

# **Software Requirements Specification & Analysis**

## **Rentree**

Rent Management System

Prepared by:

**M. Mahtab Hossain Jhalok**

**BSSE Roll: 1103**

**Md. Raisul Islam Rahat**

**BSSE Roll: 1117**

Supervised by:

**Dr.Md.Shariful Islam**

Professor,

Institute of Information Technology

University of Dhaka



Institute of Information Technology  
University of Dhaka



---

Supervisor's Signature

<b>CHAPTER-01: INTRODUCTION</b>	<b>3</b>
PURPOSE	3
INTENDED AUDIENCE	3
CONCLUSION	4
<b>CHAPTER -02: INCEPTION OF RTMS</b>	<b>4</b>
INTRODUCTION	4
LIST OF STAKEHOLDERS	5
Icebreaking	5
CONCLUSION	6
<b>Chapter 3: Elicitation of Renter</b>	<b>6</b>
Introduction	7
Collaborative Requirements Gathering	7
Quality Function Deployment	7
Normal Requirements:	8
Expected Requirements:	8
Exciting Requirements:	8
<b>Chapter 4: UseCase Scenario Of Renter</b>	<b>9</b>
<b>Management System</b>	<b>9</b>
Account management:	9
Post Management:	11
Chatbox/Contact:	12
Notification Management:	12
<b>Chapter-5: Use Case Diagram of Renter</b>	<b>12</b>
<b>Management system</b>	<b>12</b>
What is a Use Case Diagram?	13
Level 0	13
Level 1	15
Level 1.1	16
Level 1.2	18
Level 1.3	19
Level 1.4	20
<b>Activity Diagram of Renter</b>	<b>21</b>
Definition	21
Level 0	21
Level 1	22

Level 1.1	23
Level 1.1.2	24
Level 1.1.3	26
Level 1.1.4	27
<b>Swimlane Diagram of Rentree</b>	<b>28</b>
Definition of Swimlane Diagram	28
Swimlane ID (SID) 1	29
Swimlane ID (SID) 1.1	30
Swimlane ID (SID) 1.1.2	31
Swimlane ID (SID) 1.1.3	32
Swimlane ID (SID) 1.1.4	33
<b>Chapter-6: Data-Based Modelling</b>	<b>34</b>
Data modeling concept:	34
Data objects:	34
Final Data Objects:	37
Data Object Relationship:	38
Definition of ER Diagram	38
Schema Diagram :	39
<b>Chapter 7: Class-Based Modeling for Rentree</b>	<b>42</b>
CLASS-BASED MODELING CONCEPT	42
Potential nouns to be class:	42
List of verbs :	44
General classification	45
Selection Criteria	48
Selected Classes:	49
Attribute & Method Identification:	49
CRC Card:	51
Class Cards	52
CRC Diagram	57
<b>Chapter 8: Behavioral Modeling</b>	<b>64</b>
Event Table	64
State Transition	66
<b>Chapter 9: Sequence Diagram:</b>	<b>70</b>

# **CHAPTER-01: INTRODUCTION**

This chapter is a part of our software requirement specification for the project “Rent Management System”. In this chapter, we focus on the intended audience for this project.

## **PURPOSE**

This document briefly describes the Software Requirement Analysis of Renter Management System. It contains functional, non-functional and supporting requirements and establishes a requirements baseline for the development of the system. The requirements contained in the SRS are independent, uniquely numbered and organized by topic. The SRS serves as an official means of communicating user requirements to the developer and provides a common reference point for both the developer team and the stakeholder community. The SRS will evolve over time as users and developers work together to validate, clarify and expand its contents.

## **INTENDED AUDIENCE**

This SRS is intended for several audiences including the customers as well as the project designers, developers. The customer will use this SRS to verify that the developer team has created a product that the customer finds acceptable. The designers will use this SRS as a basis for creating the system’s design. The designers will continually refer back to this SRS to ensure that the system they are designing will fulfill the customer’s demands. The developers will use this SRS as a basis for developing the system’s functionality. The developers will link the requirements defined in this SRS to the software they create to ensure that they have created a software that

will fulfill all of the customer's documented requirements. When portions of the software are complete, the developer will run their tests on that software to ensure that the software fulfills the requirements documented in this SRS. The testers will again run their tests on the entire system when it is complete and ensure that all requirements documented in this SRS have been fulfilled.

## **CONCLUSION**

This analysis of the audience helped us to focus on the users who will be using our analysis. This overall document will help each and every person related to this project to have a better idea about the project.

# **CHAPTER -02: INCEPTION OF RENTREE**

## **INTRODUCTION**

Inception is the beginning phase of requirements engineering. It defines how a software project gets started and what the scope and nature of the problem to be solved are. The goal of the inception phase is to identify concurrent needs and conflicting requirements among the stakeholders of a software project. At project inception, we establish a basic understanding of the problem, the people who want a solution, the nature of the solution that is desired and the effectiveness of preliminary communication and collaborations between the other stakeholders and the software .

To establish the groundwork, we have worked with the following factors related to the inception phases:

1. List of stakeholders
2. Ice breaking
3. Multiple Viewpoints
4. Requirements questionnaire

## **LIST OF STAKEHOLDERS**

According to Sommerville and Sawyer [Som97], “Anyone who benefits in a direct or indirect way from the system which is being developed is a stakeholder.” This implies that stakeholders include the end users of the developed software as well as the people whose activities might be influenced by the tool. Towards the end of inception, the list of stakeholders is usually larger as every stakeholder is allowed to suggest one or more individuals who might be probable stakeholders for the given problem. To

identify stakeholders, we consulted some teachers and students of Dhaka University, Bangladesh and asked them the following questions:

Who will be using this web application?

Whose work will this project affect?

We identified following stakeholders for our assignment system:

1.Owner/Provider

2.Buyers/Renters

### **Icebreaking**

Icebreaking refers to the fact that to diminish the communication barrier between you and the other person. It is a crucial part since it denotes the acceptance of our proposal. We started this phase by talking with Random people. The people informally expressed their expectations. We also carried out informal meetings with people Through Online. The behavior of Buyers, Renters and sellers,Providers was positive and all of them want this change in Rentree management.

### **CONCLUSION**

The primary goal of this project is to model and design a software for those people who are related to a course. For these reasons, the software will be designed in such a way that it will bring comfort to the client who will use it. The software will be simple and user friendly. Otherwise, it will not be appreciated by the clients. The software will be designed in such a way that it takes very little time to manage. To make this software project successful, collaboration with stakeholders is the main priority. To sum up, what they want, how the software will work and how it can be more efficient than the earlier times is the main concern in this phase.

# **Chapter 3: Elicitation of Rentree**

## **Introduction**

We have seen Question and Answer (Q& A) approach in the previous chapter, where the inception phase of requirement engineering has been described. The main task of this phase is to combine the elements of problem solving, elaboration, negotiation and specification. The collaborative working approach of the stakeholders is required to elicit the requirements. We have finished the following tasks for eliciting requirements-

- \_ Collaborative Requirements Gathering
- \_ Quality Function Deployment
- \_ Usage Scenarios (Story)

## **Collaborative Requirements Gathering**

We have met with many stakeholders in the Inception phase such as the owner of some houses, buyer, and renter. These meetings created an indecisive state for us to elicit the requirements. To solve this problem, we have met with the stakeholders (who are acting a vital role in the whole process) a few times to elicit the requirements.

## **Quality Function Deployment**

Quality Function Deployment (QFD) is a process and set of tools used to effectively define customer requirements and convert them into detailed engineering specifications and plans to produce the products that fulfill those requirements. QFD is used to translate customer requirements i.e., Voice of the Customer (VOC) into measurable design targets and drive them from the

assembly level down through the sub-assembly, component, and production process levels. QFD methodology provides a defined set of matrices utilized to facilitate this progression.

### **Normal Requirements:**

- Users are sellers (owner of the house or hostel) and buyers (customer, students who look for a hostel seat, renter).
- Users can signup and login.
- Name, NID, Address, Image, Phone number, Email, District, Trade Name, Account - (Due/Received) of user is recorded. user ID is supplied by the system. Existing customers do not need to be signed up again.
- There will be different views for the seller and the buyer .
- Seller and buyer have their own seller id and buyer id which is generated by the system.
- Sellers can add their provided house or rent information in their post and create posts which will be viewed by the buyers.
- Buyer can see the posts and can select their desirable house then they can contact the seller.

### **Expected Requirements:**

- Fluent user-friendly and interactive UI.
- The system will be encrypted and secured.
- Account will be verified by sending OTP to users.
- Search options will be available for the buyers. They can search by location, price range, or room quantity.
- Backup database can be restored in case of total system failure.

### **Exciting Requirements:**

- The seller can add multiple house or rent information in a single post.
- Seller and buyer can chat through a chatbox.
- If a buyer sends a msg to the seller then a notification will be sent to the seller.

## **Chapter 4: UseCase Scenario Of Rentree**

### **Management System**

#### **Account management:**

Users must create an account to enter into the system. Any user needs to provide the following common information in order to create an account:

- o Full Name
- o Contact Number
- o Email Address
- o User Name
- o Password

After providing the following information, the user has to provide the user type. Also to be noted, a user can create only one account with the same phone number and email address. That means no more than one account can be related to any phone number or email. There will be 2 types of users: Seller and buyer. Different types of users will have to provide different information to proceed from here.

Here is the required information for each type of user:

- Seller:
- 1. Permanent house address
  - 2. Present Address

→ Buyer: 1. Present Address

For any user sign up, the phone number given will be sent an automated confirmation sms by the system.

- Update Account : Any kind of user can update some of their data, such as email, phone number, password, and username any time. However, some of the information for some kinds of users, which are considered to be more sensitive, such as business id, bank account no, etc. cannot be updated any time and has to be verified by an admin to be updated.
- Password Recovery : Any user who can recover their password through an automated process. However, for users other than customers, the system will have to be provided more information to confirm the user is legit. Once the legitimacy of a user is confirmed, the system will give the user an option to send a secret code required to reset password either by email or phone number. If password resetting turns out to be a regular habit for some users, admins will be notified and after that admin can decide if he/she wants to intervene or not.
- Login : A user can log into the system by providing username/ email/ phone number and password.

### **Welcome Screen:**

After signing in, the user will see a welcome screen . the seller will see a post creation option and the buyer will see a view post option .

### **Main Menu:**

Menu will contain various section such as user profile,notification about etc.

## **Profile:**

A profile will be created for each user. It will contain user name, email address, phone number and the profile photo of the user. A user can edit his photo and phone number in the profile section.

## **Post Management:**

❖ Seller Perspective: Seller can create posts that contain information about house, rent, or hostel. They can add multiple houses or rent information in a single post. After adding that information they can provide their contact information by those a buyer can reach them . They can update posts if they think they need to change some info or add some info . They can also delete information from a post or can delete the whole post . A post will contain :

1. Post Number
2. House / Rent / Hostel details
3. Contact Information

❖ Buyer perspective: Buyer can see posts. They can search their findings. They can sort posts by location or price range. They can also search by the number of rooms.

## **Chatbox/Contact:**

When a buyer finds his desired post then he can contact the seller by phone number or can chat through a chatbox that is in the system. They can discuss the post in this system . The chat information will be stored in the database.

## **Notification Management:**

If a buyer sends a msg to the seller then a notification will send to the seller. When the seller will log in into the system he will see the notification and view the notification. Then he can respond to the notification by replying to

the buyer. they can discuss their needs. In the same way, if the seller sends a reply and the buyer is not in the system then he will see a notification when he logged into the system.

## **Chapter-5: Use Case Diagram of Rentre**

### **Management system**

#### **What is a Use Case Diagram?**

A use case diagram is a dynamic or behavior diagram in UML. Use case diagrams model the functionality of a system using actors and use cases. Use cases are a set of actions, services, and functions that the system needs to perform. In this context, a "system" is something being developed or operated, such as a website. The "actors" are people or entities operating under defined roles within the system. Use case diagrams are valuable for visualizing the functional requirements of a system that will translate into design choices and development priorities. They also help identify any internal or external factors that may influence the system and should be taken into consideration.

#### **Primary Actor**

Primary actors interact directly to achieve the required system function and derive the intended benefit from the system. They work directly and frequently with the software. They produce some information and consume some information too.

## **Secondary Actor**

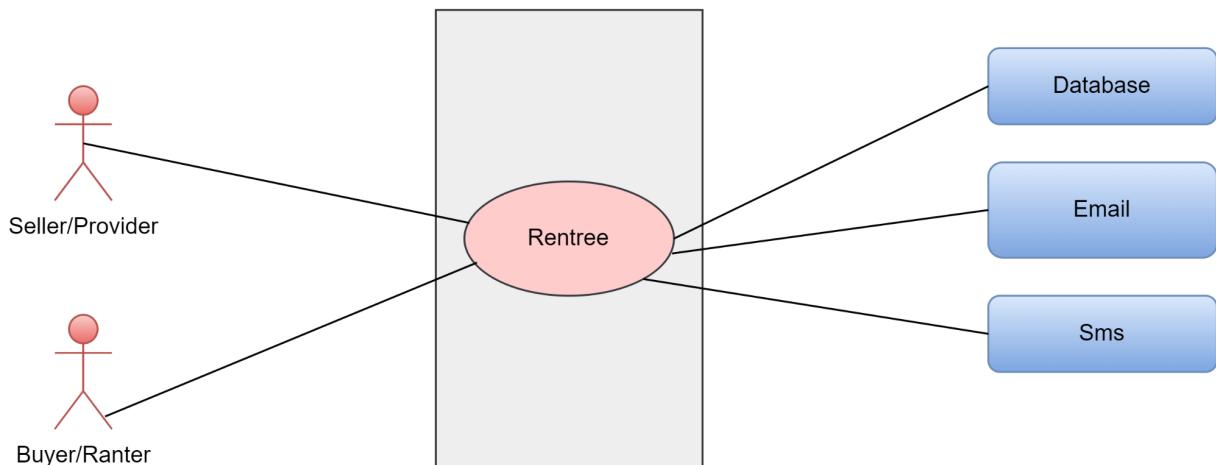
Secondary actors support the system so that primary actors can do their work. They either produce or consume information. Use Case diagram gives a non-technical view of the overall system.

### **Level 0**

Name:Renter management system

Primary Actors:Seller,Provider,Buyer,Renter

Secondary Actors:Sms,Email ,Database



### **Description of use case diagram level-0**

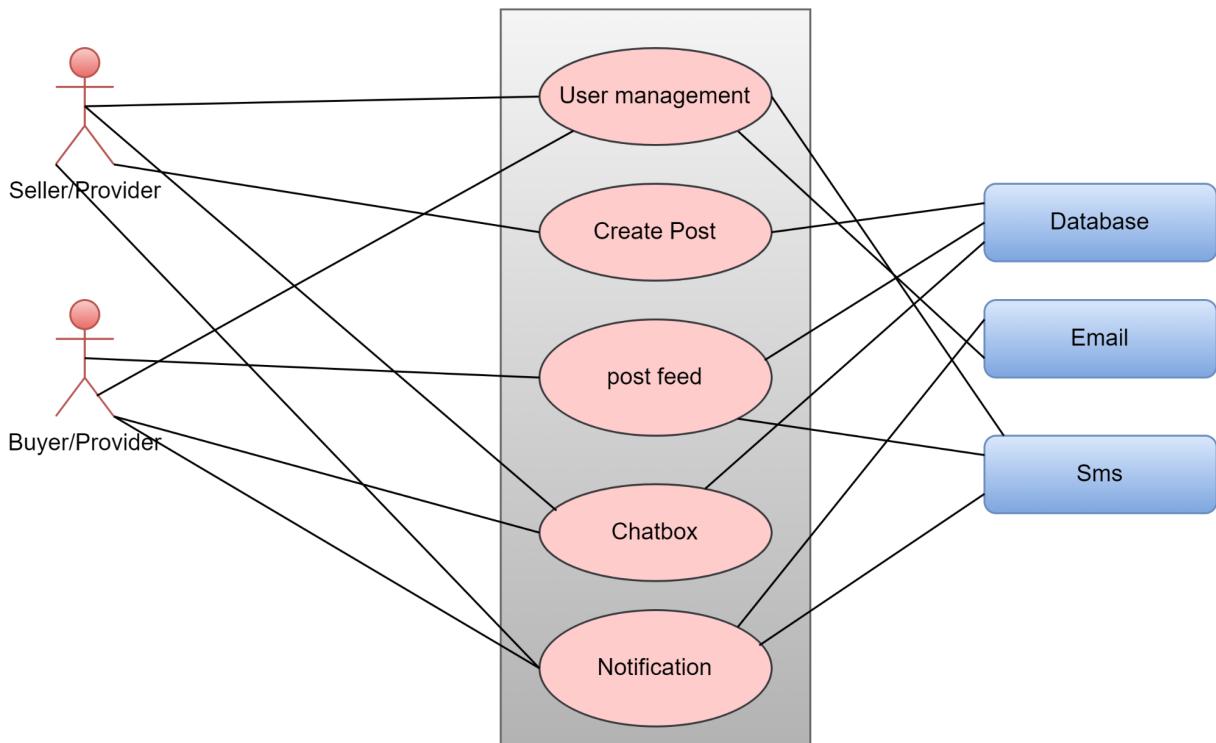
This is an overall view of all the primary and secondary actors altogether. As Buyers, Sellers, Renters,Provider are the only ones who will initiate a task for the system, they are considered as the primary actors. Buyers are the ones who will respond to the task initiated by the primary actors thus being the secondary actors. Sms, Email, Database are external software modules used by the system.

## Level 1

Name:Rentre management system

Primary Actors:Seller,Provider,Buyer,Renter

Secondary Actors:Sms,Email ,Database



## Description of use case diagram level-1

## **User Management**

Users must create an account and then log into the system to get the full functionality of the system. User can update his/her profile and can recover a password if forgotten.

## **Create post**

This section is used for Create Post.Seller or Provider Can Create Post By using this section.

## **Post feed**

This section is used for getting feedback from sellers and providers.Buyers or Renters can check the update and all information and requirements.

## **ChatBox**

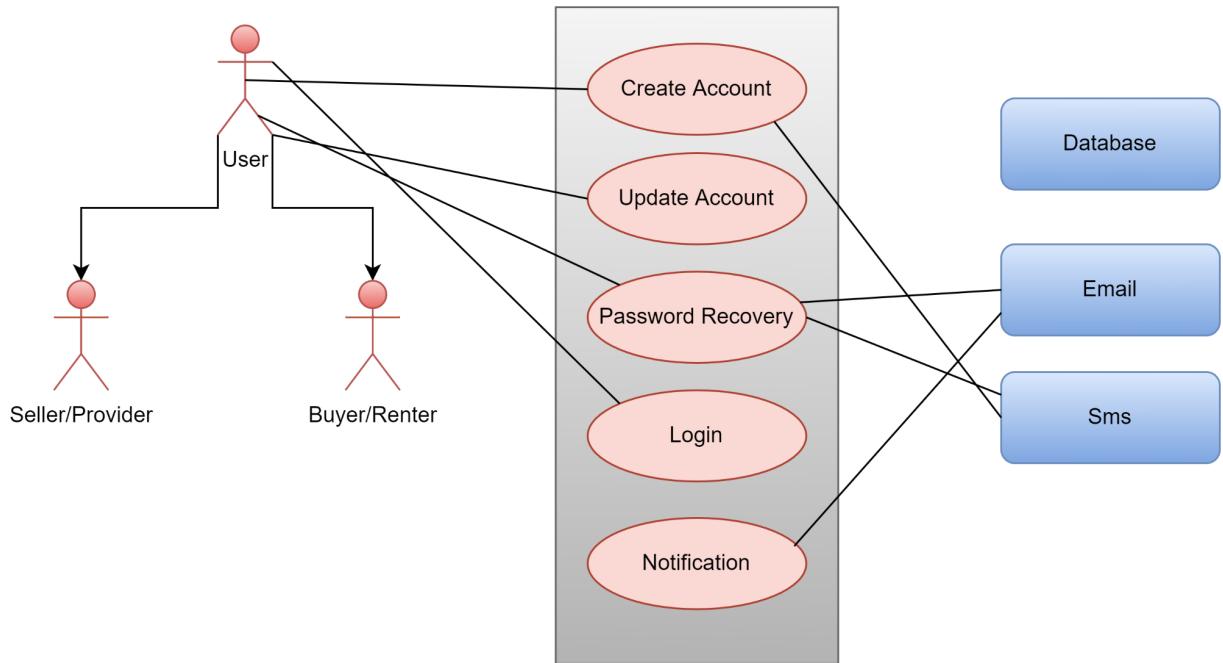
Sellers and Buyers can interact through this chatbox and share their requirements and confirm their deal

## **Level 1.1**

Name:User management

Primary Actors:User,Seller,Provider,Buyer,Renter

Secondary Actors:Sms,Email ,Database



## Description of use case diagram level-1.1

### 1.1.1 Create Account

To create an account, users must provide their information to the system.

They

must choose a user type and depending on their user type they must provide additional information.

### 1.1.2 Update Account

This section is used for updating account information .One can change their user type.

### 1.1.3. Password Recovery

A user can recover his/her password if forgotten, by using his/her email or phone number or by contacting the admin manually.

#### 1.1.4. Login

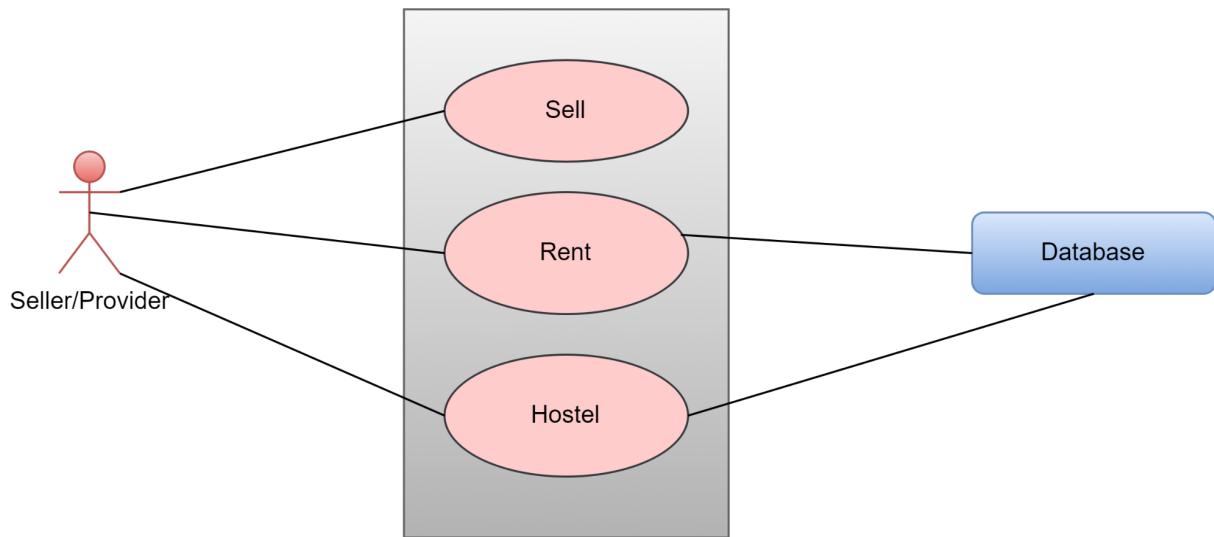
Users will log into the system by using his/her registered email-id/ phone number and password.

### Level 1.2

Name:Create Post

Primary Actors:Seller,Provider

Secondary Actors:Database



### Description of use case diagram level-1.2

#### 1.2.1 Sell

Seller can sell their house by using this section

### 1.2.2 Rent

This section is used for renting house

### 1.2.3 Hostel

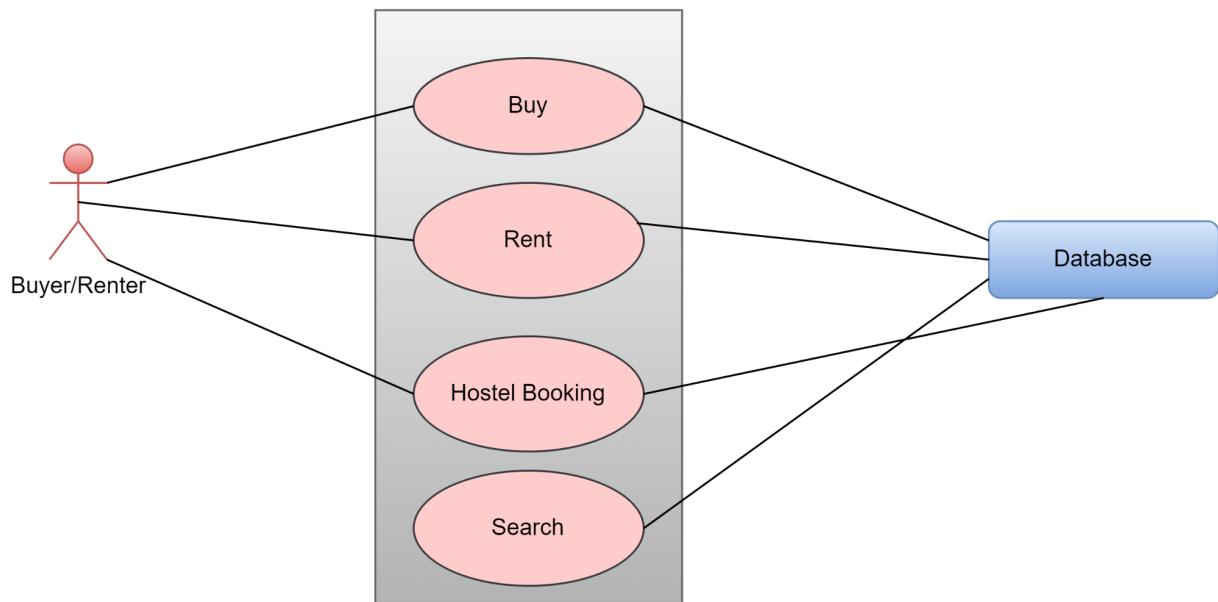
This section is used for renting a sit or room by hostel manager.

## Level 1.3

Name:Postfeed

Primary Actors:Buyer,Renter

Secondary Actors:Database



## Description of use case diagram level-1.3

### 1.3.1 Buy

Buyer can choose and buy house .

### **1.3.2 Rent**

Renter can rent house or room by using this section

### **1.3.3 Hostel Booking**

Student or bachelor can rent a sit or room .

### **1.3.4 Search**

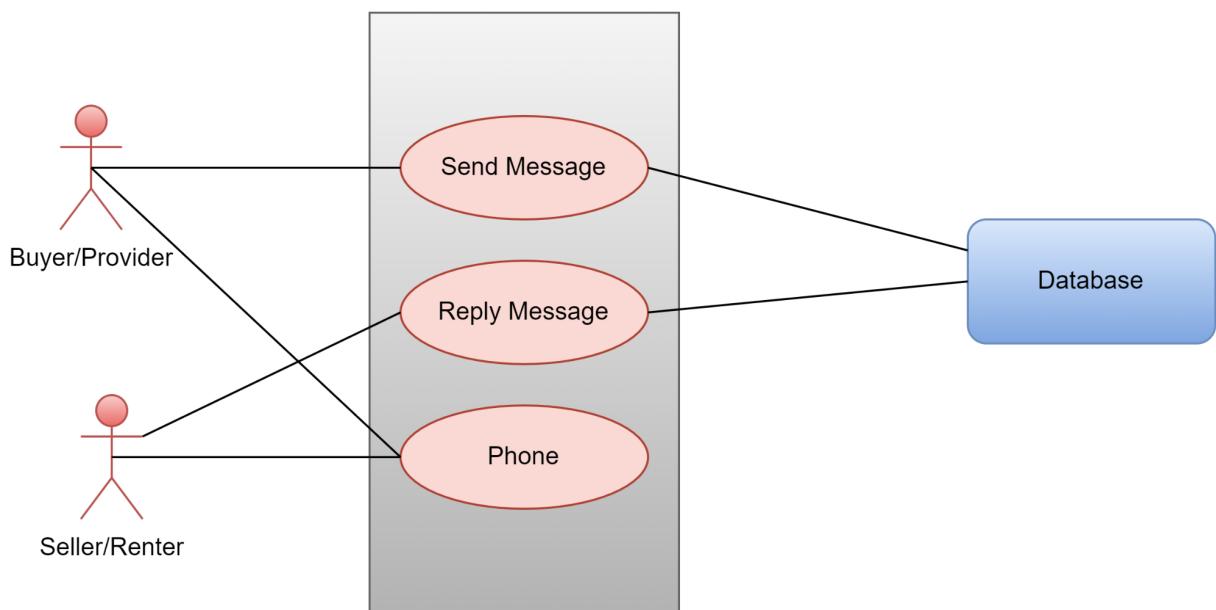
Buyers or Renters can search by location ,room quantity,price.

## **Level 1.4**

Name:ChatBox

Primary Actors:Seller,Provider,Buyer,Renter

Secondary Actors:Database



## **Description of use case diagram level-1.4**

### **1.4.1Send Message**

Buyers and Renters can send Message to owner ,provider

### **1.4.2 Reply Message**

Owner and provider can reply to their customer

### **1.4.3Phone**

Buyers or renters and sellers or providers can interact with phone by using this section

## **Activity Diagram of Rentree**

### **Definition**

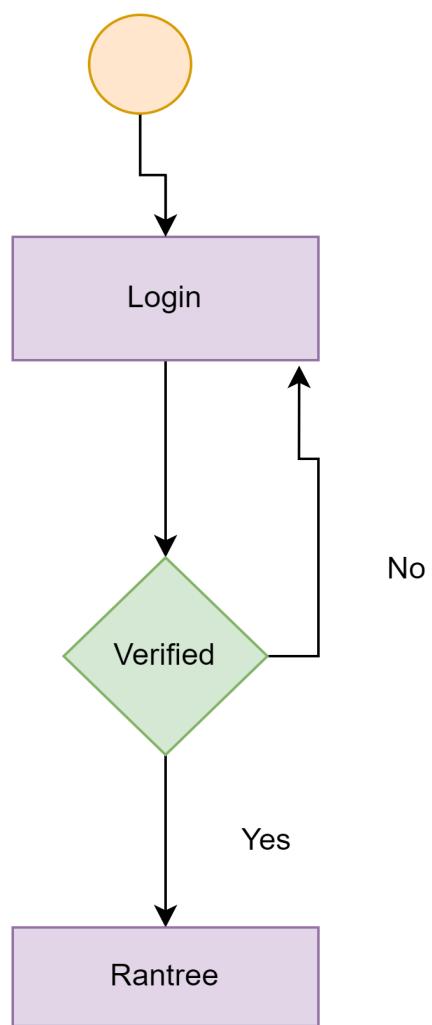
Activity diagrams are graphical representations of workflows of stepwise activities

and actions with support for choice, iteration, and concurrency.

### **Level 0**

Name: Rentree Management System

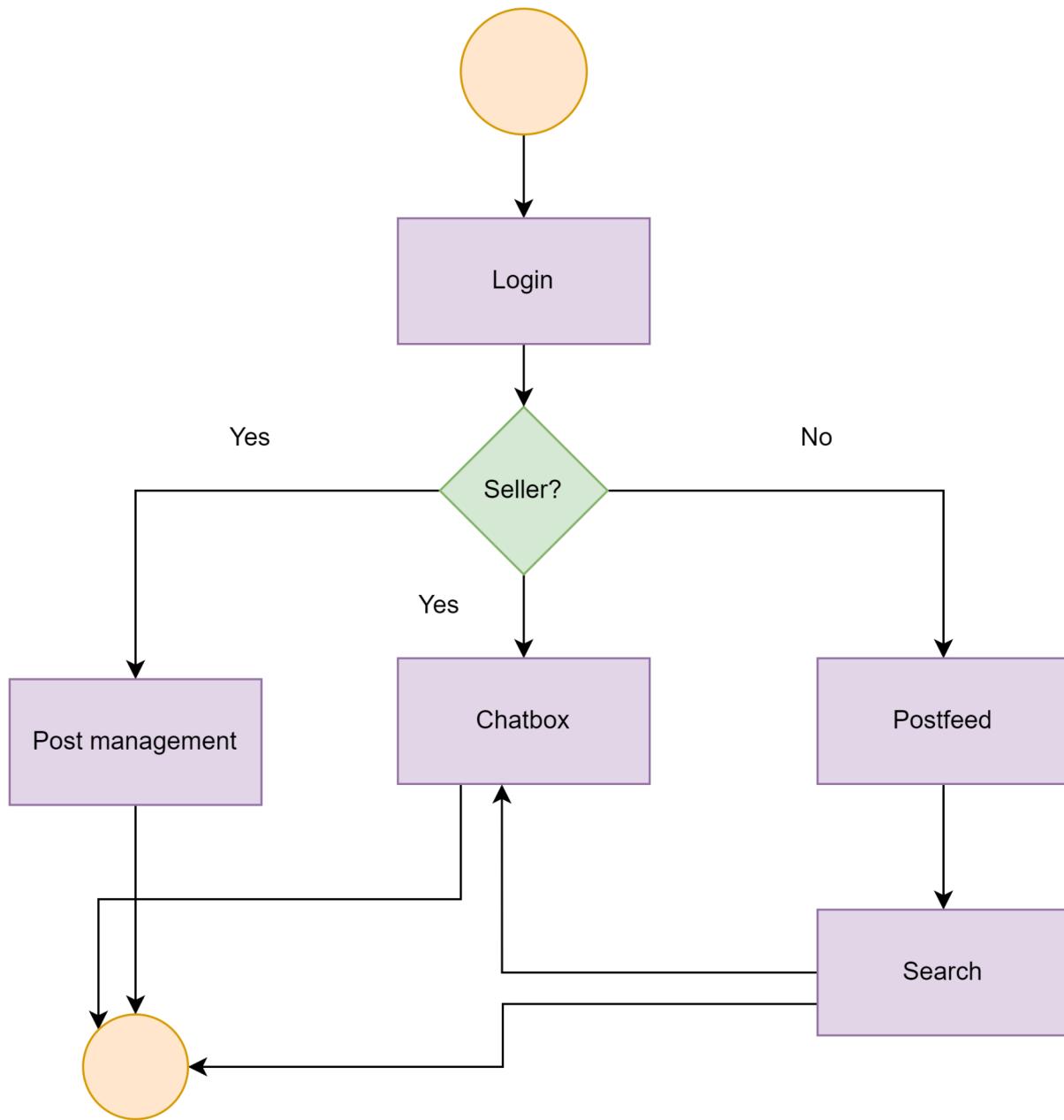
Reference: Use Case Level 0



## Level 1

Name: Rantree Management System

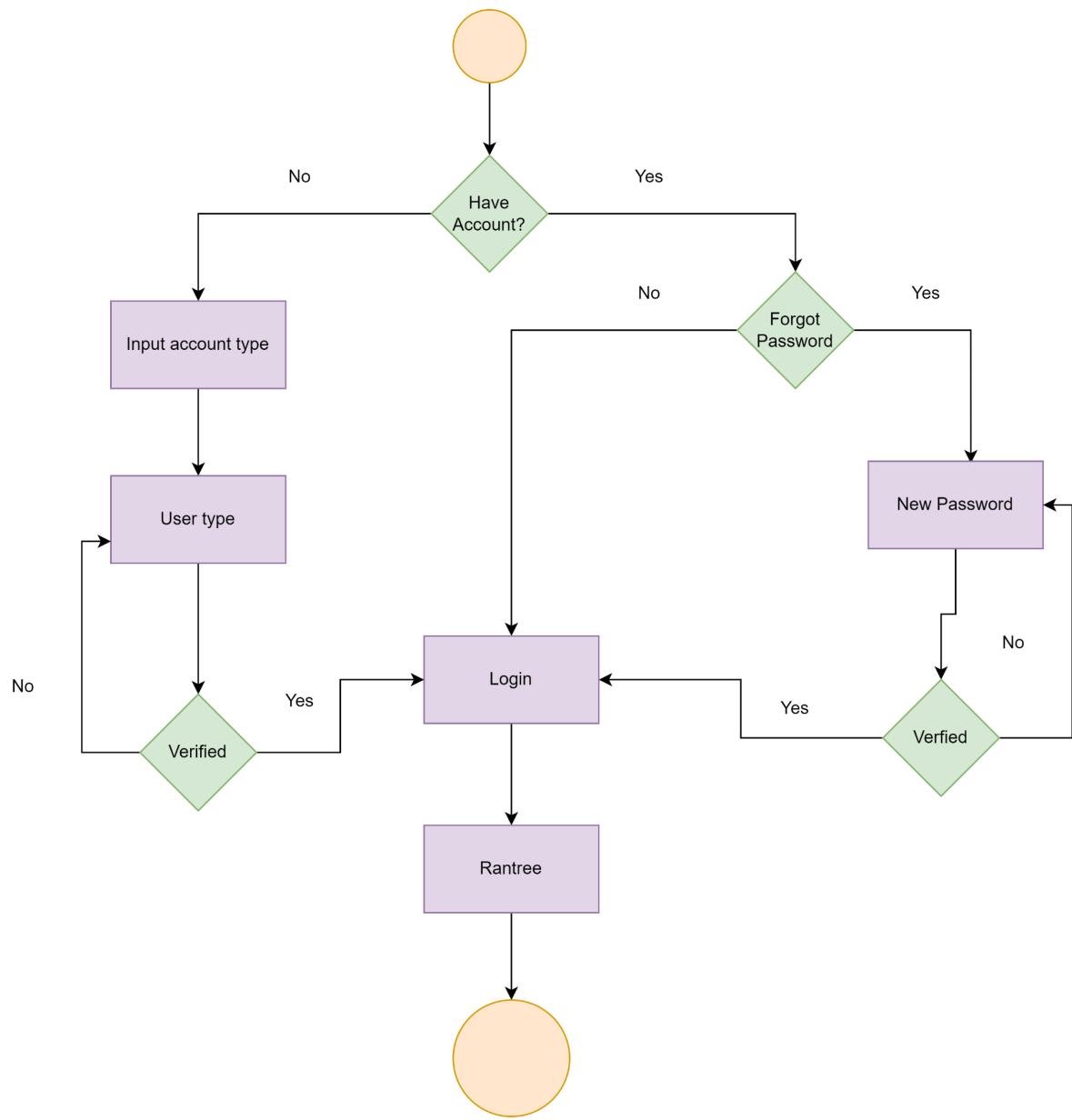
Reference: Use Case Level 1



## Level 1.1

Name: User Management

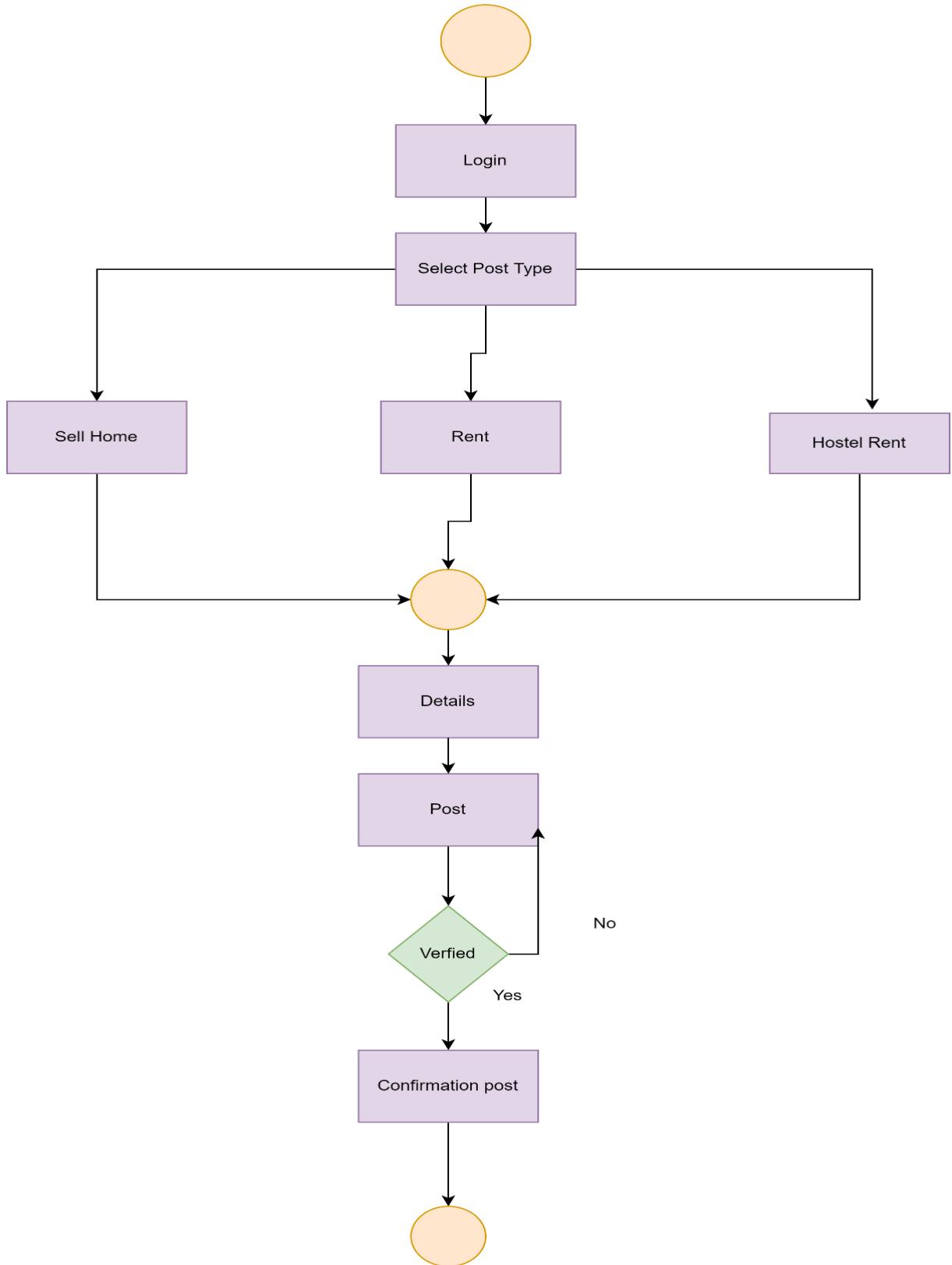
Reference: Use Case Level 1.1



### Level 1.1.2

Name:Create Post

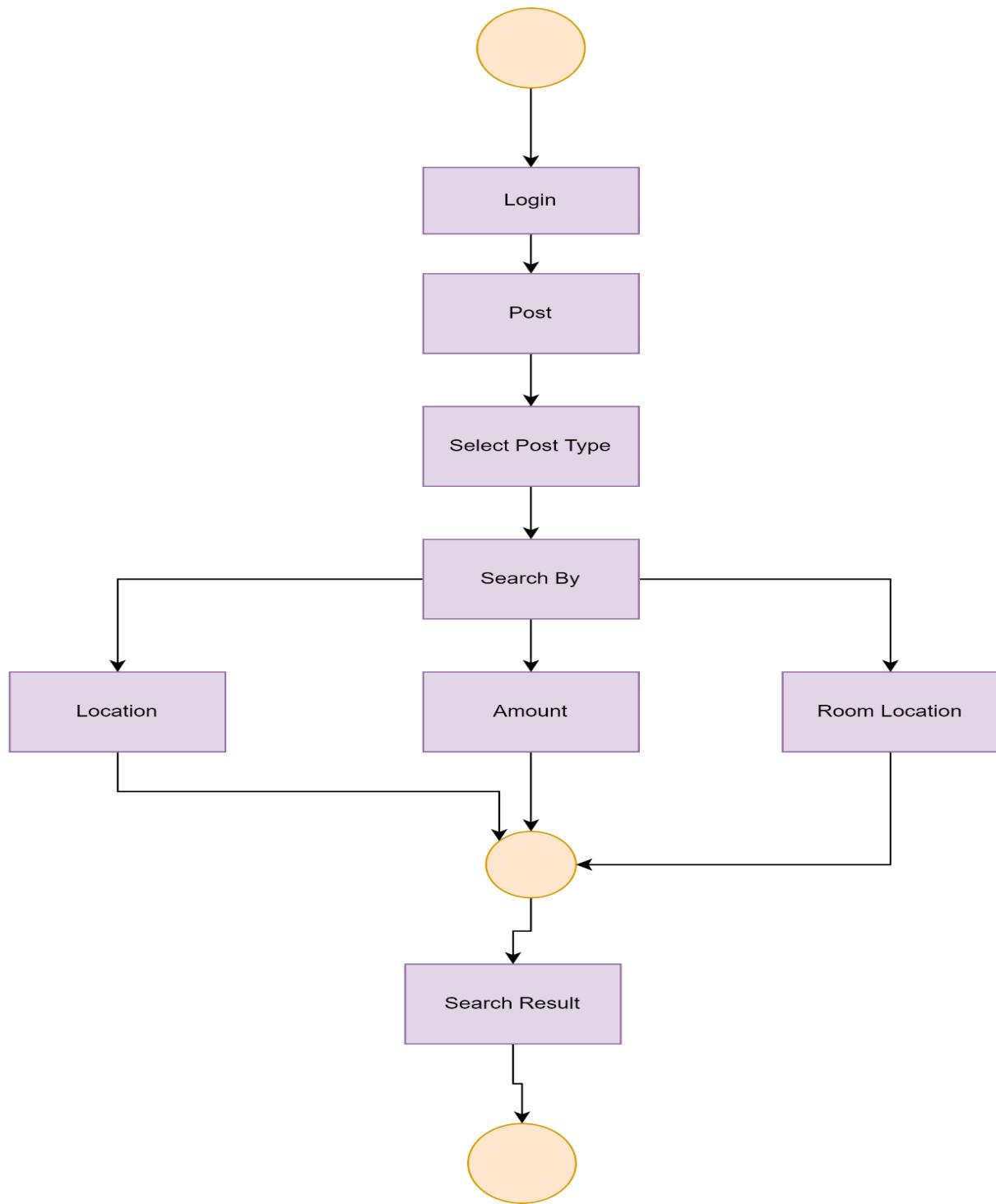
Reference: Use Case Level 1.1



### Level 1.1.3

Name: Postfeed

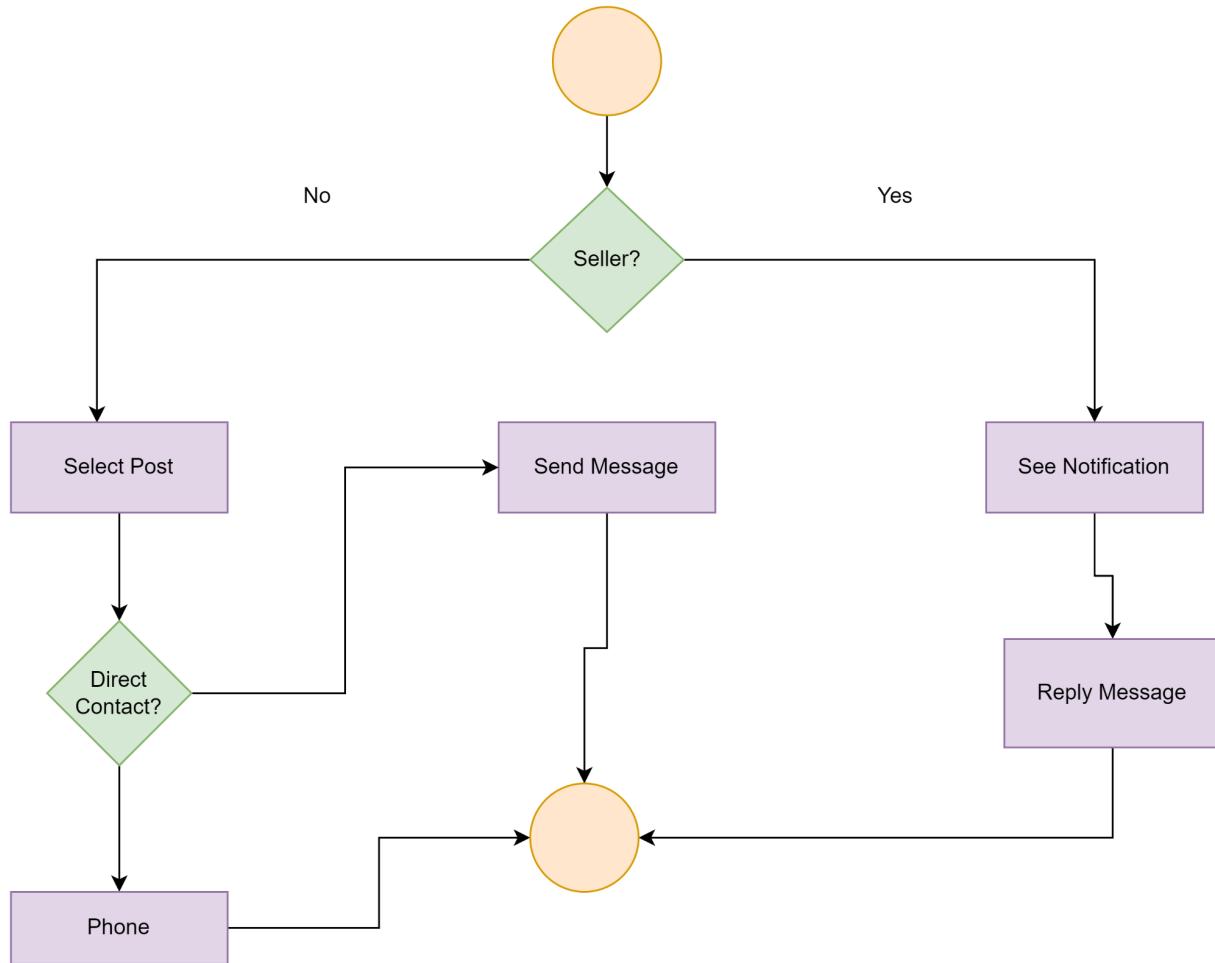
Reference: Use Case Level 1.1.3



## Level 1.1.4

Name: Chatbox

Reference: Use Case Level 1.1.4



## **Swimlane Diagram of Rentrée**

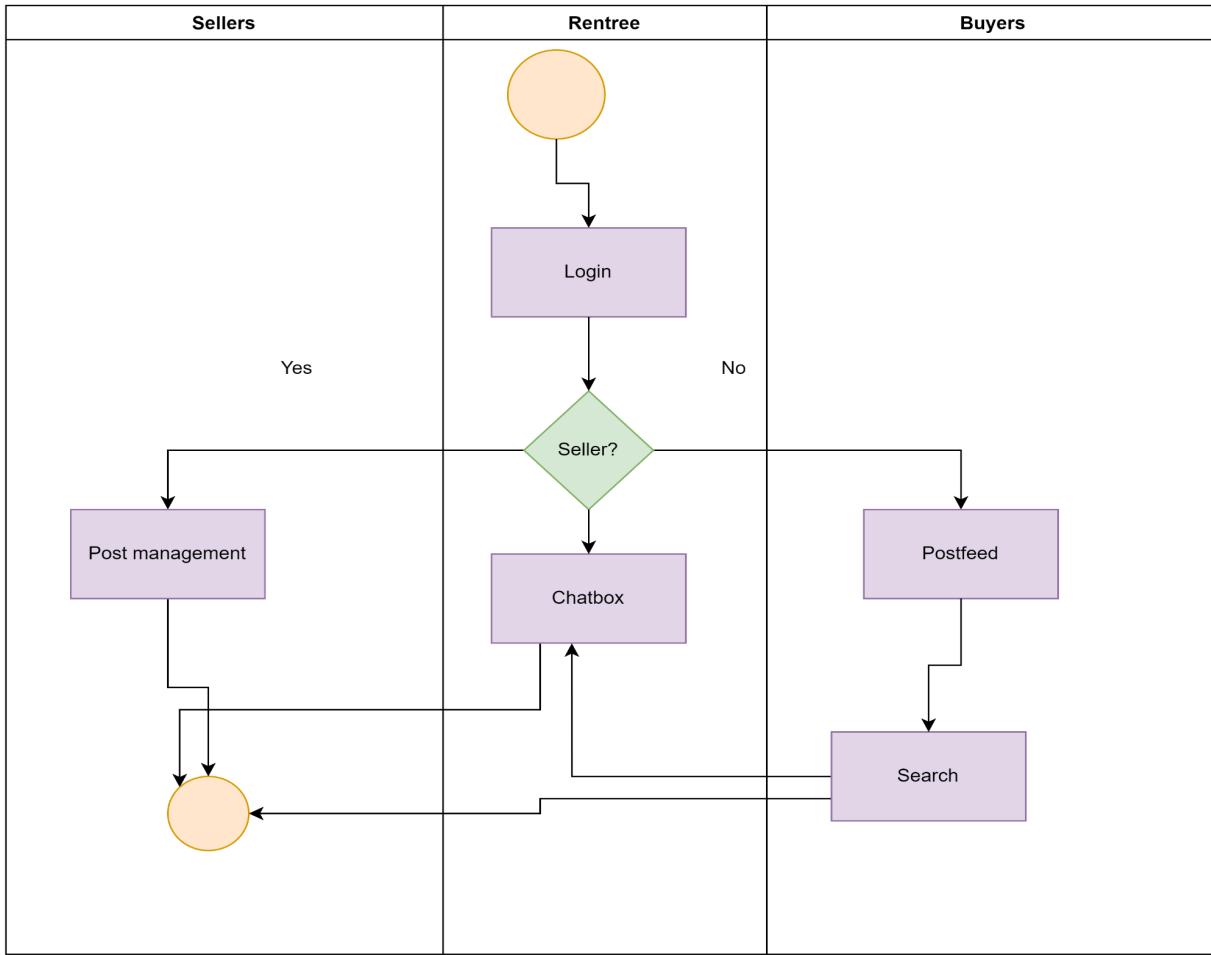
### **Definition of Swimlane Diagram**

If software requirements include the necessity to create, extend or interact with a database or complex data structures need to be constructed and manipulated, then the software team chooses to create data models as part of overall requirements modeling. The entity-relationship diagram (ERD) defines all data objects that are processed within the system, the relationships between the data objects, and the information about how the data objects are entered, stored, transformed, and produced within the system.

#### **Swimlane ID (SID) 1**

Name: User Management

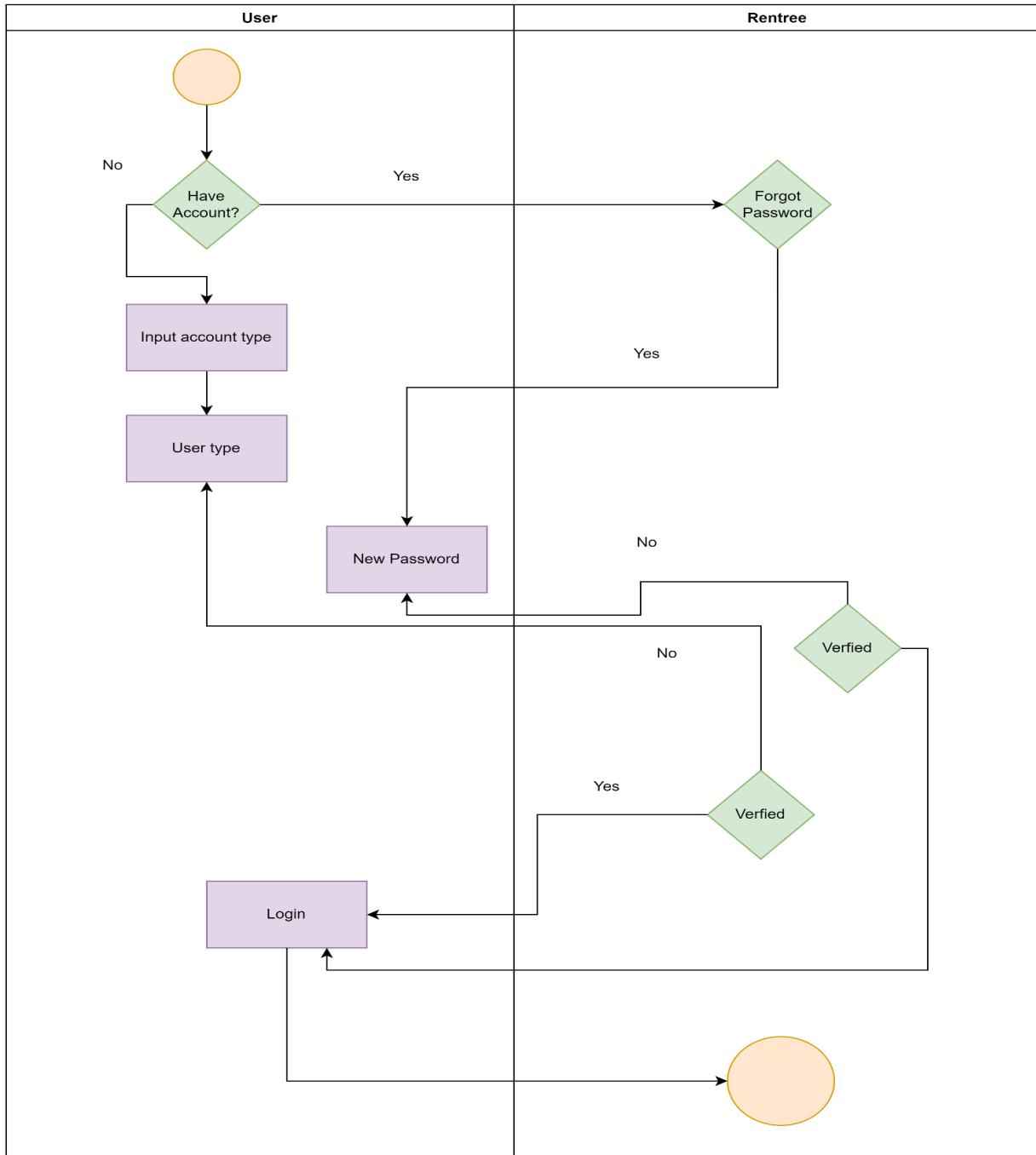
Reference: Use Case & Activity Level 1



## Swimlane ID (SID) 1.1

Name: User Management

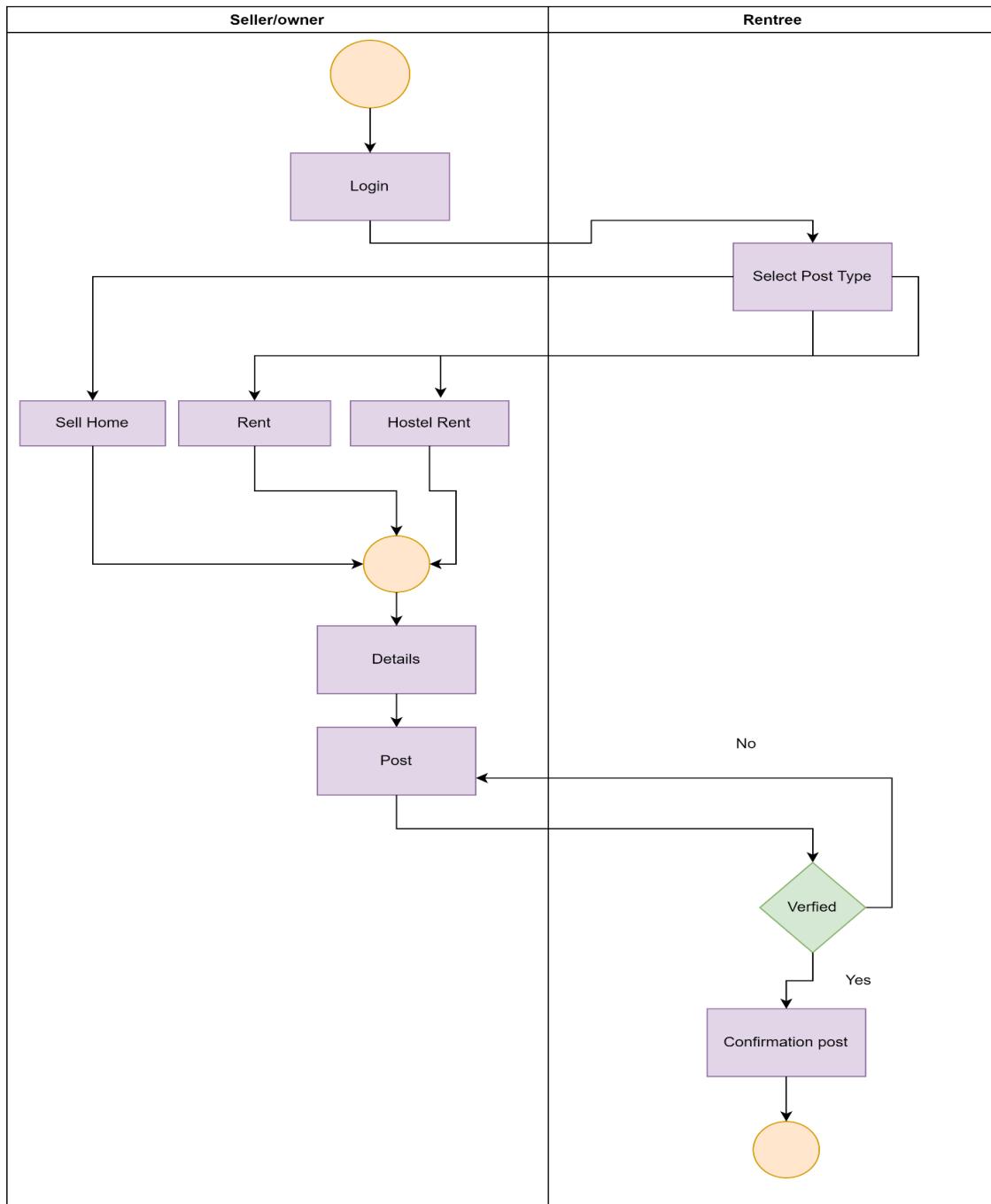
Reference: Use Case & Activity Level 1.1



## Swimlane ID (SID) 1.1.2

Name: Create Post

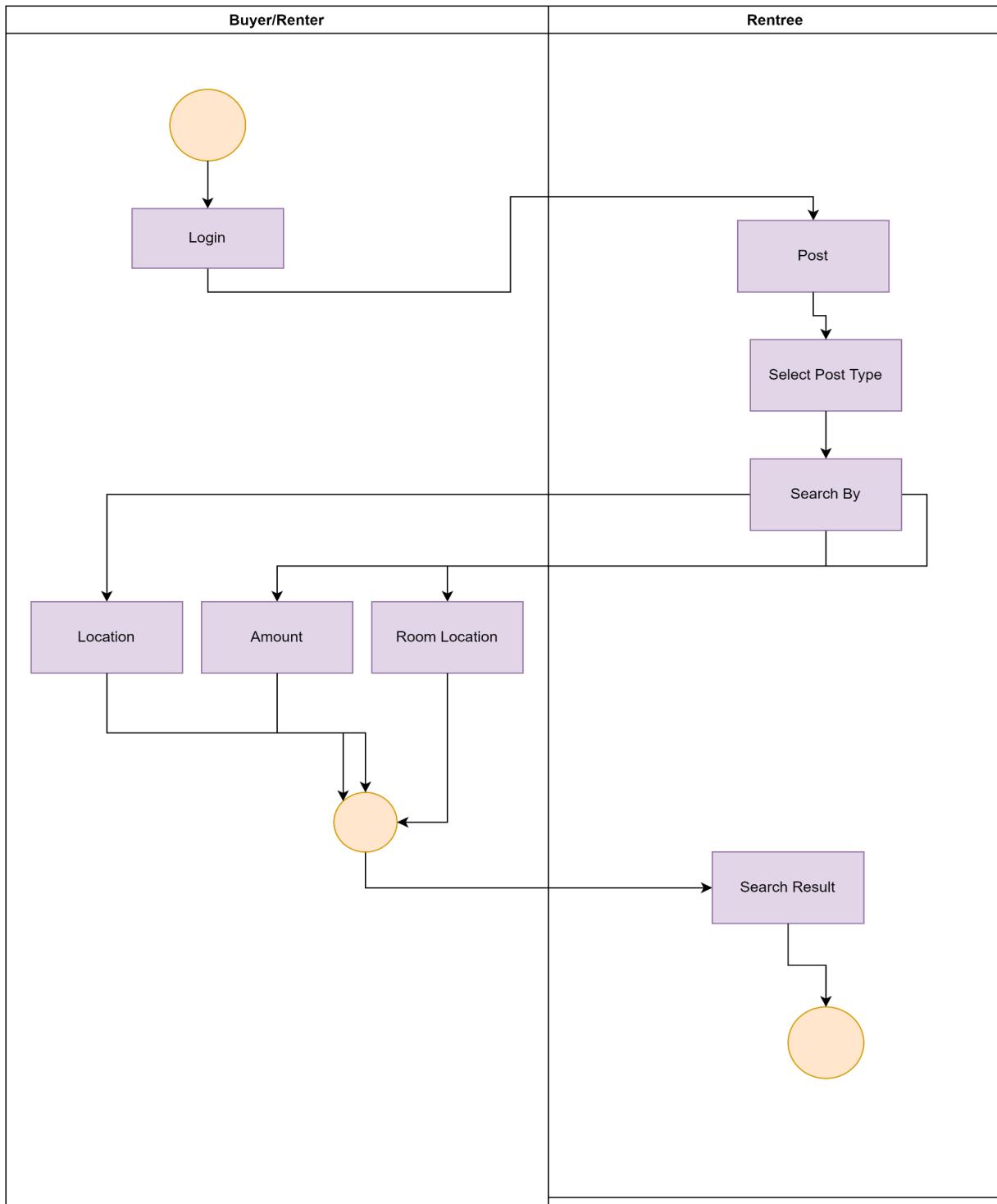
Reference: Use Case & Activity Level 1.1.2



### Swimlane ID (SID) 1.1.3

Name: Postfeed

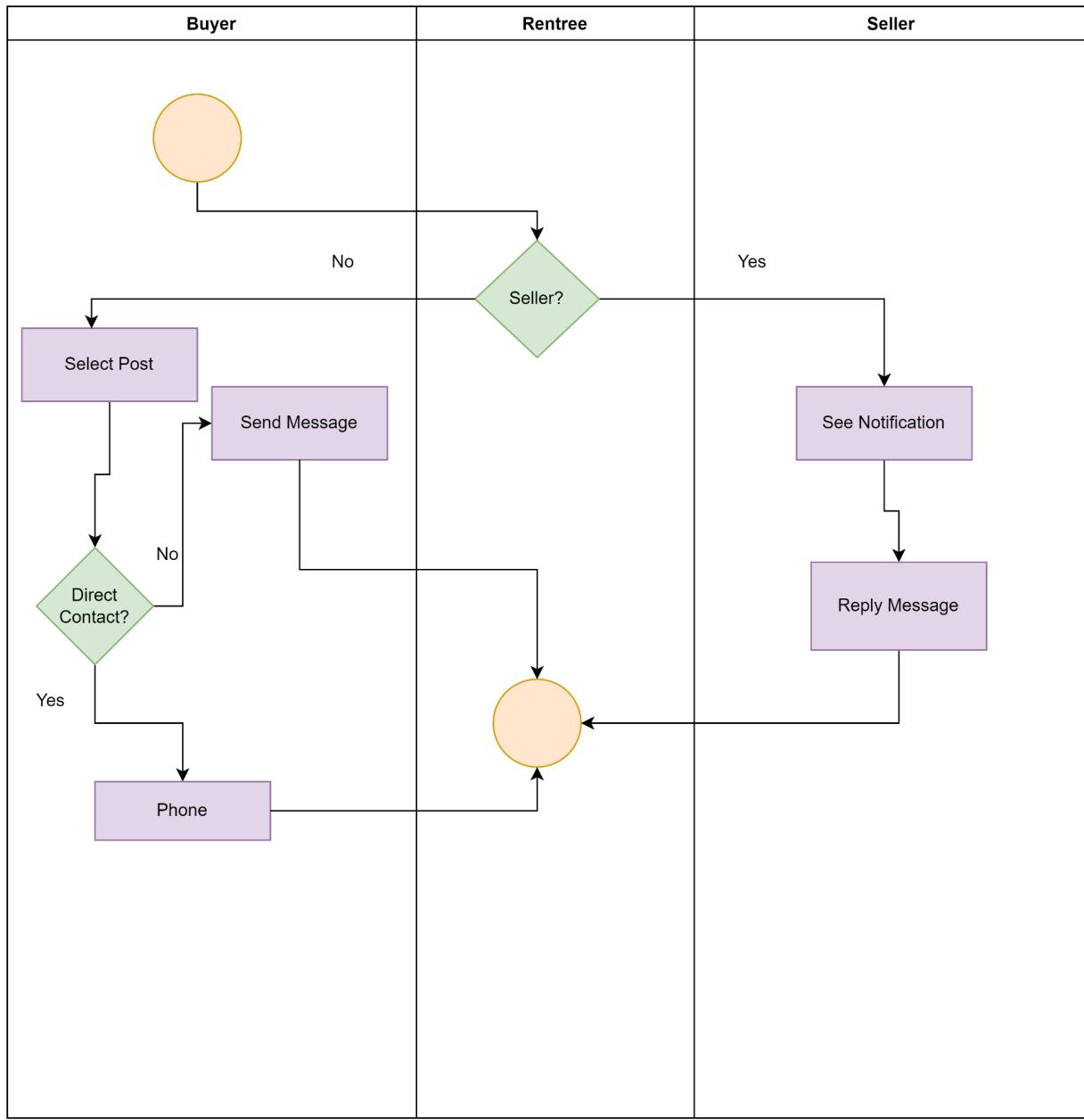
Reference: Use Case & Activity Level 1.1.3



## Swimlane ID (SID) 1.1.4

Name: Chatbox

Reference: Use Case & Activity Level 1.1.4



# **Chapter-6: Data-Based Modelling**

## **Data modeling concept:**

If software requirements include the necessity to create, extend or interact with a database or complex data structures need to be constructed and manipulated, then the software team chooses to create data models as part of overall requirements modeling. The entity-relationship diagram (ERD) defines all data objects that are processed within the system, the relationships between the data objects, and the information about how the data objects are entered, stored, transformed, and produced within the system.

## **Data objects:**

A data object is a representation of composite information that must be understood by the software. Here, composite information means information that has a number of different properties or attributes. A data object can be an external entity, a thing, an occurrence, a role, an organizational unit, a place or a structure.

## **Data Object Identification:**

<b>Sl</b>	<b>Nouns</b>	<b>Problem/ Solution Space</b>	<b>Attributes</b>
1.	Account	s	2,9,10,11,12,13,14,15,16,17,20,25
2.	Account-id	s	
3.	User	p	
4.	Buyer	p	2,9,10,11,12,13,14,15,16,17,20,25
5.	Seller	p	2,9,10,11,12,13,14,15,16,17,20,25
6.	Software	p	
7.	Software management	p	
8.	information	p	

9.	user-roles	s	
10.	Full Name	s	
11.	Contact	s	
12.	Temporary Address	s	
13.	Permanent Address	s	
14.	NID No	s	
15.	Photo	s	
16.	Email Address	s	
17.	Password	s	
18.	profile	p	
19.	change	p	
20..	username	s	
21.	Customer	s	2,10,11,12,13,14,15,16,17,20,25
22..	Post	s	2,24,36,38,39,44,46,47,48
23.	Post Details	s	
24.	Post ID	s	
25.	ID	s	
26.	details	p	
27.	Room quantity	s	
28.	Seat per room	s	
29.	Area	s	
30.	width	s	
31.	height	s	
32.	Location	s	

33.	Price	s	
34.	Pictures	s	
35.	Videos	s	
36.	Post Type	s	
37.	Search	p	
38.	Phone	p	
39.	Number	p	
40.	sms	s	
41.	OTP	s	
42.	status	s	
43.	confirmed	p	
44.	Date	s	
45.	website	p	
46.	Hostel	s	28,29,30,31,32,33,34,35
47.	Rent	s	27,29,30,31,32,33,34,35
48.	House	s	27,29,30,31,32,33,34,35
49.	time	p	
50.	Seller name	s	
51.	Buyer name	s	
52.	update	p	
53.	app	p	
54.	access	p	
55.	stakeholders	p	
56.	growth	p	

57.	data	p	
58.	cloud	s	
59.	Recovery	p	
60.	chat	p	4,5,62,63
61.	notification	s	2,4,5,24,62
62.	message	s	
63.	Reply	s	

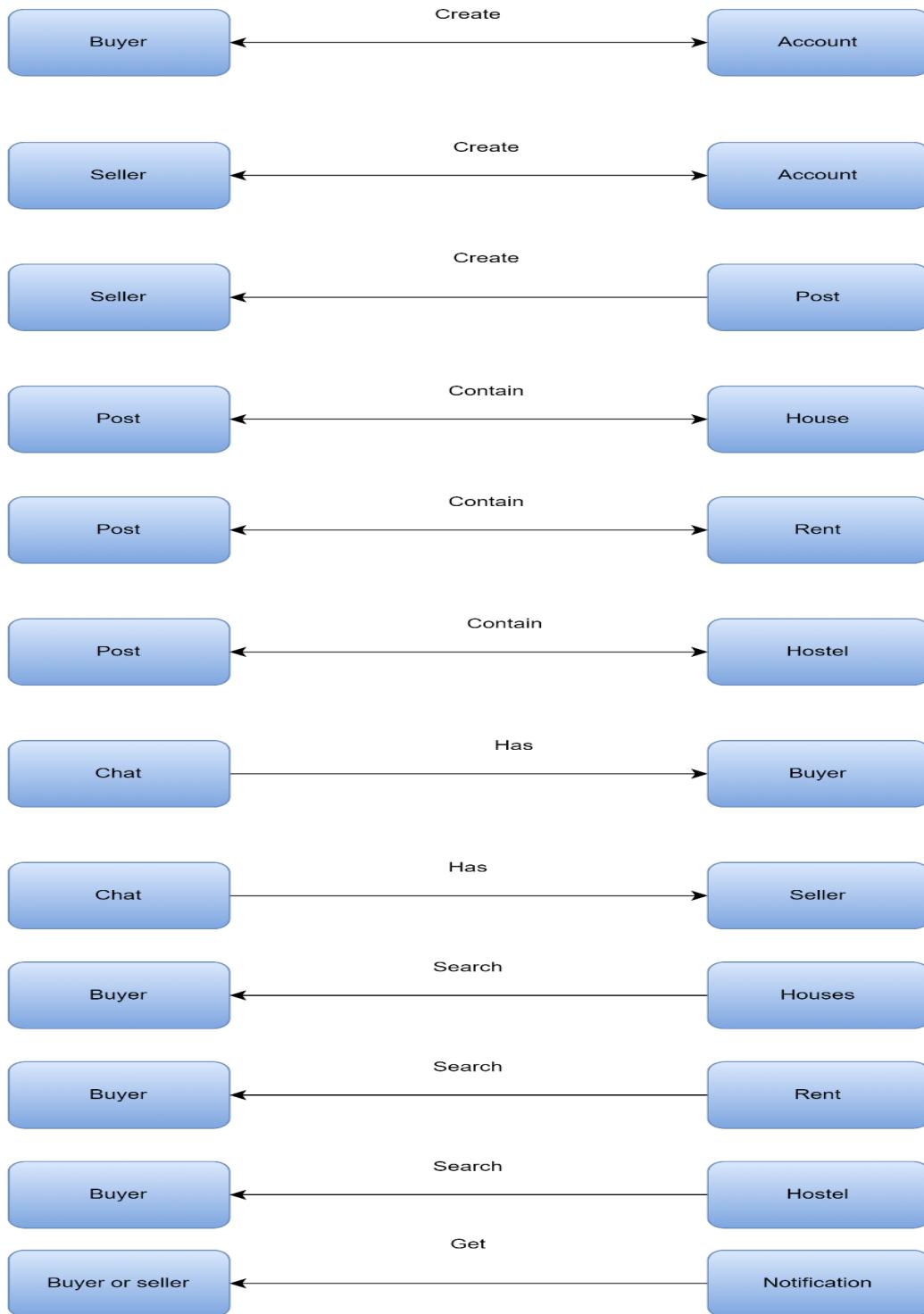
### **Final Data Objects:**

1. Buyer
2. Seller
3. Post
4. Hostel
5. House
6. Rent
7. Chat
8. Notification

### **Data Object Relationship:**

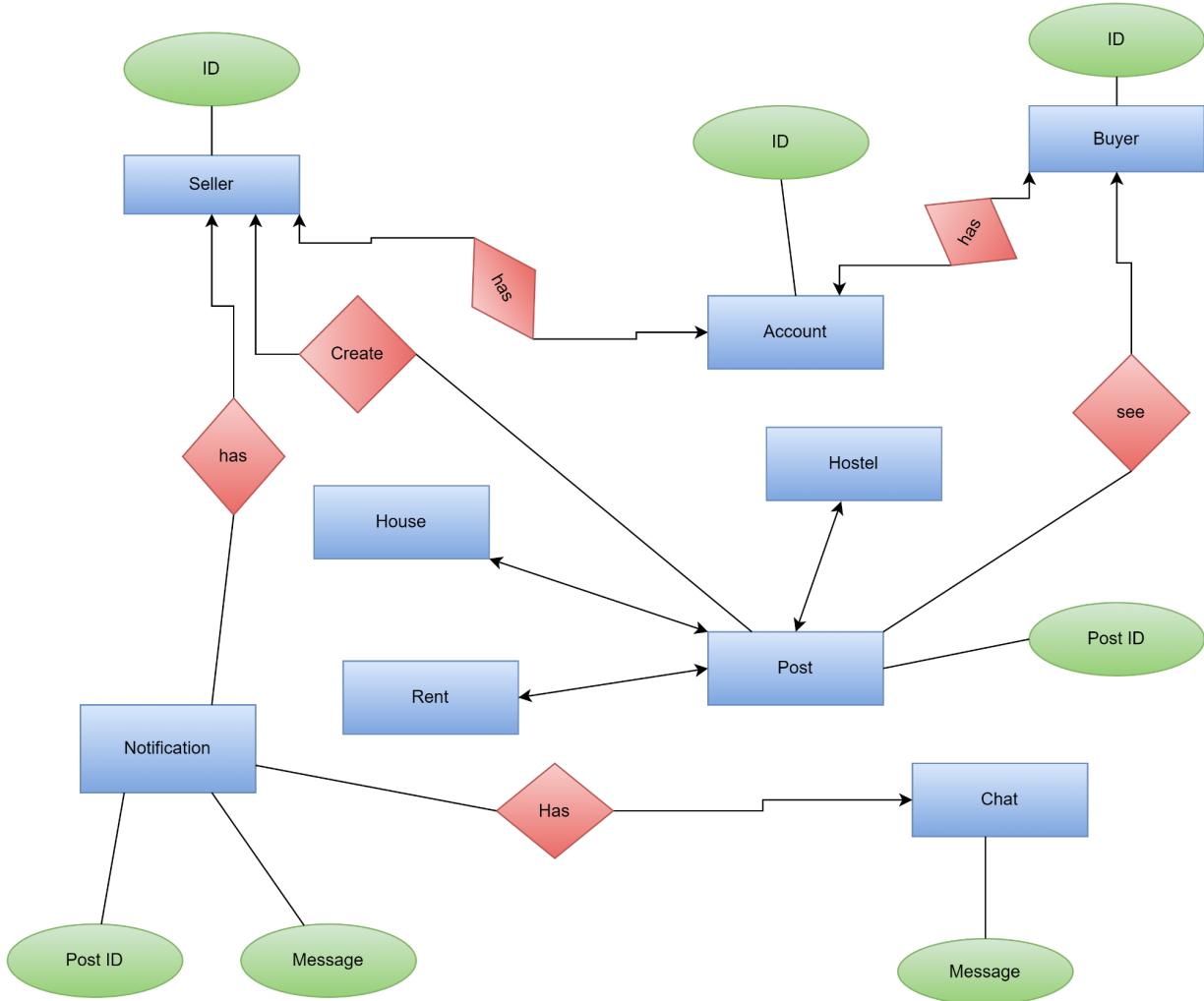
1. Buyer can create an account ( one to one relationship)
2. The seller can create an account ( one to one relationship)
3. The seller can create many posts ( one to many relationship)
4. Post can contain house information (one to one relationship)
5. Post can contain rent information ( one to one relationship)
6. Post can contain hostel information ( one to one relationship)
7. Chat has messages of the buyer ( Many to one relationship)
8. Chat has reply of Seller ( Many to one relationship)
9. Buyers can search for houses ( one to many relationship)

10. Buyers can search for rent ( one to many relationship)
11. Buyers can search for hostel ( one to many relationship)
12. Buyer or Seller has notification of messages (one to many )



## Definition of ER Diagram

An entity-relationship diagram (ERD) shows the relationships of entity sets stored in a database. By defining the entities, their attributes, and showing the relationships between them, an ER diagram illustrates the logical structure of databases. ER diagrams are used to sketch out the design of a database.



## Schema Diagram :

Data Object	Attributes	Type	Size
Buyer	-account_id -buyer_id -user_roles -full_name -contact -address -nid_no -photo -email_address -password	VARCHAR VARCHAR VARCHAR VARCHAR VARCHAR VARCHAR VARCHAR VARCHAR VARCHAR Text	50 50 50 50 50 50 50 50 50
Seller	-account_id -seller_id -user_roles -full_name -contact -address -nid_no -photo -email_address -password	VARCHAR VARCHAR VARCHAR VARCHAR VARCHAR VARCHAR VARCHAR VARCHAR VARCHAR Text	50 50 50 50 50 50 50 50 50
Post	-account_id -post_id -post_type -phone -date -house -rent -hostel	VARCHAR VARCHAR VARCHAR VARCHAR VARCHAR VARCHAR	50 50 50 50 50 50
House	-room_quantity -area -width	VARCHAR VARCHAR VARCHAR	50 50 50

	<ul style="list-style-type: none"> <li>-length</li> <li>-location</li> <li>-price</li> <li>-pictures</li> <li>-videos</li> </ul>	Varchar Varchar	50 50
Rent	<ul style="list-style-type: none"> <li>-room_quantity</li> <li>-area</li> <li>-width</li> <li>-length</li> <li>-location</li> <li>-price</li> <li>-pictures</li> <li>-videos</li> </ul>	Varchar Varchar Varchar Varchar Varchar	50 50 50 50 50
Hostel	<ul style="list-style-type: none"> <li>-seat per room</li> <li>-area</li> <li>-width</li> <li>-length</li> <li>-location</li> <li>-price</li> <li>-pictures</li> <li>-videos</li> </ul>	Varchar Varchar Varchar Varchar Varchar	50 50 50 50 50
Notification	<ul style="list-style-type: none"> <li>-account_id</li> <li>-post_id</li> <li>-buyer_id</li> <li>-seller_id</li> <li>-message</li> </ul>	Varchar Varchar Varchar Varchar varchar	50 50 50 50 50
Chat	<ul style="list-style-type: none"> <li>-buyer_id</li> <li>-seller_id</li> <li>-message</li> </ul>	Varchar Varchar varchar	50 50 50

# Chapter 7: Class-Based Modeling for Renter

## CLASS-BASED MODELING CONCEPT

This chapter is intended to describe the class-based modeling of Renter. Class-based modeling represents the objects that the system will manipulate, the operations that will be applied to the objects, relationships between the objects, and the collaborations that occur between the classes that are defined.

### Potential nouns to be class:

Sl	Nouns	Sl	Nouns
1.	Account	15.	Photo
2.	Account-id	16.	Email Address
3.	Seller	17.	Password
4.	Buyer	18.	profile
5.	Buyer Id	19.	change
6.	Seller Id	20..	username
7.	Post	21.	Post ID
8.	Post Details	22..	ID
9.	user-roles	23.	Room quantity
10.	Full Name	24.	House
11.	Contact	25.	time
12.	Temporary Address	26.	update
13.	Permanent Address	27.	stakeholders
14.	NID No	28.	details

<b>Sl</b>	<b>Nouns</b>	<b>Sl</b>	<b>Nouns</b>
29.	Area		
30.	width		
31.	height		
32.	Location		
33.	Price		
34.	Pictures		
35.	Videos		
36.	Post Type		
37.	Search		
38.	Phone		
39.	Rent		
40.	Hostel		
41.	OTP		
42.	SMS		
43.	confirmation		
44.	Date		
45.	Recovery		
46.	cloud		
47.	data		

### **List of verbs :**

<b>Sl</b>	<b>Verb</b>	<b>Sl</b>	<b>Verb</b>
1.	Create Account	14.	Search by Amount
2.	Delete Account	15.	Search by Room quantity
3.	Edit Account	16.	Update post
4.	Generate Account ID	17.	View history
5.	Generate buyer ID	18.	Backup data
6.	Generate seller ID	19.	Recover data
7.	Create Post	20..	See post
8.	Delete post	21.	login
9.	Generate Post ID	22..	logout
10.	Verify Account	23.	Insert pictures
11.	Notify Message	24.	Insert videos
12.	Account reset	25.	Select Post type
13.	Search by location	26.	reply

## General classification

Candidate classes were then characterized in seven general classes. The seven general characteristics are as follows:

1. External entities
2. Things
3. Events
4. Roles
5. Organizational units
6. Places
7. Structures

<b>SI</b>	<b>Noun</b>	<b>General Classifications</b>
1.	Account	5,7
2.	Account-id	2
3.	Seller	4,5,7
4.	Buyer	4,5,7
5.	Buyer Id	2
6.	Seller Id	2
7.	Post	2,3,7
8.	Post Details	2
9.	user-roles	2
10.	Full Name	2
11.	Contact	2
12.	Temporary Address	2
13.	Permanent Address	2
14.	NID No	2
15.	Photo	2
16.	Email Address	2
17.	Password	2
18.	profile	2
19.	change	2
20.	username	2
21.	Post ID	2

22..	ID	<b>2</b>
23.	Room quantity	<b>2</b>
24.	House	<b>2,7</b>
25.	time	<b>2</b>
26.	update	<b>2</b>
27.	stakeholders	<b>2</b>
28.	details	<b>2</b>
29.	Area	<b>2</b>
30.	width	<b>2</b>
31.	height	<b>2</b>
32.	Location	<b>2</b>
33.	Price	<b>2</b>
34.	Pictures	<b>2</b>
35.	Videos	<b>2</b>
36.	Post Type	<b>2</b>
37.	Search	<b>3</b>
38.	Phone	<b>2</b>
39.	Rent	<b>2,7</b>
40.	Hostel	<b>2,7</b>
41.	OTP	<b>1</b>
42.	SMS	<b>1</b>
43.	confirmation	<b>1</b>
44.	Date	<b>2</b>

45.	Recovery	<b>3</b>
46.	cloud	<b>1</b>
47.	data	<b>5</b>

## Selection Criteria

The candidate classes are then selected as classes by six Selection Criteria. A candidate class generally becomes a class when it fulfills around three Characteristics.

1. Retain information
2. Needed services
3. Multiple attributes
4. Common attributes
5. Common operations
6. Essential requirements

Potential general classified nouns to become a class after selection criteria :

<b>SI</b>	<b>Noun</b>	<b>Selection Criteria</b>
1.	Account	<b>1,2,3,4,5,6(selected)</b>
2.	OTP	<b>2,6</b>
3.	Seller	<b>1,2,3,4,5,6(selected)</b>
4.	Buyer	<b>1,2,3,4,5,6(selected)</b>

5.	Post details	<b>3,4</b>
6.	Cloud	<b>2</b>
7.	Post	<b>1,3,4,5,6(selected)</b>
8.	Rent	<b>1,3,4,5(selected)</b>
9.	House	<b>1,3,4,5(selected)</b>
10.	Hostel	<b>1,3,4,5(selected)</b>
11.	chat	<b>2,6(selected)</b>
12.	email	<b>2,5,6(selected)</b>
13.	Notificaion	<b>2,5,6(selected)</b>
14.	search	<b>1,2,3,6(selected)</b>

### **Selected Classes:**

1. Account
2. Seller
3. Buyer
4. Notificaton
5. Chat
6. Search
7. Post
8. House
9. Hostel
10. Rent
11. SMS

### **Attribute & Method Identification:**

<b>Class Name</b>	<b>Attributes</b>	<b>Method</b>
Account	-accountID	+create_account()

	<ul style="list-style-type: none"> <li>-FullName</li> <li>-username</li> <li>-emailAddress</li> <li>-phoneNumber</li> <li>-password</li> <li>-address</li> <li>-photo</li> <li>-NID</li> <li>-Account type</li> </ul>	<ul style="list-style-type: none"> <li>+view_account_info()</li> <li>+recover_password()</li> <li>+update_account()</li> <li>-generate_ID()</li> </ul>
Seller	<ul style="list-style-type: none"> <li>-user role</li> <li>-seller_id</li> </ul>	+view-info()
Buyer	<ul style="list-style-type: none"> <li>-user role</li> <li>-buyer_id</li> </ul>	+view-info()
Notification	<ul style="list-style-type: none"> <li>-notice</li> <li>-id</li> </ul>	<ul style="list-style-type: none"> <li>+view()</li> <li>+delete()</li> </ul>
Chat	<ul style="list-style-type: none"> <li>-id</li> <li>-post_id</li> <li>-buyer_id</li> <li>-seller_id</li> </ul>	<ul style="list-style-type: none"> <li>+view()</li> <li>+send_msg()</li> </ul>
Search	<ul style="list-style-type: none"> <li>-location</li> <li>-room_quantity</li> <li>-price</li> <li>-posts</li> </ul>	<ul style="list-style-type: none"> <li>+show_result()</li> <li>+set_location()</li> <li>+set_room_quantity()</li> <li>+set_price()</li> </ul>
Post	<ul style="list-style-type: none"> <li>-account_id</li> <li>-post_id</li> <li>-post_type</li> <li>-phone</li> <li>-date</li> <li>-house</li> <li>-rent</li> <li>-hostel</li> </ul>	<ul style="list-style-type: none"> <li>+create_post()</li> <li>+delete_post()</li> <li>+update_post()</li> </ul>
Rent	<ul style="list-style-type: none"> <li>-room_quantity</li> <li>-area</li> <li>-width</li> <li>-length</li> </ul>	<ul style="list-style-type: none"> <li>+get_info()</li> <li>+set_info()</li> </ul>

	<ul style="list-style-type: none"> <li>-location</li> <li>-price</li> <li>-pictures</li> <li>-videos</li> <li>-type</li> </ul>	
House	<ul style="list-style-type: none"> <li>-room_quantity</li> <li>-area</li> <li>-width</li> <li>-length</li> <li>-location</li> <li>-price</li> <li>-pictures</li> <li>-videos</li> <li>-type</li> </ul>	<ul style="list-style-type: none"> <li>+get_info()</li> <li>+set_info()</li> </ul>
Hostel	<ul style="list-style-type: none"> <li>-seat per room</li> <li>-area</li> <li>-width</li> <li>-length</li> <li>-location</li> <li>-price</li> <li>-pictures</li> <li>-videos</li> <li>-type</li> </ul>	
SMS	-OTP	+send-OTP()

## CRC Card:

Class	Responsibility	Collaborator
Account	Creating user accounts Updating account information	user
Seller	Creating Seller Id Updating account information	seller

Buyer	Creating Buyer Id Updating account information	buyer
Notification	Creates notice Updates notice Deletes notice Sends mail	Database User
Chat	Contact with seller Send msg to seller Reply to buyer	Database user phone
Search	Find desired posts Sort post	Database User
Post	Create Post Delete Post Update post	Seller Database House Rent Hostel
Rent	Stores Rent information	Seller
House	Stores House information	Seller
Hostel	Stores Hostel information	Seller
SMS	Verify account by OTP	System

## Class Cards

Account	
Attributes	Method
-accountID -FullName -username -emailAddress	+create_account() +view_account_info() +recover_password()

-phoneNumber -password -address -photo -NID -Account type	+update_account() -generate_ID()
<b>Responsibilities</b>	<b>Collaborator</b>
Creating user accounts Updating account information	user

<b>Seller</b>	
<b>Attributes</b>	<b>Method</b>
-user role -seller_id	+view-info()
<b>Responsibilities</b>	<b>Collaborator</b>
Creating Seller Id Updating account information	seller

<b>Buyer</b>	
<b>Attributes</b>	<b>Method</b>
-user role -buyer_id	+view-info()
<b>Responsibilities</b>	<b>Collaborator</b>
Creating Buyer Id Updating account information	buyer

<b>Notification</b>	
<b>Attributes</b>	<b>Method</b>
-notice -id	+view() +delete()

<b>Responsibilities</b>	<b>Collaborator</b>
Creates notice Updates notice Deletes notice Sends mail	Database User

<b>Chat</b>	
<b>Attributes</b>	<b>Method</b>
-id -post_id -buyer_id -seller_id	+view() +send_msg()
<b>Responsibilities</b>	<b>Collaborator</b>
Contact with seller Send msg to seller Reply to buyer	Database user phone

<b>Search</b>	
<b>Attributes</b>	<b>Method</b>
-location -room_quantity -price -posts	+show_result() +set_location() +set_room_quantity() +set_price()
<b>Responsibilities</b>	<b>Collaborator</b>
Find desired posts Sort post	Database User

<b>Post</b>	
<b>Attributes</b>	<b>Method</b>

<ul style="list-style-type: none"> <li>-account_id</li> <li>-post_id</li> <li>-post_type</li> <li>-phone</li> <li>-date</li> <li>-house</li> <li>-rent</li> <li>-hostel</li> </ul>	<ul style="list-style-type: none"> <li>+create_post()</li> <li>+delete_post()</li> <li>+update_post()</li> </ul>
<b>Responsibilities</b>	<b>Collaborator</b>
Create Post Delete Post Update post	Seller Database House Rent Hostel

<b>Rent</b>	
<b>Attributes</b>	<b>Method</b>
<ul style="list-style-type: none"> <li>-room_quantity</li> <li>-area</li> <li>-width</li> <li>-length</li> <li>-location</li> <li>-price</li> <li>-pictures</li> <li>-videos</li> </ul>	<ul style="list-style-type: none"> <li>+get_info()</li> <li>+set_info()</li> </ul>
<b>Responsibilities</b>	<b>Collaborator</b>
Stores Rent information	Seller

<b>House</b>	
<b>Attributes</b>	<b>Method</b>
-room_quantity	+get_info()

-area -width -length -location -price -pictures -videos -Type	+set_info()
<b>Responsibilities</b>	<b>Collaborator</b>
Stores House information	Seller

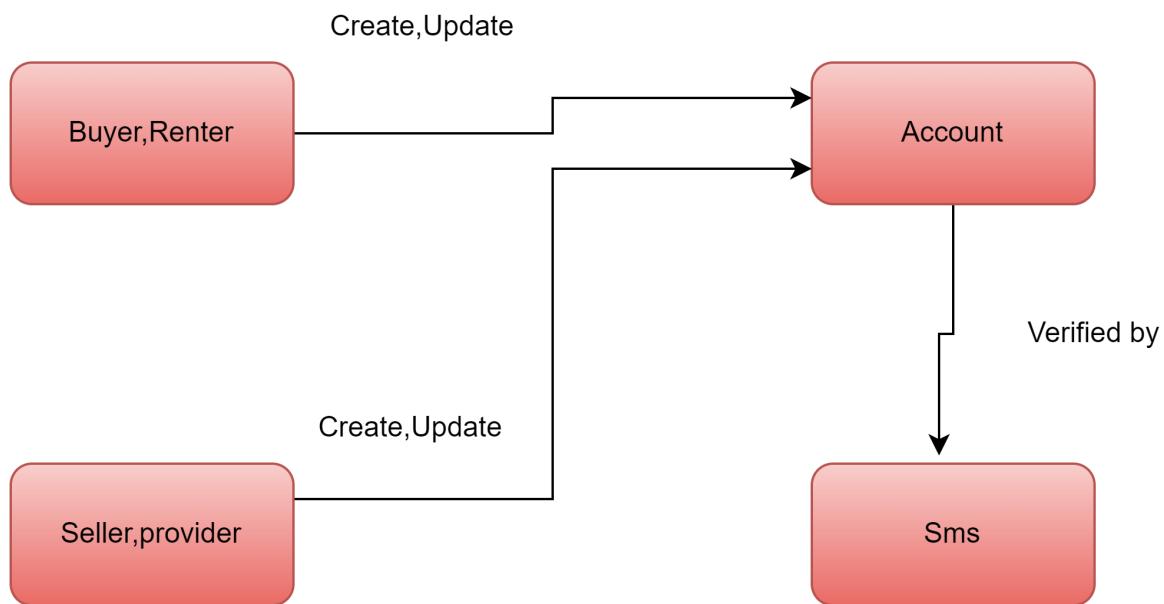
<b>Hostel</b>	
<b>Attributes</b>	<b>Method</b>
-seat per room -area -width -length -location -price -pictures -videos -type	+get_info() +set_info()
<b>Responsibilities</b>	<b>Collaborator</b>
Stores Hostel information	Seller

<b>SMS</b>	
<b>Attributes</b>	<b>Method</b>
-OTP	+send-OTP()
<b>Responsibilities</b>	<b>Collaborator</b>
Verify account by OTP	System

# CRC Diagram

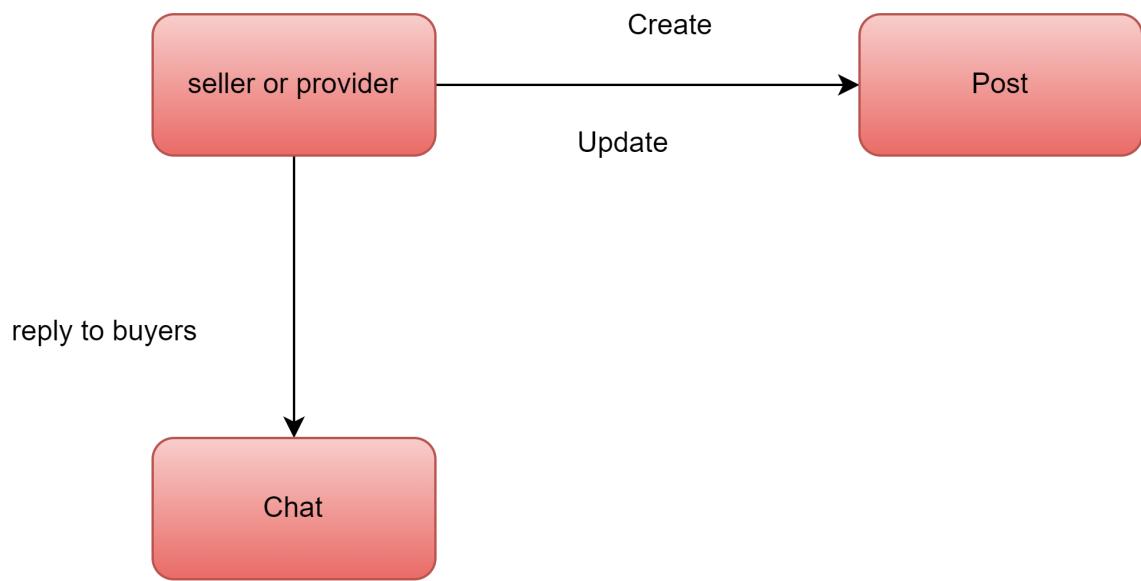
**Diagram ID: 1**

**Name:Account**



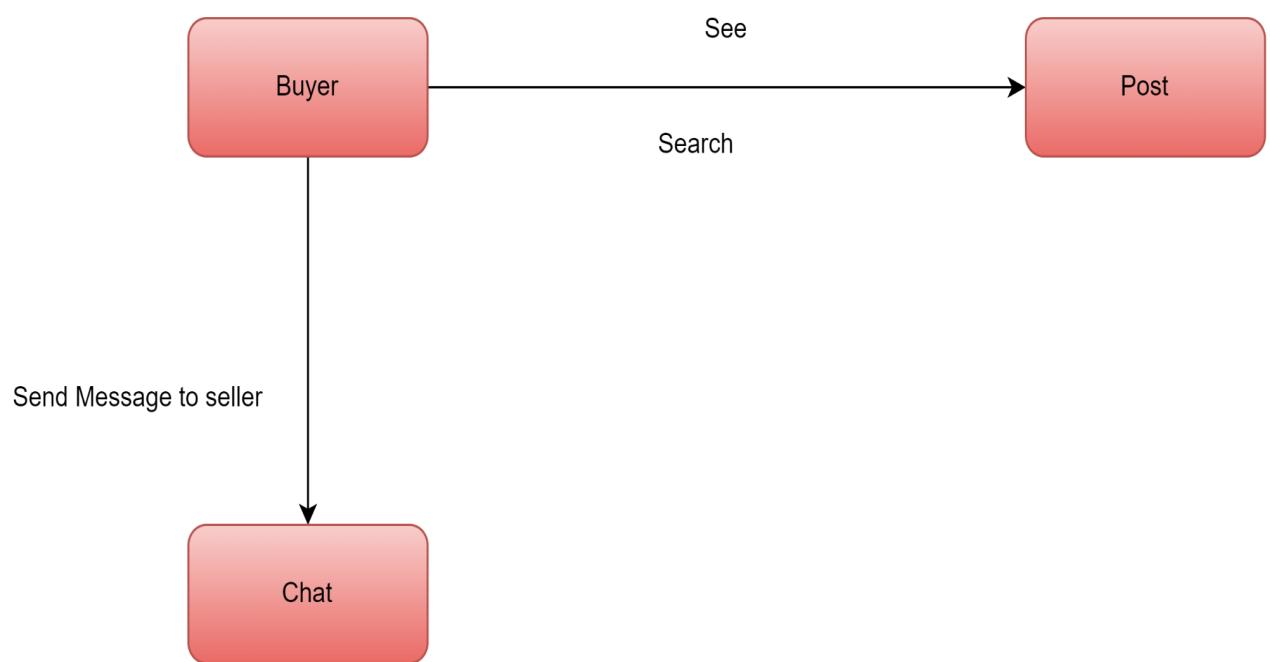
**Diagram ID: 2**

**Name:Seller**



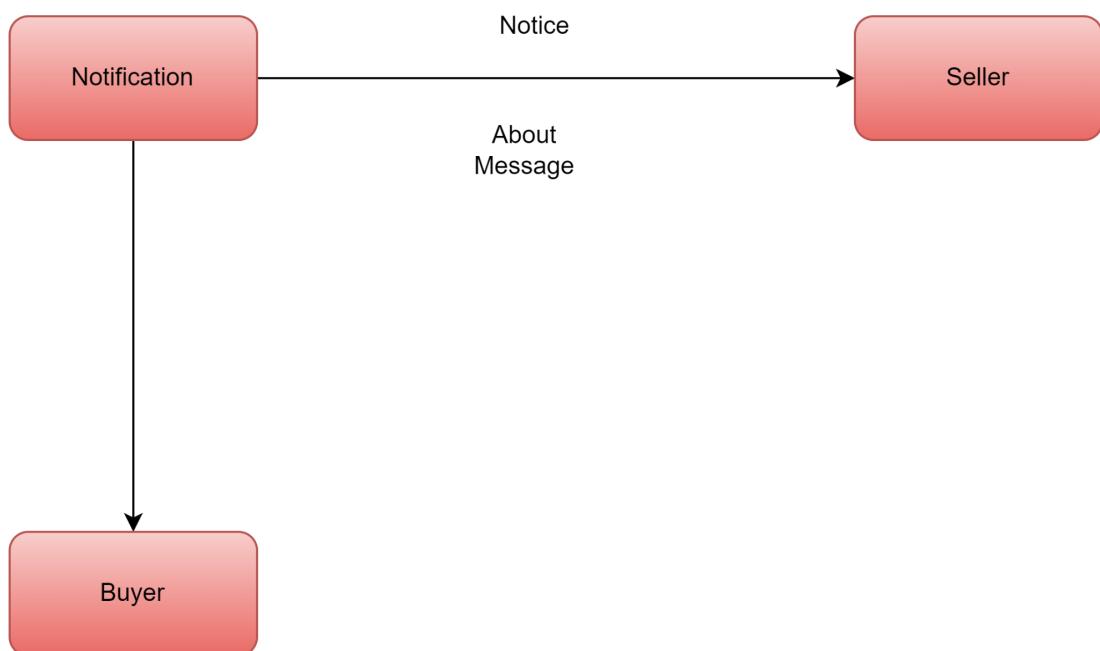
**Diagram ID: 3**

**Name:Buyer**



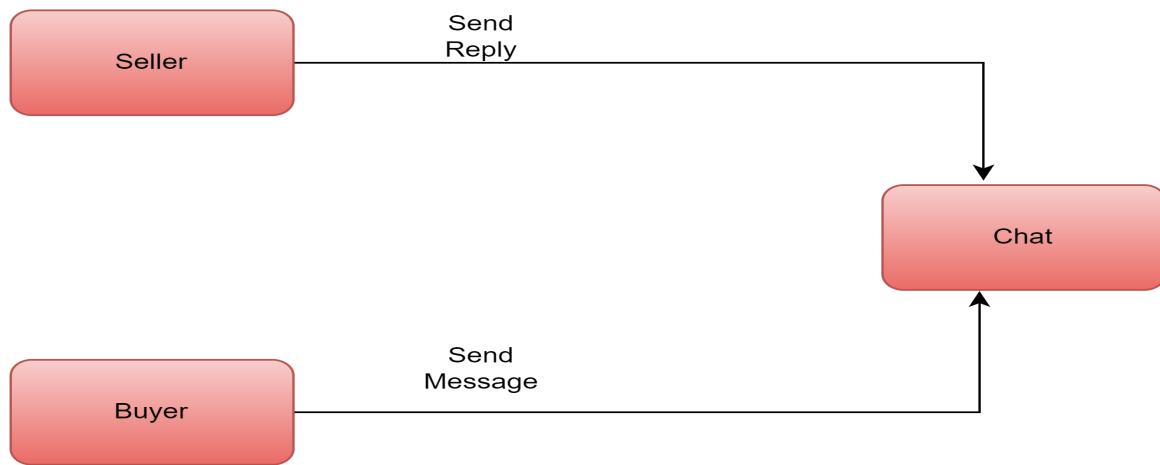
**Diagram ID: 4**

**Name:Notification**



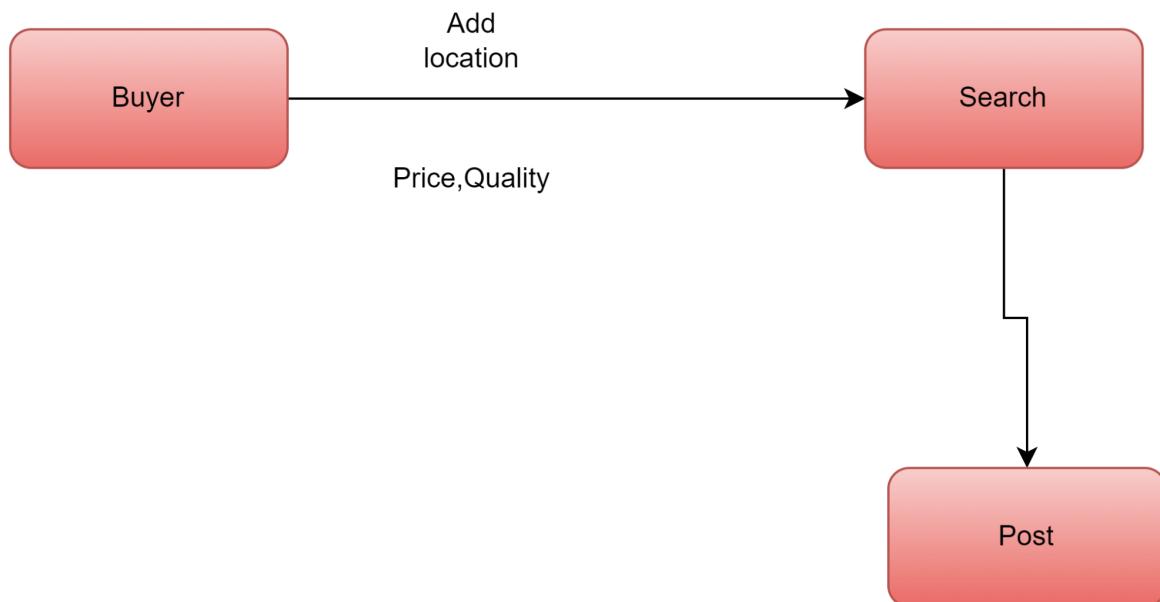
**Diagram ID: 5**

**Name:Chat**



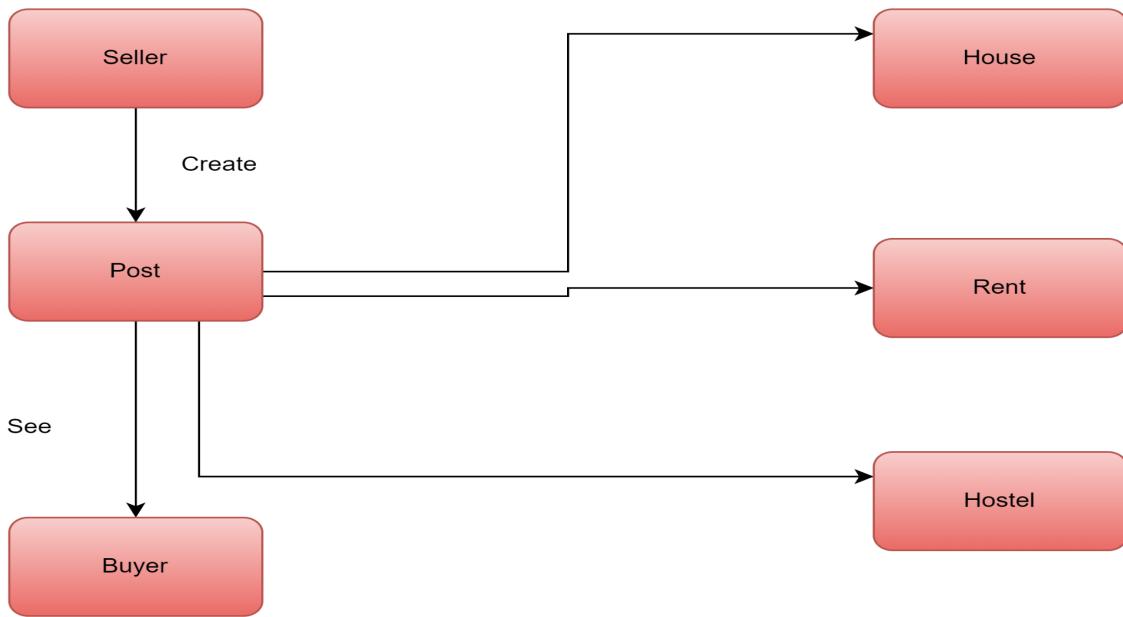
**Diagram ID: 6**

**Name:Search**



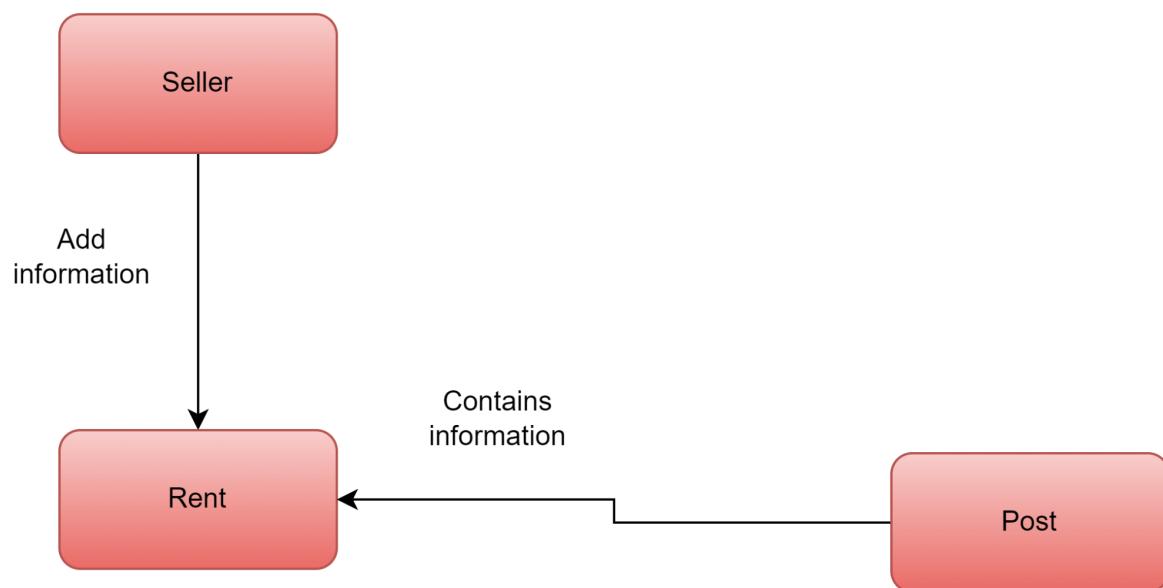
**Diagram ID: 7**

**Name:Post**



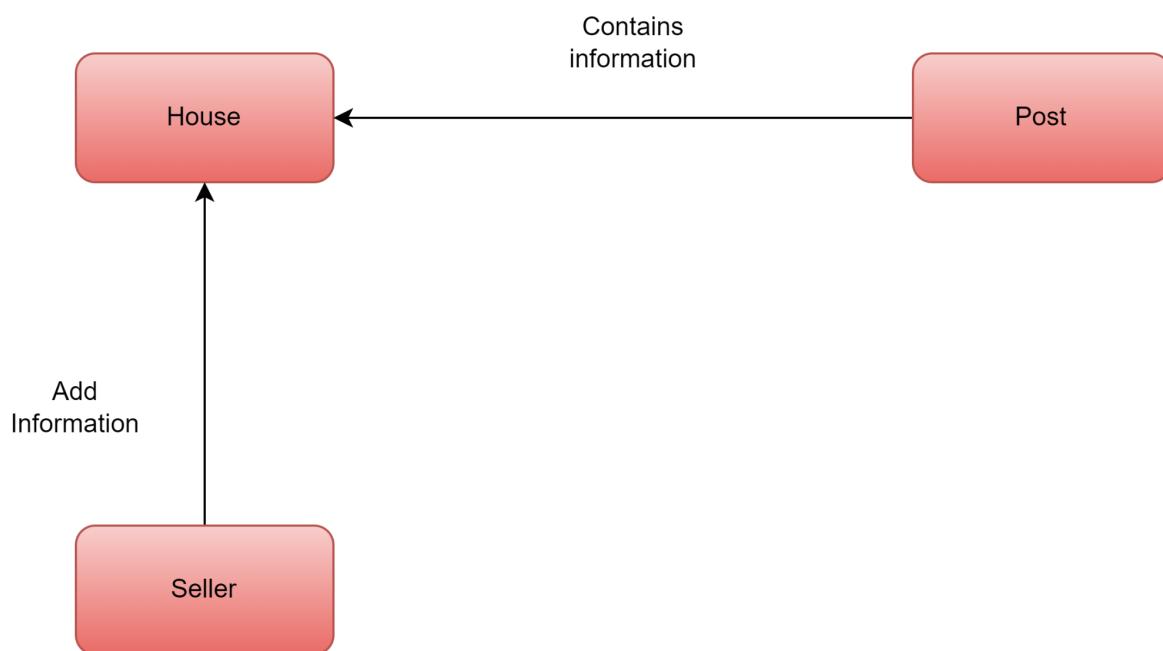
**Diagram ID: 8**

**Name:Rent**



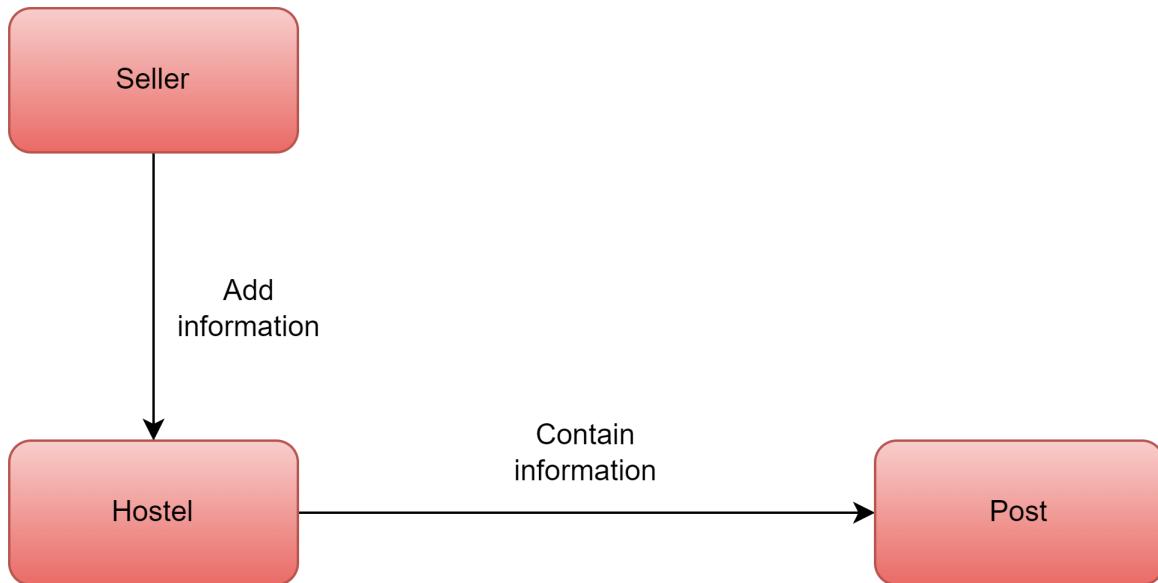
**Diagram ID: 9**

**Name:House**



**Diagram ID: 10**

**Name:Hostel**



**Diagram ID: 11**

**Name:Verification**



# Chapter 8: Behavioral Modeling

## Concept of Behavioral Modeling

The behavioral model indicates how software will respond to external events or stimuli. In the context of behavioral modeling, two different characterizations of states must be considered: (1) the state of each class as the system performs its function and (2) the state of the system as observed from the outside as the system performs its function.

## State Transition Diagram

One component of a behavioral model is a UML state diagram that represents active states for each class and the events (triggers) that cause changes between these active states.

## Event Table

SL NO.	Event	State Name	Initiate	Collaborator	Associated method
1.	Will create an account	create_account	user	sms,email	+create_account() +verify_info() +notify_user() +send_confirmation()
2.	Will be able to update information	Update_info	user	email,sms	+update_info() +setFull_name() +setMobile_number() +setEmail_address() +set_permanentAddress() +set_photo()

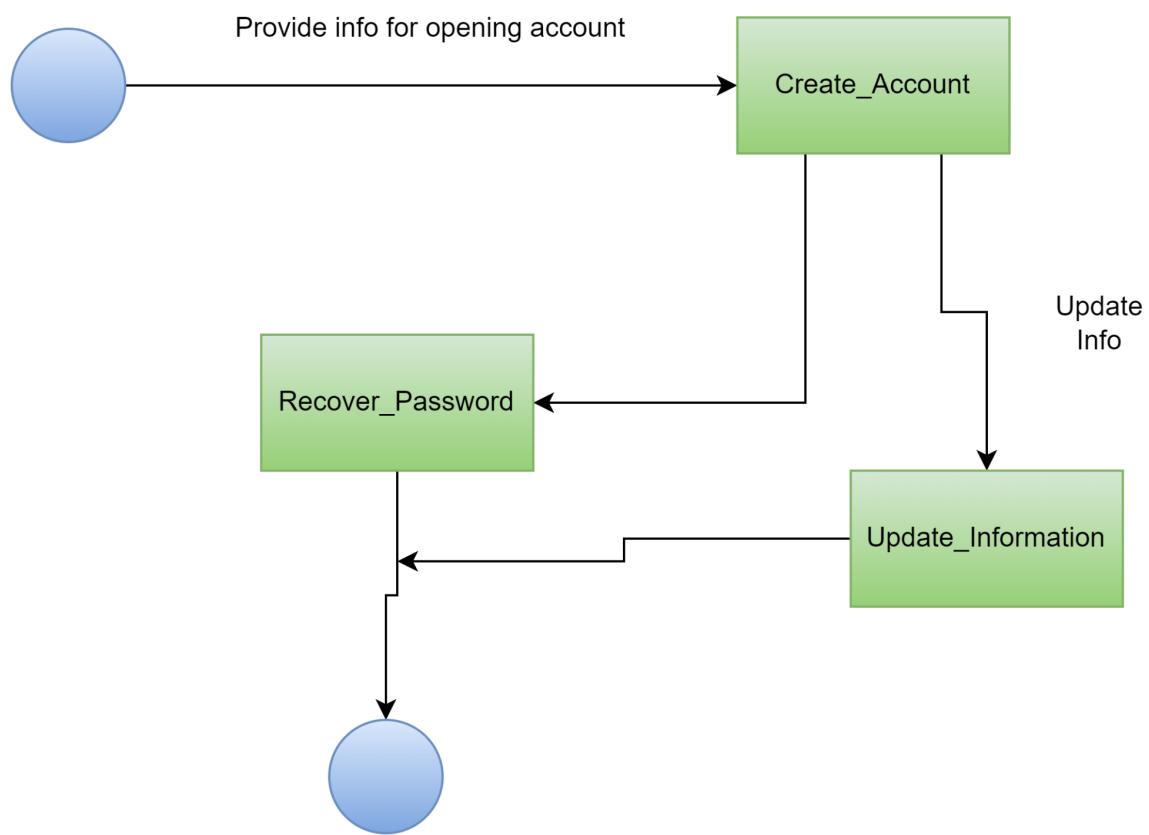
3.	Will be able to recover password	Recover_password	user	SMS, Email	+recover_password() +send_otp() +send_link()
4.	Will create a post	create_post	seller	post,house,rent,hostel	+create_post()
5.	Will search post	search_post	buyer	post,search	+show_result() +set_location() +set_room_quantity() +set_price()
6.	Will update a post	update_post	seller	post	+update_post()
7.	Will delete a post	delet_post	seller	post	+delete_post()
8.	Will send mesage	send_msg	buyer	post,chat,seller	+send_msg()
9.	Will reply message	reply	seller	post,chat,buyer	+reply_msg()
10.	Will receive notification	Receive_notification	seller,buyer	post,chat	+send_msg() +reply_msg() +view_notification()
11.	Generate buyer id	generate_buyer_id	buyer	account,user_role	+generate_id()
12.	Generate seller id	generate_seller_id	seller	account,user_role	+generate_id()
13.	Will add house details	add_house_info	seller	house	+set_room_quantity() +set_area() +set_width

					+set_length +set_location +set_price +set_pictures +set_videos +set_type
14.	Will add rent details	add_rent_in fo	seller	rent	+set_room_quantity() +set_area() +set_width +set_length +set_location +set_price +set_pictures +set_videos +set_type
15.	Will add hostel details	add_hostel_info	seller	house	+set_seat_per_room() +set_area() +set_width +set_length +set_location +set_price +set_pictures +set_videos +set_type

### State Transition Diagram

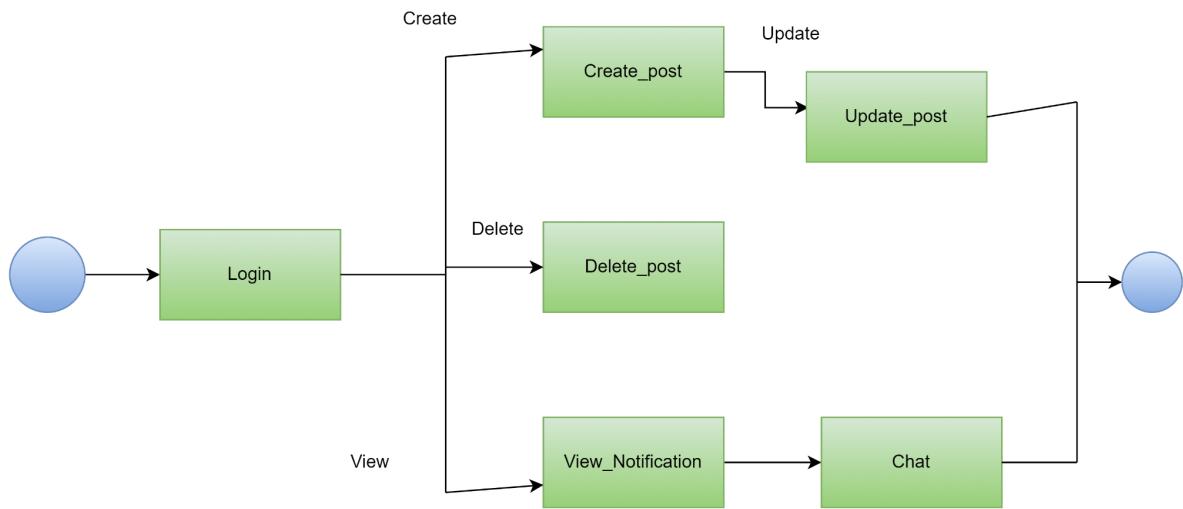
**ID :1**

**Name:User Account**



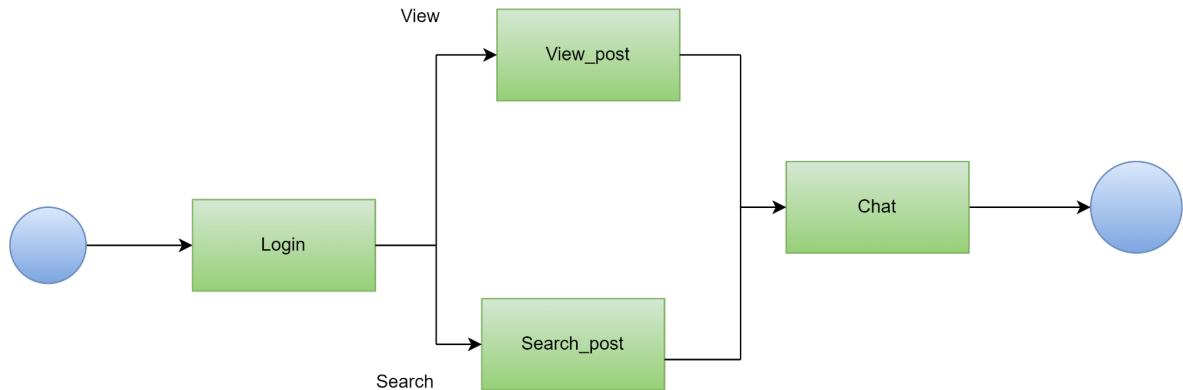
**ID :2**

**Name:Seller**



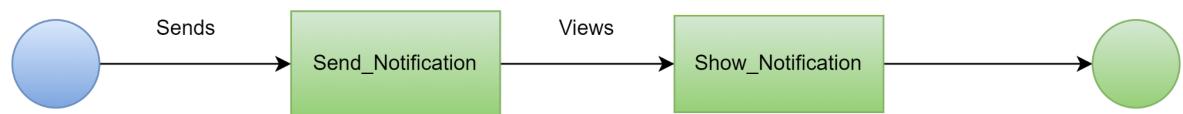
**ID:3**

**Name:Buyer**



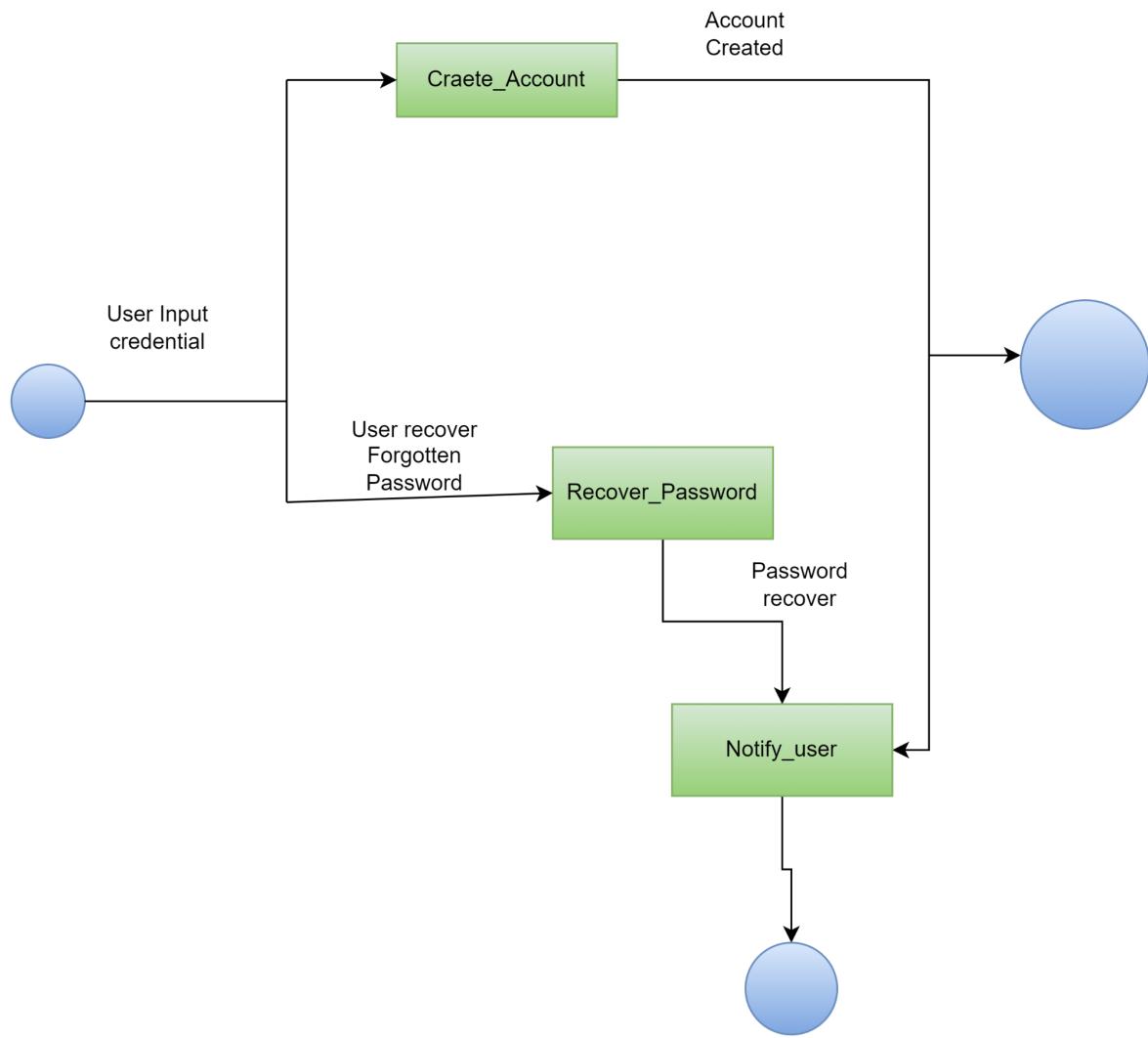
**ID:4**

**Name:Notification**



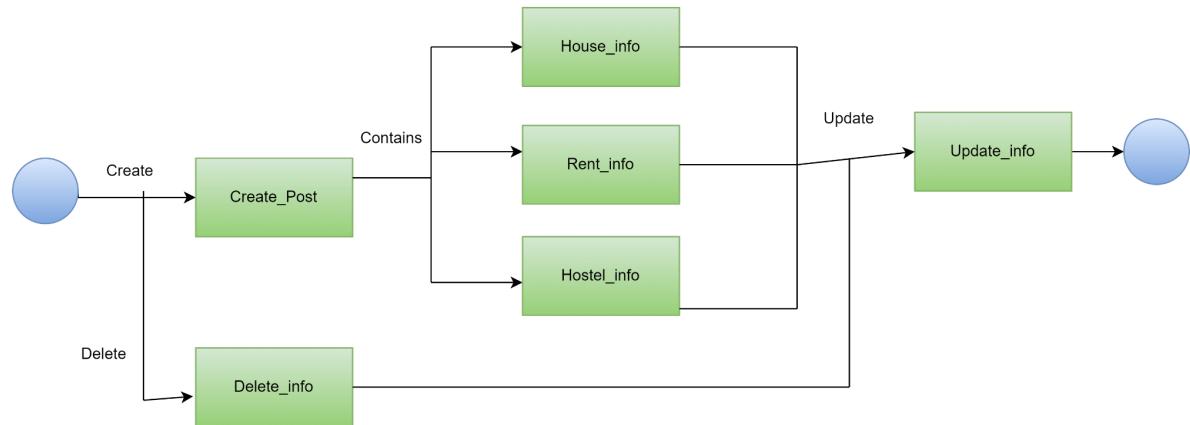
**ID:5**

**Name:SMS**



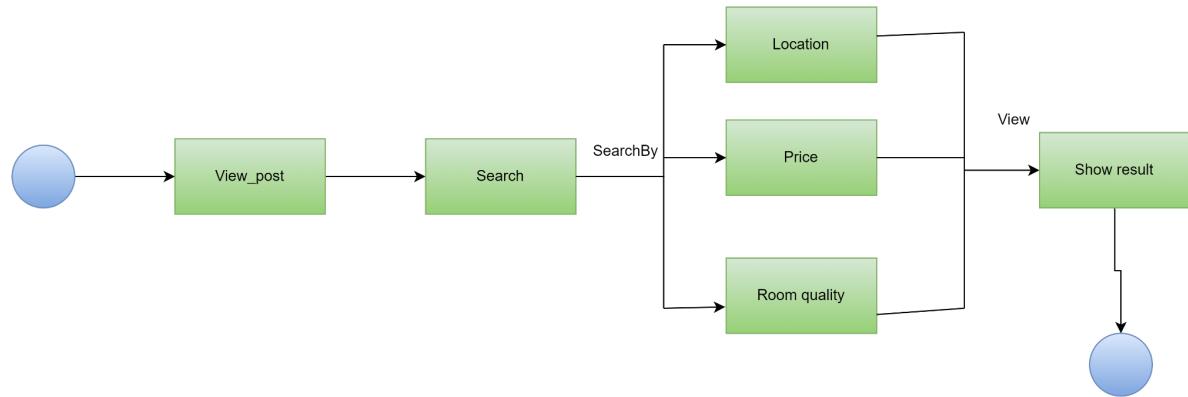
**ID:6**

**Name:Post**



**ID:7**

**Name:Search**



## Chapter 9: Sequence Diagram:

A sequence diagram is a Unified Modeling Language (UML) diagram that illustrates the sequence of messages between objects in an interaction

