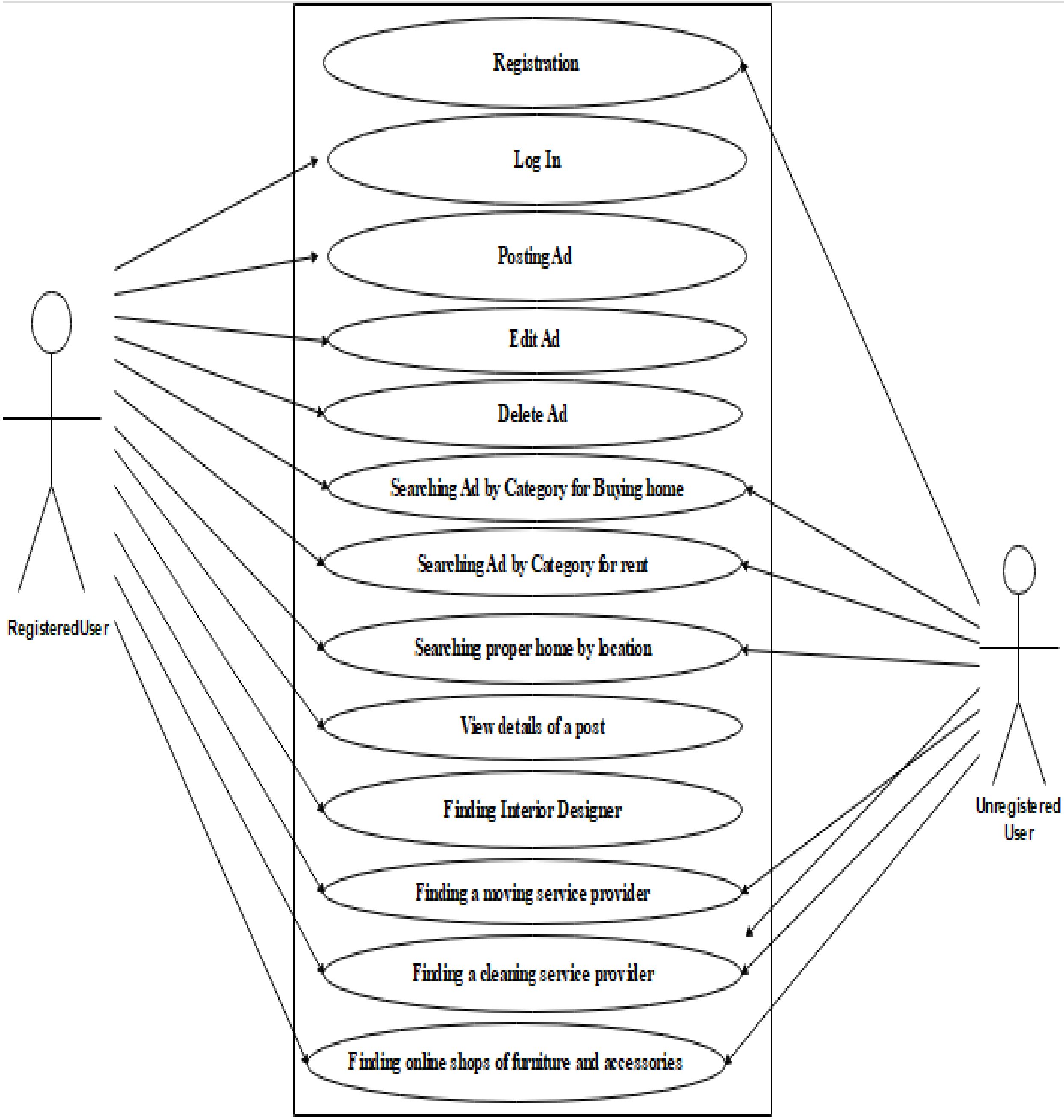
Registration
Log In
Posting Ad
Edit Ad
Delete Ad
Searching Ad by Category for Buying home
Searching Ad by Category for rent
Searching proper home by location
View details of a post
Finding Interior Designer
Finding a moving service provider
Finding a cleaning service provider
Finding online shops of furniture and accessories
RegisteredUser
Unregistered User

Figure 2: Use case diagram of ‘Humble Abode’ System

2.3. Activity Diagram:

An activity diagram is basically a flow chart that shows the flow from one activity to another. It is a visual representation of workflows in a system. The activity diagrams of each use case is given in figure 3-15:

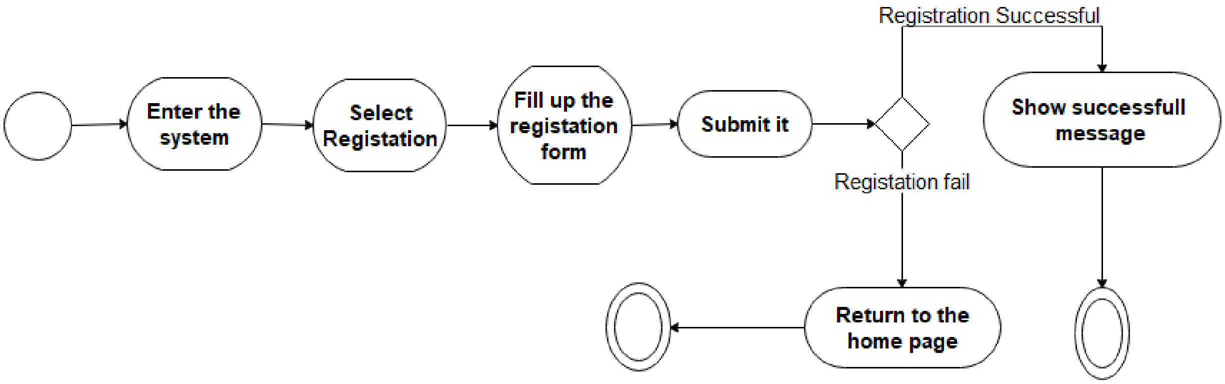

Figure3: Activity diagram of UC1(Registration)

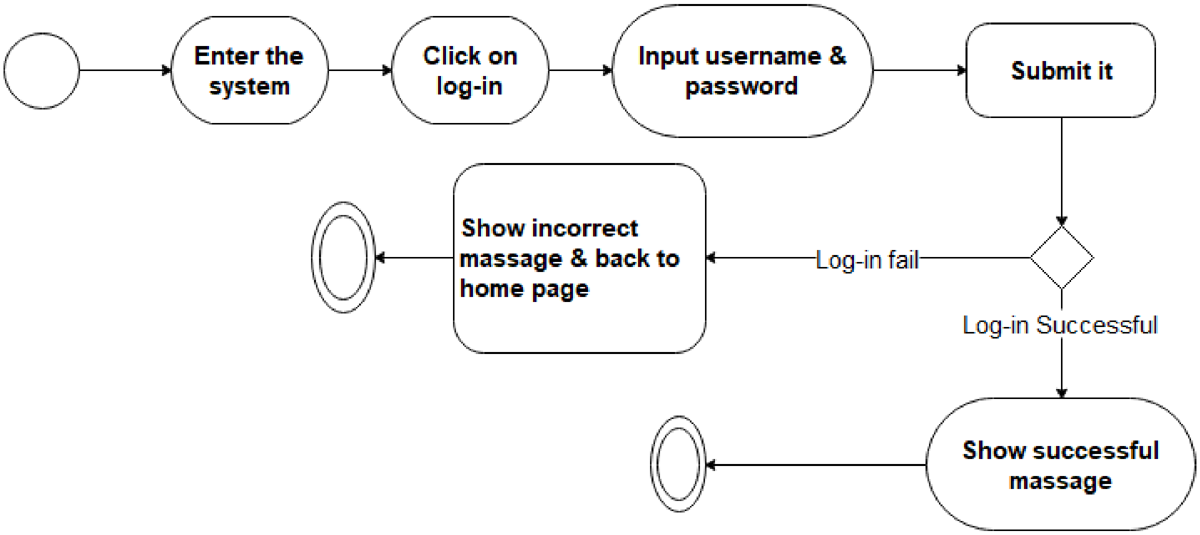

Figure4: Activity diagram of UC3(Log-in)

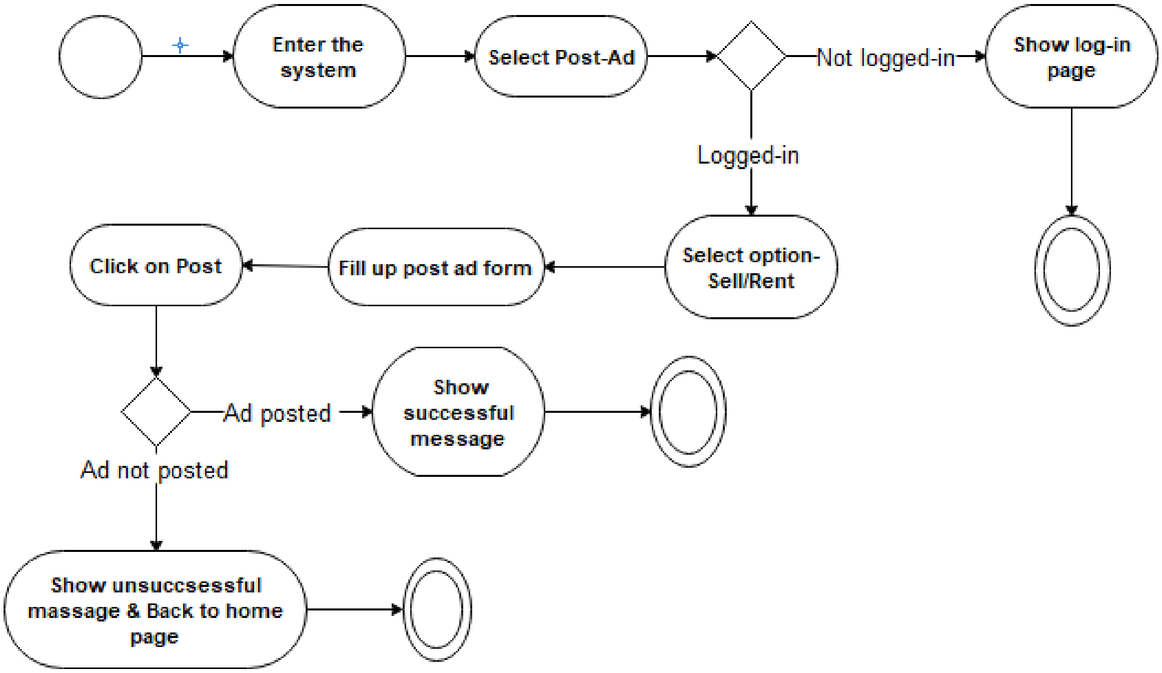

Figure5: Activity diagram of UC3(Post Ad)

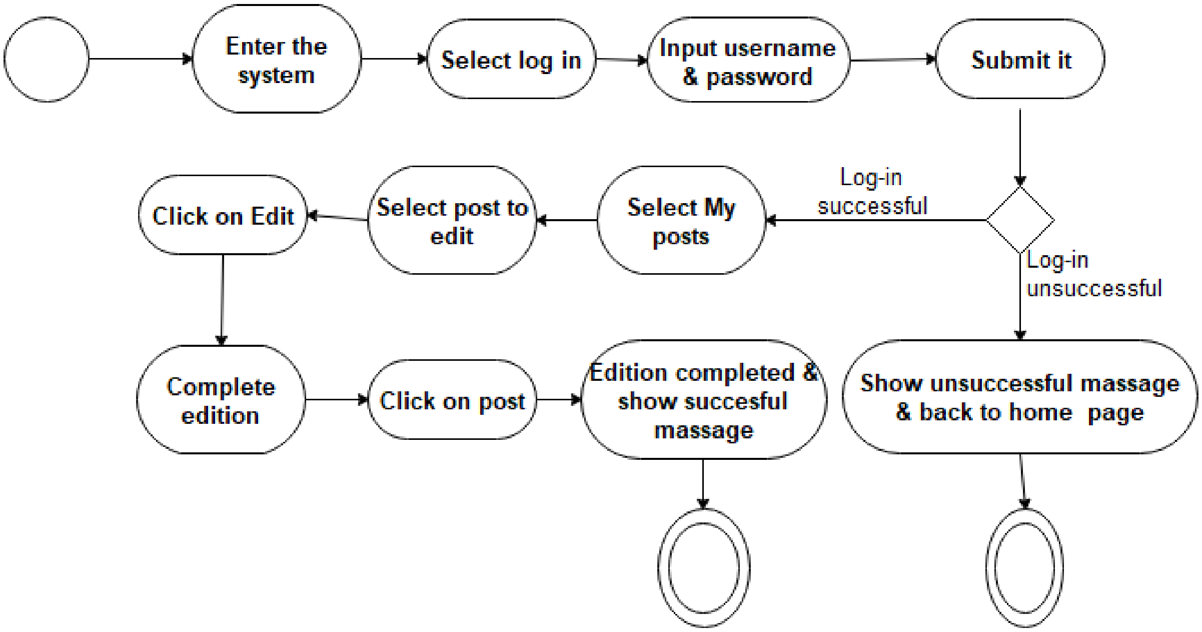

Figure6: Activity diagram of UC4(Edit Ad)

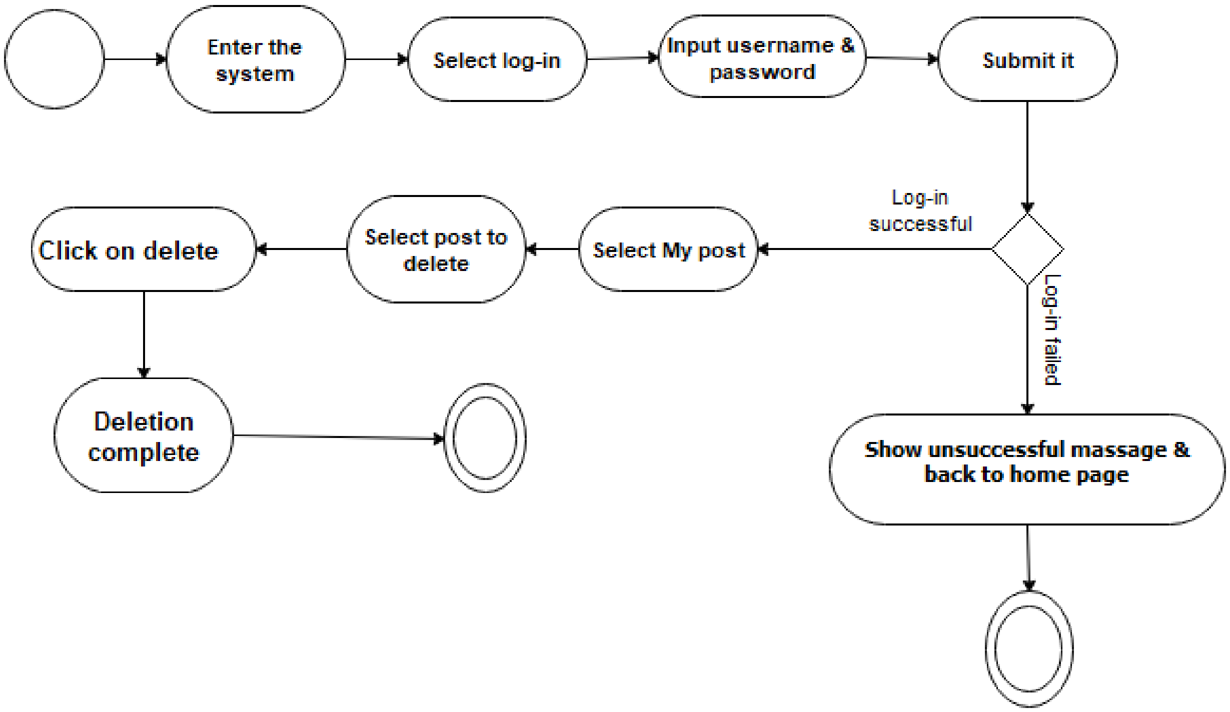

Figure7: Activity diagram of UC5(Delete Ad)

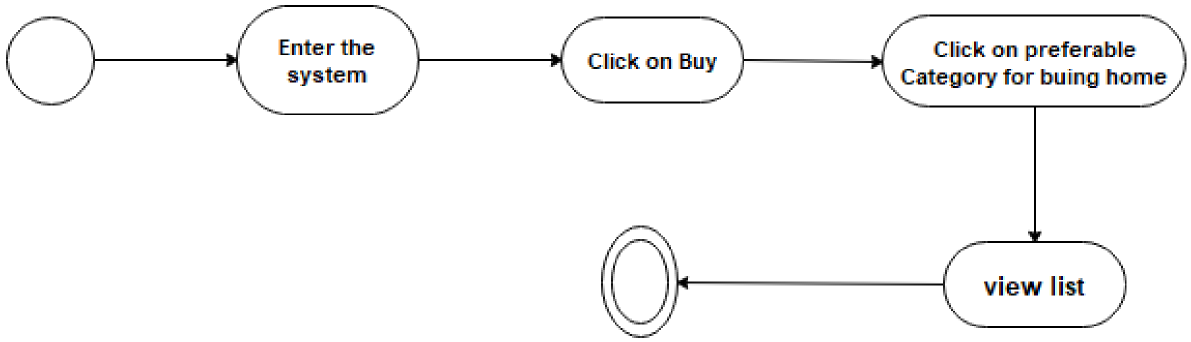

Figure 8: Activity diagram of UC6(Searching Ad by Category for Buying home)

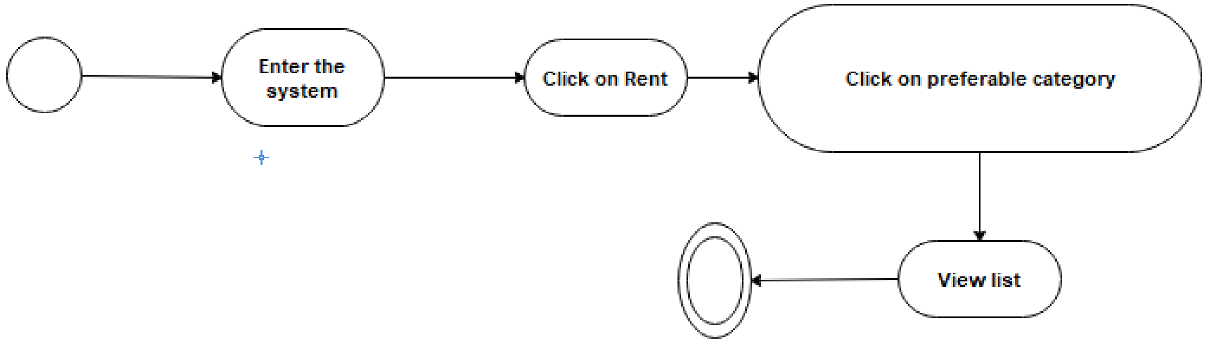

Figure 9: Activity diagram of UC7(Searching Ad by Category for Rent)

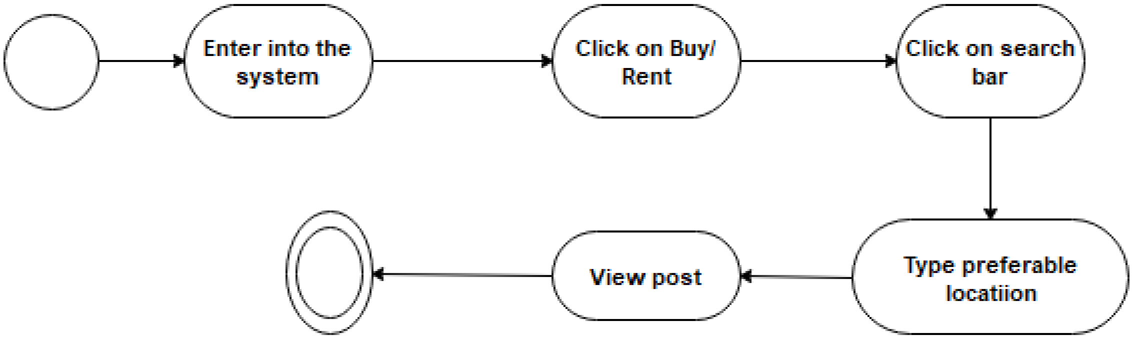

Figure 10: Activity diagram of UC8(Searching proper home by location)

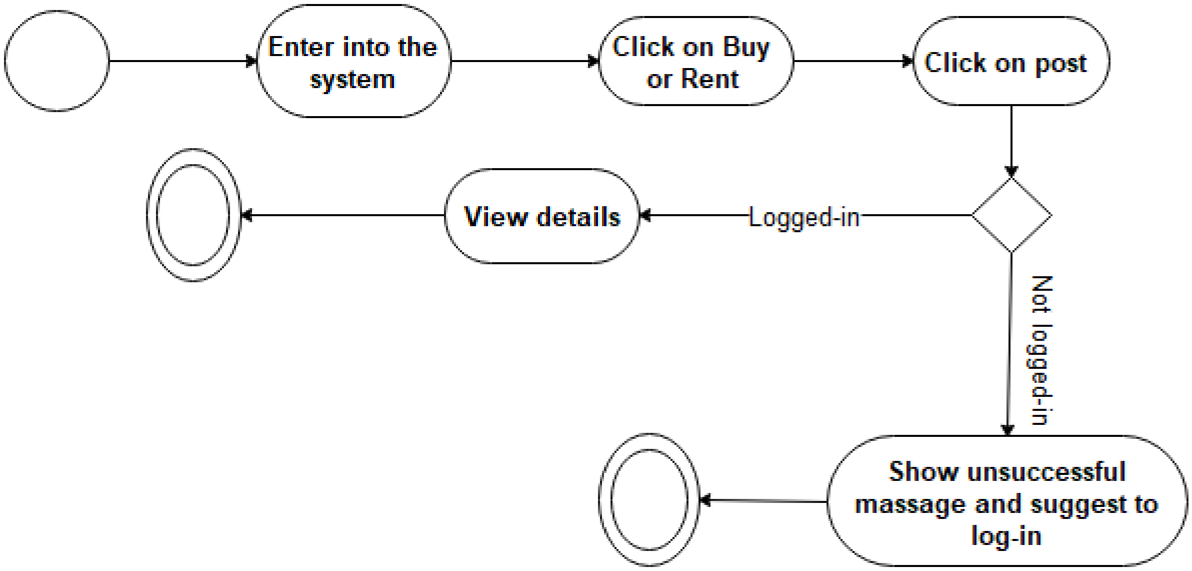

Figure 11: Activity diagram of UC9(View details of a post)

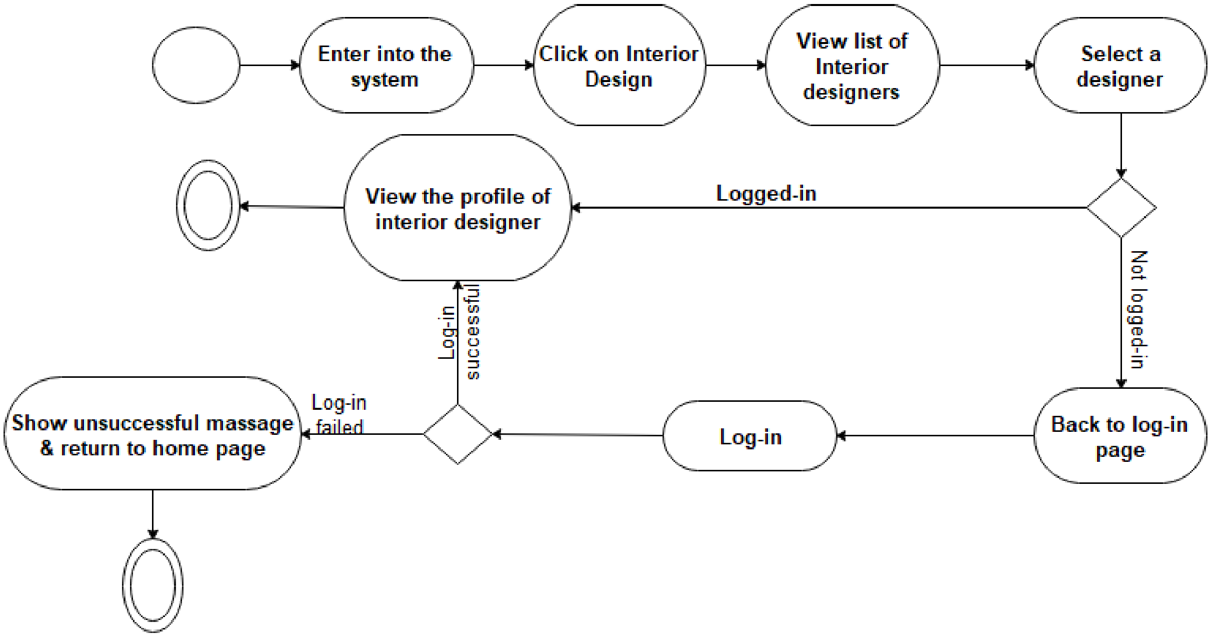

Figure 12: Activity diagram of UC10(Finding Interior Designer)

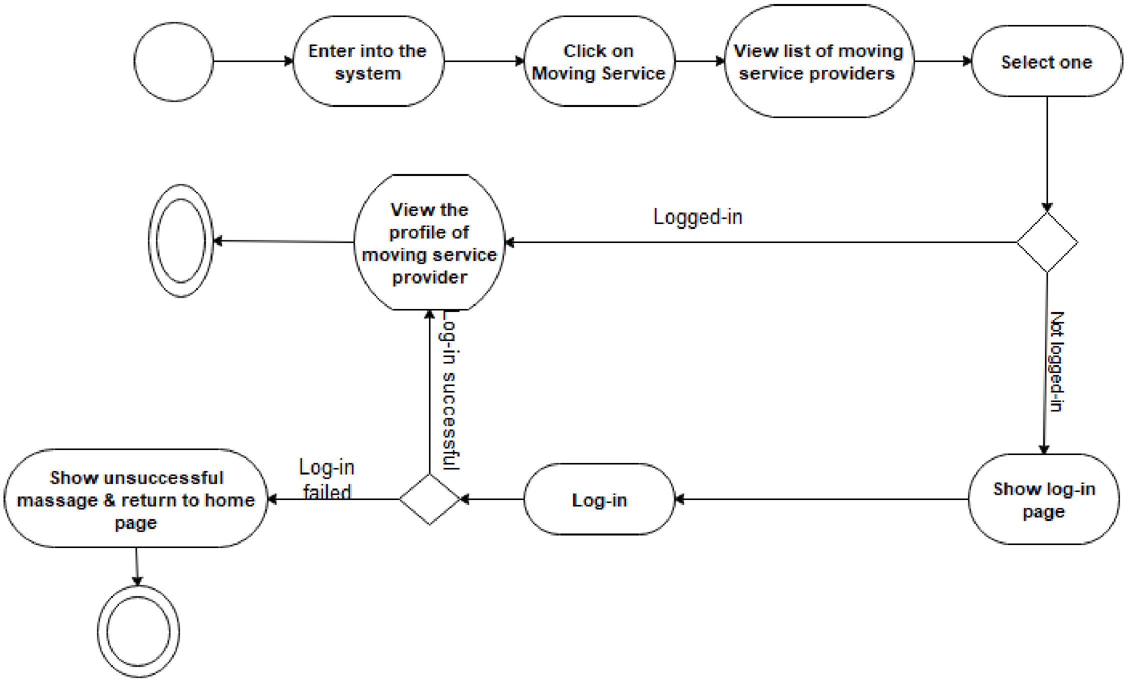

Figure 13: Activity diagram of UC11(Finding a moving service provider)

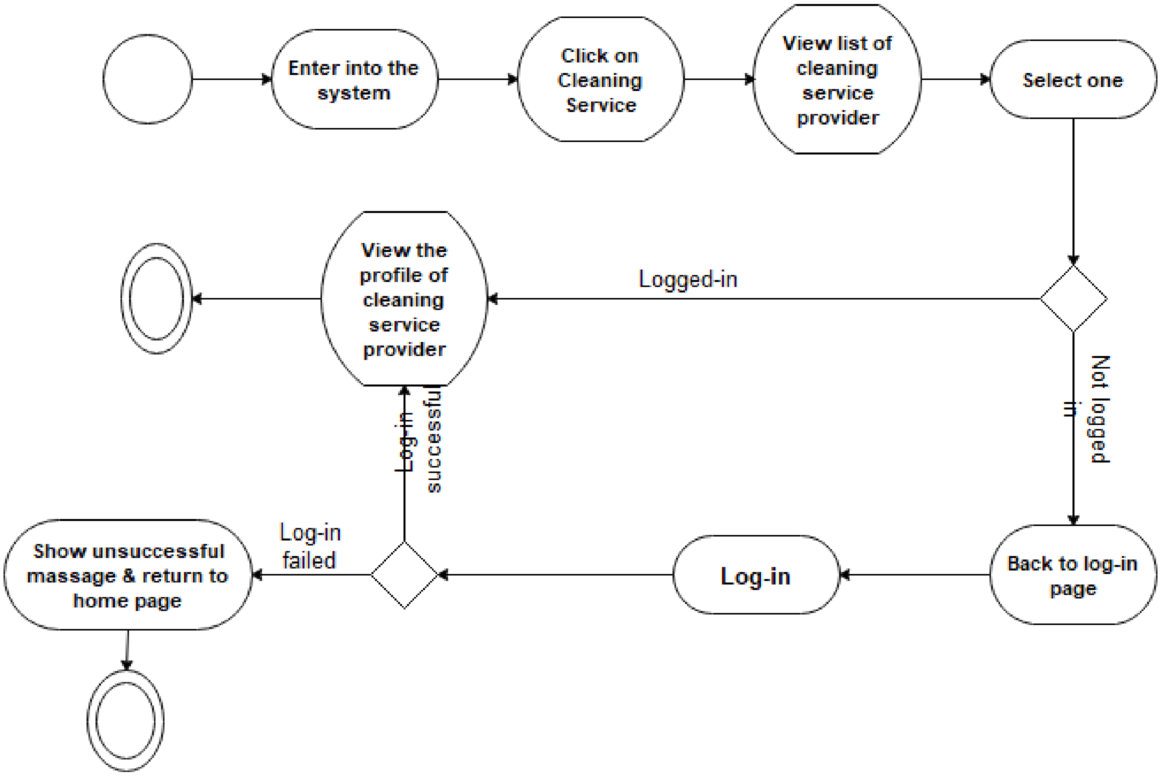
Enter into the system
Click on Cleaning Service
View list of cleaning service provider
Select one
Logged-in
View the profile of cleaning service provider
Not logged in
Log-in successful
Log-in failed
Show unsuccessful massage & return to home page
Log-in
Back to log-in page

Figure 14: Activity diagram of UC12(Finding a cleaning service provider)

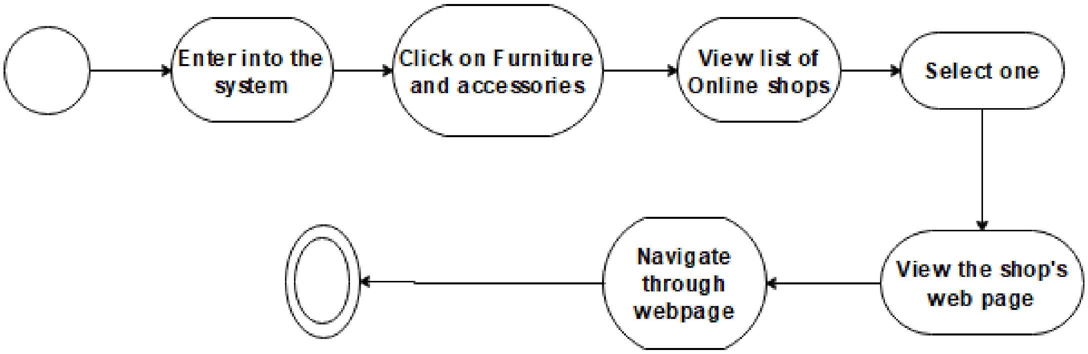

Figure 15: Activity diagram of UC13(Finding online shops of furniture & accessories)

3. System Architecture Style

An architectural pattern is a concept that solves and declines some essential cohesive elements of software architecture. It defines the overall shape and structure of software applications. It is a concept that solves and delineates some cohesive elements of software architecture. For our android project, we find that the MVC pattern is a perfect fit because internal architecture of this app closely resembles this pattern.

The Model-View-Controller shortly known as the MVC is a software architectural design for implementing the user interfaces on computers. MVC patterns separate input, processing and output of an application. This model is divided into three interconnected parts called the model, the view and the controller. All of the three above given components are built to handle some specific development aspects of any web or software application. In the MVC development, controller receives all requests for the application and then instruct the model to prepare any information required by the view. The view uses that data prepared b-y the controller to bring the final output. The basic diagram for MVC pattern is in fig 15:

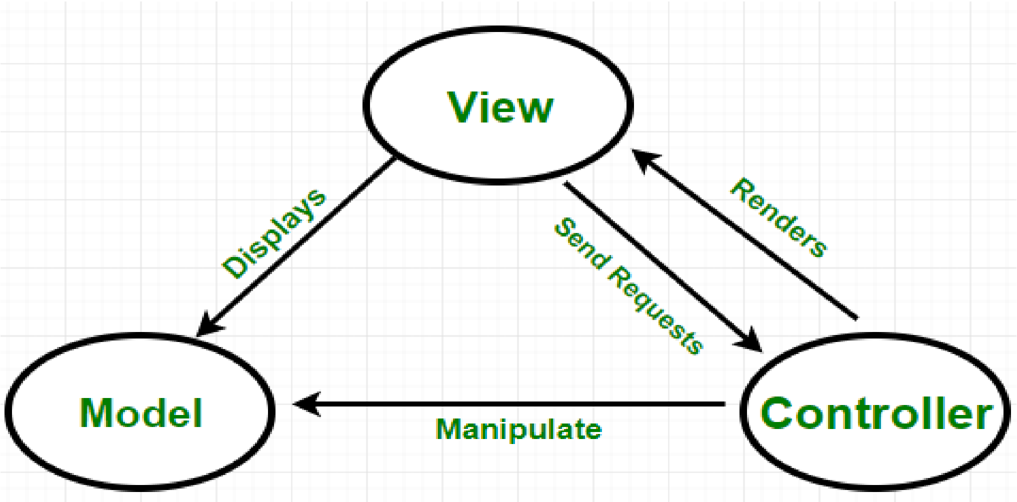

Fig 15: MVC Pattern

We have chosen MVC pattern for our application as it is simple, clean, readable and testable, and does not require any additional libraries and project architecture closely resembles to MVC pattern. According to MVC pattern three different parts of our system are:

Model: This level is very important as it represents the data to the user and defines where the applications data objects are stored. In our application, there is a database which contains system data and model classes for manipulating data.

View: A view is a visual representation of the MVC model. This level creates an interface to allow the actual output to the user. In our android application, there are activities that show the data to the user. So, activity classes like display, files are put in view component. View handles requests from the user and informs the controller.

Controller: Controller manages user interaction and passes the interaction to the view. It acts as a link between the user and the model. There are classes to interact with user which are considered as components of this level.

MVC pattern for our system is shown in fig 16:

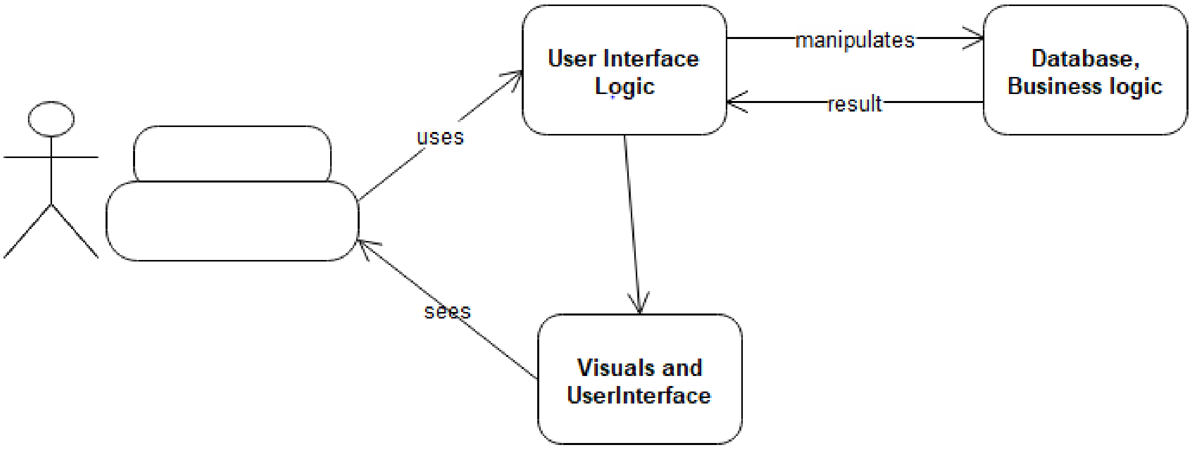

Fig 16: MVC pattern in our system

Important advantages of MVC model and causes behind choosing it:

- Faster development process.
- Ability to provide multiple view.
- Modification does not affect the entire model.
- Friendly development platform and supportive for inexperienced developers.
- MVC pattern fits for application requiring strict maintainability and test ability standards.

4. Conceptual Class Diagram:

A conceptual model captures the important concepts and relationships in some domain. Concepts are represented by classes, while relationships are represented by associations. This system’s conceptual class diagram is show in figure 16:

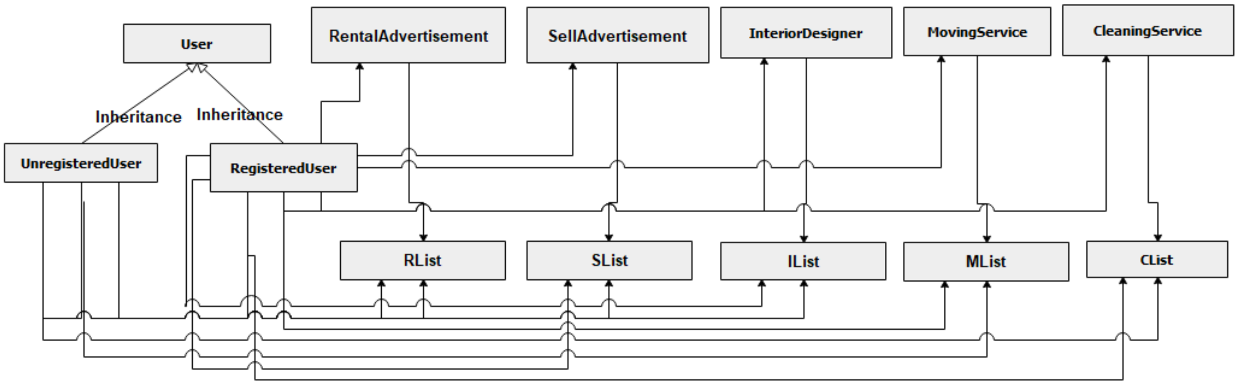

Figure 16: Conceptual Class Diagram

5. Sequence Diagram:

A sequence diagram is used to model the series of interactions that take place in a particular use case, describing the flow of messages, events and actions between objects. The sequence diagrams for each use case are presented in figure 17- 28.

UC 01: Registration(Unregistered Users)

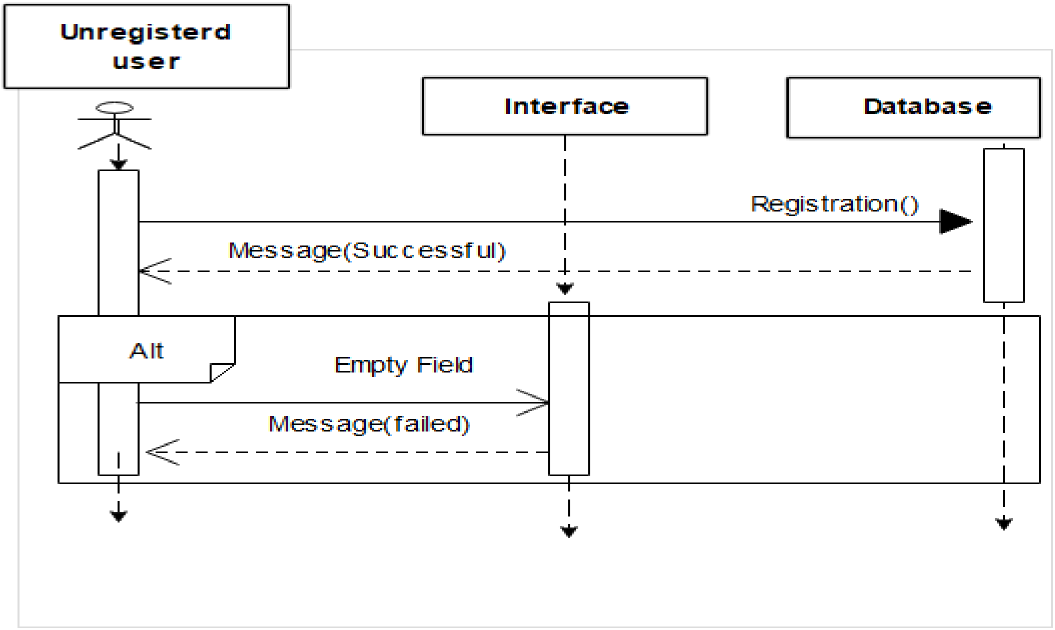

Figure17: Sequence diagram for Registration (Unregistered Users)

UC 02: Log-In(Registered Users)

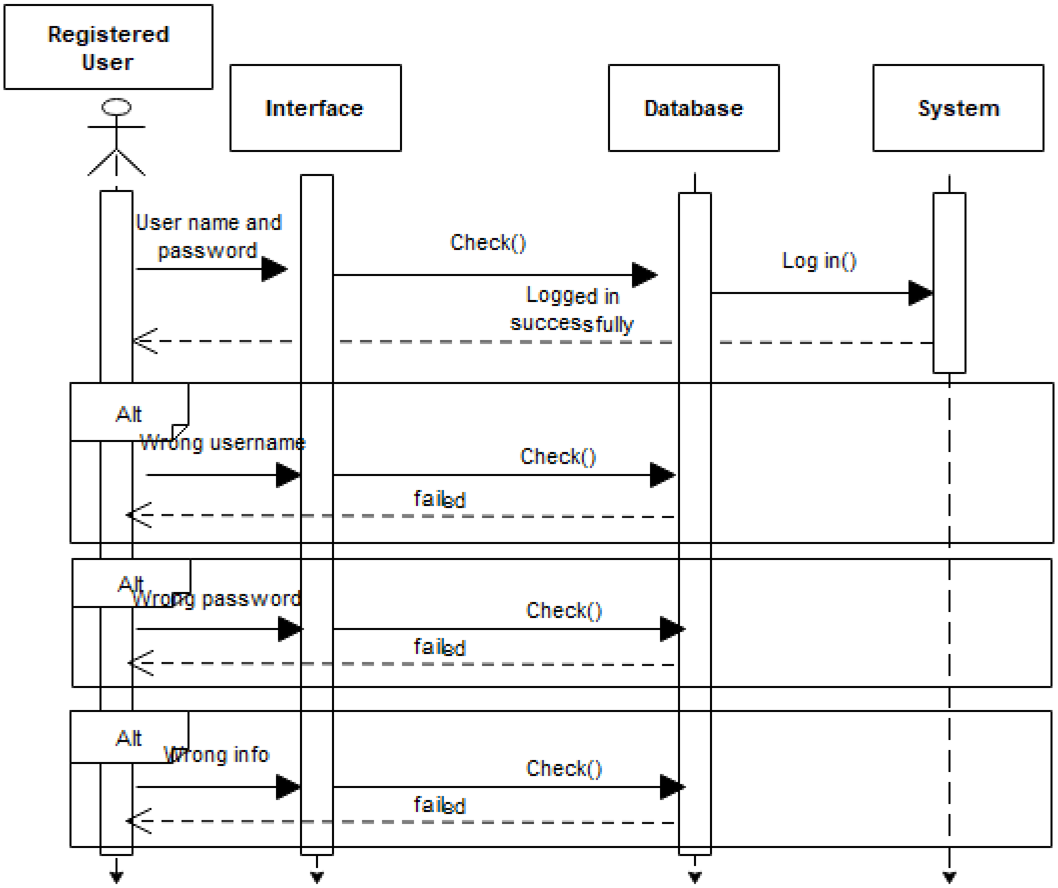

Figure18: Sequence diagram for Log-In(Registered Users)

UC 03:Post Ad(Registered Users)

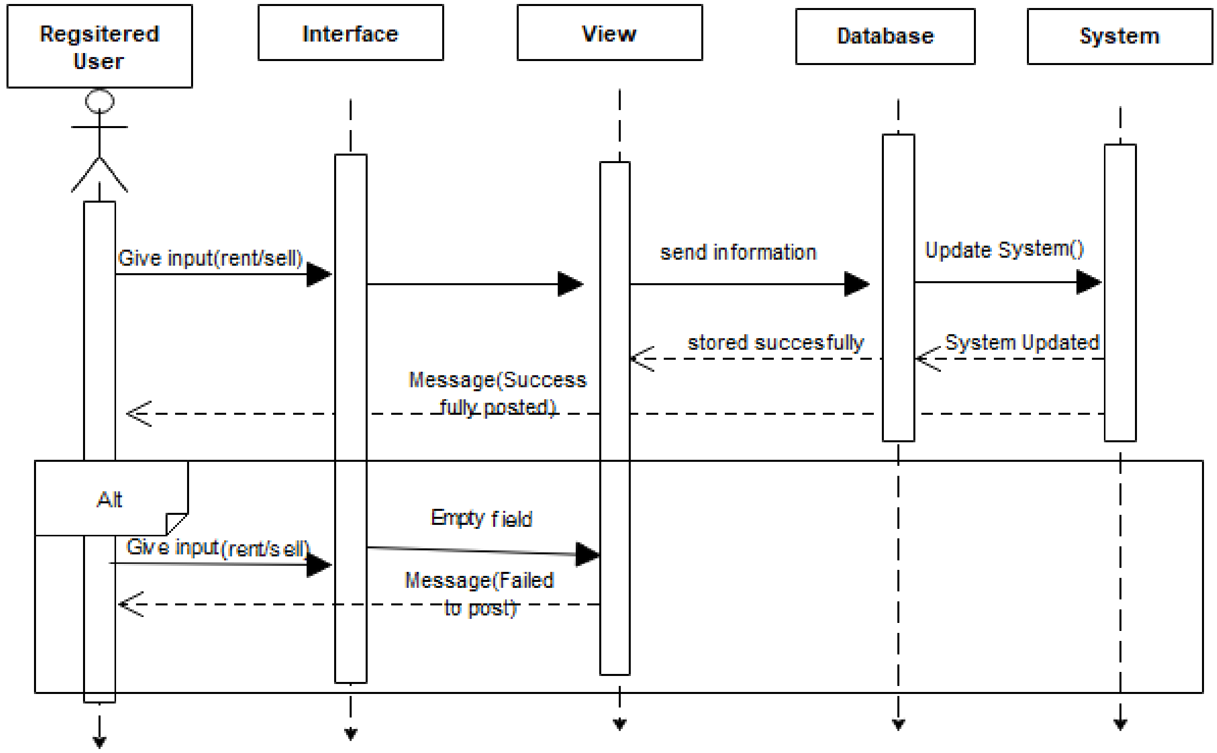

Figure18: Sequence diagram for Post Ad(Registered Users)

UC 04:Edit Ad(Registered Users)

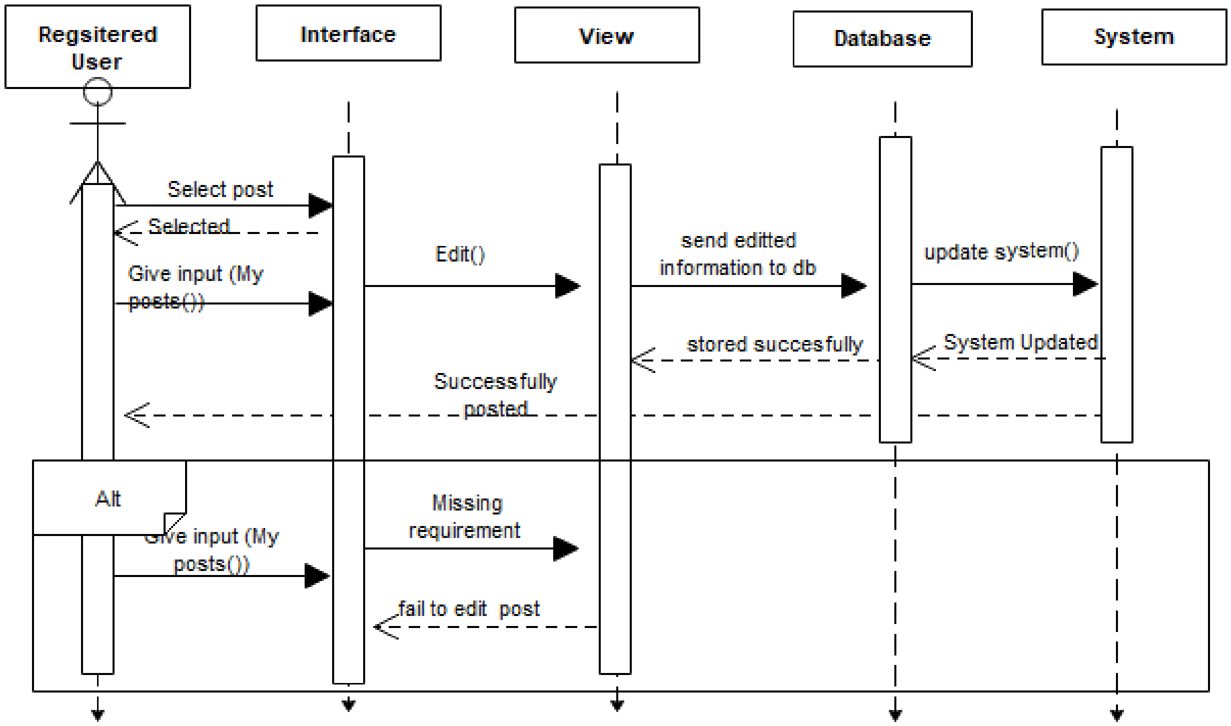

Figure19: Sequence diagram for Edit Ad(Registered Users)

UC 05.Delete Ad(Registered Users)

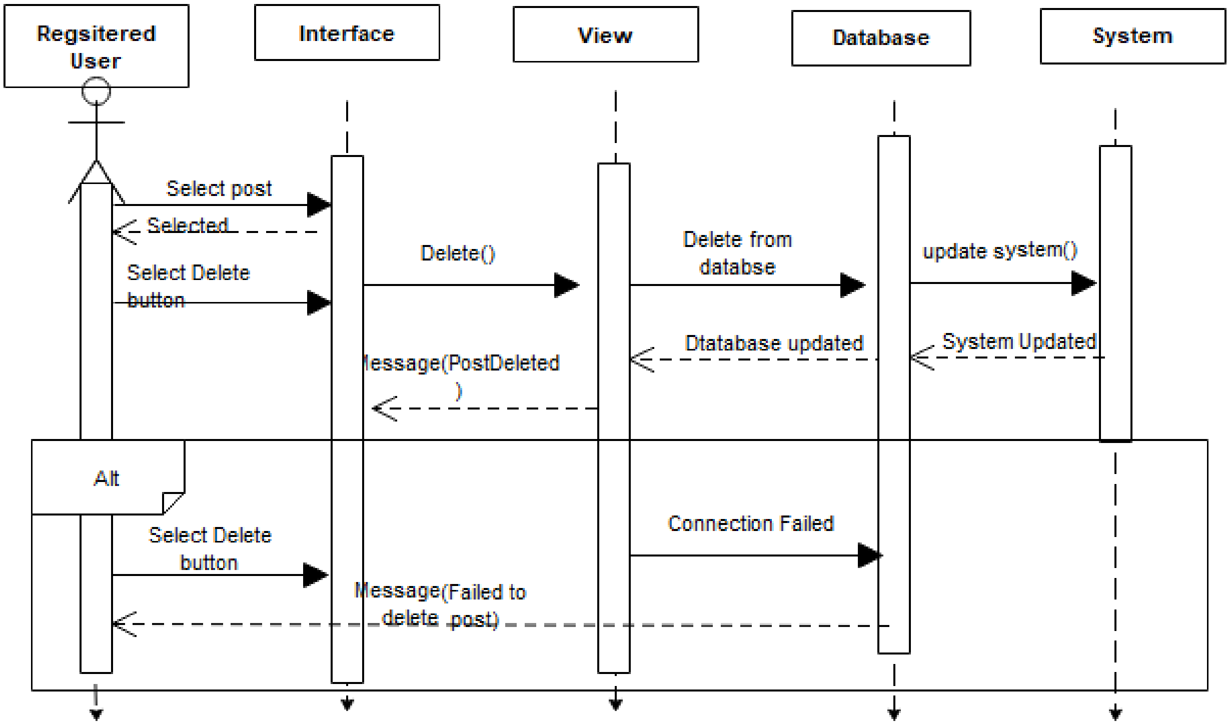

Figure 20: Sequence diagram for Delete Ad(Registered Users)

UC 06.Searching Ad by category for buying home (Registered Users/Unregistered Users)

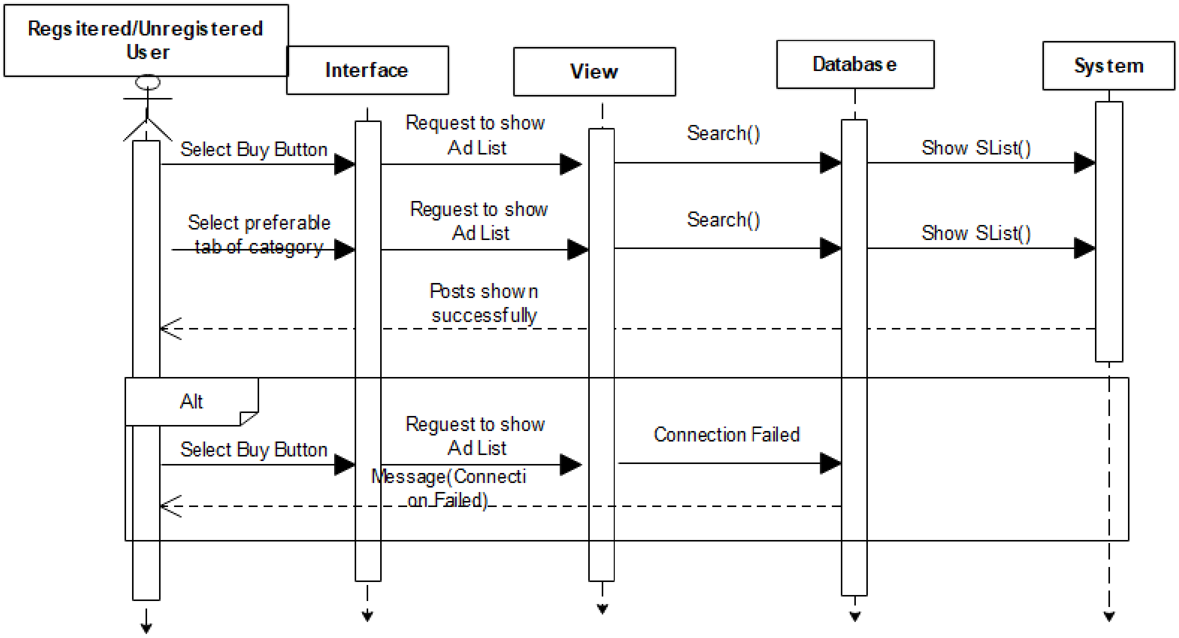

Figure21: Sequence diagram for Searching Ad by category for buying home(RegisteredUsers/Unregistered Users)

UC 07.Searching Ad by category for Renting (Registered Users/Unregistered Users)

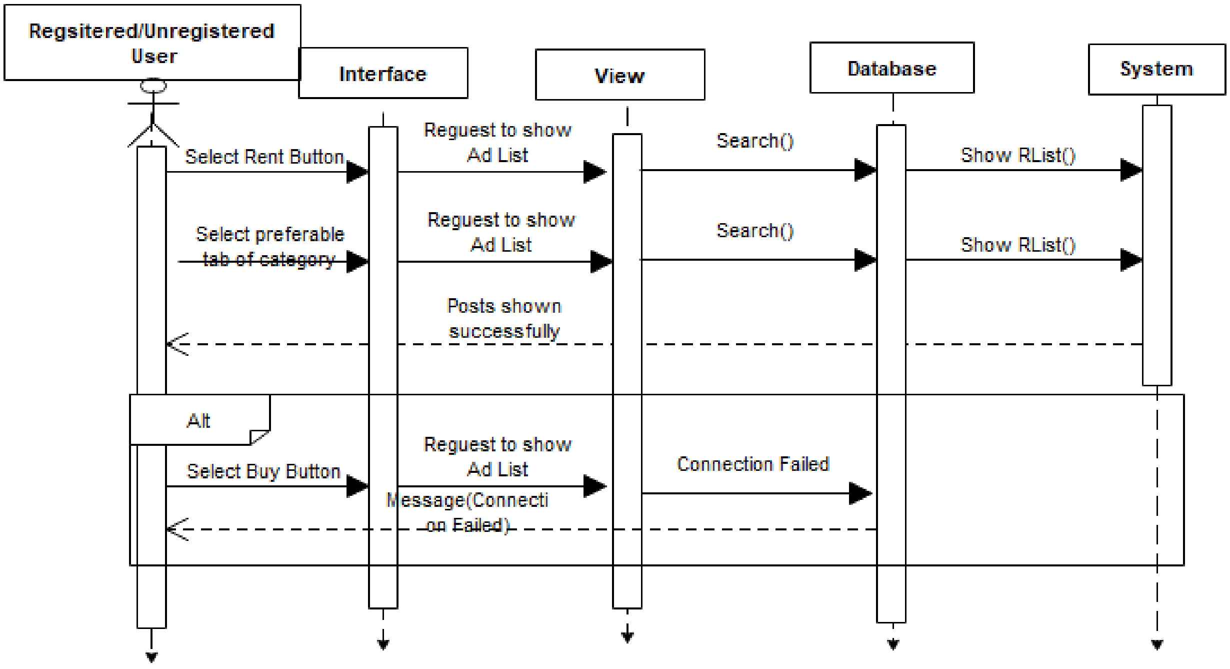

Figure22: Sequence diagram for Searching Ad by category for Renting (Registered User/Unregistered Users)

UC8: Searching Ad by Location(Registered Users/Unregistered Users)

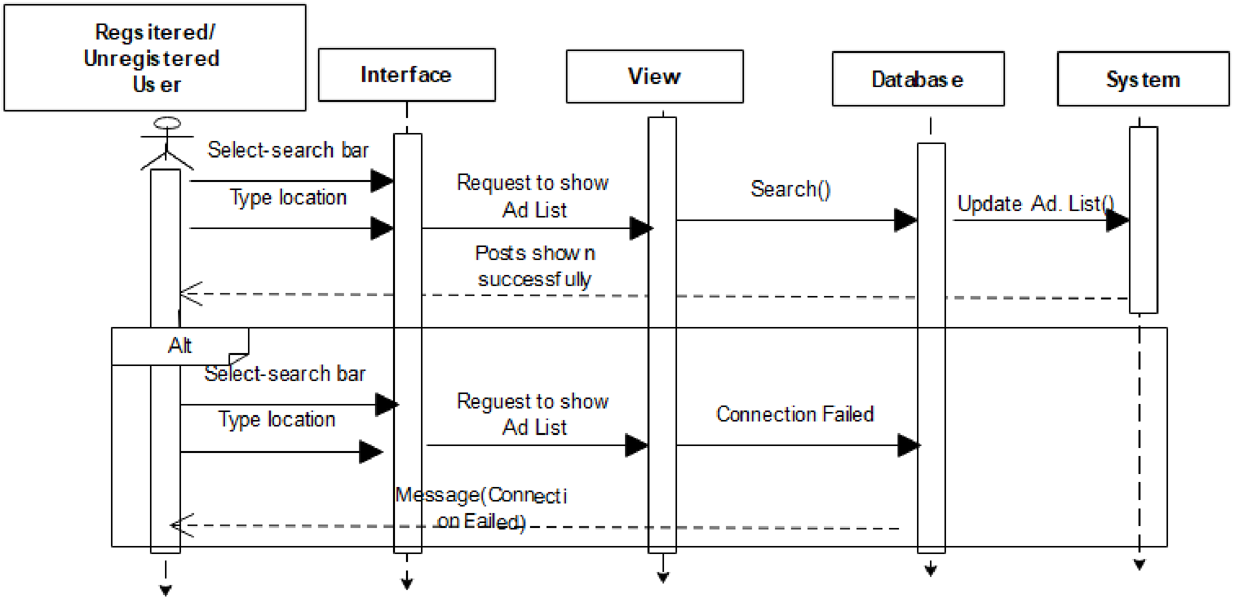

Figure23: Sequence diagram for Searching Ad by location (Registered User /unregistered user)

UC9: View Details of a Post (Registered Users)

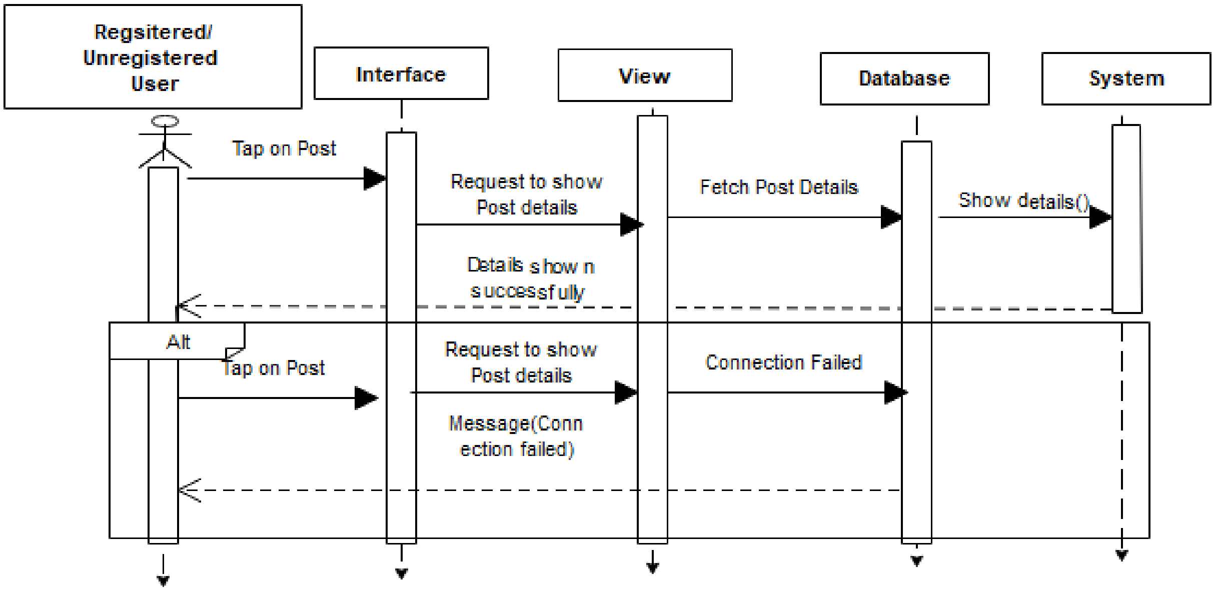

Figure24: Sequence diagram for View Details of a Post (Registered Users)

UC10: Finding Interior Designer (Registered Users/Unregistered Users)

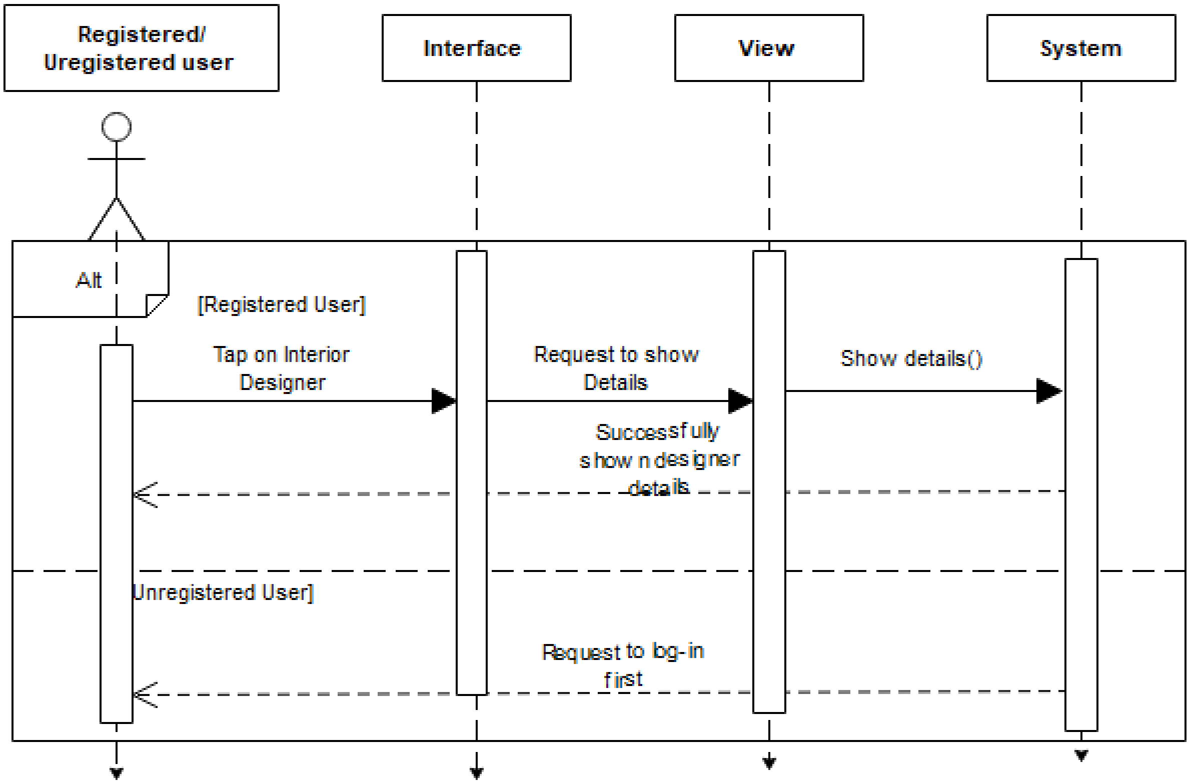

Figure25: Sequence diagram for Finding Interior Designer (Registered Users/Unregistered Users)

UC11: Finding a Moving Service (Registered Users/Unregistered Users)

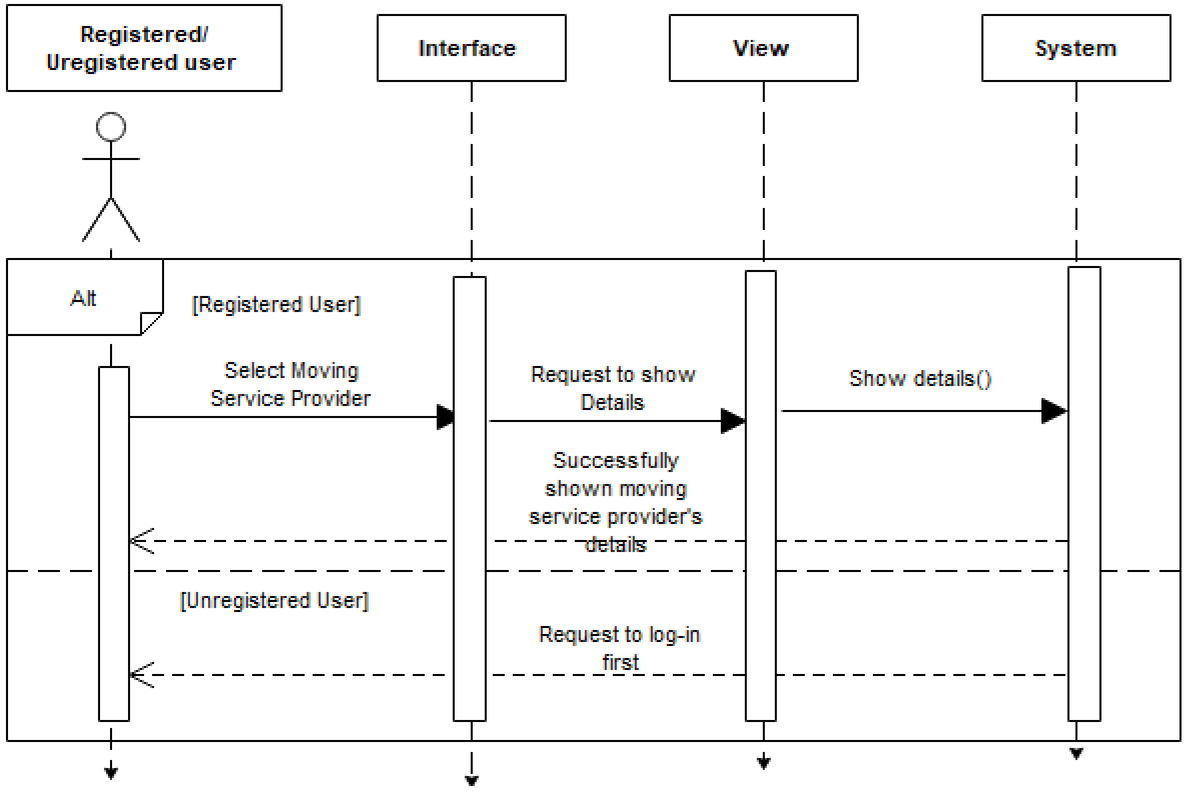

Figure26: Sequence diagram for Finding Interior Designer (Registered Users/Unregistered Users)

UC12: Finding Cleaning Service (Registered Users/Unregistered Users)

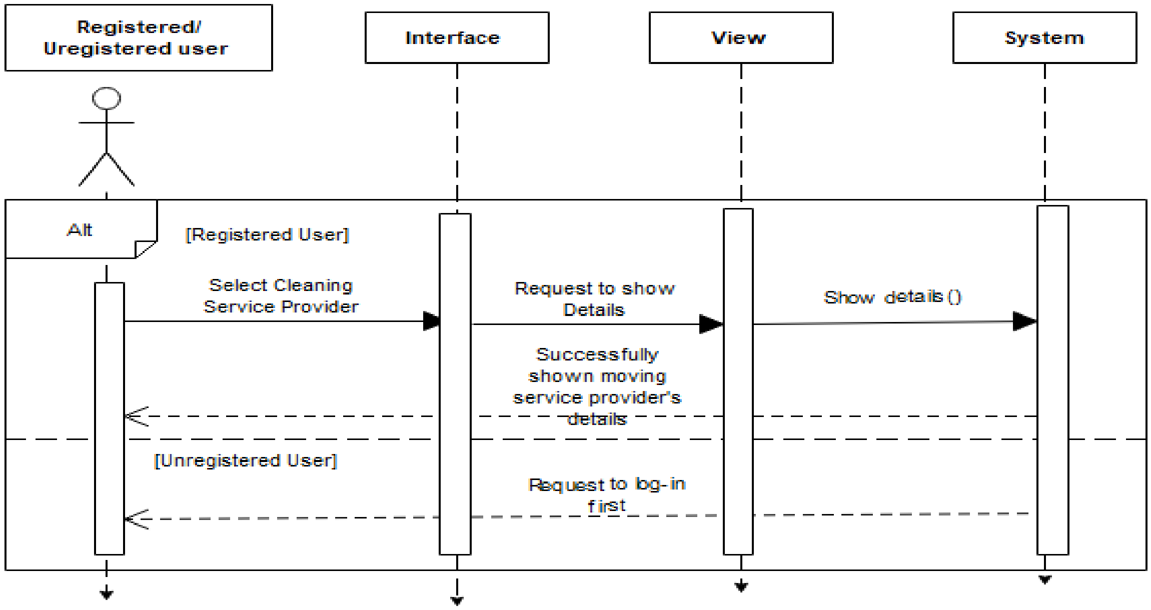

Figure27: Sequence diagram for Finding Cleaning Service (Registered Users/Unregistered Users)

UC13: Finding Furniture and accessories shop (Registered Users/Unregistered Users)

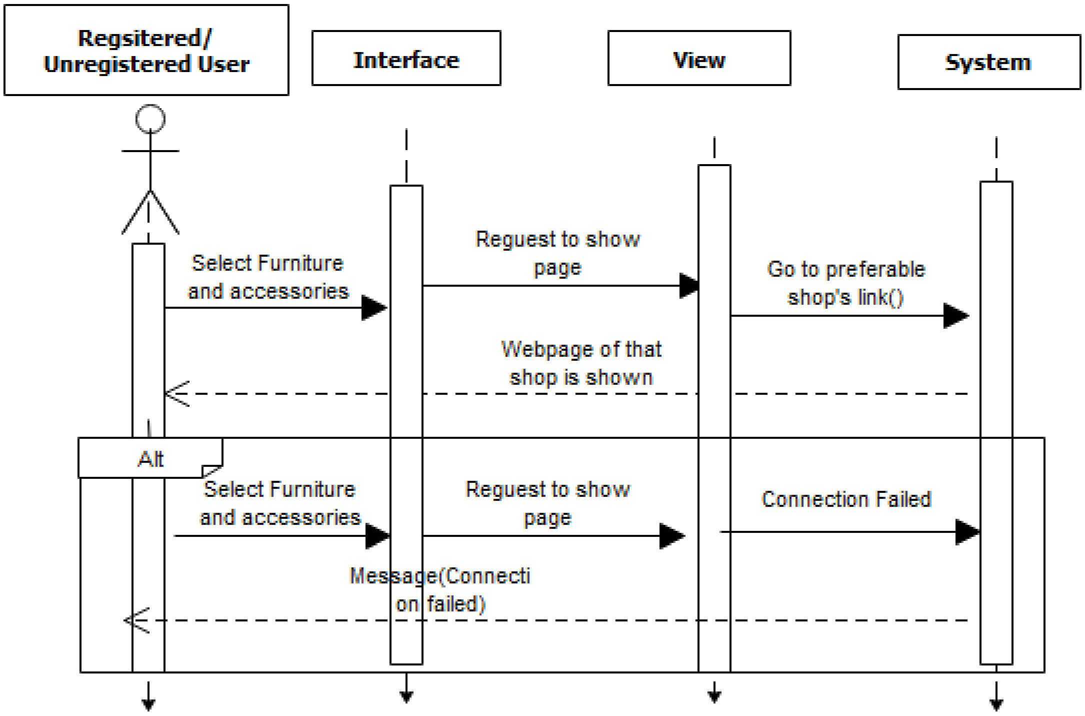
Regsitered/ Unregistered User
Interface
View
System
Select Furniture and accessories
Reguest to show page
Go to preferable shop's link()
Webpage of that shop is shown
Alt
Select Furniture and accessories
Reguest to show page
Connection Failed
Message(Connection failed)

Figure28: Sequence diagram for Finding Furniture and accessories shop (Registered Users/Unregistered Users)

6. State Machine Diagram:

State machine diagram is a behavior diagram which shows discrete behavior of a part of designed system through finite state transitions.

UC 01: Registration(Unregistered Users)

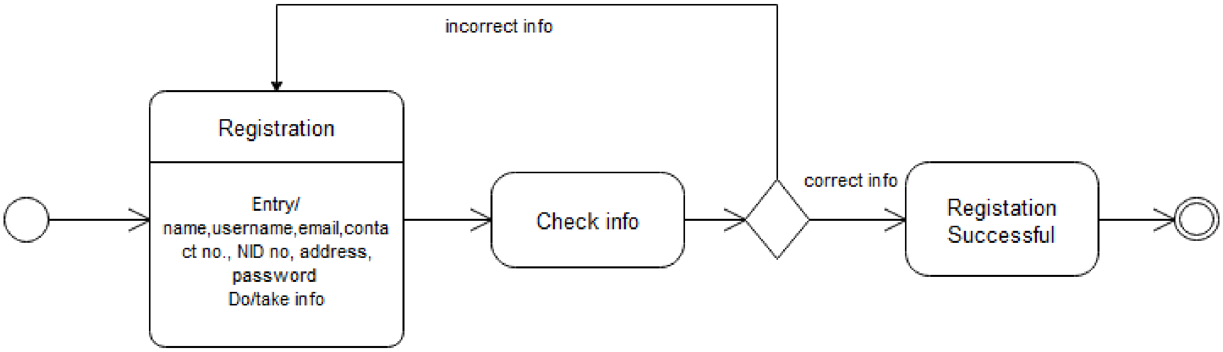

Figure29: State machine diagram for Registration (Unregistered Users)

UC 02: Log-In(Registered Users)

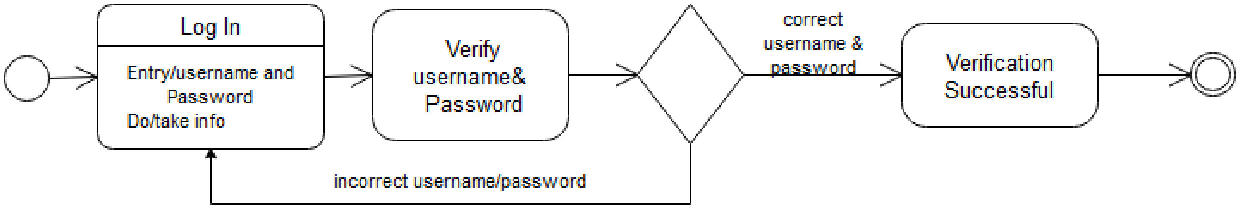
Log In
Entry/username and Password
Do/take info
Verify username& Password
correct username & password
Verification Successful
incorrect username/password

Figure30: State machine diagram for Log-In(Registered Users)

UC 03:Post Ad(Registered Users)

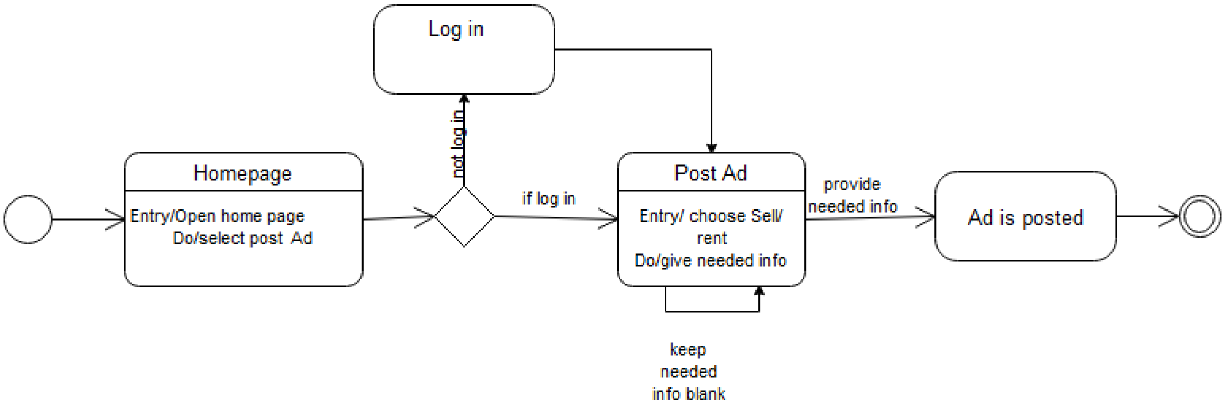

Figure31: State machine diagram for Post Ad(Registered Users)

UC 04:Edit Ad(Registered Users)

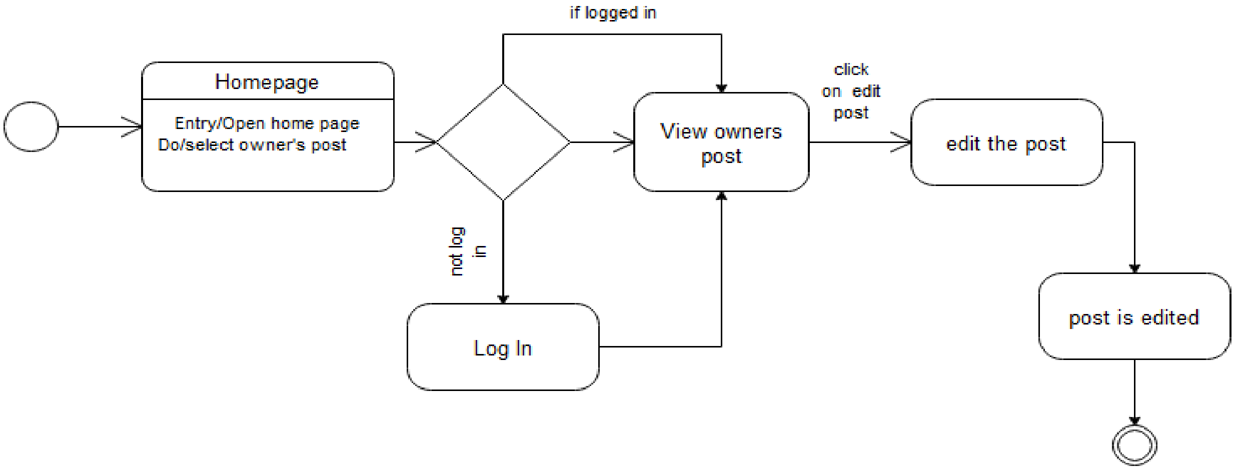

Figure32: State machine diagram for Edit Ad(Registered Users)

UC 05.Delete Ad(Registered Users)

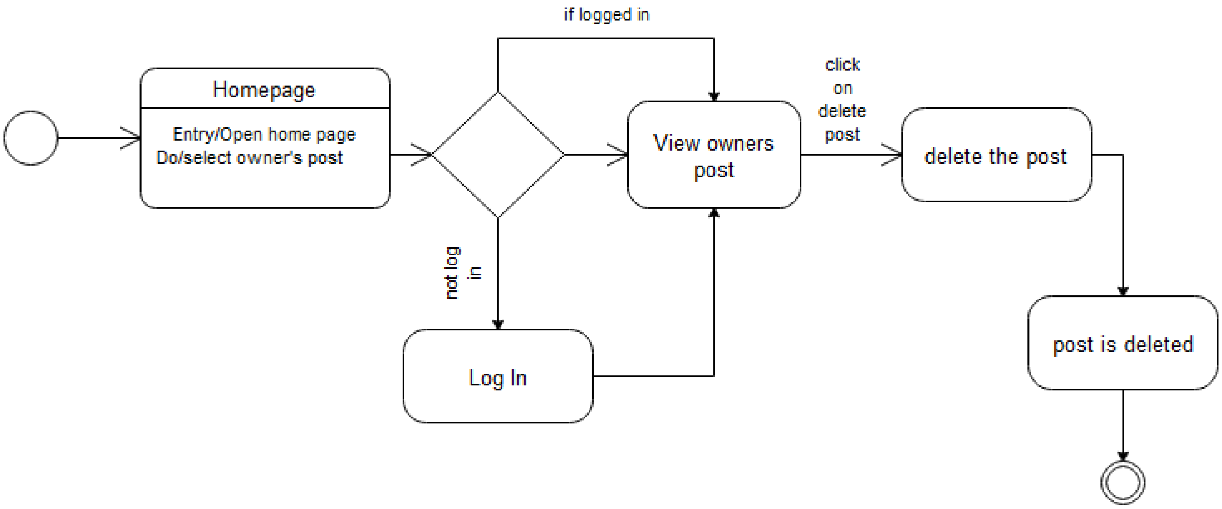

Figure33: State machine diagram for Delete Ad(Registered Users)

UC 06.Searching Ad by category for buying home (Registered Users/Unregistered Users)

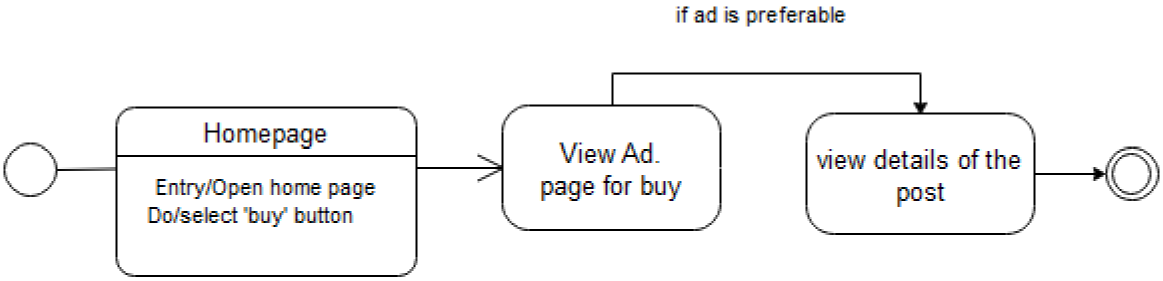

Figure34: State machine diagram for Searching Ad by category for buying home (Registered Users/Unregistered Users)

UC 07.Searching Ad by category for Renting (Registered Users/Unregistered Users)

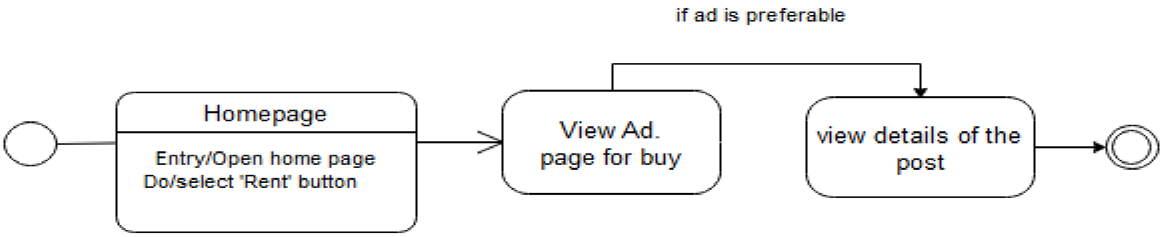

Figure35: State machine diagram for Searching Ad by category for Renting (Registered User / Unregistered Users)

UC8: Searching Ad by Location(Registered Users/Unregistered Users)

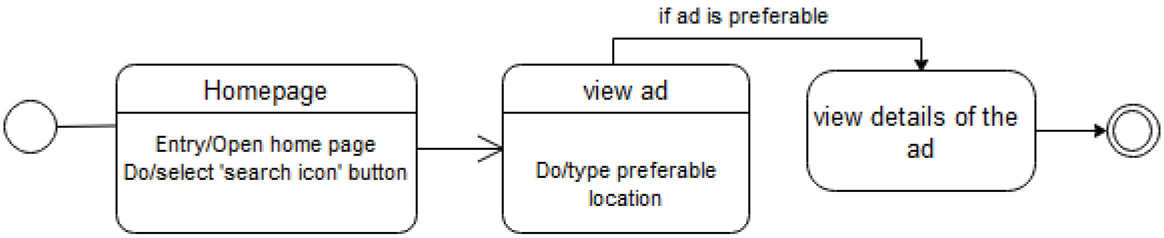

Figure36: State machine diagram for Searching Ad by location (Registered User / Unregistered Users)

UC9: View Details of a Post (Registered Users)

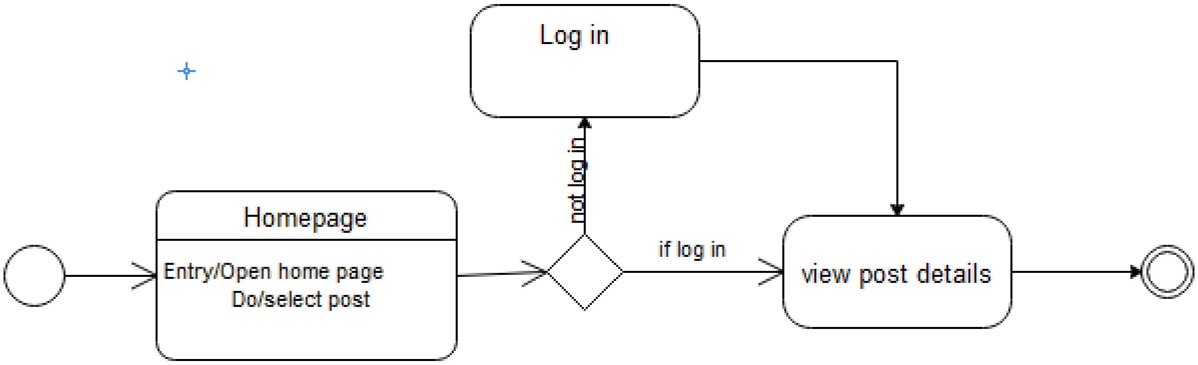

Figure37: State machine diagram for View Details of a Post (Registered Users)

UC10: Finding Interior Designer (Registered Users/Unregistered Users)

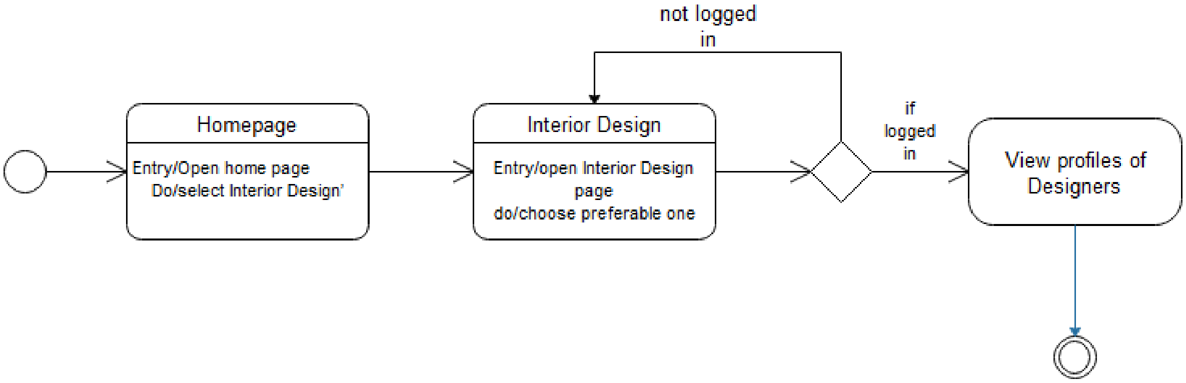

Figure38: State machine diagram for Finding Interior Designer (Registered Users/Unregistered Users)

UC11: Finding a Moving Service (Registered Users/Unregistered Users)

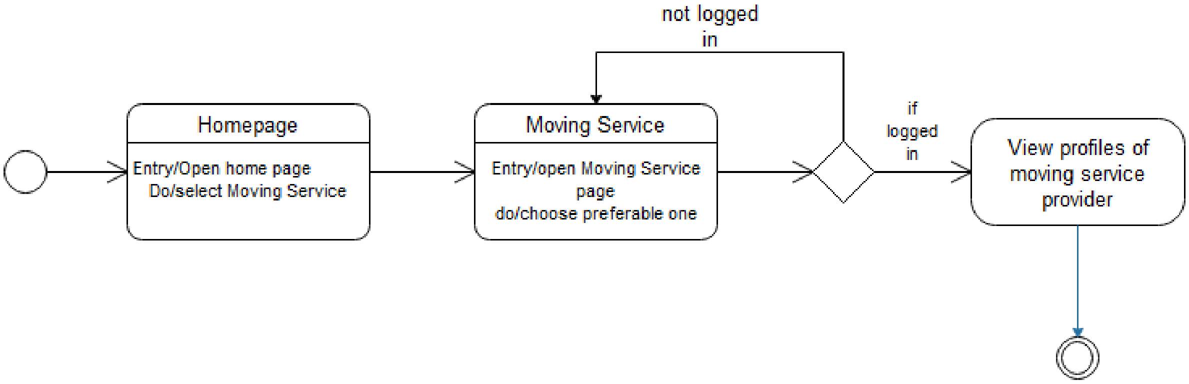

Figure39: State machine diagram for Finding Interior Designer (Registered Users/Unregistered Users

UC12: Finding Cleaning Service (Registered Users/Unregistered Users)

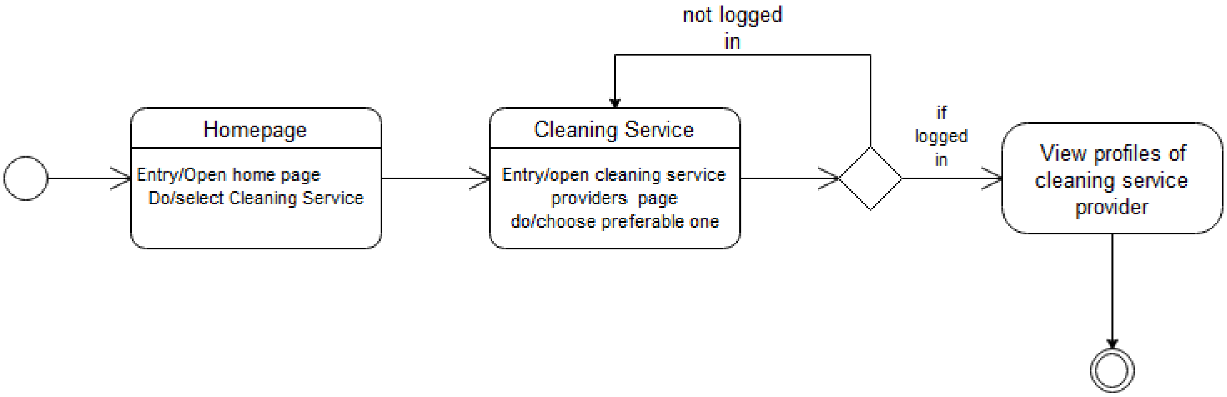
not logged in
Homepage
Entry/Open home page
Do/select Cleaning Service
Cleaning Service
Entry/open cleaning service providers page
do/choose preferable one
if logged in
View profiles of cleaning service provider

Figure40: State machine diagram for Finding Cleaning Service (Registered Users/Unregistered Users)

UC13: Finding Furniture and accessories shop (Registered Users/Unregistered Users)

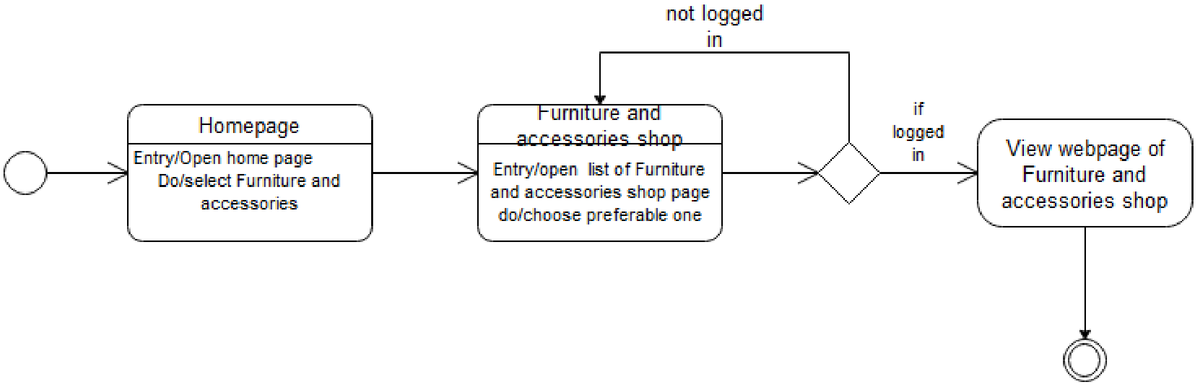
not logged in
Homepage
Entry/Open home page
Do/select Furniture and accessories
Furniture and accessories shop
Entry/open list of Furniture and accessories shop page
do/choose preferable one
if logged in
View webpage of Furniture and accessories shop

Figure41: State machine diagram for Finding Furniture and accessories shop(Registered Users/Unregistered Users)

7. Detail Class Diagram:

A detailed class diagram shows the system's classes, their attributes, operations and the relationship between objects. The detailed class diagram of "Humble Abode" is shown in figure 42:

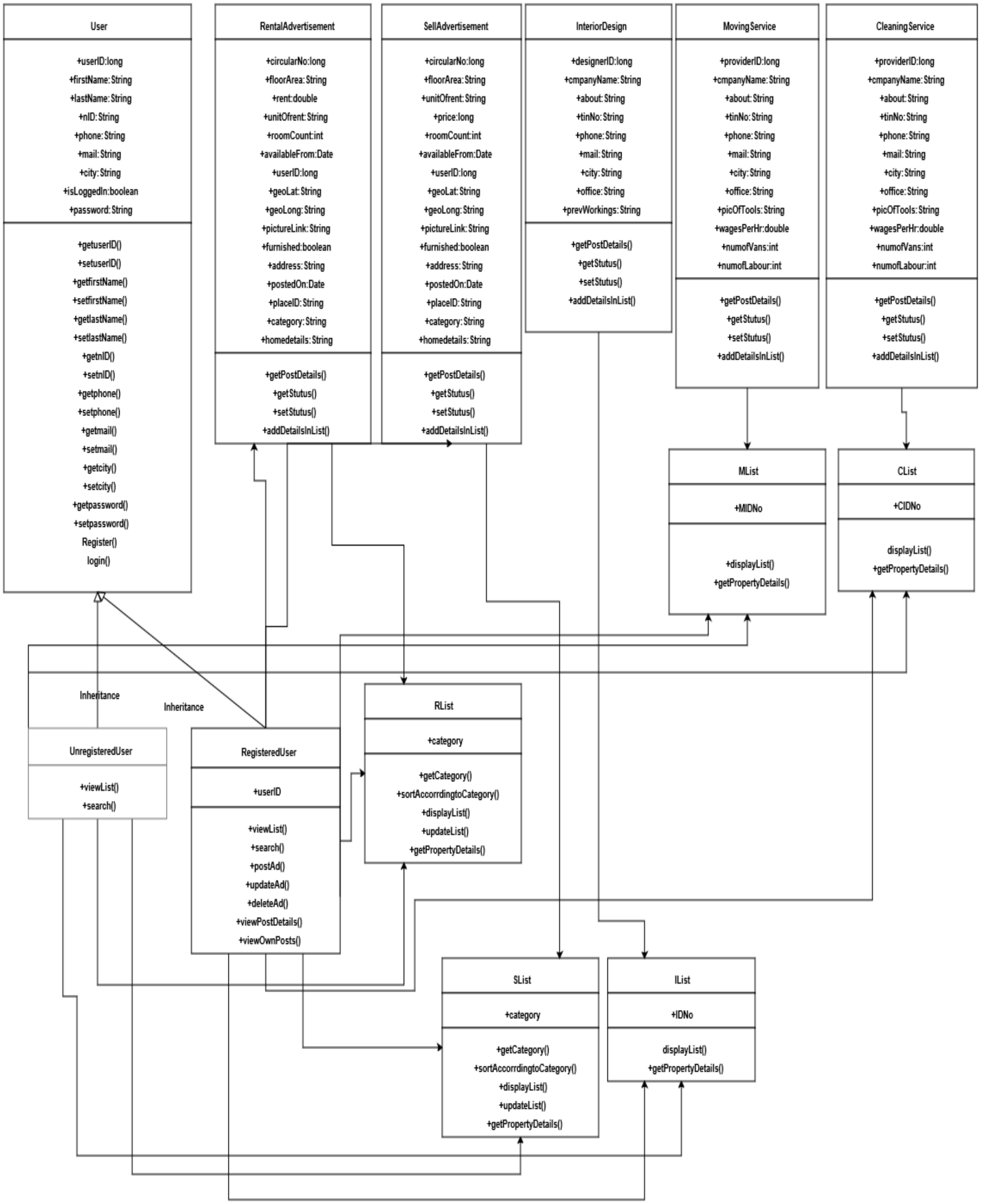

Figure42: Detail class diagram

8. Conclusion:

The detailed design document enlists enough and necessary study on different context models, class diagrams, activity diagrams, sequence diagrams, state machine diagram, system architecture style, that are required for the project development. In order to derive the design, clear and thorough understanding of each class activity is needed. Moreover, analyzing on life cycle of each class to build a strong design/structure is needed. This document will help the stakeholders to have an abstract idea of the model of the proposed system. So, the system design document is an unavoidable deliverable for developing successful software.