Institute of Information Technology

University of Dhaka

Software Project Lab-2

Software Requirements Specification and Analysis Report

Rental Application

Submitted By

Md Shakibul Islam Shakib

Roll: 1404

&

Nandan Bhowmick

Roll: 1436

Supervised by:

Dr. Rezvi Shahariar

Assistant Professor

IIT,DU

(Supervisor's signature)

Submission Date: 09/12/2024

Index

Software Requirements Specification

Introduction

A house rental app connects property owners and renters, simplifying the rental process with property listings, area-based categorization, real-time chat, secure bookings, and rent prediction. Renters can browse, filter, and book properties effortlessly, while hosts manage listings efficiently. It ensures convenience, transparency, and trust for seamless housing solutions.

Purpose

The purpose of a house rental app is to streamline the process of finding and renting properties, making it accessible and efficient for both renters and property owners. It bridges the gap by offering features like detailed property listings, area-based categorization, real-time communication, and secure payment options. For renters, the app simplifies property search, enhances decision-making with clear descriptions and predictive pricing, and facilitates direct interactions with hosts. For owners, it provides an organized platform to showcase properties and connect with potential tenants. By ensuring transparency, trust, and convenience, the app aims to transform the traditional rental experience into a seamless digital journey.

Stakeholders

1.Renters

2.Landlords

Viewpoints:

Renters – Users who are searching for apartments or houses to rent. They interact with the application to find properties, contact landlords, and possibly book tours or send rental applications.

Landlords/Property Owners – Individuals or businesses that list their properties for rent on the platform. They manage property listings, respond to renter inquiries, and approve or decline rental applications.

QFD(Quality Function Deployment)
Normal:

- **User Registration & Profiles**: Users (landlords/renters) can register, log in, and create profiles with basic details.
- **Property Listings**: Landlords can list, update, and delete properties with details like location, price, bedrooms, availability, and images.Admin can delete if not feasible.
- **View Listings**: Landlords can view their added and reserved properties; renters can view their previous rentals and manage their wishlist.
- **Search by Area**: Users can choose specific areas of Dhaka city (e.g., Mohammadpur, Dhanmondi) and view available properties in that area.

- **Advanced Search Filters**: Renters can search properties based on preferences (location, price, bedrooms) and advanced filters (pet-friendly, furnished).
- **Property Details**: Renters can view detailed property information, including images, amenities, and landlord contact details.

Expected:

- **Sending Notification:**Give notification when a property is available.

Exciting:

- **In-platform chat**: Renters can send messages to the landlords through the platform to inquire about properties.
- **Integrated map:** An integrated map to allow users to view the exact location of the property, enhancing user experience.
- **Rent Calculator**: Users can input bedrooms,bathrooms,location etc
- **Property Visit Scheduling**: Renters can select a date from a calendar to schedule a property visit.

User Story

The rental market in Dhaka city presents significant challenges for both renters and landlords. With high demand, rapidly changing availability, and limited access to trustworthy information, finding an ideal rental property can be a daunting task. Renters often face difficulties in locating properties that match their budget, preferred location, and desired amenities. Additionally, the process of managing inquiries, scheduling visits, and communicating with landlords is often fragmented and inefficient. On the other hand, landlords face their own set of obstacles, such as showcasing properties effectively, building credibility, and responding promptly to tenant inquiries. These issues create friction in the rental process, making it time-consuming and stressful. Our application is designed to solve these challenges, offering a streamlined platform that improves the rental experience for both renters and landlords.

User Registration & Login:

- **Renter**: Renters create personalized accounts by registering with their basic details, such as full name, email, phone number, and a profile picture. If they forget the password, an OTP will be sent to their email which according to status they will have to enter within a certain timestamp for verification.
- **Landlord**: Landlords are initially registered as normal users/renters and are considered landlords when they add or host a property to rent in the application. There is no separate registration process for landlords. Profiles include options to upload verification documents for credibility, building trust with renters. A dashboard allows landlords to manage their listed properties, check tenant inquiries, and view analytics such as property views and engagement metrics. Landlords can login to the website the same way renters do.

Property Listings:

- Renter: Renters maintain a personalized list of properties, including their wishlist and rental history, with information like start and end dates, landlord contact details, and rent agreements.

- Landlord: Landlords create property listings with comprehensive details. These include location, price, property type, bedrooms and bathrooms, amenities, availability, and media. Landlords can edit, update, or delete listings directly through their dashboard.

View Listings:

- Renter: Renters maintain a personalized list of properties, including a wishlist for saving interesting properties they might consider in the future and a rental history for reference.
- Landlord: Landlords access a categorized view of their properties, including active listings, reserved listings, and rented out listings with tenant details and contract dates.

Search by Area:

- Renter: Renters can narrow their search by selecting specific areas of Dhaka, such as Mohammadpur, Dhanmondi, or Gulshan. The feature includes a dropdown or interactive map to refine the search further based on local landmarks, schools, or transport hubs.
- Landlord: Listings are categorized by location, ensuring that properties appear in search results when renters select specific neighborhoods. This improves discoverability for location-specific properties.

Advanced Search Filters:

- Renter: Renters use filters to streamline their property search, such as budget, property type, amenities, bedrooms and bathrooms, and proximity to specific landmarks or public transportation.
- Landlord: Tagged filters on listings ensure that properties meet specific renter needs, appearing in targeted search results.

Property Details:

- Both renters and landlords can view in-depth property details for various listings. These details include descriptions of amenities, rules (e.g., no smoking), number of bedrooms, bathrooms, and landlord profiles with contact information.

Rent Calculator:

- Renter: Renters can estimate monthly rent by entering property preferences such as the number of bedrooms and bathrooms, preferred area, and desired amenities. The calculator provides an estimated monthly rent based on market trends, helping renters plan their budget.
- Landlord: Landlords can use the calculator to get an idea of how much to ask for rent, ensuring their property is competitively priced based on current market trends and renter expectations.

Property Visit Scheduling:

- Renter: Renters can select a date and time from a provided calendar and time picker respectively to schedule a property visit. This feature ensures that renters can plan visits smoothly and stay informed about their upcoming appointments.
- Landlord: Landlords receive a detailed email notification whenever a renter schedules a visit to one of their properties. This helps landlords stay aware of upcoming visits and manage their property viewings efficiently.

Sending Notifications:

- Landlord: Landlords receive email notifications when renters add a property to their wishlist or schedules/books a date to visit the property.

In-Platform Chat:

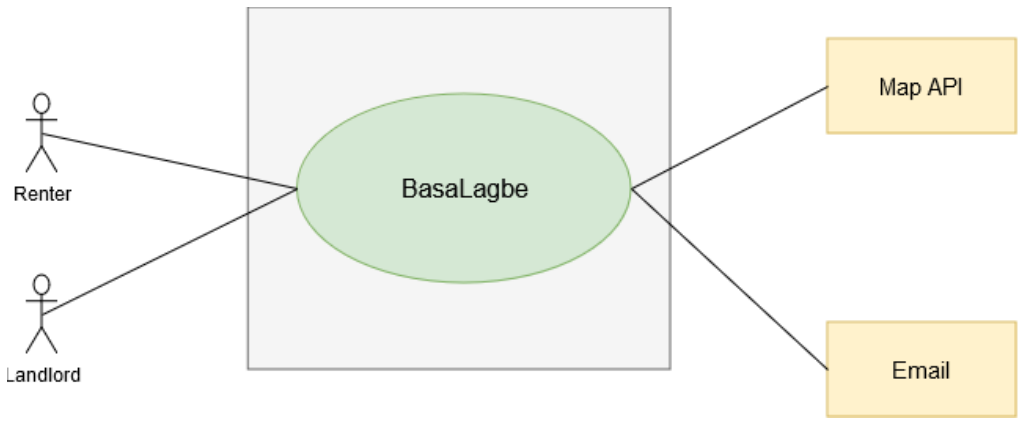
- Renter: Renters can chat directly with landlords to inquire about property details, schedule visits, and negotiate terms.
- Landlord: Landlords can communicate with renters, answer queries, and share additional details such as property documents or policies.

Integrated Map:

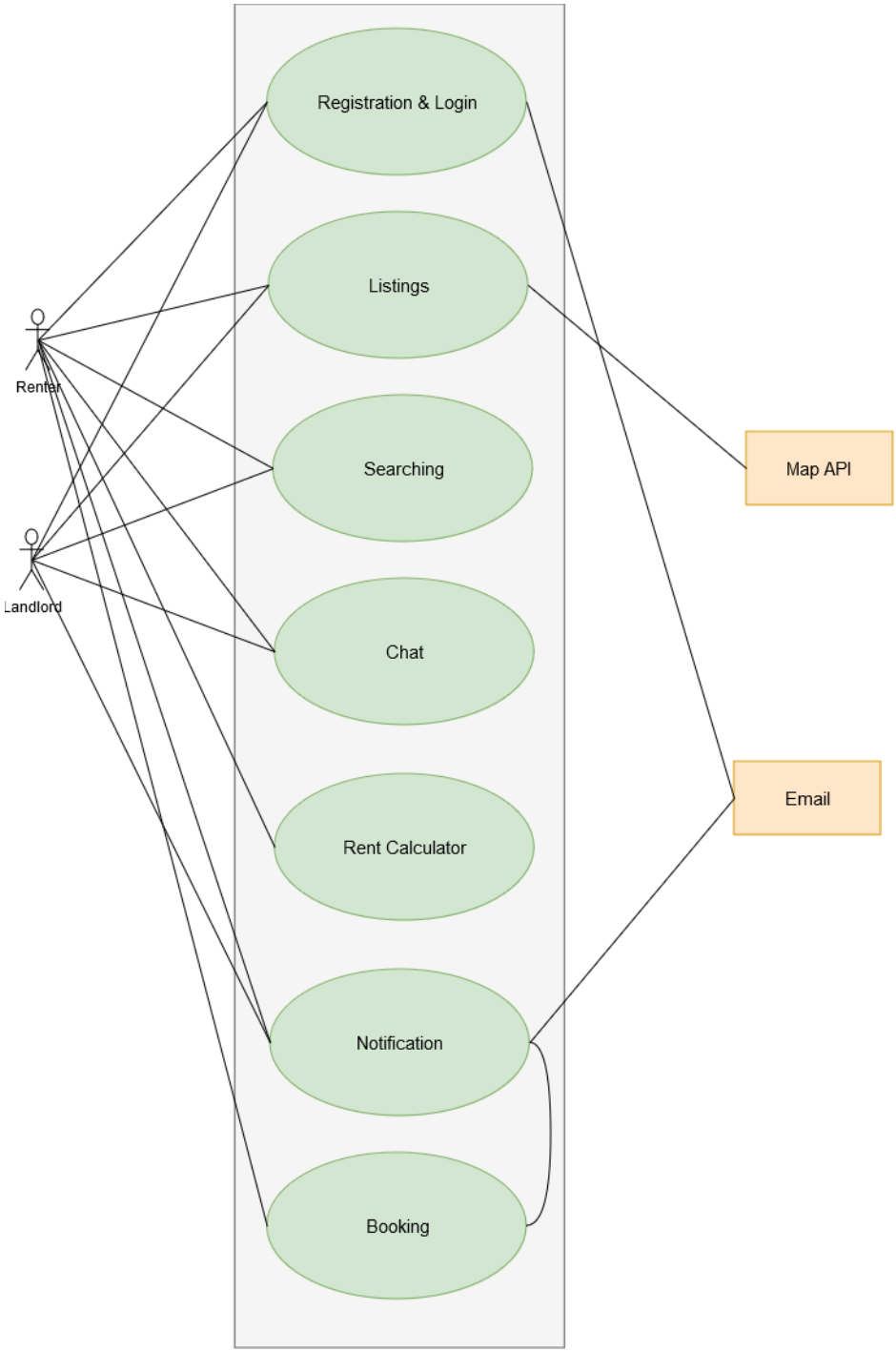
- **Renter**: Renters can view an interactive map that shows the exact location of the property.
- **Landlord**: Landlords can add the exact location of their property to the map when creating a property listing. This ensures that renters can easily locate the property on the map, improving its visibility and appeal.

1.Use Case Diagram

Level 0:
Name: BasaLagbe
Primary actors: Renters, Landlords, Admin
Secondary actors: Map API, Email

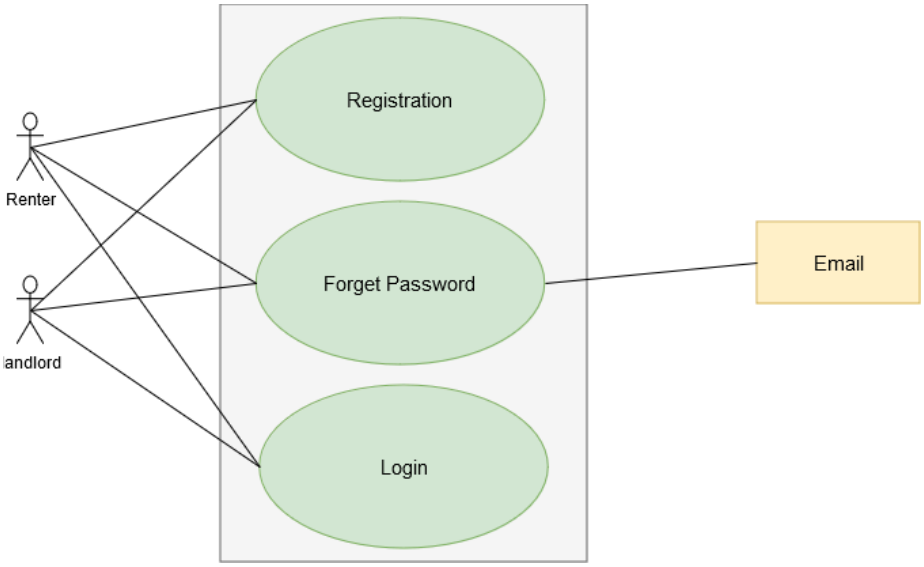


Level 1:
Primary actors: Renters, Landlords, Admin
Secondary actor: Map API, Email



Level 1.1: Registration
Primary actors: Renter, Landlord

Secondary actress: Email



Registration

- Action:
  - The renter or landlord provides personal details (e.g., name, email, password).
  - The system validates the input and checks if the email is unique.
- Reply:
  - A success message is shown upon successful registration.
  - If there's an error (e.g., duplicate email), an error message is displayed.

Login

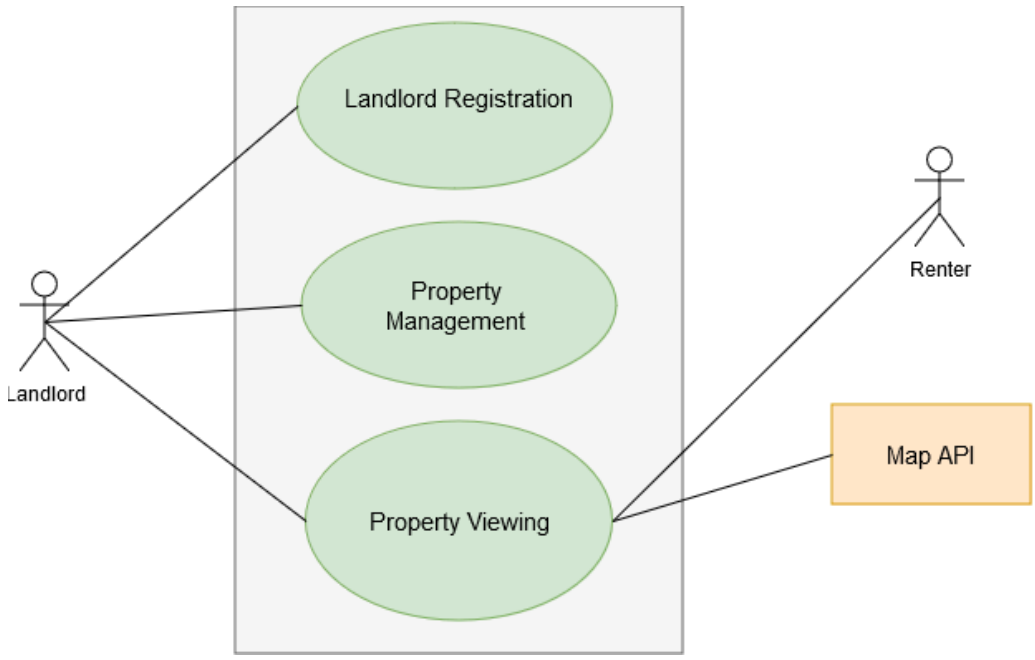
- Action:
  - The renter or landlord enters their email and password.
  - The system verifies the credentials against the database.
- Reply:
  - Successful login leads to dashboard access.
  - If login fails, an error message is displayed (e.g., "Invalid credentials").

Forget Password

- Action:
  - The renter or landlord enters their registered email.
  - The system checks if the email exists in the database.
  - If valid, a password recovery email is sent.

- Reply:
    - A confirmation message is shown (e.g., "Recovery email sent").
    - If the email is not registered, an error message is displayed (e.g., "Email not found").

Level 1.2: Listings:
Primary actors: Landlord
Secondary actors: Renter, Map API



Landlord Registration

- Action:
    - The landlord enters their details (e.g., name, email, password, contact info).
    - The system validates the input and checks for duplicate records.
- Reply:
    - Successful registration message: "Account created successfully."
    - Error message for issues like duplicate email or invalid inputs: "Email already exists" or "Please enter valid information."

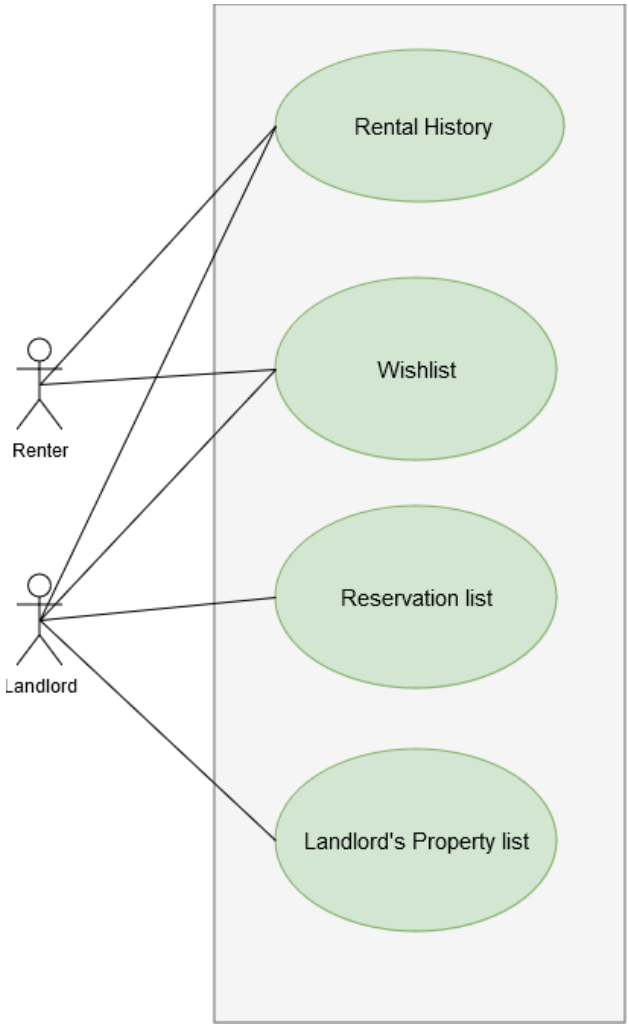
Property Management

- Action:
  - The landlord or admin:
    - Adds new properties (e.g., address, rent, features).
    - Updates existing property details (e.g., availability, price).
    - Deletes properties no longer available.
- Reply:
  - Success messages:
    - "Property added successfully."
    - "Property details updated."
    - "Property removed successfully."
  - Error messages for validation or system errors:
    - "Unable to add property. Please try again."
    - "Invalid details. Check inputs."

Property Viewing

- Action:
  - The renter searches for properties based on preferences (e.g., location, price).
  - The system retrieves matching properties and displays them.
  - A Map API is used to show property locations.
- Reply:
  - Search results: List of properties with details (e.g., name, rent, address).
  - Map interface showing locations.
  - Error message if no properties match the search: "No properties found matching your criteria."

Level 1.2.2: View listings & details
Primary actors: Landlord, renter



Rental History

- Action:
    - The renter views their past rental records (e.g., property details, rental period, payments).
- Reply:
    - Displays a list of past rentals with details like dates, amounts paid, and property addresses.
    - Error message if no rental history exists: "No rental records found.**"**

---

2. Wishlist

- Action:
    - The renter adds properties to their wishlist for later review.

    - The renter views or removes properties from the wishlist.
- Reply:
    - Success message for actions:
        - "Property added to wishlist."
        - "Property removed from wishlist."
    - Displays the wishlist with property details.
    - Error message if no properties are in the wishlist: "Your wishlist is empty."

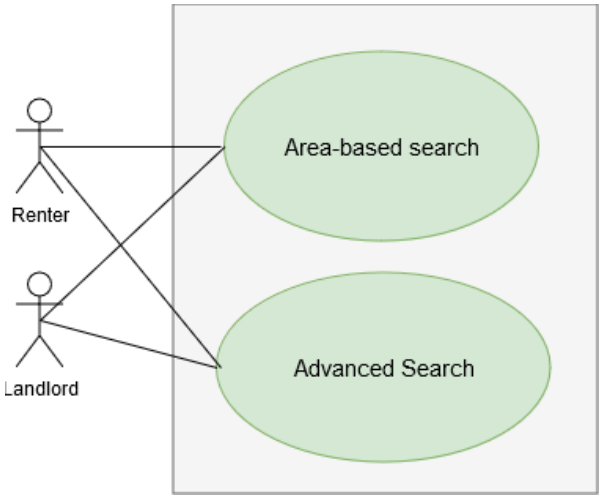
Reservation List

- Action:
    - The landlord views a list of renters who have requested reservations for their properties.
    - The landlord accepts or rejects reservation requests.
- Reply:
    - Displays the list of reservations with renter details (e.g., name, date, status).
    - Success message for actions:
        - "Reservation accepted."
        - "Reservation rejected."
    - Error message if no reservations are pending: "No reservation requests found."

Landlord's Property List

- Action:
    - The landlord views a list of all properties they have listed.
    - The landlord updates property details or removes a property from the list.
- Reply:
    - Displays the property list with details (e.g., address, rent, availability status).
    - Success message for updates or removals:
        - "Property updated successfully."
        - "Property removed successfully."
    - Error message if no properties are listed: "You have no properties listed."

Level 1.3: Searching:
Primary actors: Renter, landlord



Area-Based Search

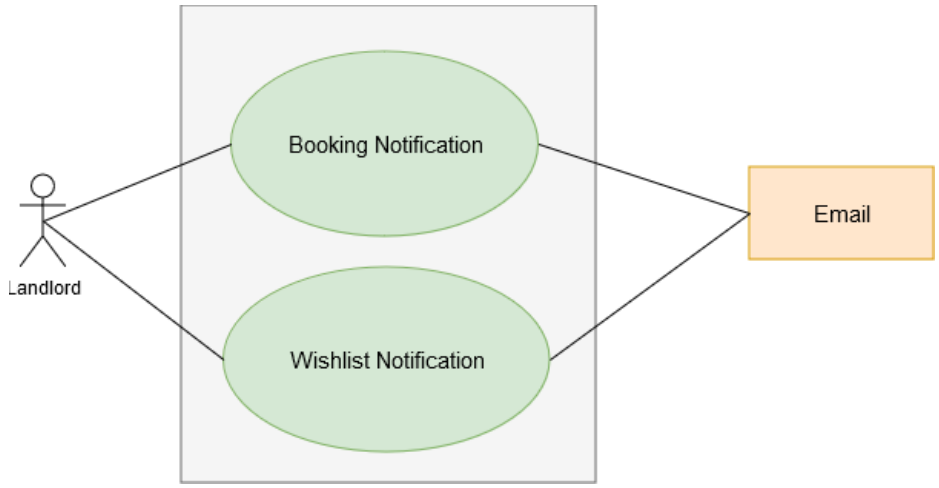
- Action:
    - The renter selects a specific area or city (e.g., Dhaka neighborhoods like Gulshan, Banani, etc.).
    - The system fetches properties available in the selected area.
- Reply:
    - If properties are available:
        - Displays a list of properties with details (e.g., name, rent, size, amenities, and images).
        - Provides a map view showing property locations (if integrated with Map API).
    - If no properties are available:
        - Displays a message: "No properties found in this area. Please try another location."

Advanced Search

- Action:
    - The renter applies additional filters, such as:
        - Price range: Min and max rent.
        - Number of bedrooms: Studio, 1BHK, 2BHK, etc.

        - Property type: Apartment, house, shared rooms.
        - Amenities: Parking, elevator, pool, etc.
        - Availability date: Immediate or future availability.
        - Furnishing: Fully furnished, semi-furnished, or unfurnished.
    - The system processes these filters and retrieves properties that match all criteria.
- Reply:
    - If properties match the filters:
        - Displays the list of properties with detailed information and images.
        - Provides a comparison option (if supported) to evaluate properties side-by-side.
    - If no properties match:
        - Displays a message: "No properties found matching your criteria. Try modifying your filters."

Level 1.4: Notifications and Booking
Primary actors: Landlord
Secondary actors: Email



WishList Notifications

Action:

- Trigger:
    - A renter adds a landlord's property to their wishlist.
- System Action:
    - The system sends a notification to the landlord informing them about the addition.

Reply:

- Notification Content:
    - "A renter has added your property [Property Name] to their wishlist."
    - Option for the landlord:
        - "View Property Details" (to see which property was added).
        - Optionally view a list of renters who added the property to their wishlists (if the system supports this).

Schedule Notifications

Action:

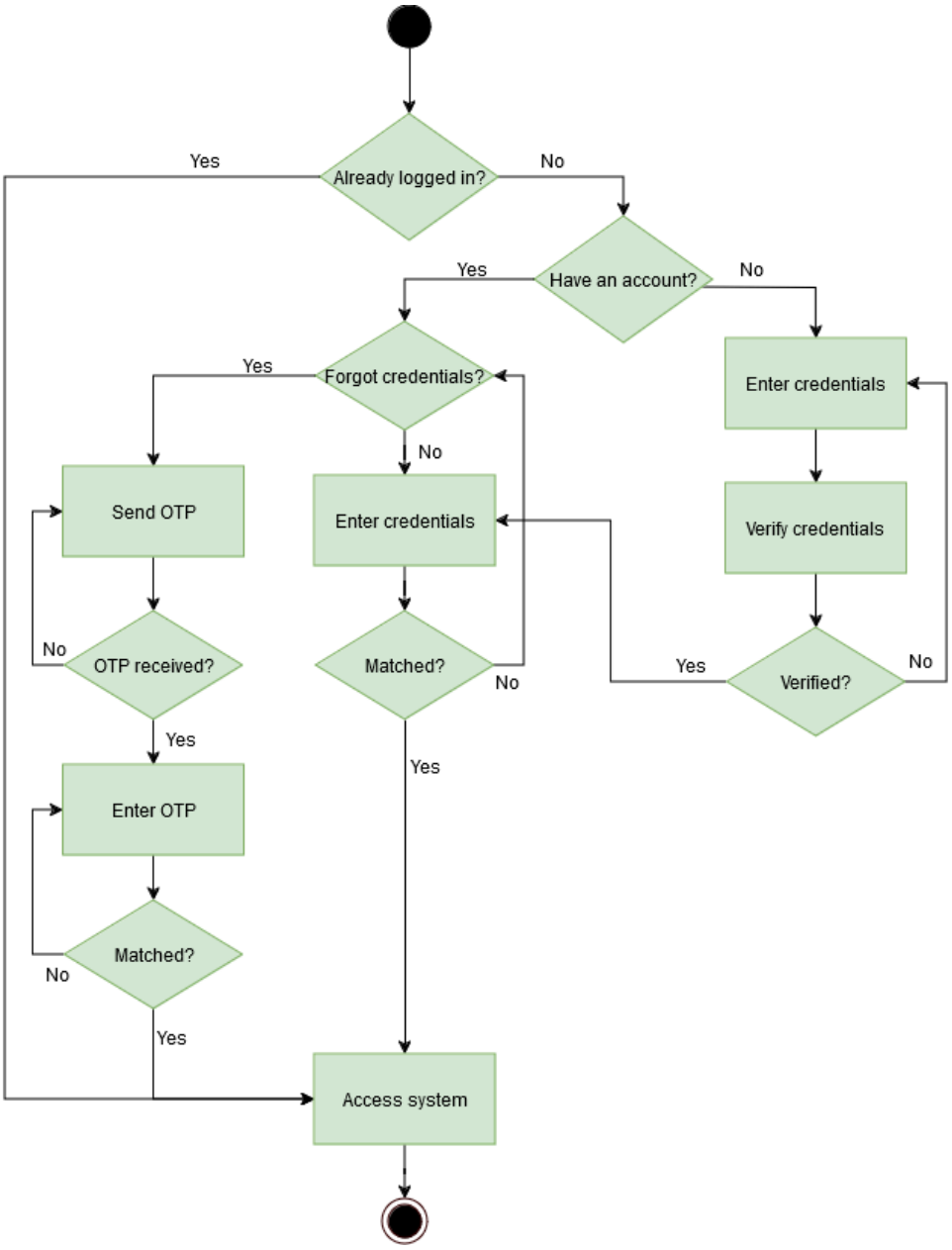
- Trigger:
    - A renter schedules a visit for a property listed by the landlord.
- System Action:
    - Send a notification to the landlord about the scheduled visit.

Reply:

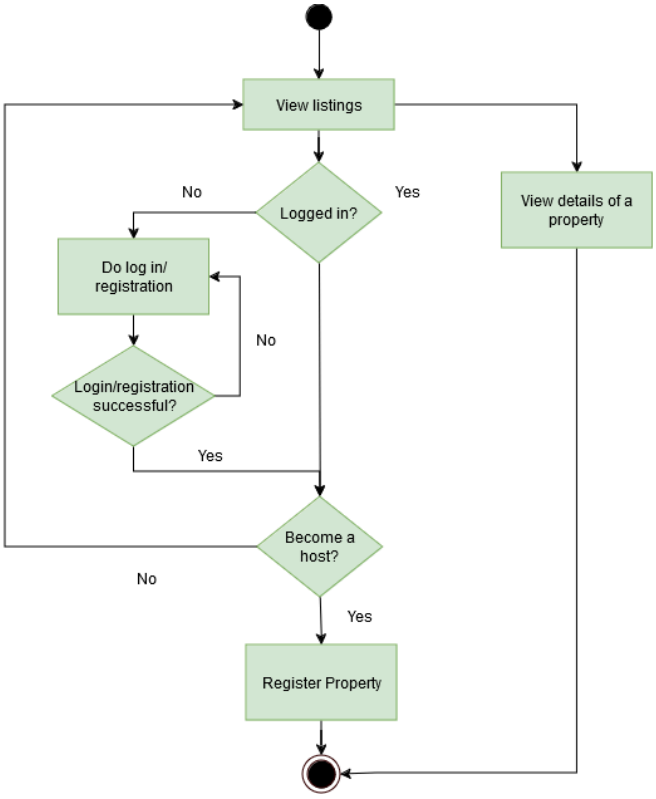
- For New Visit Requests:
    - Notification Content:
        - "A renter has scheduled a visit for your property [Property Name] on [Date & Time]."
        - Options for the landlord:
            - "Approve Schedule" (confirms the visit).
            - "Propose New Time" (suggests a new schedule).
- For Cancellations:
    - Notification Content:
        - "The visit to [Property Name] scheduled for [Date & Time] has been canceled by the renter."
        - Options for the landlord:
            - "View Property Details."
            - "Contact Renter" (if system allows direct communication).

2.Activity Diagram

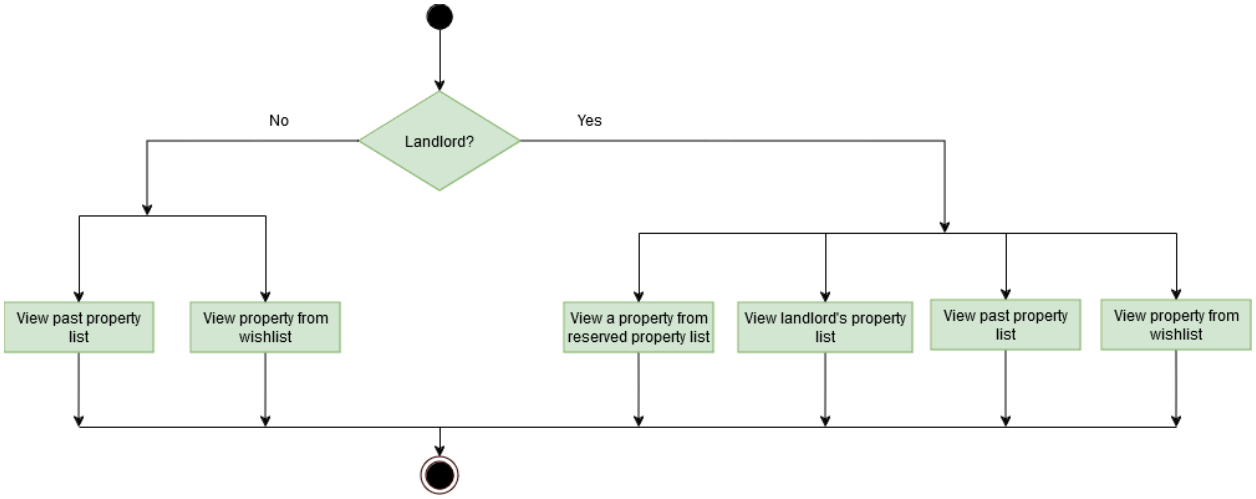
Level: 1.1
Name: Registration



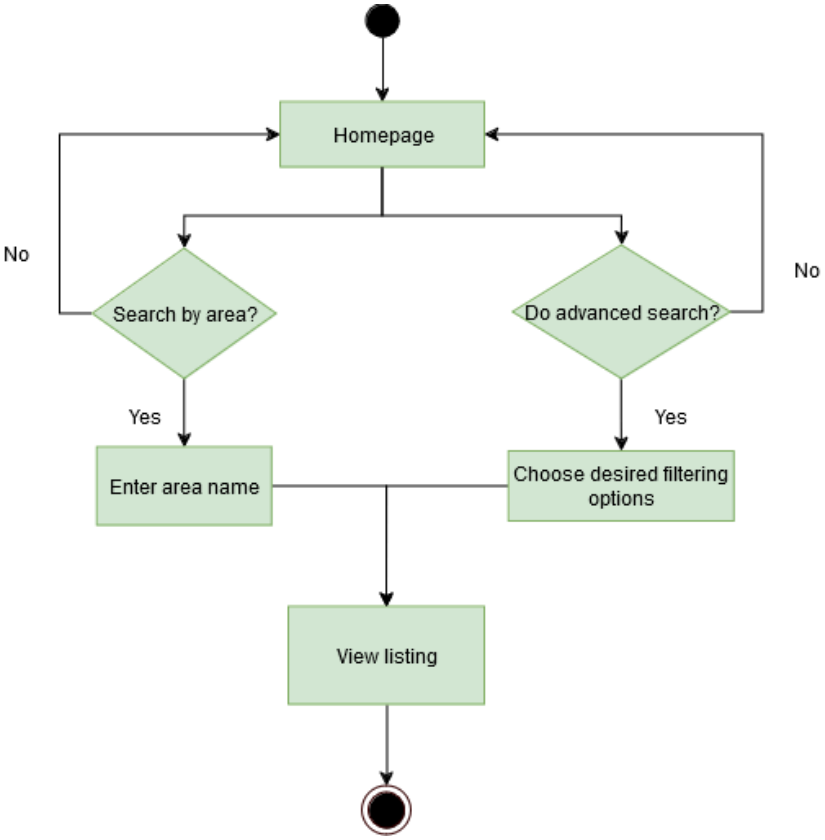
Level: 1.2
Name: Listings



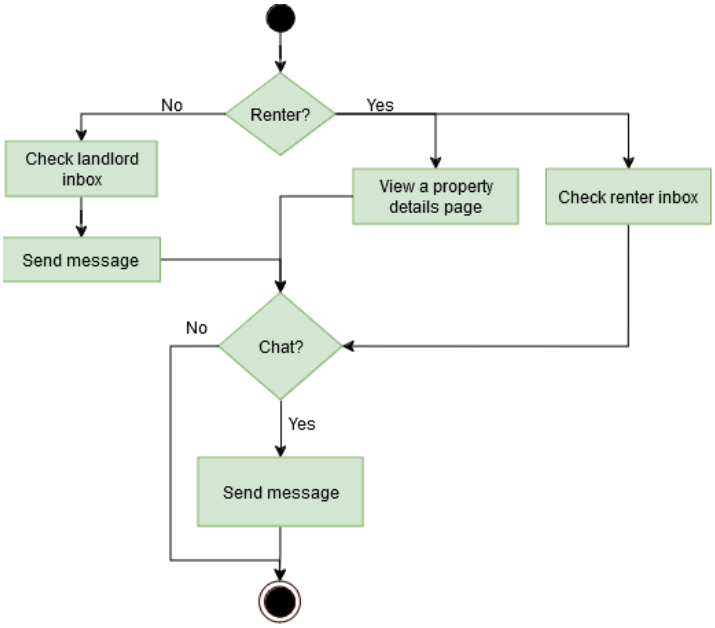
Level: 1.2.2
Name: View Listings and Details



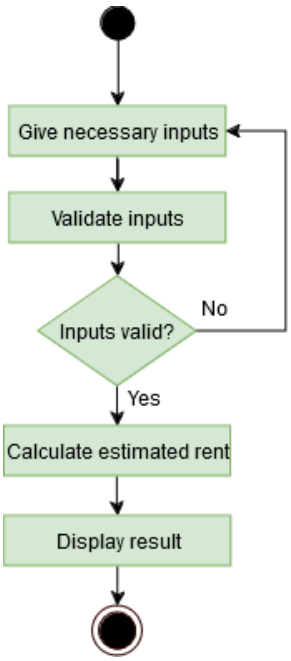
Level: 1.3
Name: Searching



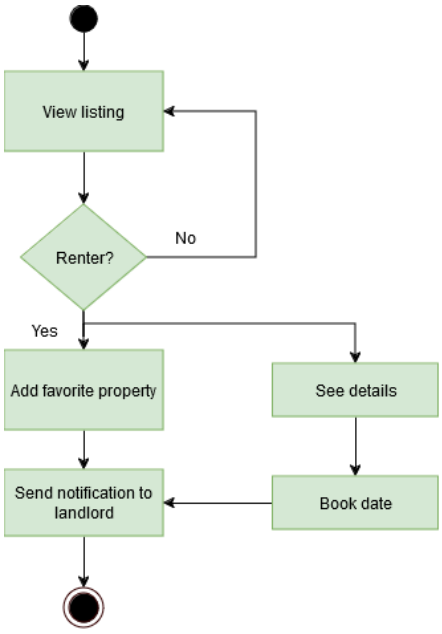
Level: 1.4
Name: Chat



Level: 1.5
Name: Rent Calculator

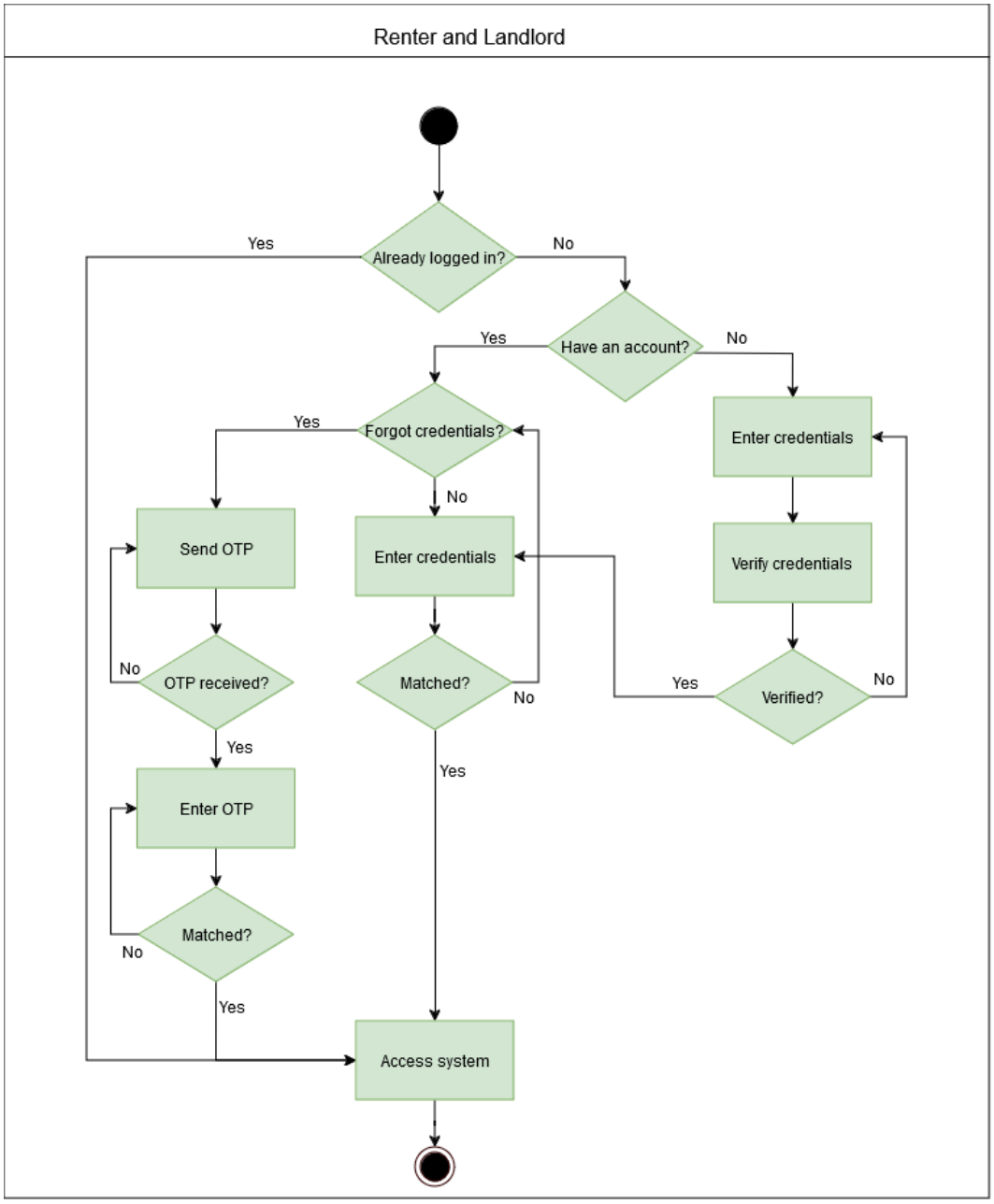


Level: 1.6
Name: Notification



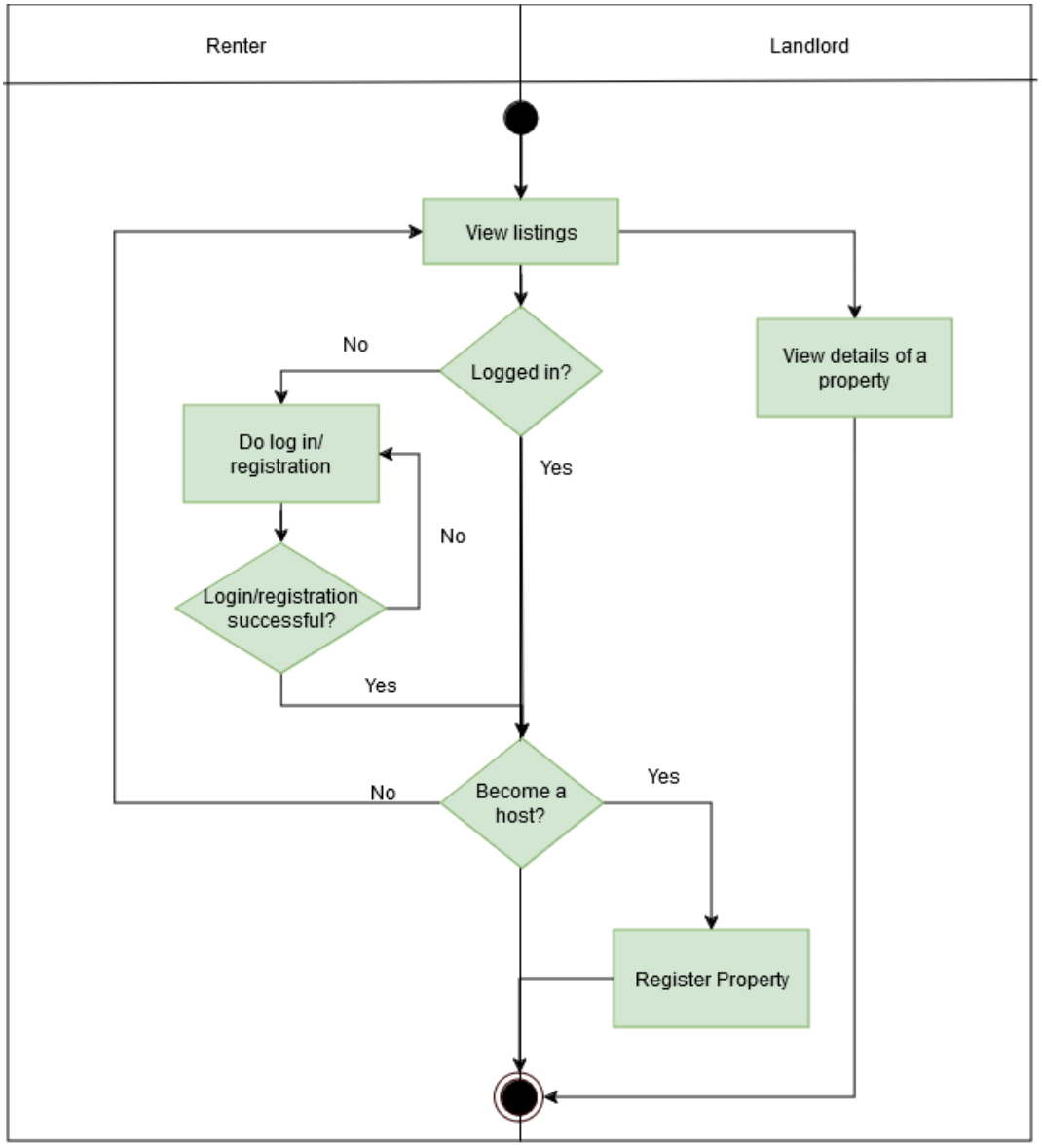
3.Swimlane Diagram

Level: 1.1
Name: Registration



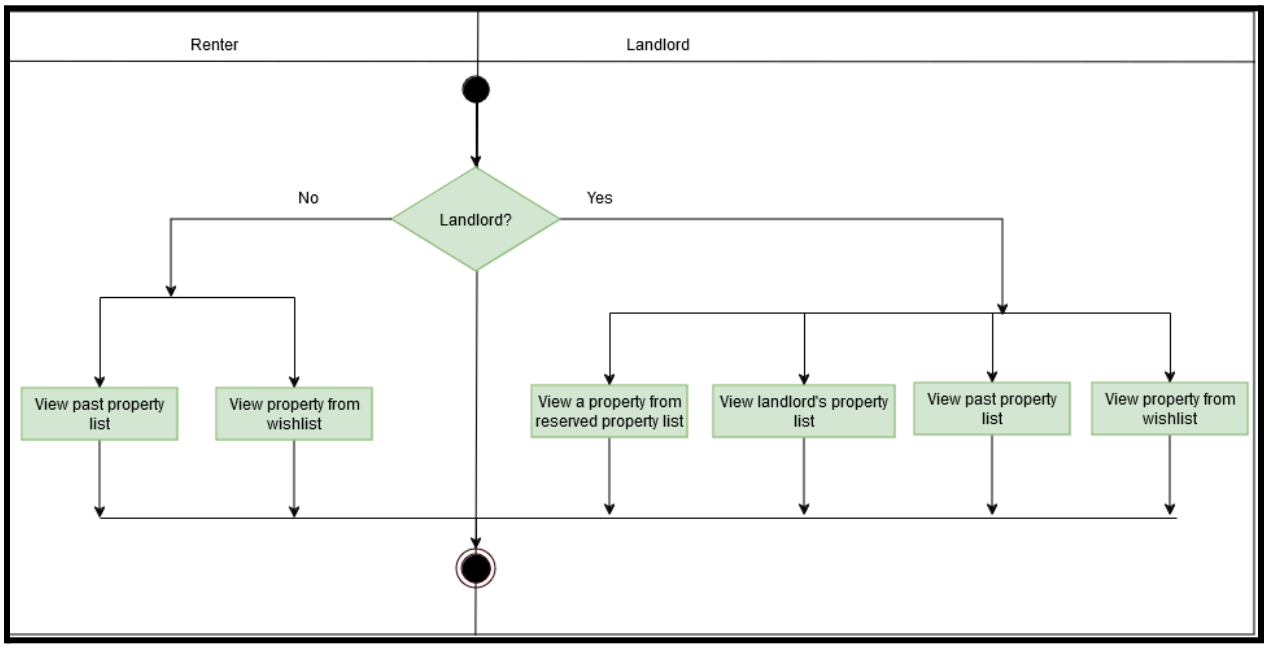
Level: 1.2

Name: Listings



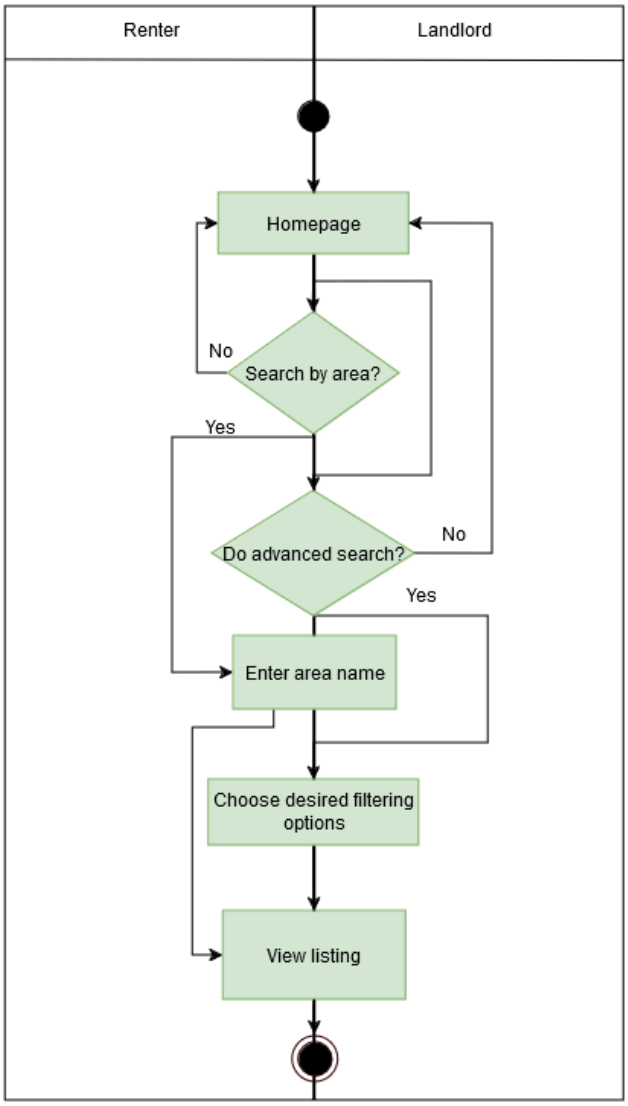
Level: 1.2.2

Name: View Listings and Details



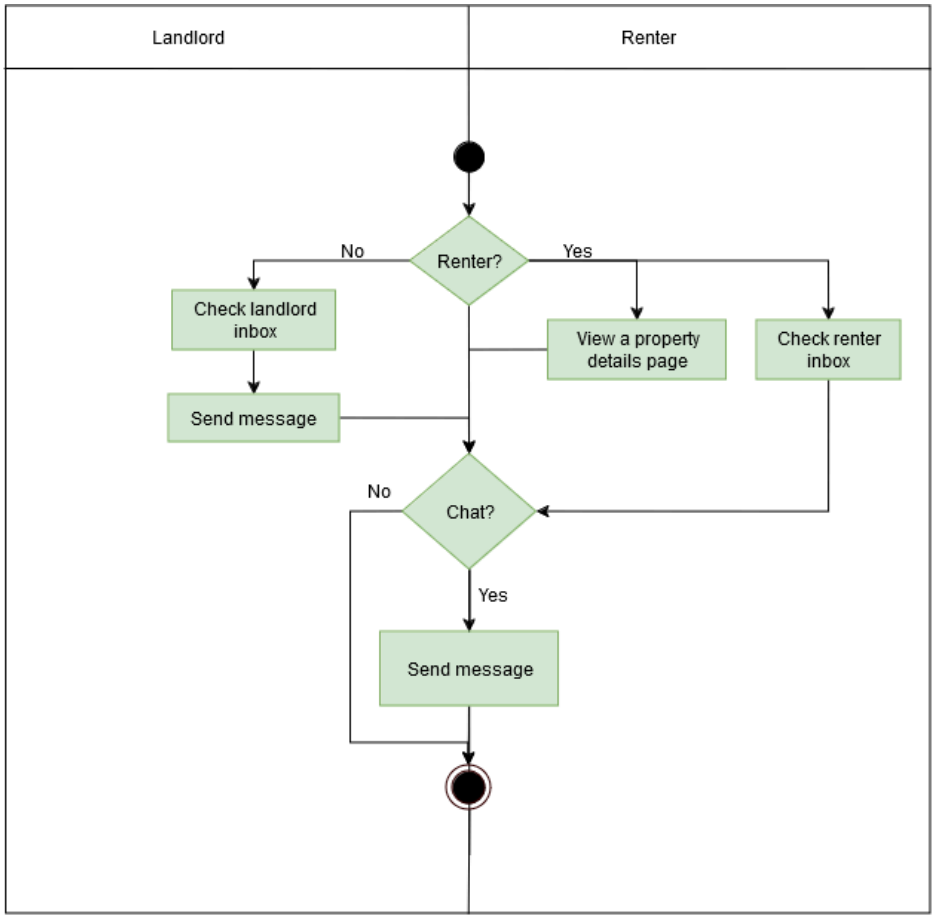
Level: 1.3

Name: Searching



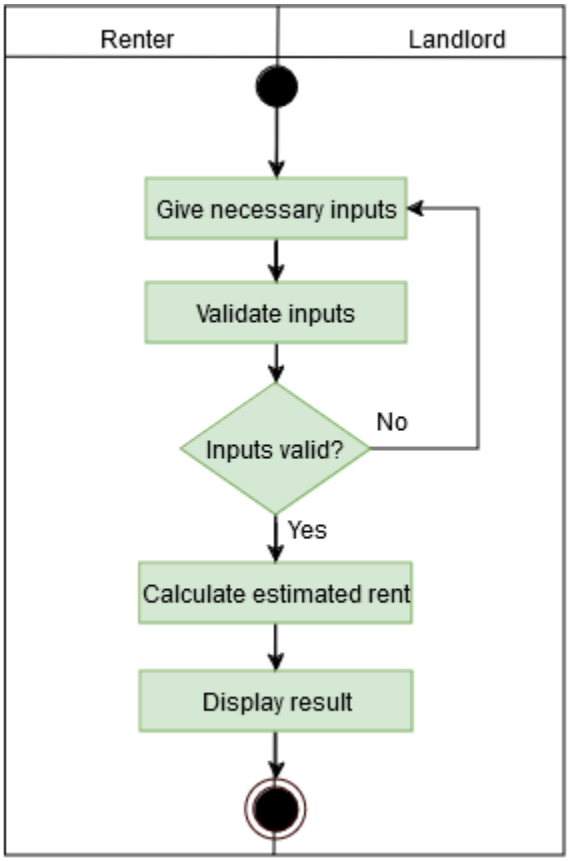
Level: 1.4

Name: Chat



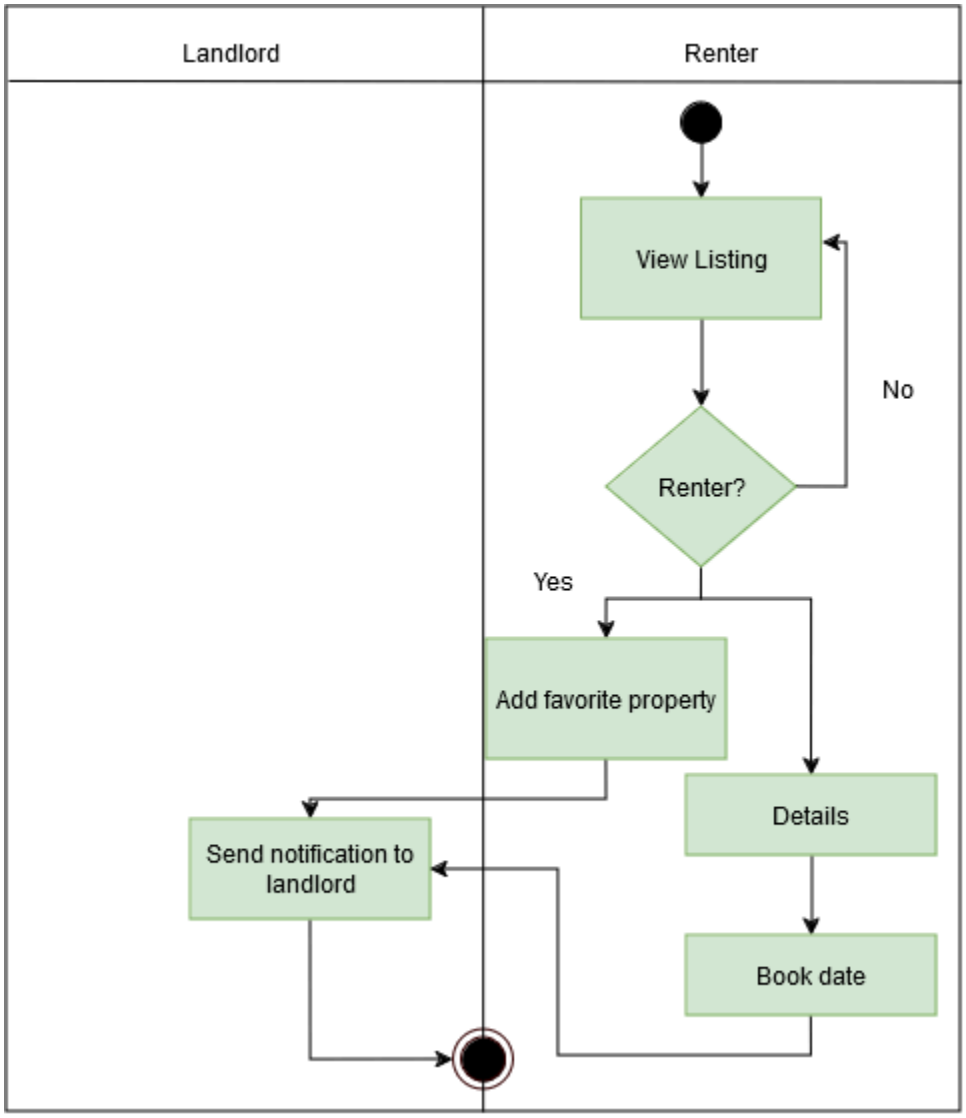
Level: 1.5

Name: Rent Calculator



Level: 1.6

Name: Notification



4.Database Modeling

Data Object Identification:

SL	Nouns	Problem/Solution Space	Attributes
1	Admin	S	
2	Renter	S	4,5,6,7,8,22,26,33
3	Landlord	S	4,5,6,7,9,23,25,26,33
4	Name	S	
5	Email	S	
6	Photo	S	
7	password	S	
8	renter_id	S	
9	landlord_id	S	
10	Type	S	
11	Room_no	S	
12	Bathroom_no	S	
13	amenities	S	
14	pictures	S	
15	house _size	S	
16	Profile	P	
17	Location	S	
18	features	P	
19	Price	S	
20	Availability	S	

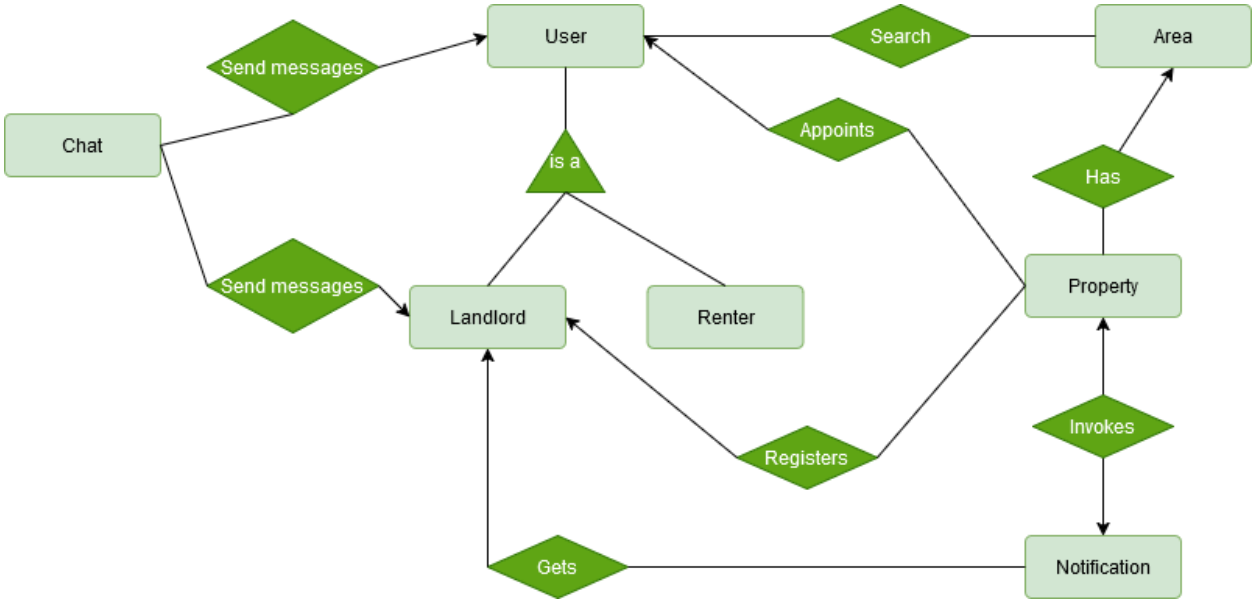
21	Information	P	
22	wishlist	S	
23	Reservation List	S	
24	listings	P	
25	Property_list	S	
26	Message	S	
27	Schedule	S	8,9,34,35,46
28	Accessibility	P	
29	Notification	S	8,9,26
30	Map	S	
31	Details	P	
32	Calculator	S	
33	Contact_details	S	
34	Property_id	S	
35	Date	S	
36	Description	S	
37	Home	P	
38	Preferences	P	
39	Person	P	
40	Surroundings	P	
41	Property	S	9,10,11,12,13,14,15,17,19,20,30,34,36
42	Chat	S	8,9,26
43	OTP	S	8,44,45
44	Timestamp	S	
45	Status	S	

46	Time	S	
47	Area	S	50,4
48	Longitude	S	
49	Latitude	S	
50	Description		

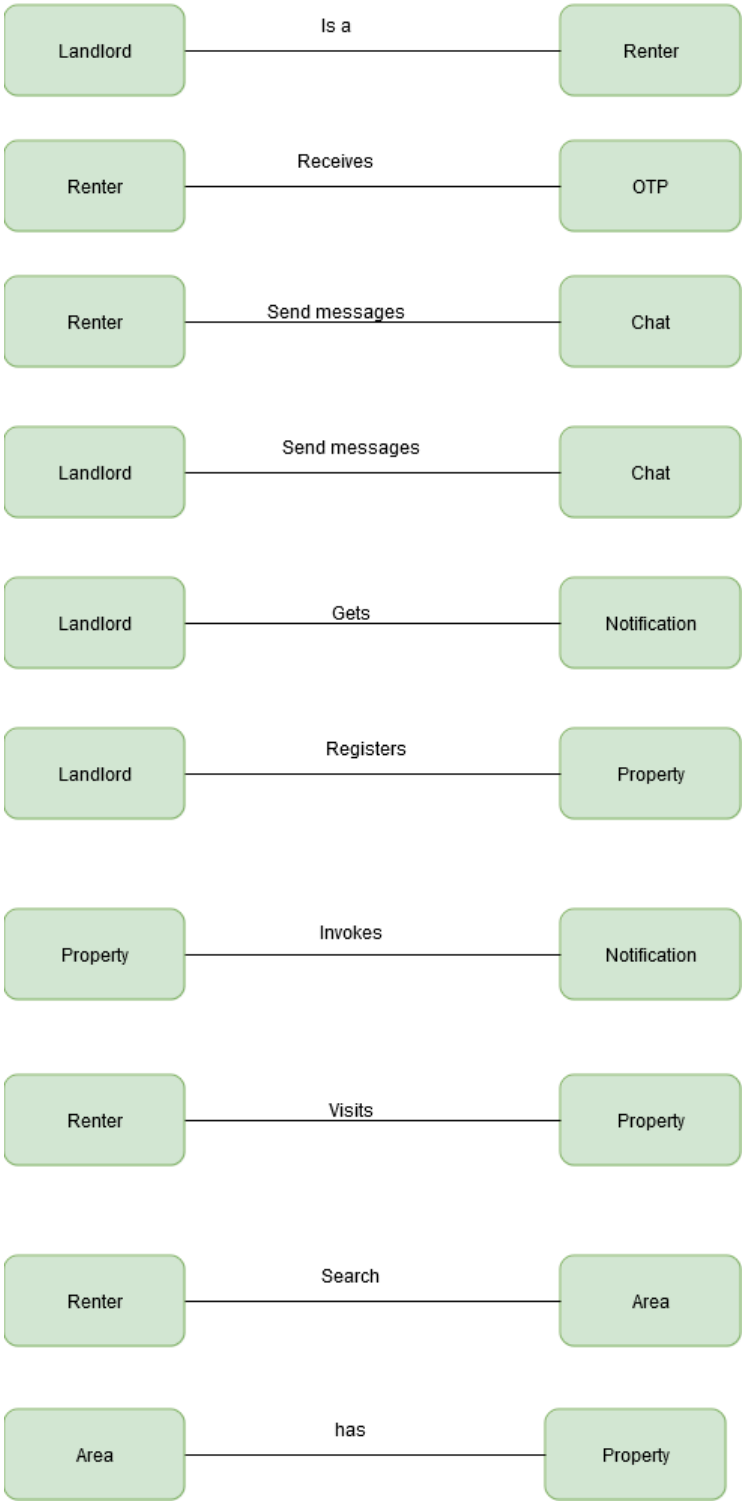
Data Objects:

1. Renter
2. Landlord
3. Property
4. Chat
5. Notification
6. Area
7. OTP

ER Diagram:


User
Search
Area
Send messages
Chat
Appoints
is a
Has
Send messages
Landlord
Renter
Property
Invokes
Registers
Gets
Notification

Relationship among data objects:



Schema:

Name	Attributes	Key Type(PK/FK)	Type	Size
Renter	renter_id name email photo password wishlist contact_details	PK	NUMBER VARCHAR VARCHAR VARCHAR VARCHAR TEXT VARCHAR	11 100 100 255 100 - 255
Landlord	landlord_id name email photo password property_list reservation_list contact_details	PK	NUMBER VARCHAR VARCHAR VARCHAR VARCHAR TEXT TEXT VARCHAR	11 100 100 255 100 - - 255
Property	property_id type room_no bathroom_no amenities pictures house_size location price availability description latitude longitude	PK	NUMBER VARCHAR NUMBER NUMBER TEXT VARCHAR NUMBER VARCHAR DECIMAL BOOLEAN TEXT VARCHAR VARCHAR	11 50 3 3 - 255 6 100 10,2 1 - 255 255
Chat	chat_id renter_id landlord_id message	PK FK FK	NUMBER NUMBER NUMBER TEXT	11 11 11 -
Visit	visit_id renter_id landlord_id property_id date time accessibility	PK FK FK FK	NUMBER NUMBER NUMBER NUMBER DATE TIME VARCHAR	11 11 11 11 - 100

Notification	notification_id Recipient_id Message Timestamp is_read	PK FK	NUMBER NUMBER TEXT TIMESTAMP BOOLEAN	11 11 - - 1
OTP	renter_id timestamp status	FK	NUMBER NUMBER BOOLEAN	11 11 1
Area	property_ids description area_name	 PK	NUMBER TEXT VARCHAR	- - 100

5.CLASS BASED MODELING:

CLASS BASED MODELING CONCEPT:

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.

Analysis Class Identification:

SL	Nouns
1	Admin
2	Renter
3	Landlord
4	Name
5	Email
6	Photo
7	password
8	renter_id
9	landlord_id
10	Type
11	Room_no
12	Bathroom_no
13	amenities

14	pictures
15	house _size
16	Profile
17	Location
18	features
19	Price
20	Availability
21	Information
22	wishlist
23	Reservation List
24	listings
25	Property_list
26	Message
27	Schedule
28	Accessibility
29	Notification
30	Map
31	Details
32	Rent Calculator
33	Contact_details
34	Property_id
35	Date
36	Description
37	Home
38	Preferences

39	Person
40	Surroundings
41	Property
42	Chat
43	Status
44	OTP
45	Timestamp
46	Notification
47	Service
48	Search Service
49	Area
50	longitude
51	Latitude
52	Calculator

List of Verbs in User Story:

SL	Verbs
1	Register
2	Log in
3	Create

4	Personalize
5	Access
6	List
7	Update
8	Delete
9	Include
10	Advertise
11	View
12	Manage
13	Choose
14	Filter
15	Match
16	Find
17	Make
18	Input
19	Estimate
20	Select
21	Schedule
22	Receive
23	Become
24	Keep
25	Obtain
26	Switch
27	Show
28	Understand

29	Help
30	Communicate
31	Enable

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

SL	Nouns	General Classifications
1	Admin	1, 4, 5
2	Renter	1, 4, 3
3	Landlord	1, 4, 5
4	Name	2
5	Email	2
6	Photo	2,
7	password	2
8	renter_id	2
9	landlord_id	2,
10	Type	7
11	Room_no	7

12	Bathroom_no	7
13	amenities	7, 2
14	pictures	2, 7
15	house _size	2, 7
16	Profile	2, 7, 4
17	Location	6, 7
18	features	7, 2
19	Price	2, 7
20	Availability	3, 2
21	Information	3, 2
22	wishlist	2, 3, 7
23	Reservation List	3, 7, 2
24	listings	2, 5,7
25	Property_list	2, 5,7
26	Message	2, 3
27	Schedule	3, 7, 2
28	Accessibility	7
29	Notification	3, 2
30	Map	6, 2
31	Details	2, 3
32	Rent Calculator	2,7
33	Contact_details	2
34	Property_id	2, 7
35	Date	3, 2
36	Description	2

37	Home	6
38	Preferences	2, 3
39	Person	1, 4
40	Surroundings	6, 7
41	Property	2, 6, 7
42	Chat	2, 3,7
43	Status	2
44	OTP	2,3,7
45	Timestamp	2
46	Notifications	2,3,7
47	Search service	1,2,3
48	Service	1,2,3
49	Conversation	2,3,7

Potential To Be A Class:

- Admin
- Renter
- Landlord
- Profile
- Wishlist
- Reservation List
- Listings
- Property_list
- Schedule
- Message
- Property
- OTP
- Notifications

- Service
- Search service
- Conversation

Analysis: As rent calculator has a specific and crucial role in our system, it will also be considered as a potential class.

- Rent calculator

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:

SL	Nouns	Selection Criteria
1.	Admin	
2.	Renter	1, 2, 3, 4, 5
3.	Landlord	1, 2, 3, 4, 5
4.	Wishlist	1, 2, 3, 4, 5
5.	Reservation List	1, 2, 3,4, 5

6.	Listings	1, 2, 3, 6
7.	Property_list	1, 2, 3,4, 5
8.	Schedule	1, 2, 3,4, 5
9.	Message	2, 3, 4, 6
10.	Property	6
11	Profile	1,3,5
12	OTP	1,2,3,4,5
13	Notifications	1,2,3,4,5
14	Service	2,5,6
15	Search Service	2,5,6
16	Conversation	2,3,4,6
17	Rent Calculator	2,3,6

Selected Class:

1. User
2. Landlord
3. Renter
4. Property
5. WishlistManager
6. ReservationManager
7. ListingManager
8. ScheduleManager
9. NotificationManager
10. ConversationManager
11. OTPManager
12. SearchService

13. RentCalculator

Attributes and Method Identification:

User	-userID -name -email -photo -password -contactDetails	+addDetails() +search(criteria[]) +estimateRent(criteria[]) +viewNotifications()
Renter		+makeReservation() +cancelReservation()
Landlord		+hostProperty(property) +removeProperty(propertyID) +viewReservations() +viewPropertyList()
Property	-userID -propertyID -type -roomNo -bathroomNo -amenities -pictures -housesize -location -price -availability -description -latitude -longitude	+registerProperty(details) +updateProperty(propertyID, details) +viewDetails()
WishlistManager	-wishlists{}	+addToWishlist(userID,propertyID) +removeFromWishlist(userID,prope

		rtyID) +viewWishlist(userID) +createNotification(landlordID, type, notificationMessage)
ReservationManager	-reservations{}	+addReservations(renterID,landlordID,propertyID) +cancelReservation(reservationID) +viewReservations(renterID, landlordID)
ListingManager	-listings{}	+createListing(property) +updateListing(propertyID,details) +deleteListing(propertyID) +searchListings(criteria[]) +viewListings()
ScheduleManager	-schedules{}	+createSchedule(renterID,landlordID) +viewSchedule(scheduleID) +createNotification(landlordID, type, notificationMessage)
NotificationManager	-notifications{}	+createNotification(landlordID, type, notificationMessage) +getNotifications(landlordID)
ConversationManager	-conversations{}	+sendMessage(renterID, landlordID, messageContent) +viewAllMessage(conversationID)
OTPManager	-OTPs{}	+generateOTP(renterID) +validateOTP(renterID, OTP) +expireOTP(renterID)
SearchService		+searchProperties(listings, criteria[])
RentCalculator		+estimateRent(criteria[])

Class Cards:
ID:1

Class: Renter	
Responsibilities	**Collaborators**
UpdatProfile	Renter
WishList	Renter
Viewlist	Renter

ID:2

Class: Landlord	
Responsibilities	**Collaborators**
Host a property	Landlord
Remove a property	Landlord
See the property list	Landlord
View notification	landlord

ID:3

Class: WishList	
Responsibilities	**Collaborators**
Add	renter,landlord
Remove	Renter,landlord
View	Renter,landlord
Create notification	renter,landlord

ID:4

Class: ReservationList	
Responsibilities	**Collaborators**
Add	renter
cancel	landlord
view	landlord

ID:5

Class: PropertyList	
Responsibilities	**Collaborators**
remove	landlord
Get property	landlord
create_list	landlord

ID:6

Class: Property	
Responsibilities	**Collaborators**
Register Property	landlord
Update Property	landlord
Availability	
View details	landlord

ID:7

Class: Listings	
Responsibilities	**Collaborators**
View listings	searchService
Search listings	searchService
Create listings	landlord
Update listing	landlord
Delete listing	landlord

ID:8

Class: Schedule	
Responsibilities	**Collaborators**
Create Schedule	Renter
View schedule	landlord,renter
Create notification	renter,landlord

ID:9

Class: Message	
Responsibilities	**Collaborators**
View message	Renter,Landlord

ID:10

Class: Conversation	
Responsibilities	**Collaborators**
Send message	Renter, landlord
View messages	Renter, landlord

ID:11

Class: OTP	
Responsibilities	**Collaborators**
Generate OTP	system
Validate OTP	system
Expire OTP	system

ID:12

Class: Service	
Responsibilities	**Collaborators**
Register	renter,landlord
Log in	renter,landlord
logout	renter,landlord
Authentication	renter,landlord

ID:13

Class: Notification	
Responsibilities	**Collaborators**
create	renter,landlord
get notification	landlord

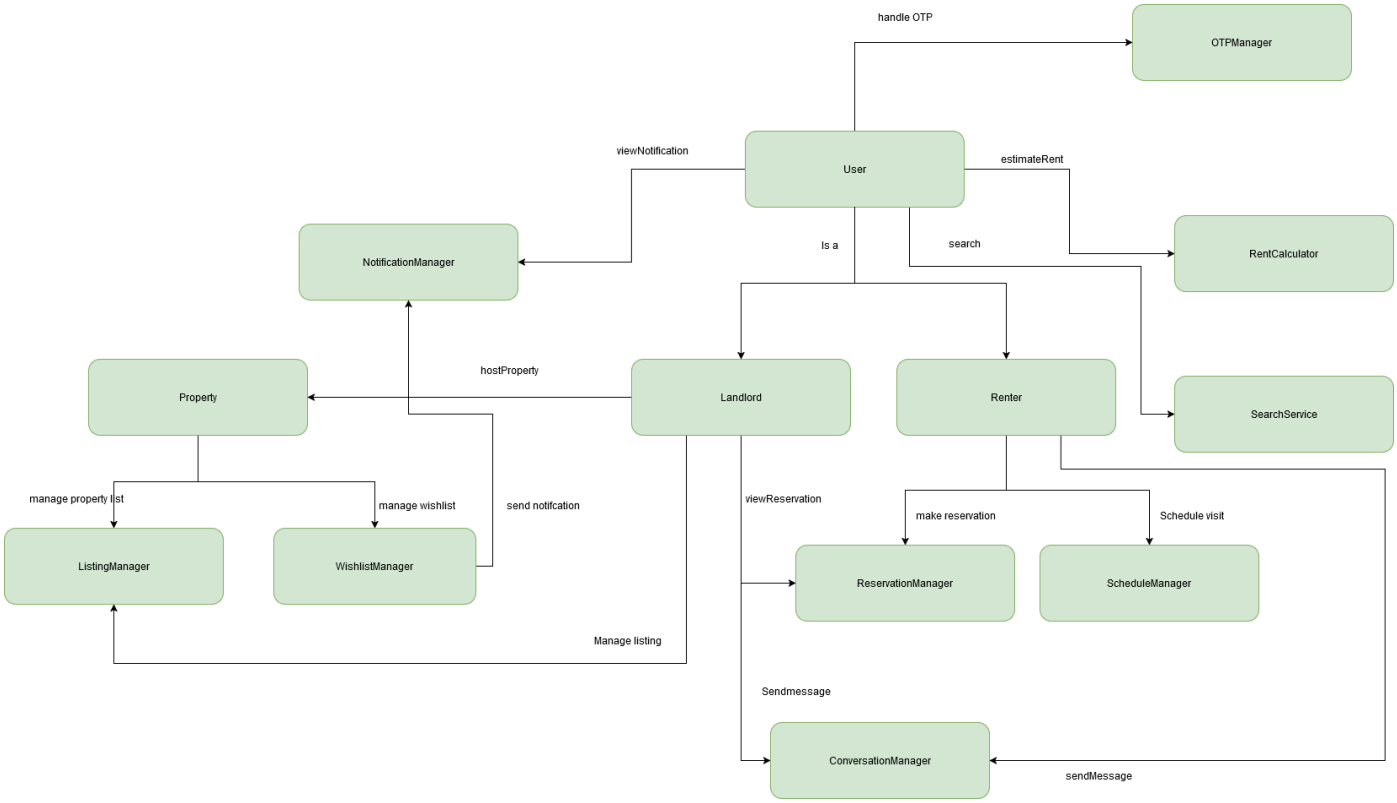
ID:14

Class:Searchservice	
Responsibilities	**Collaborators**
Search Properties	Renter,Landlord

ID:15

Class:RentCalculator	
Responsibilities	**Collaborators**
Estimate rent of property	Renter,Landlord

CRC Diagram:
ID:1



6.Behavioral Modeling

Behavioral modeling is the process of predicting future behavior using current and relevant data on consumer and company spending. There are two distinct characterizations of states that must be taken into account in behavioral modeling. First, the state of each class as the system executes its task, and then the system's state as seen from the outside while the system executes its task.

6.1 State Transition Diagram

A UML state diagram, which depicts the active states for each class and the occasions (triggers) that induce changes between these active states, is a part of a behavioral model. Here we've listed the most important events that trigger the change of states.

List of Events

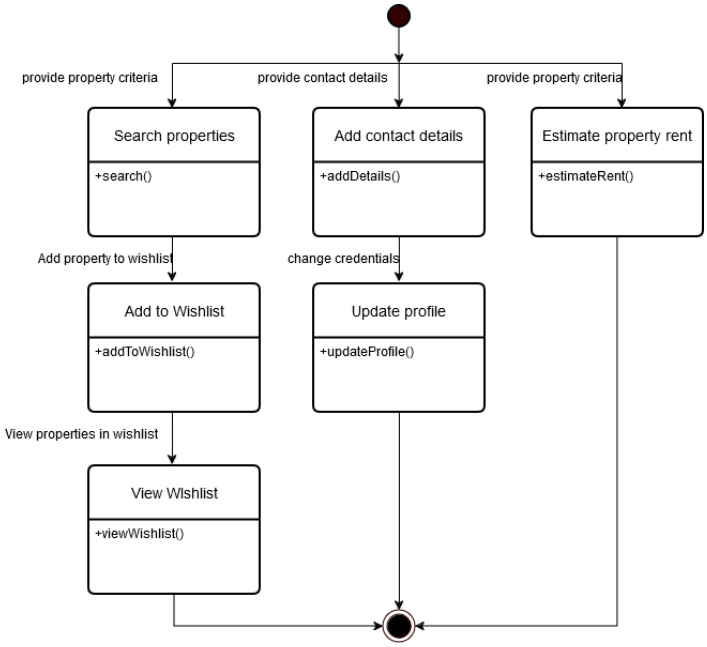
SI	Initiator Class	Event	Collaborator Class
01	Renter	Update profile	
02	Renter	Add details	
03	Renter	View wishlist	WishList
04	Renter	Search properties	SearchService, Listings
05	Landlord	Host property	Property_list, Listings

06	Landlord	Remove property	Property_list
07	Landlord	See reservation list	Reservation_list
08	Landlord	See wishlist	See wishlist
09	Landlord	View property details	Property
10	Landlord	View notifications	Notification
11	Service	Register	Renter, Landlord
12	Service	Login	Renter, Landlord
13	Service	Logout	
14	Service	Authenticate	OTP
15	Wishlist	Add to wishlist	Renter, Landlord, Property
16	Wishlist	Remove from wishlist	Renter, Landlord, Property
17	Wishlist	Create notification	Notification
18	Reservation_list	Add reservation	Renter, Landlord, Property
19	Reservation_list	Cancel reservation	Landlord
20	Reservation_list	View reservation details	Landlord
21	Property_list	Create property list	Landlord
22	Property_list	Remove property	Landlord
23	Property_list	Get properties	Landlord, Property
24	Property	Register property	Landlord, Listings
25	Property	Mark property as available	
26	Listings	Create listing	Property

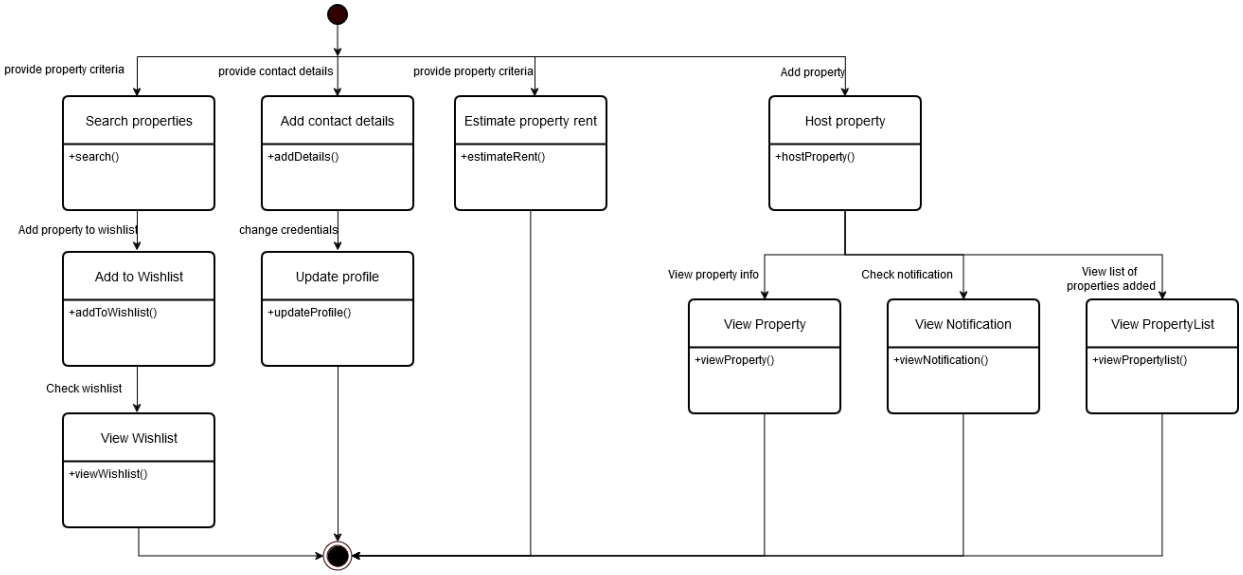
27	Listings	Update listing	Property
28	Listings	Delete listing	
29	Listings	Search listings	SearchService
30	Listings	View listings	
31	Schedule	Create schedule	Renter, Landlord, Property
32	Schedule	View schedule	
33	Schedule	Create notification	Notification
34	Message	View message	Conversation
35	Conversation	Send message	Message
36	Conversation	View all messages	Message
37	OTP	Generate OTP	
38	OTP	Validate OTP	
39	OTP	Expire OTP	
40	Rent Calculator	calculate	

6.2 State Transition Diagram:

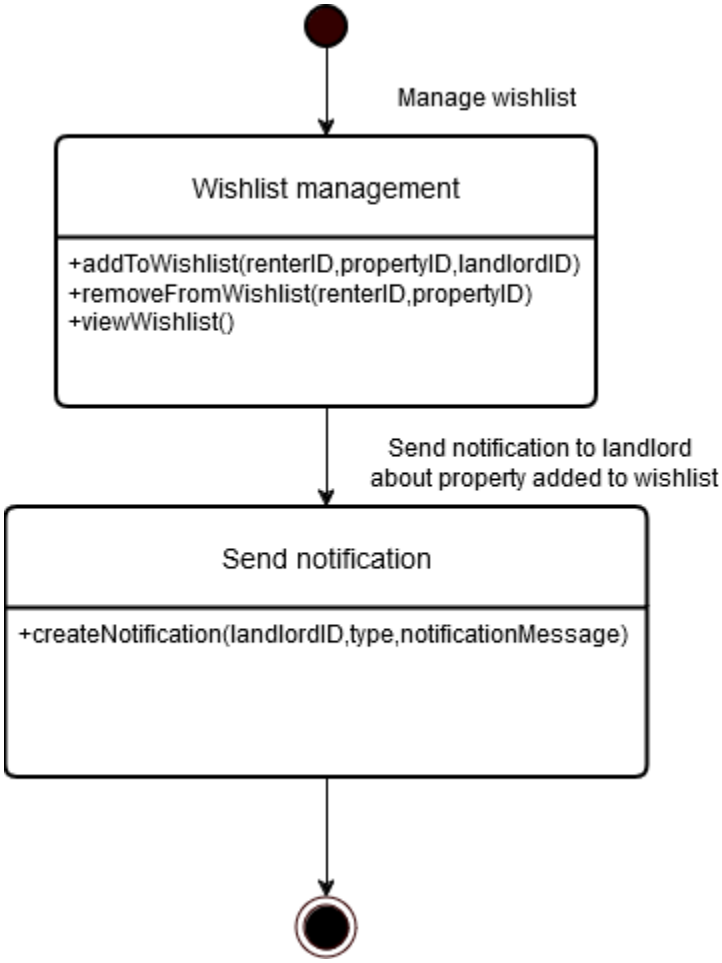
ID:1


provide property criteria
provide contact details
provide property criteria
Search properties
+search()
Add contact details
+addDetails()
Estimate property rent
+estimateRent()
Add property to wishlist
change credentials
Add to Wishlist
+addToWishlist()
Update profile
+updateProfile()
View properties in wishlist
View WIshlist
+viewWishlist()

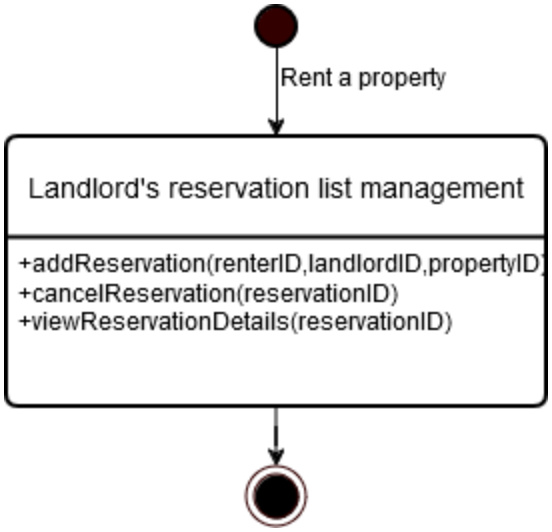
ID:2


provide property criteria
provide contact details
provide property criteria
Add property
Search properties
+search()
Add contact details
+addDetails()
Estimate property rent
+estimateRent()
Host property
+hostProperty()
Add property to wishlist
change credentials
Add to Wishlist
+addToWishlist()
Update profile
+updateProfile()
View property info
Check notification
View list of properties added
View Property
+viewProperty()
View Notification
+viewNotification()
View PropertyList
+viewPropertylist()
Check wishlist
View Wishlist
+viewWishlist()

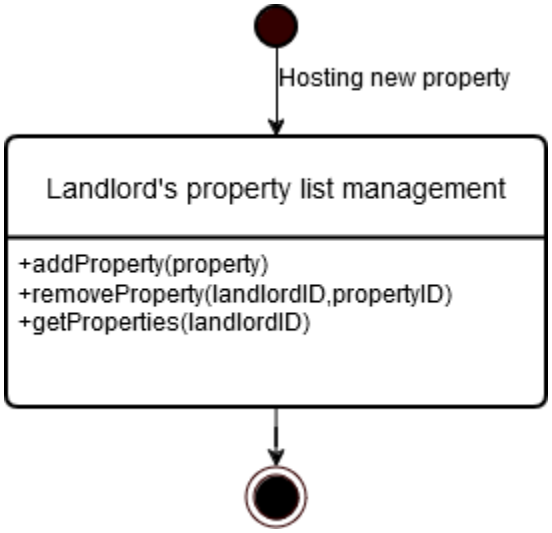
ID:3


Manage wishlist
Wishlist management
+addToWishlist(renterID,propertyID,landlordID)
+removeFromWishlist(renterID,propertyID)
+viewWishlist()
Send notification to landlord
about property added to wishlist
Send notification
+createNotification(landlordID,type,notificationMessage)

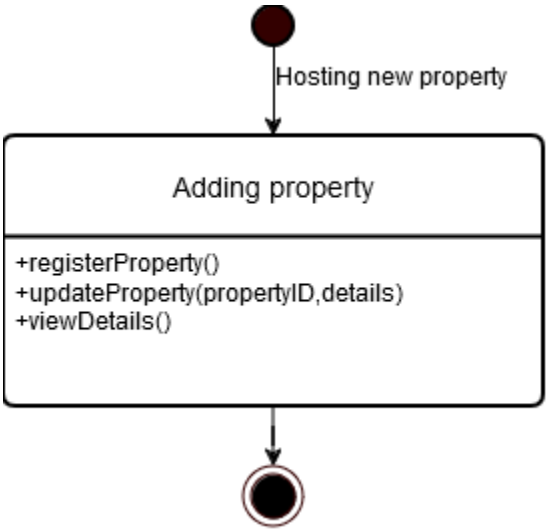
ID:4


Rent a property
Landlord's reservation list management
+addReservation(renterID,landlordID,propertyID)
+cancelReservation(reservationID)
+viewReservationDetails(reservationID)

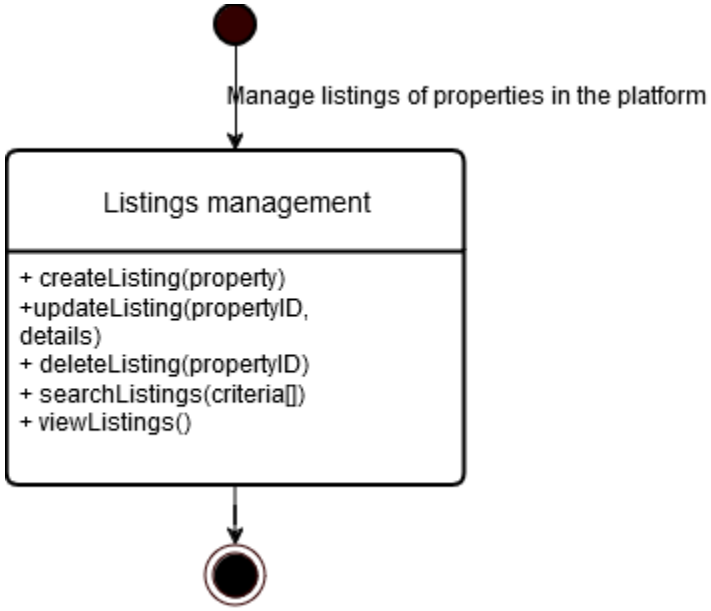
ID:5


Hosting new property
Landlord's property list management
+addProperty(property)
+removeProperty(landlordID,propertyID)
+getProperties(landlordID)

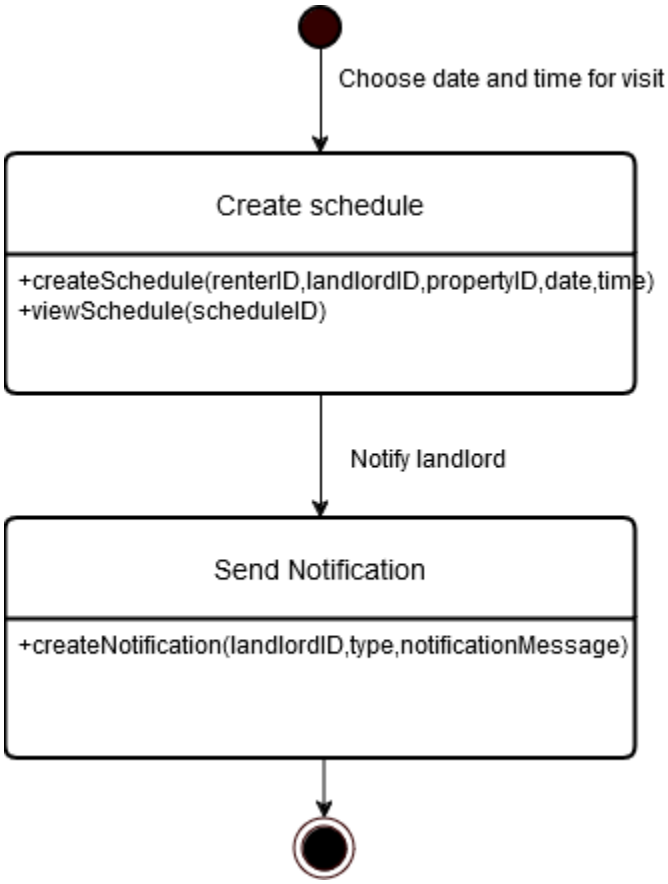
ID:6


Hosting new property
Adding property
+registerProperty()
+updateProperty(propertyID,details)
+viewDetails()

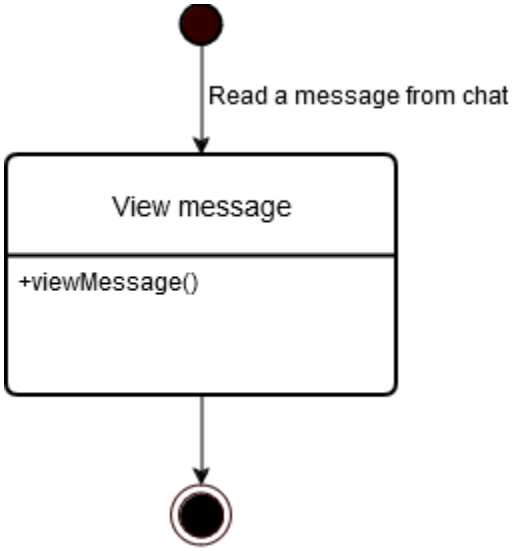
ID:7


Manage listings of properties in the platform
Listings management
+ createListing(property)
+updateListing(propertyID,
details)
+ deleteListing(propertyID)
+ searchListings(criteria[])
+ viewListings()

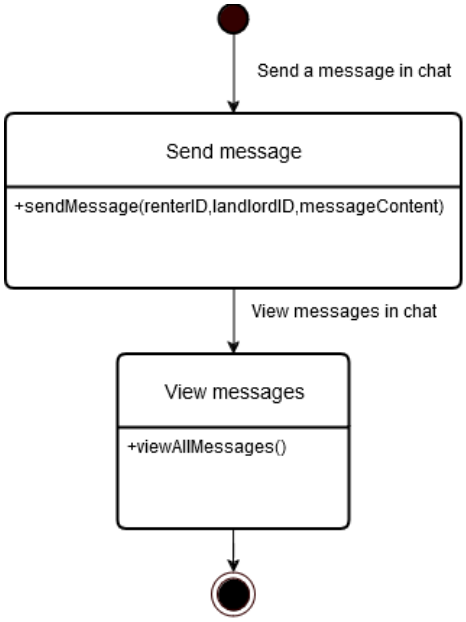
ID:8


Choose date and time for visit
Create schedule
+createSchedule(renterID,landlordID,propertyID,date,time)
+viewSchedule(scheduleID)
Notify landlord
Send Notification
+createNotification(landlordID,type,notificationMessage)

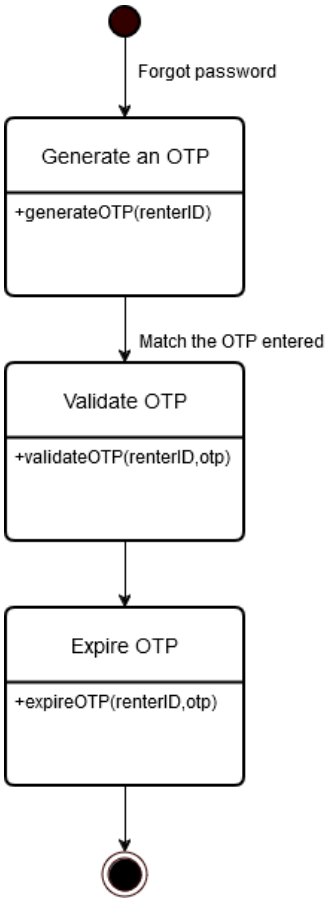
ID:9


Read a message from chat
View message
+viewMessage()

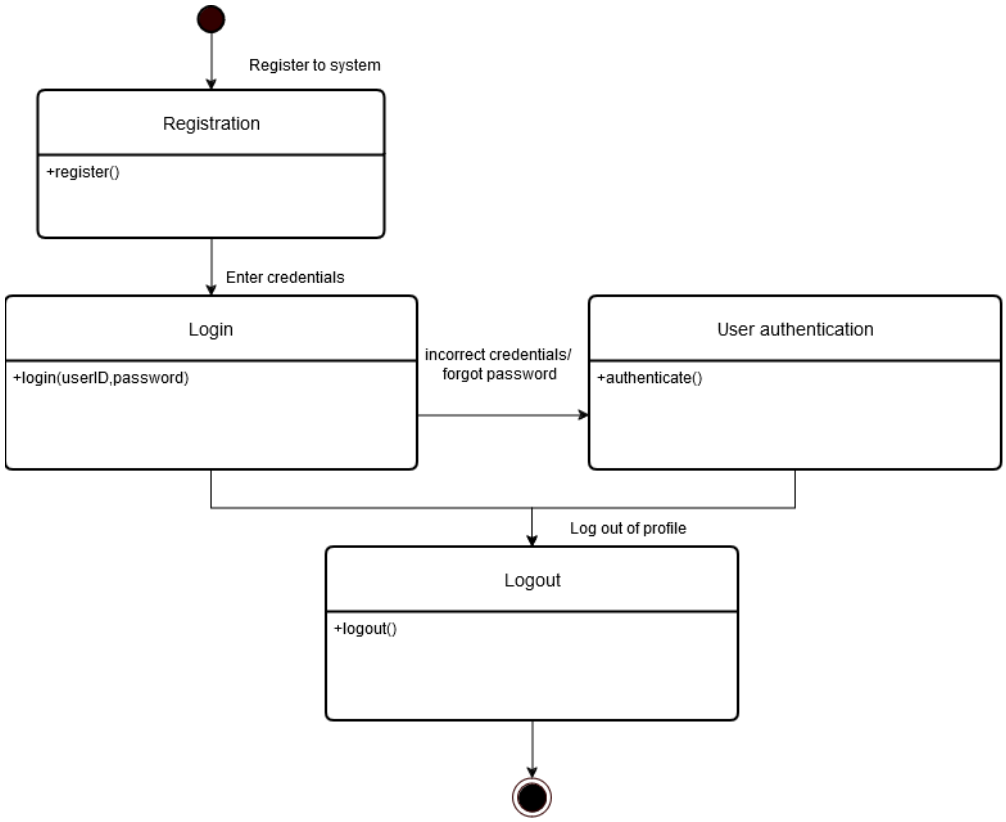
ID:10


Send a message in chat
Send message
+sendMessage(renterID,landlordID,messageContent)
View messages in chat
View messages
+viewAllMessages()

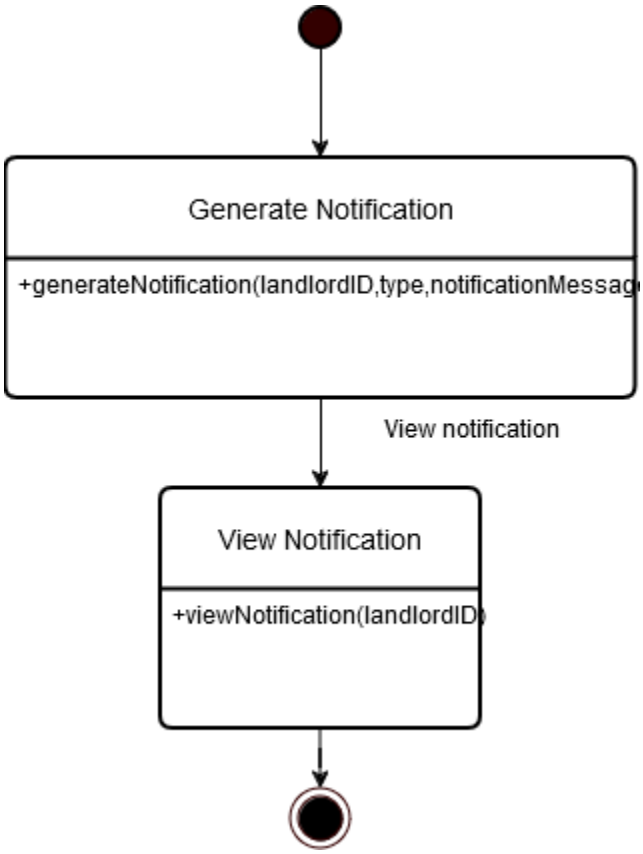
ID:11


Forgot password
Generate an OTP
+generateOTP(renterID)
Match the OTP entered
Validate OTP
+validateOTP(renterID,otp)
Expire OTP
+expireOTP(renterID,otp)

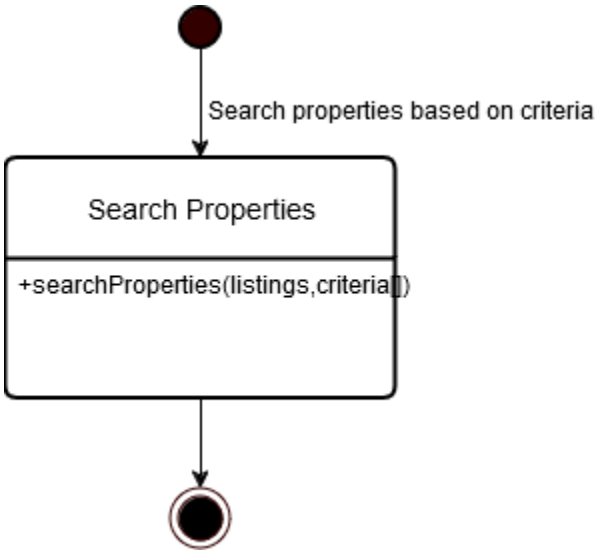
ID:12


Register to system
Registration
+register()
Enter credentials
Login
+login(userID,password)
incorrect credentials/
forgot password
User authentication
+authenticate()
Log out of profile
Logout
+logout()

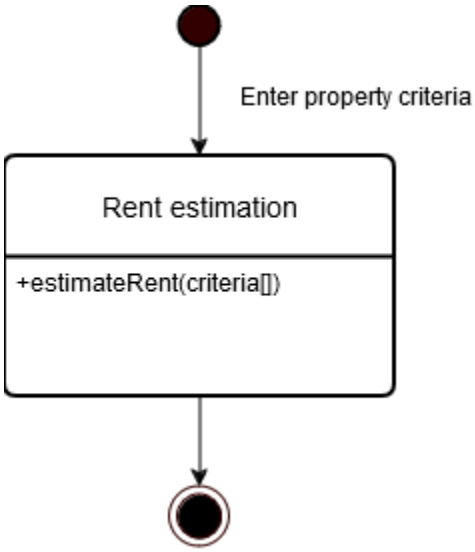
ID:13


Generate Notification
+generateNotification(landlordID,type,notificationMessag
View notification
View Notification
+viewNotification(landlordID

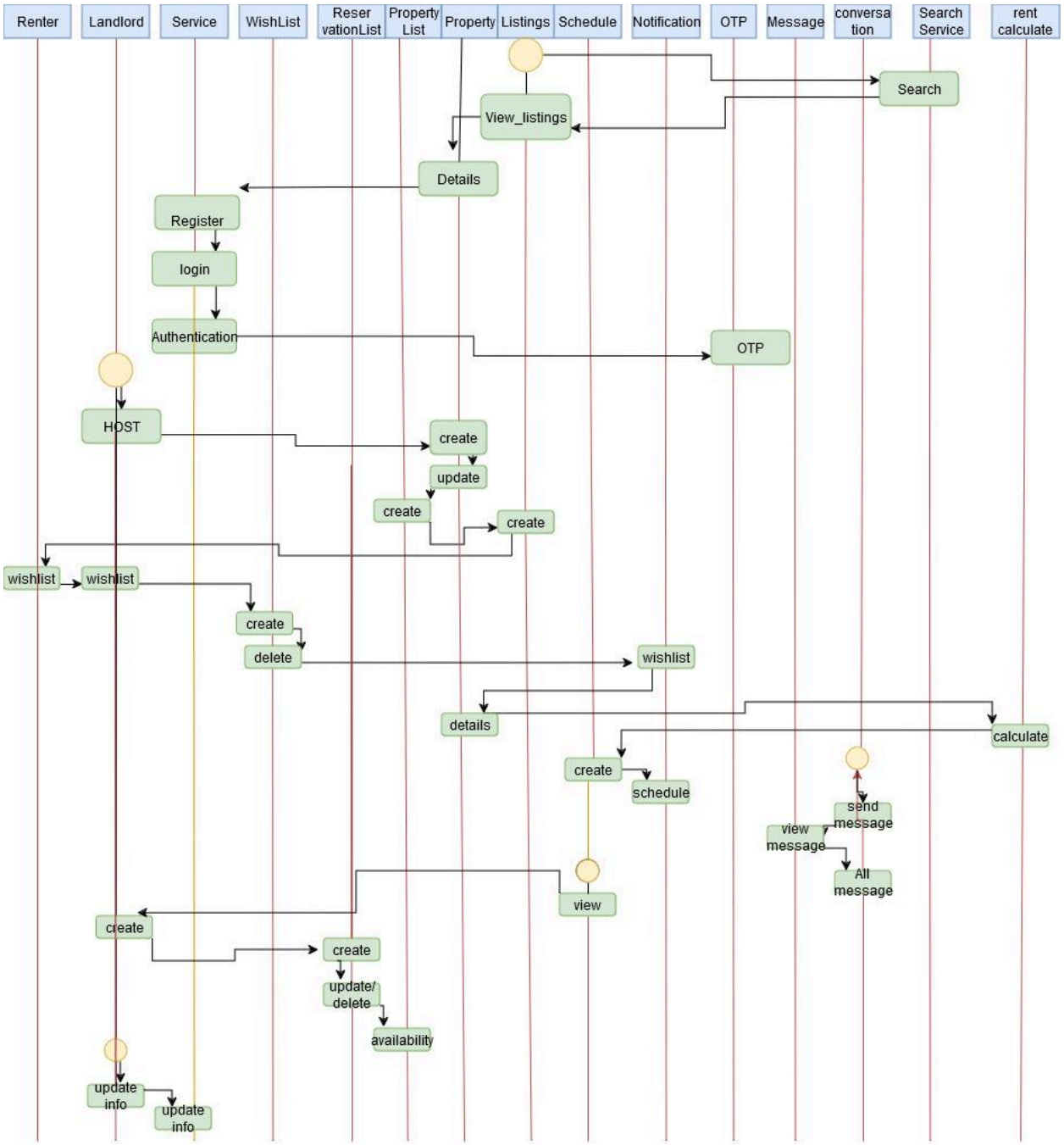
ID:14


Search properties based on criteria
Search Properties
+searchProperties(listings,criteria[])

ID:15


Enter property criteria
Rent estimation
+estimateRent(criteria[])

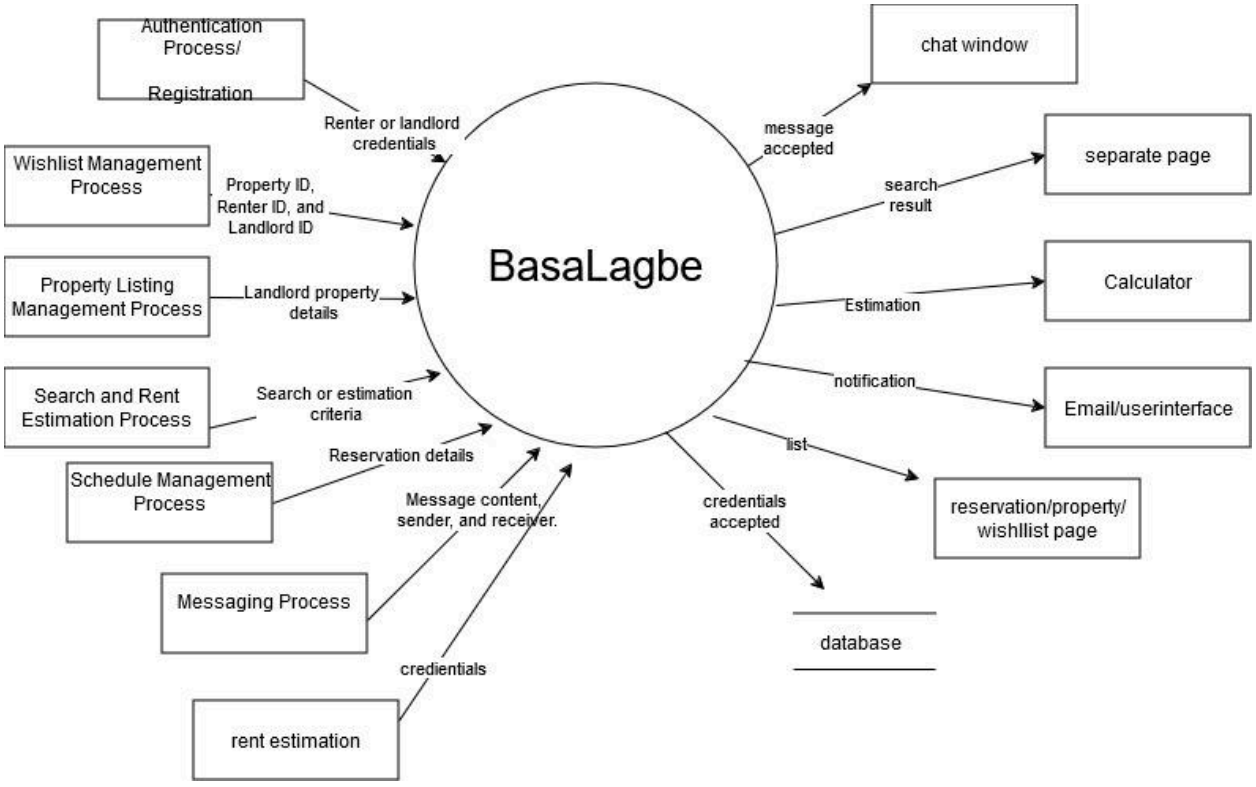
Sequence Diagram



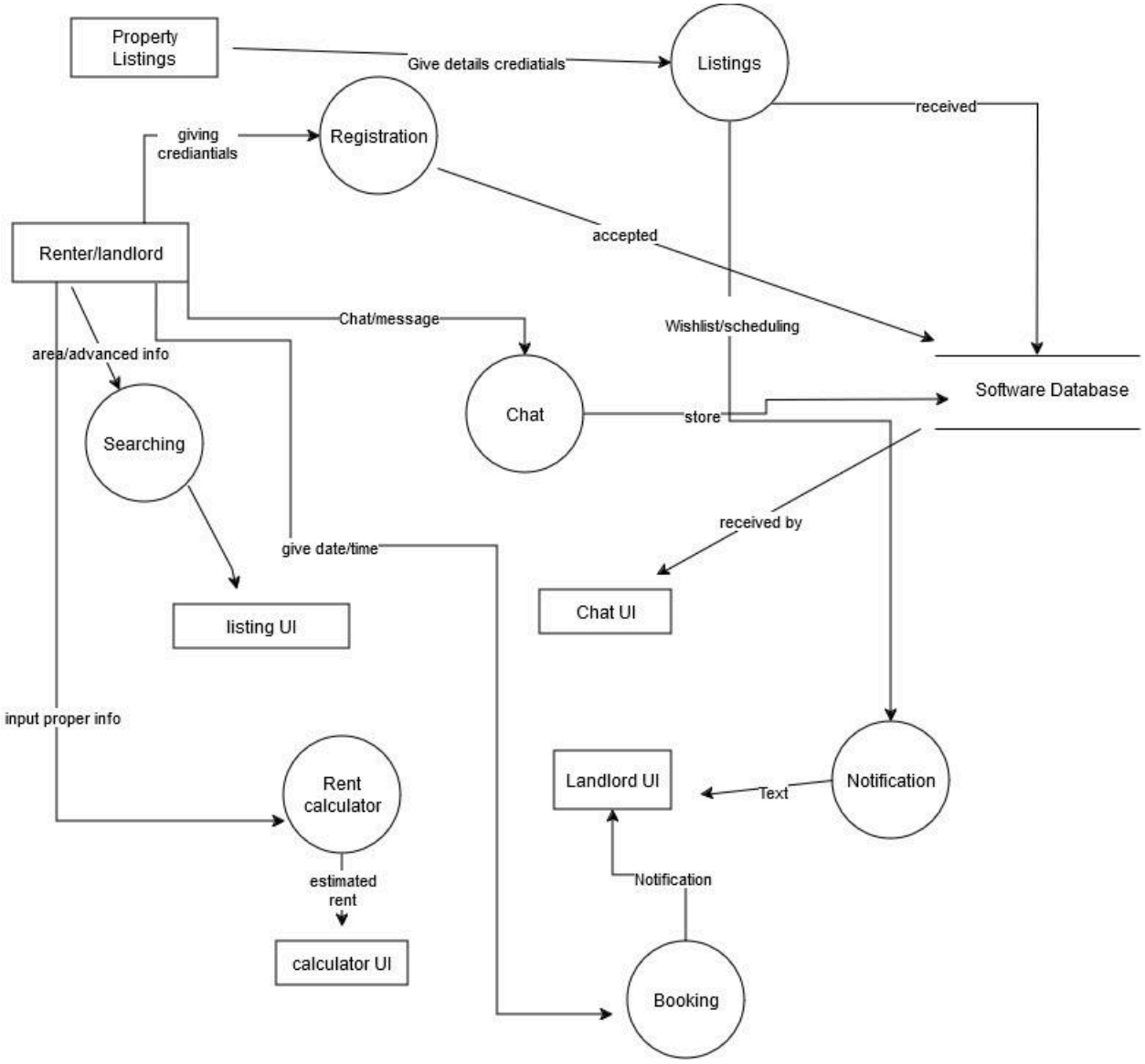
7. Data Flow Diagram

Data Flow Diagram (DFD) represents the flow of data within information systems. Data Flow Diagrams (DFD) provide a graphical representation of the data flow of a system that can be understood by both technical and non-technical users. The models enable software engineers, customers, and users to work together effectively during the analysis and specification of requirements.

LEVEL 0:



LEVEL 1


Property Listings
Give details crediatials
Listings
received
giving crediantials
Registration
accepted
Renter/landlord
Chat/message
Wishlist/scheduling
area/advanced info
Chat
Software Database
store
Searching
received by
give date/time
Chat UI
listing UI
input proper info
Rent calculator
Landlord UI
Text
Notification
estimated rent
Notification
calculator UI
Booking