Design Document

for

Real-Estate Broker

Version 1.0

Prepared by

Group: 14 Group Name: Dangling Pointers

Chinmay Hiran Pillai	200298	chinmay20@iitk.ac.in
Shivang Pandey	200941	shivangp20@iitk.ac.in
Armeet Singh Luthra	200185	armeet20@iitk.ac.in
Arth Banka	200191	arth20@iitk.ac.in
Vaishali Rawat	201080	vaishalir20@iitk.ac.in
Utpal Dwivedi	211132	utpaldwi21@iitk.ac.in
Praveen Raj	210763	praveenr21@iitk.ac.in
Abhishek Kumar	210039	akumar21@iitk.ac.in
Pravallika Mudunuru	220814	pmudunuru22@iitk.ac.in
Gaddam Shiva Leela	220392	gaddamshiv22@iitk.ac.in

Course: CS253

Mentor TA: *Vaibhav*

Date: 09/02/2024

Index

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
1.0	Chinmay Hiran Pillai, Shivang Pandey, Armeet Luthra, Arth Banka, Vaishali Rawat, Praveen Raj, Utpal Dwivedi, Pravallika Mudunuru, Gaddam Shivaleela	Design Document for Real-Estate Broker	8/2/24

1 Context Design

1.1 Context Model

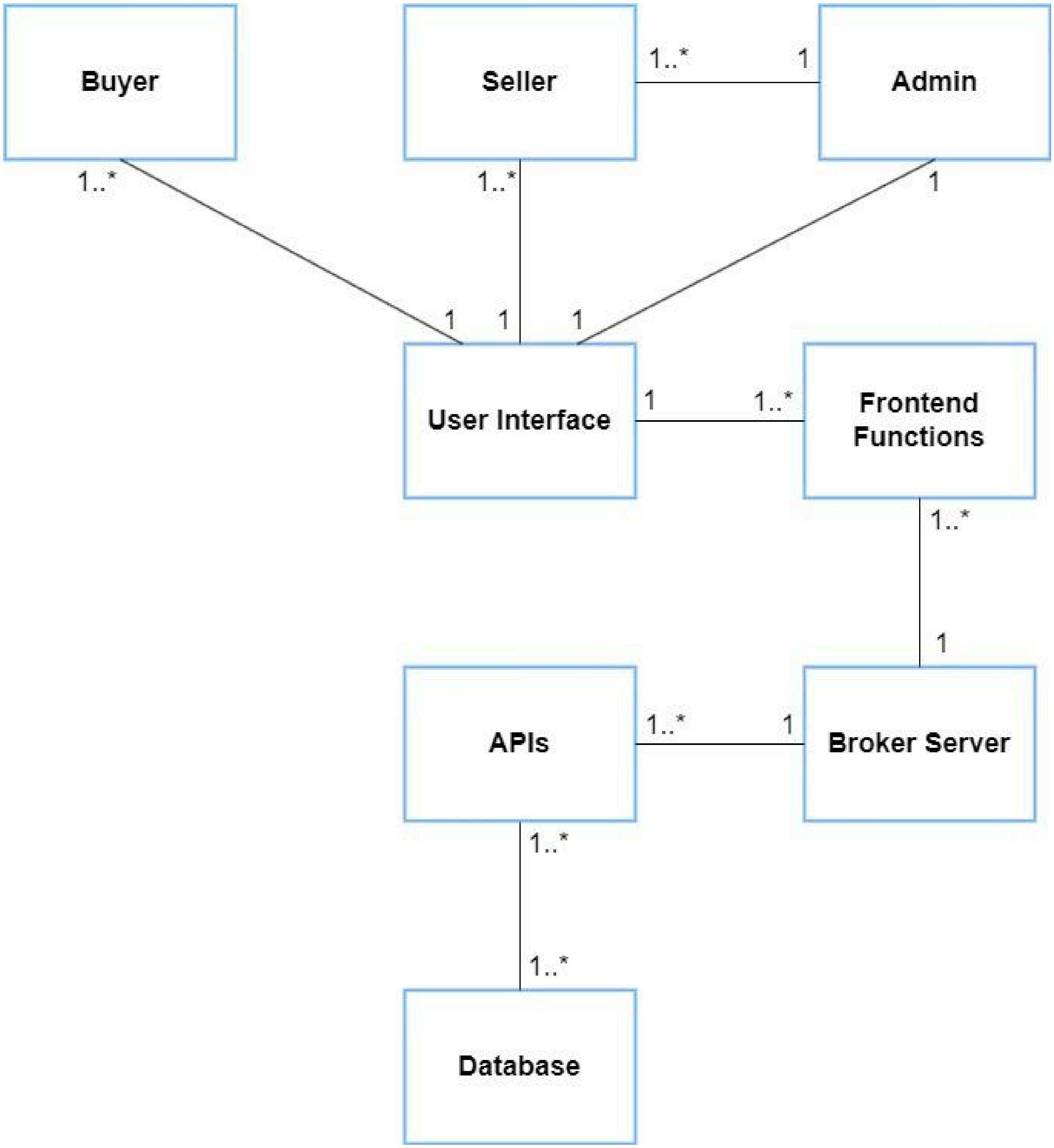
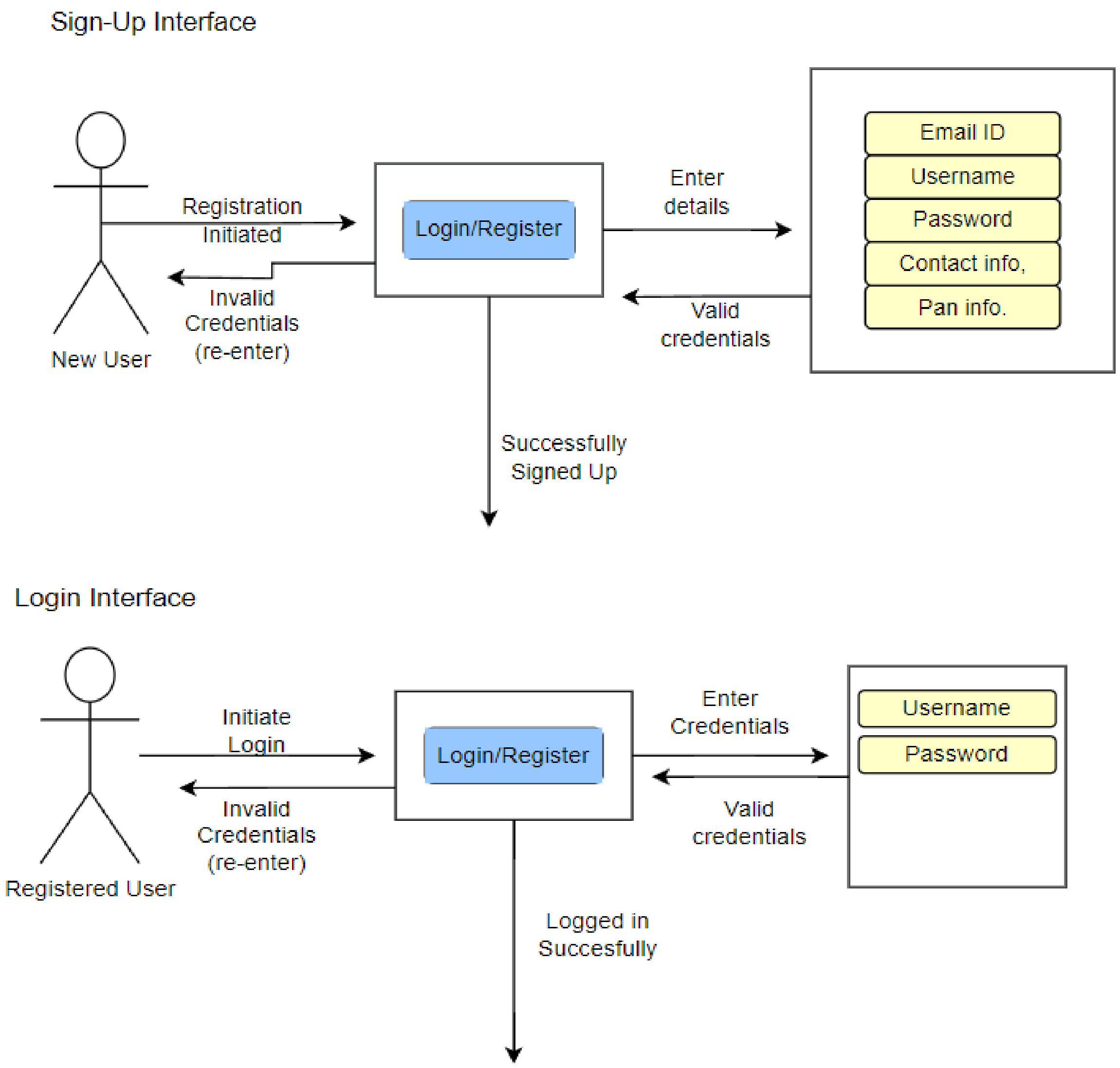


Fig 1.1 – Context Model of the System

1.2 Human Interface Design

1.2.1 Log In and Sign-up Interface:

A new user can register themselves via the sign-up page, this requires one to enter rudimentary details (username, password and contact info) as well as their PAN information. After external e-KYC authentication, the user is registered with the website and in case of erroneous details, they are prompted to re-enter their credentials.
A registered user can now easily access their account by entering their username and password.


Sign-Up Interface

Login Interface

Fig 1.2 a) Log in/ Sign up Interface

1.2.2 Home Page and Navigation Bar

The user is now transported to the home page. This page has a bird's eye view of all the properties listed for sale as well as an easy access navigation bar that can be used to implement a variety of actions. The listed properties have a view option for any interested user which takes one to a more exclusive property details page.

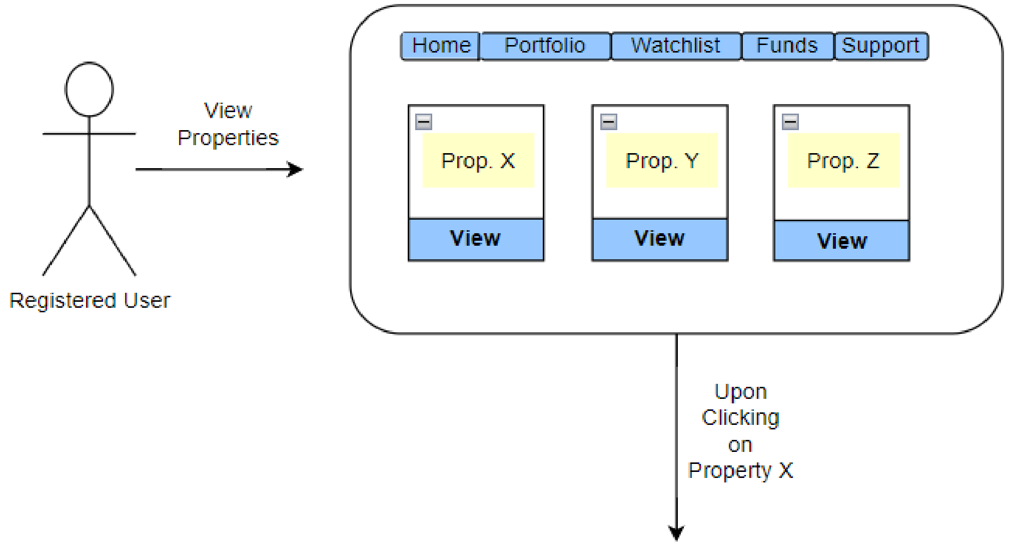
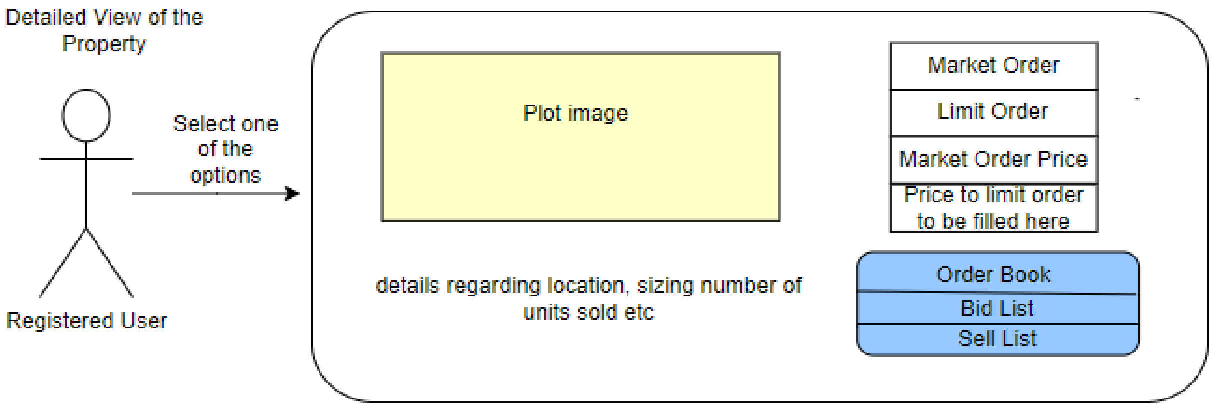


Fig 1.2 b) Home Page

1.2.3 Property Details

The detailed view of the property, apart from having a picture of the plot also consists of details like location, sizing, no. of units bought and for sale et all. The user, depending on his/her interest may choose to add the property to watchlist or use the available buy/sell options. This page also consists of an order book which displays the top five buy bids and the five lowest sell bids on the given property. This will help the user make informed decisions about trading keeping in mind the current state of the market.



1.2.4 Option to Limit Order:

The user may choose to set a buy/sell value on a given property by using the limit order option. This prompts the user to fill in the desired price. In case of a buy order, if the user has insufficient funds in their account, they will be alerted about the same and only after the addition of funds (this by accessing their funds from the navigation bar on the home page) will they be allowed to exercise this option. Consequently, the property is added to/ removed from the user's portfolio and their current funds are suitably altered.

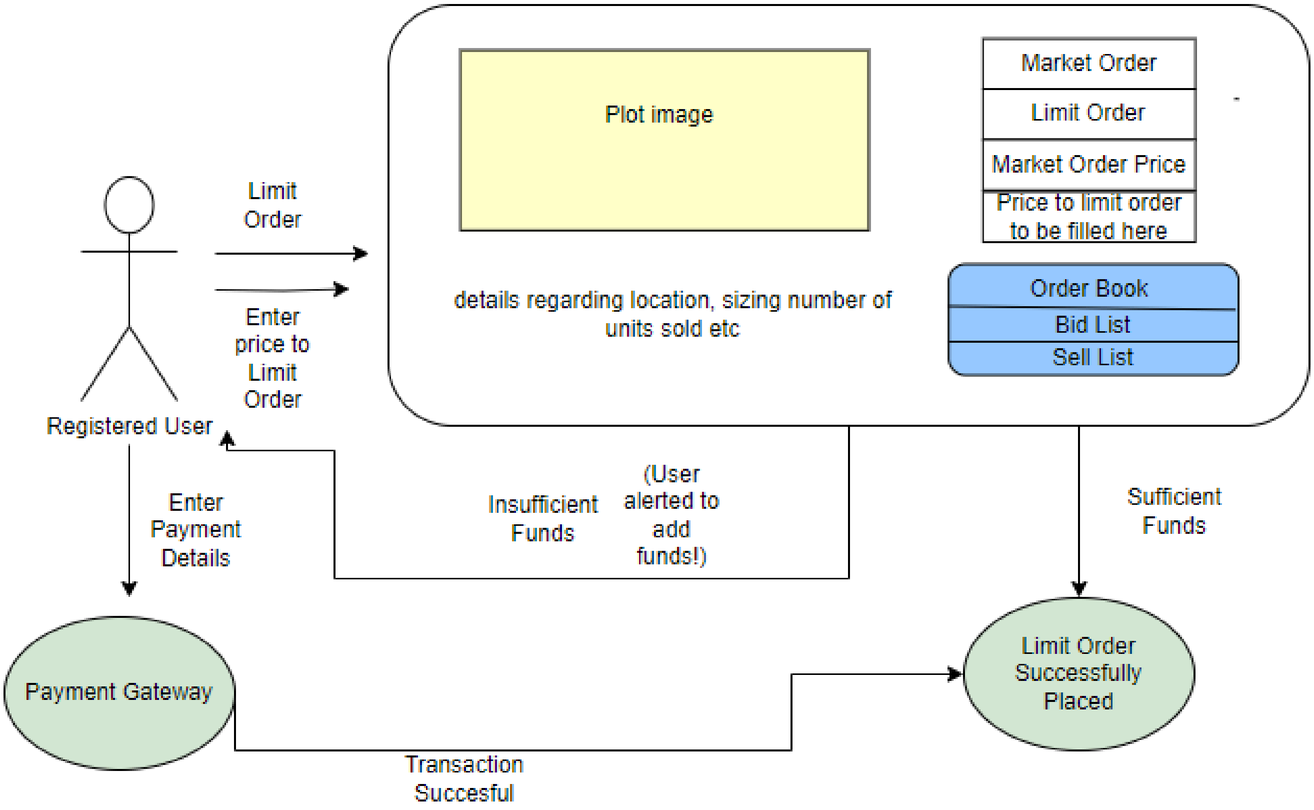


Fig 1.2 d) Limit Order

1.2.5 Option to Market Order:

This option allows the user to buy/sell a property at the current market price, the interface is exactly like the limit order option except in that the user is not expected to fill a form to set the price.

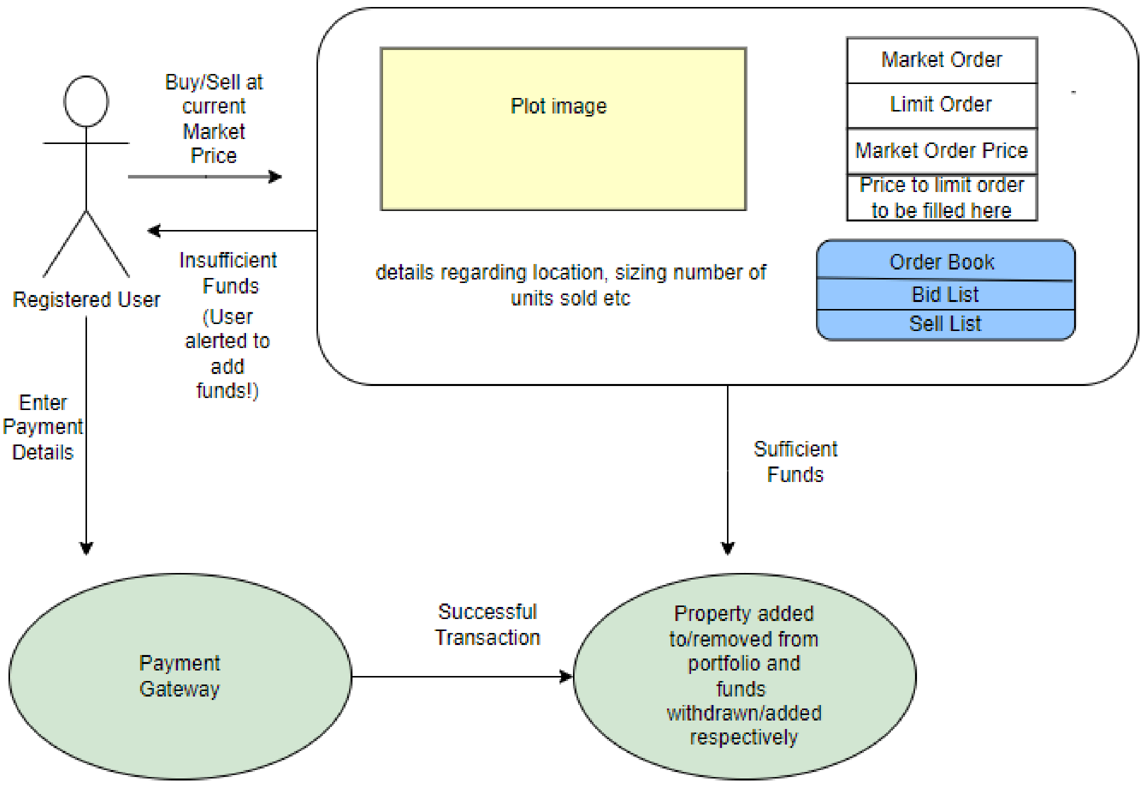

Buy/Sell at current Market Price
Registered User
Insufficient Funds
(User alerted to add funds!)
Enter Payment Details
Plot image
details regarding location, sizing number of units sold etc
Market Order
Limit Order
Market Order Price
Price to limit order to be filled here
Order Book
Bid List
Sell List
Sufficient Funds
Payment Gateway
Successful Transaction
Property added to/removed from portfolio and funds withdrawn/added respectively

Fig 1.2 e) Market Order

1.2.6 Navigation Bar:

The home page navigation bar has a variety of options for the user to modify and view their account statistics.

- *Portfolio*: The portfolio page displays all the properties currently owned by the customer. Clicking on a certain property would take the user to its detailed view from where they might choose to implement the sell option if need be.

- *Watchlist*: The watchlist will showcase all the properties with their current status that a user has added to the watchlist in the past.

- *Funds*: This page may be used to view current available funds a well as add (this, through a payment gateway) and withdraw funds from their account whenever needed.

- *Support*: This feature is added for user convenience in addressing queries as well and resolving problems by providing the contact information (mailing address, customer care helpline, etc.) to the user.

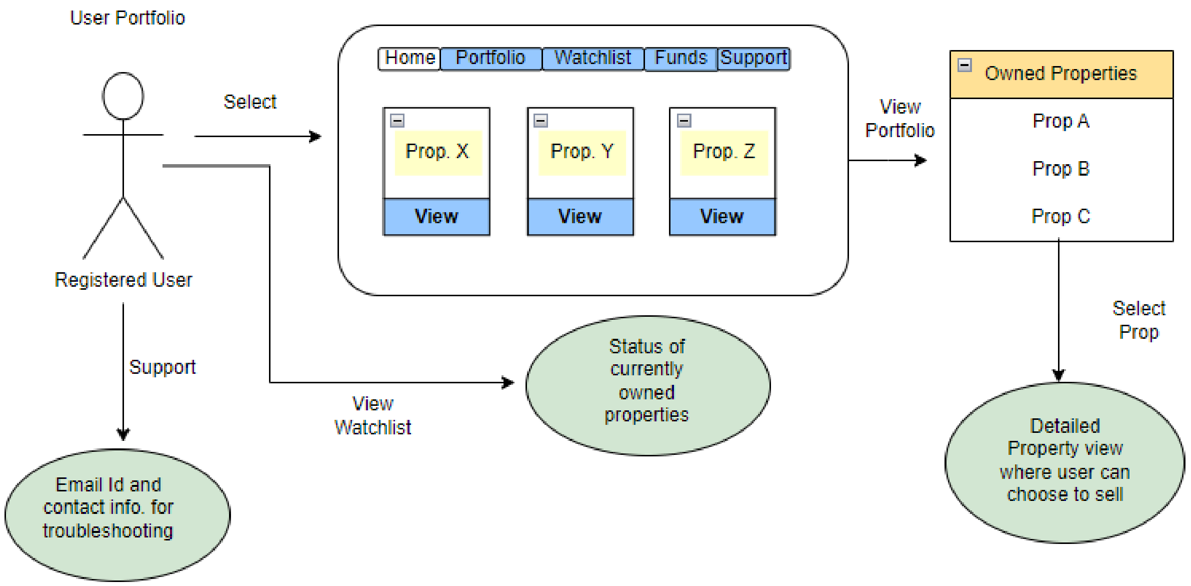


Fig 1.2 f) Portfolio, Support and Watchlist Options

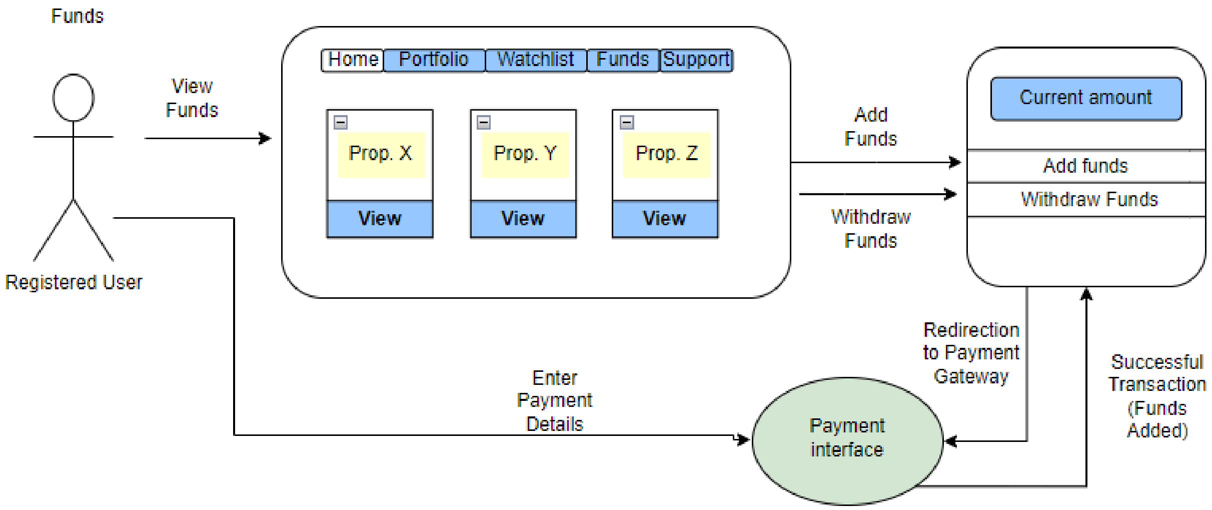


Fig 1.2 g) User Funds

2 Architecture Design

We will utilise a layered architecture for our system. This is an appropriate model for our system since it enables different developer teams to develop all layers simultaneously without worrying about the other layers. This will improve efficiency, prevent bottlenecks and save development time.

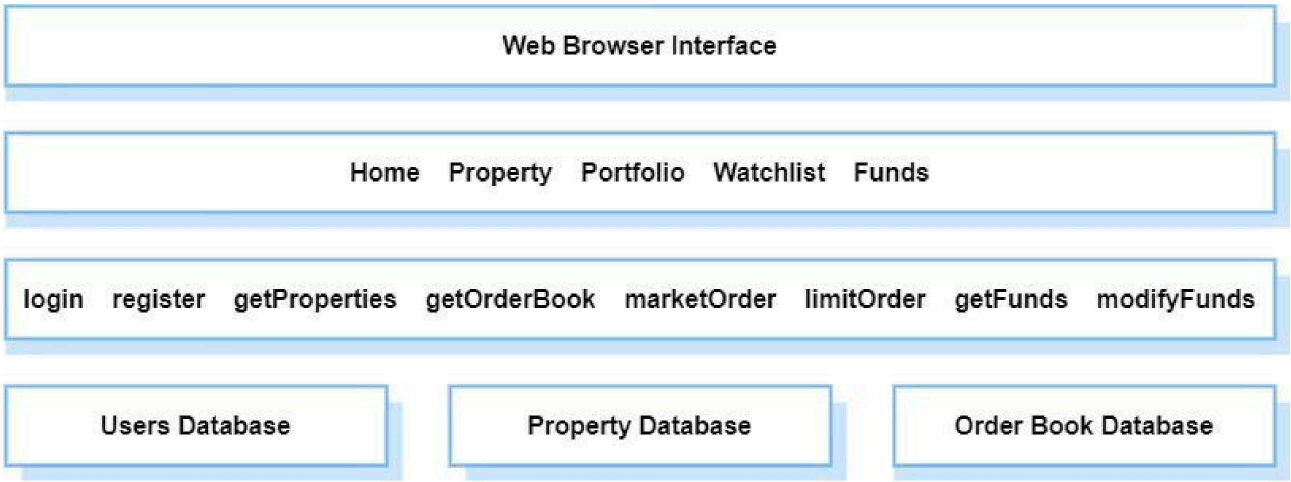

Web Browser Interface
Home Property Portfolio Watchlist Funds
login register getProperties getOrderBook marketOrder limitOrder getFunds modifyFunds
Users Database
Property Database
Order Book Database

Fig 2.1 – Layered Architecture Model of the System

The first layer is the web browser. The user's interaction with the browser calls and displays the appropriate frontend component from the second layer. Once on a specific component, specific interactions with the frontend will result in API (Application Programming Interface) calls made through HTTP protocol from the client to the server. The corresponding API in the third layer will respond to this request by first making a write/query request to the appropriate database i.e. the fourth layer, processing the data returned by the database and then respond to the frontend layer with the appropriate response.

This architecture will enable us to meet our non-functional requirements in the following ways:

1. Data Protection*: The one and only entry point to the database will be through the secure server.*
2. Reliability*: The clear separation between the different layers enables easier debugging reducing potential downtime.*
3. Atomicity in Database: Since the database is a separate layer, we can utilise a SQL/NoSQL database with inbuilt atomicity.

3 Object Oriented Design

3.1 Use Case Diagrams

3.1.1 Types of relationships:

1) <<include>>*:* It indicates that a use case includes the functionality of another use case. It is denoted by a dashed arrow pointing from the including use case to the included use case. Whenever a use case always follows from the implementation of another use case they are related by include relationship.

2) <<extend>>*:* It illustrates that a use case can be extended by another use case under specific conditions. It is represented by a dashed arrow with the keyword "extend." This relationship is useful for handling optional or exceptional behavior.

3) Generalization*:* It indicates that one use case is a specialized version of another. It is represented by an arrow pointing from the specialized use case to the general use case.

4) Association*:* This relationship signifies that the actor is involved in the functionality described by the use case.It is depicted by a line connecting the actor to the use case.

3.1.2 Diagrams

1) Login Page:

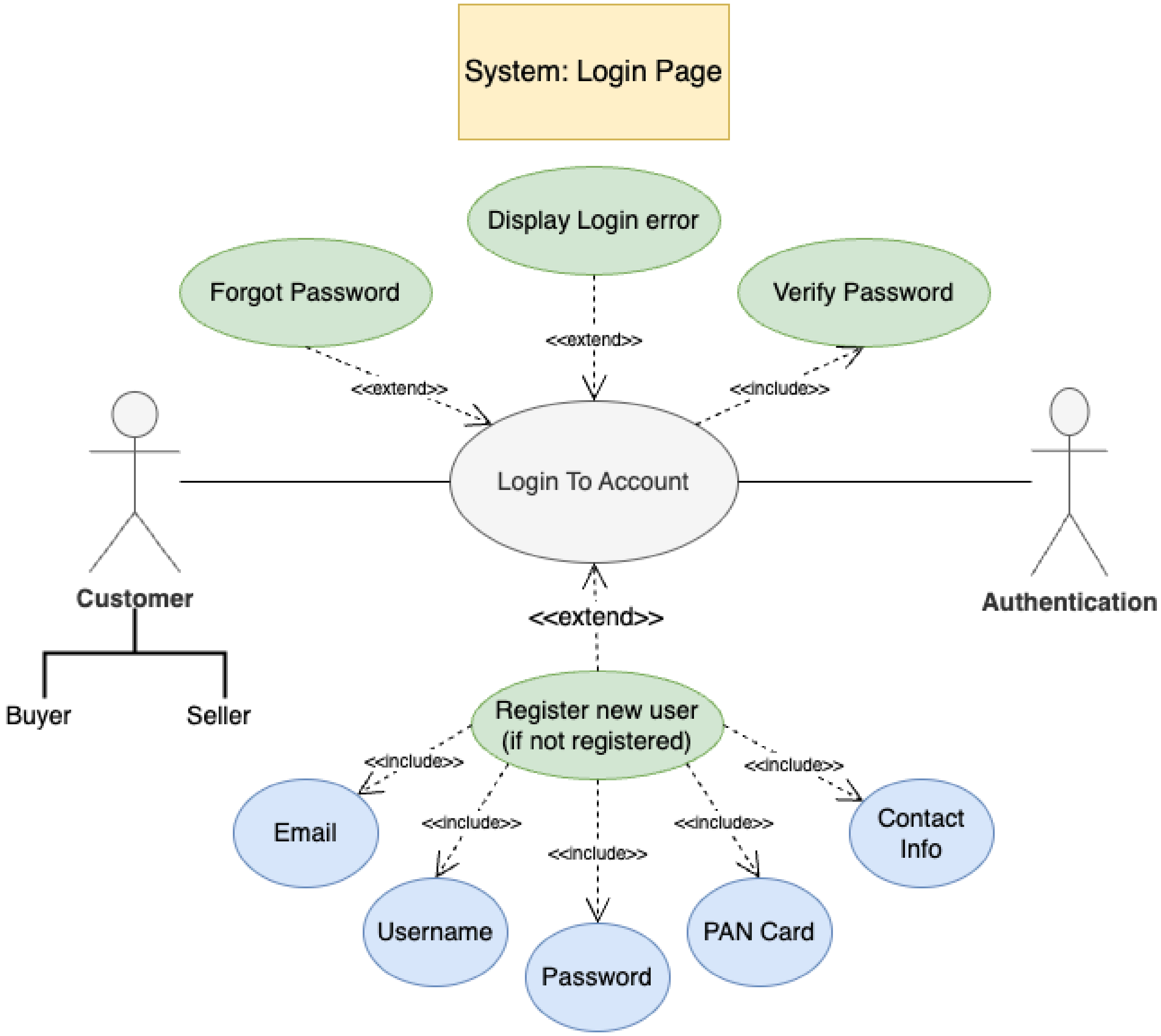


Fig. 3.1 a) Login Page

2) User Dashboard

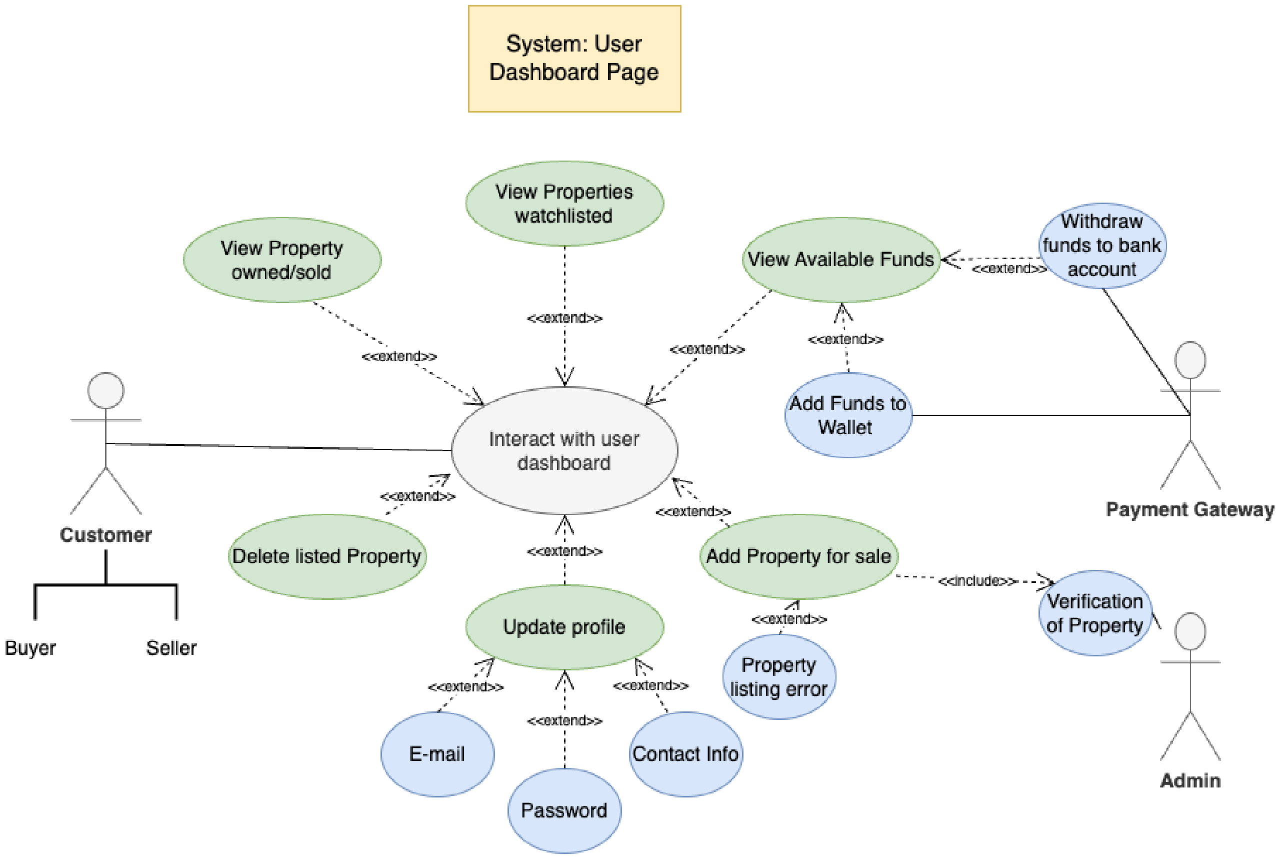


Fig. 3.1 b) User Dashboard

3) Home Page

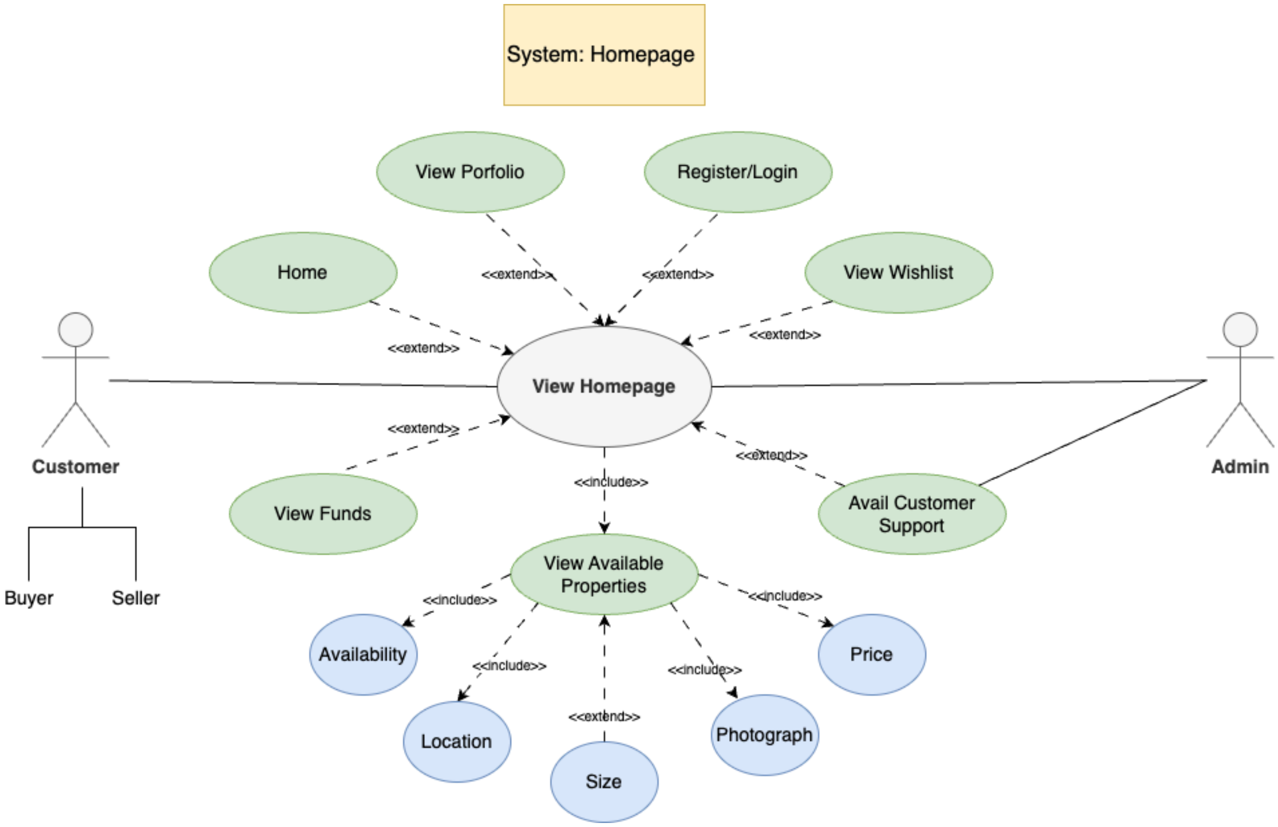


Fig. 3.1 c) Home Page

4) Property Page

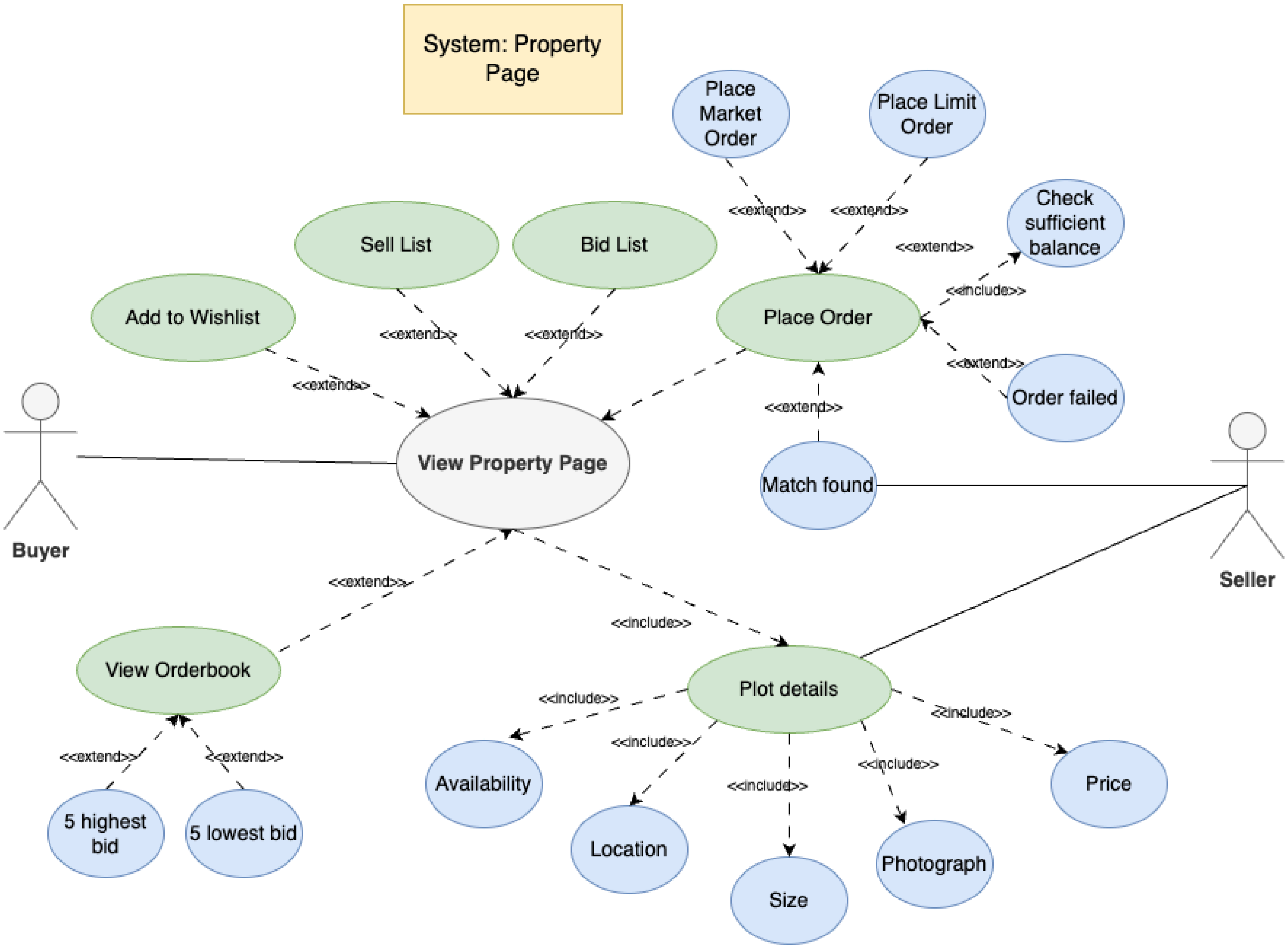


Fig. 3.1 d) Property Page

5) Support Page

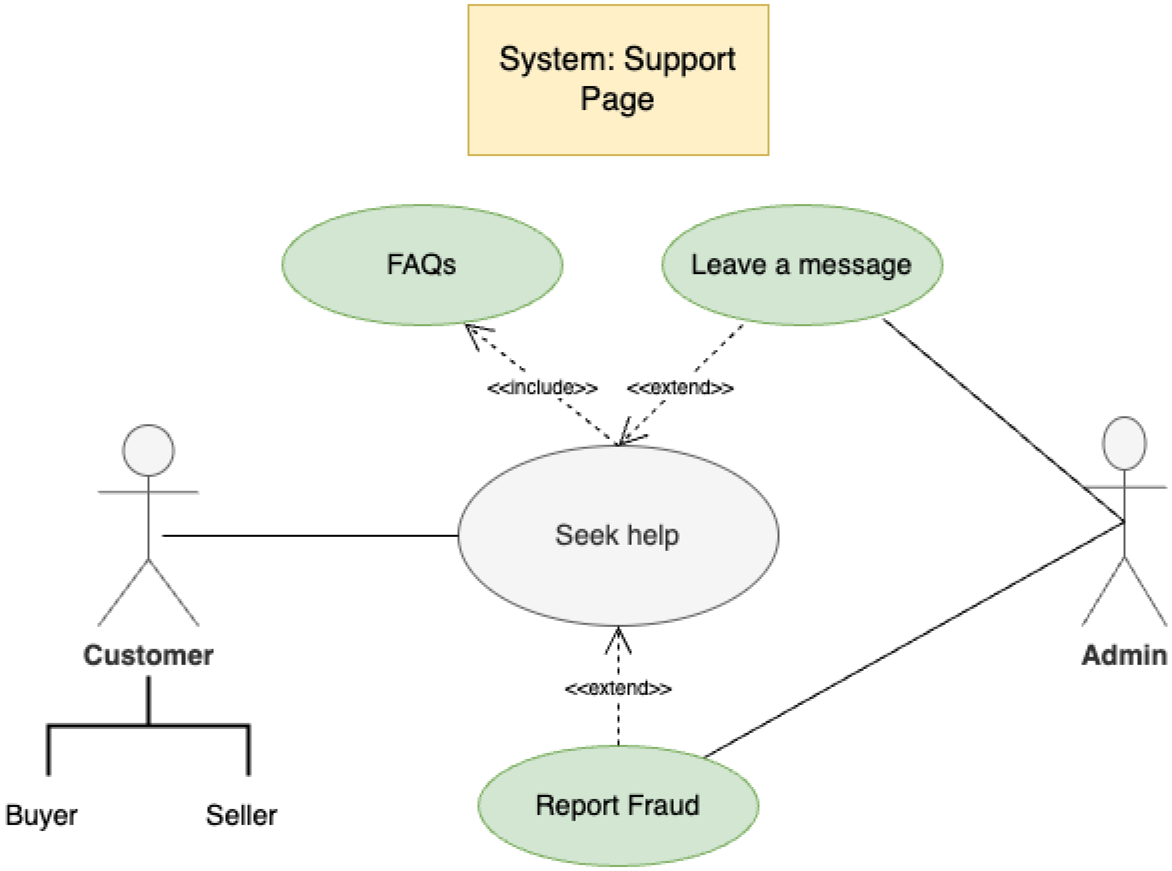
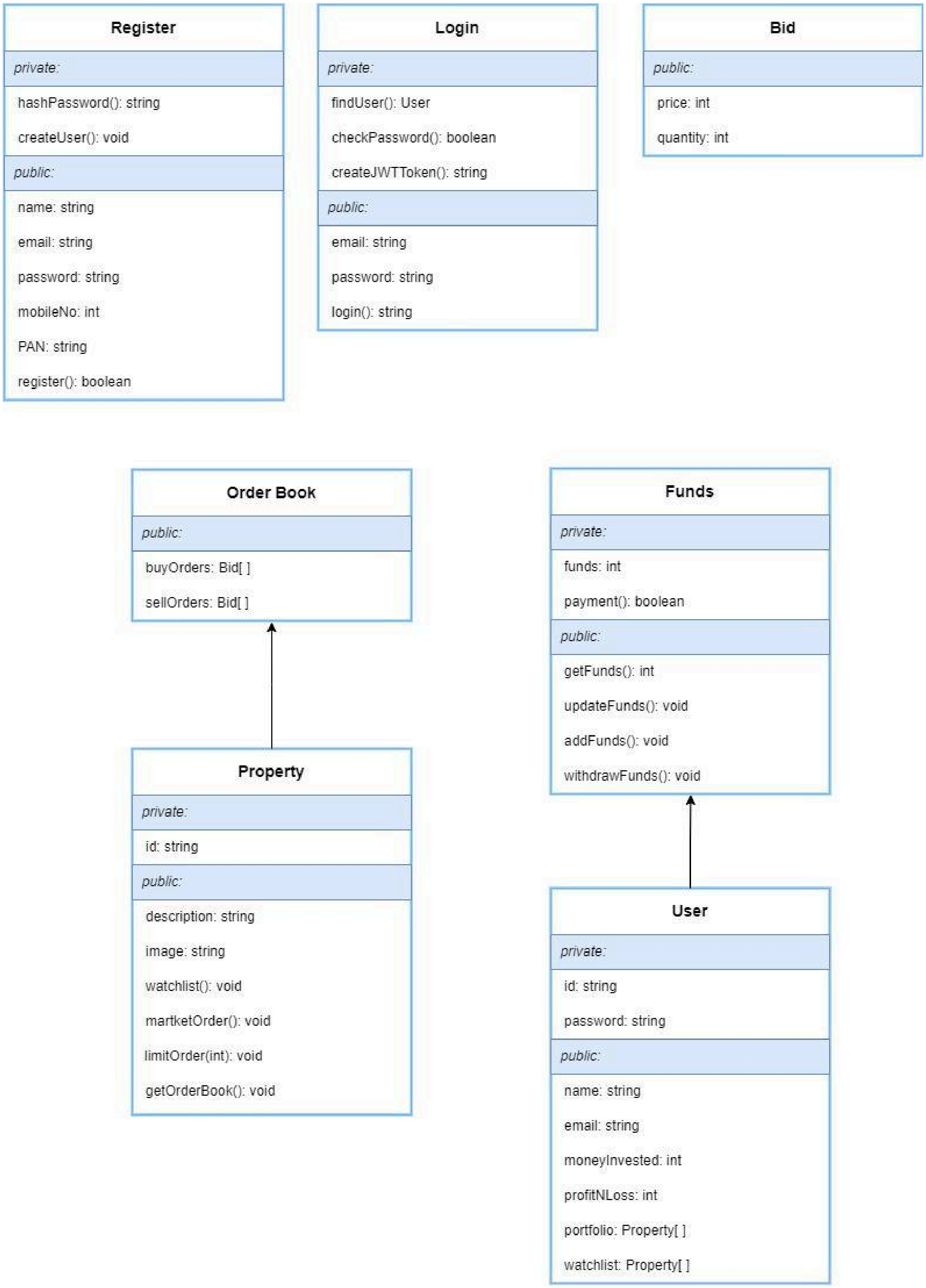
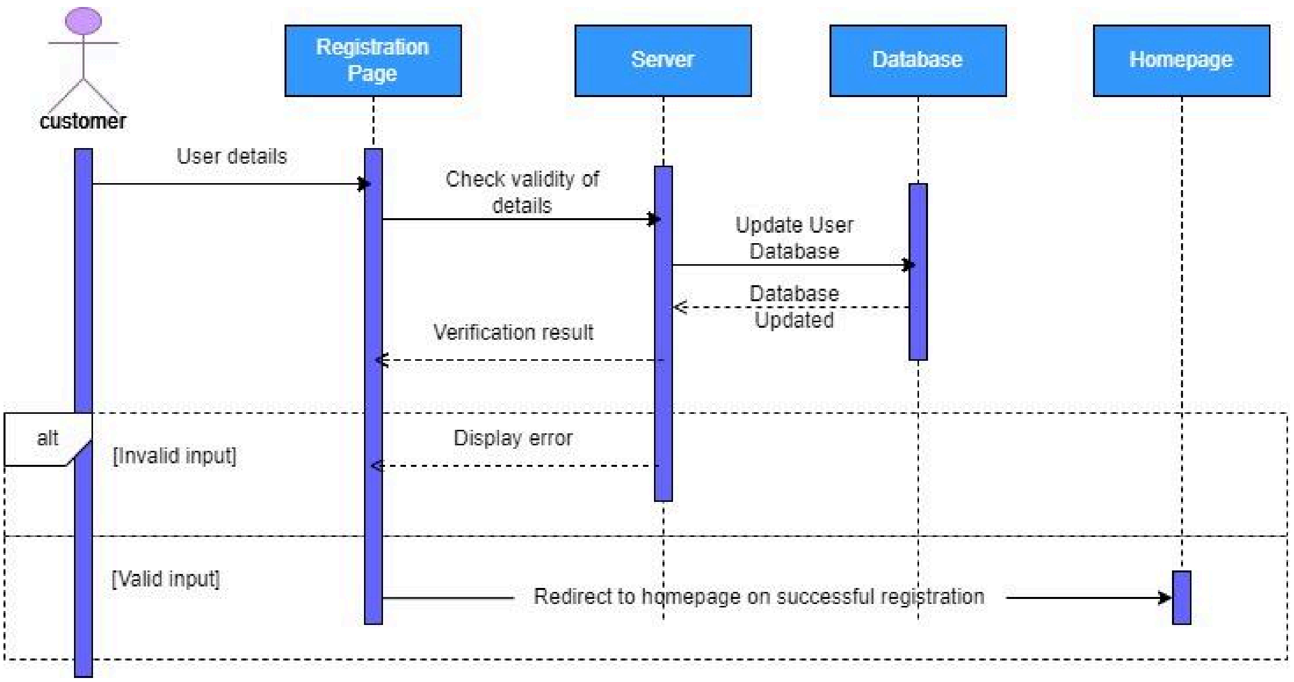


Fig. 3.1 e) Support Page

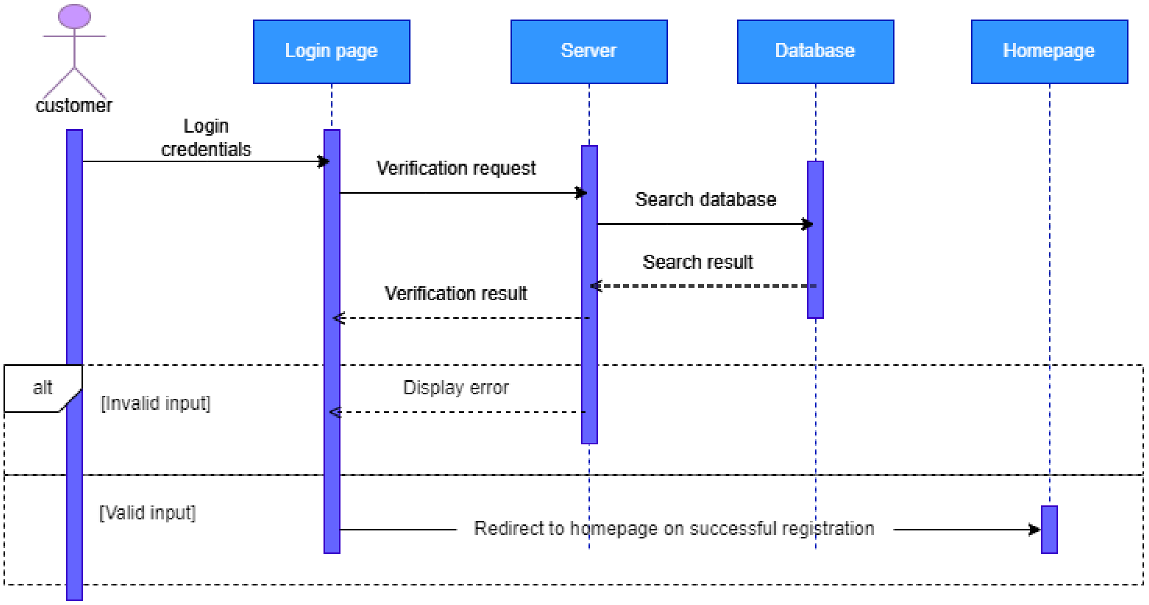
3.2 Class Diagrams



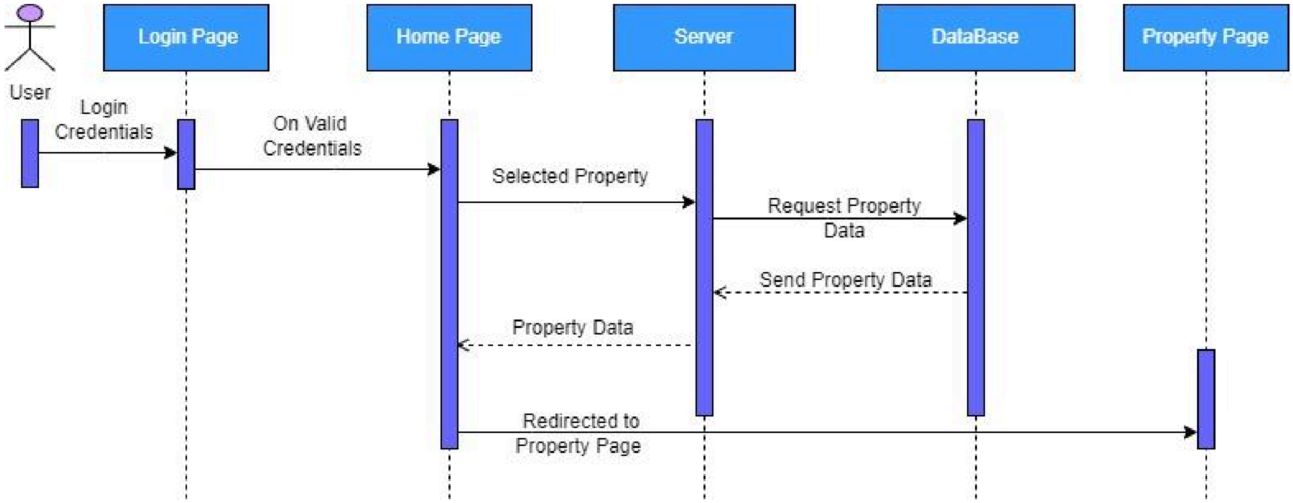
3.3 Sequence Diagrams



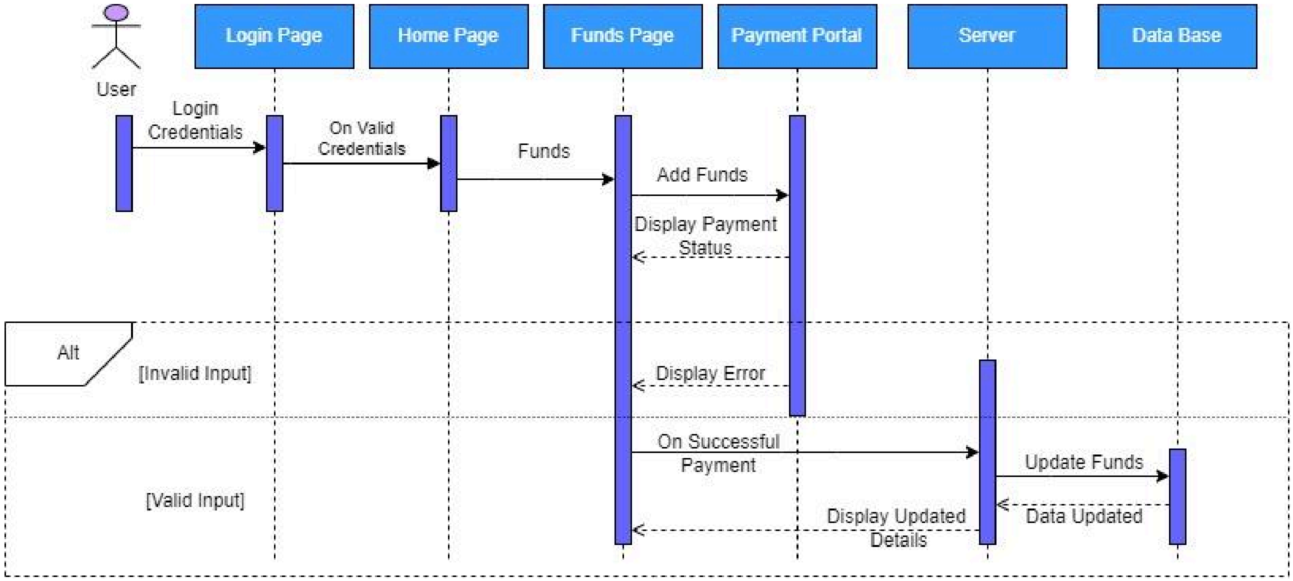
Sequence of Registration



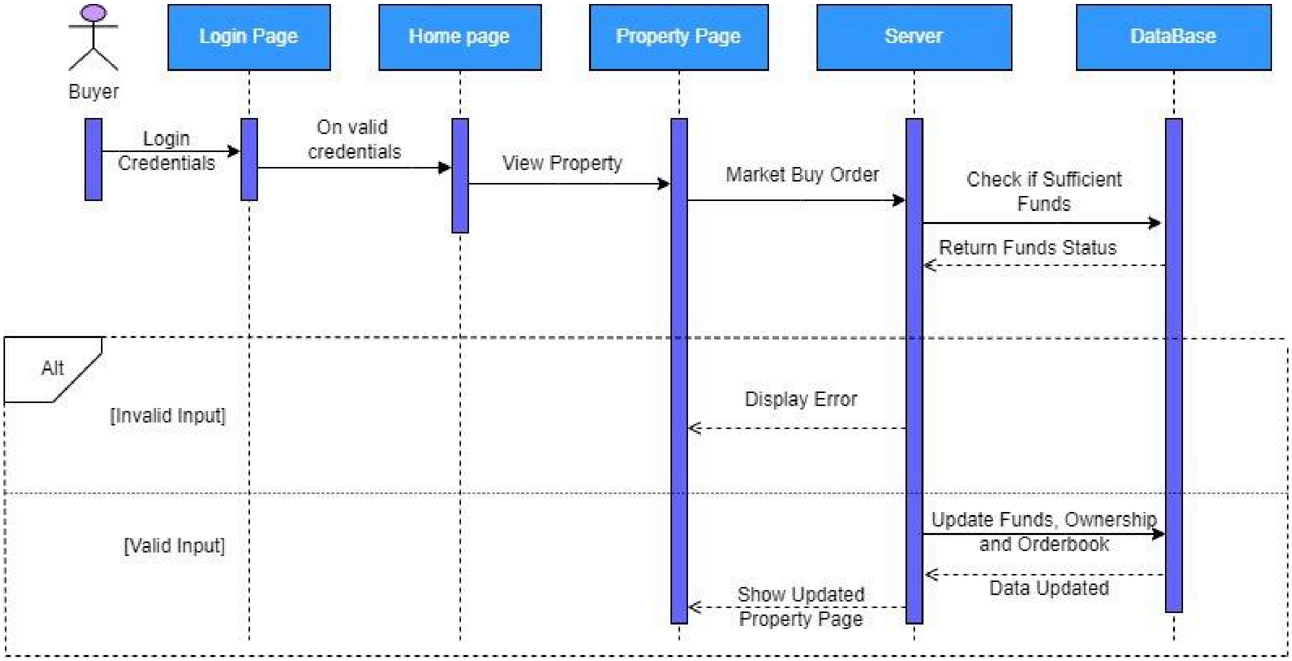
Sequence of Login



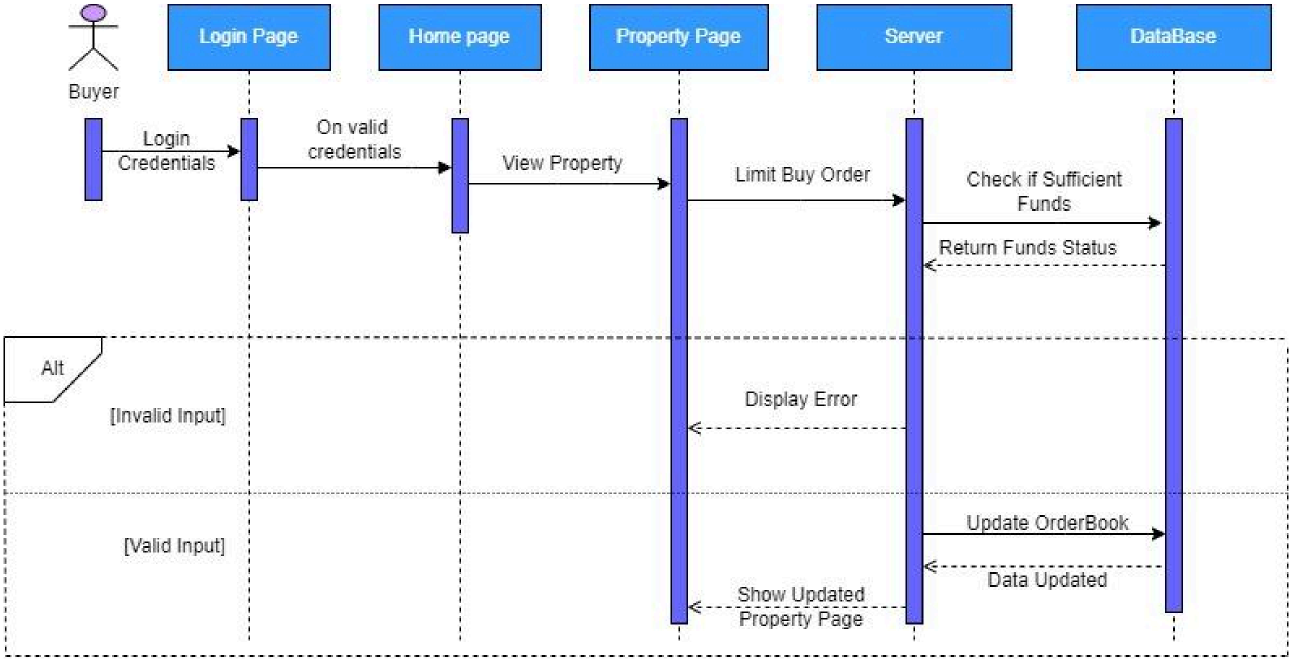
Sequence of viewing property page



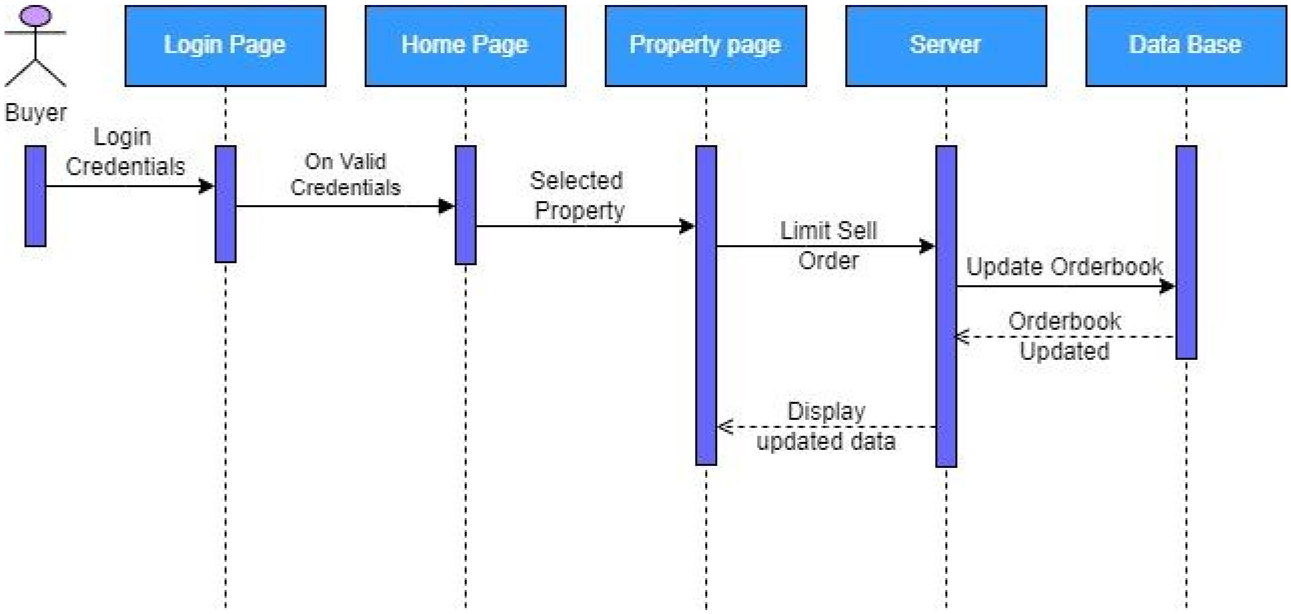
Sequence of adding funds



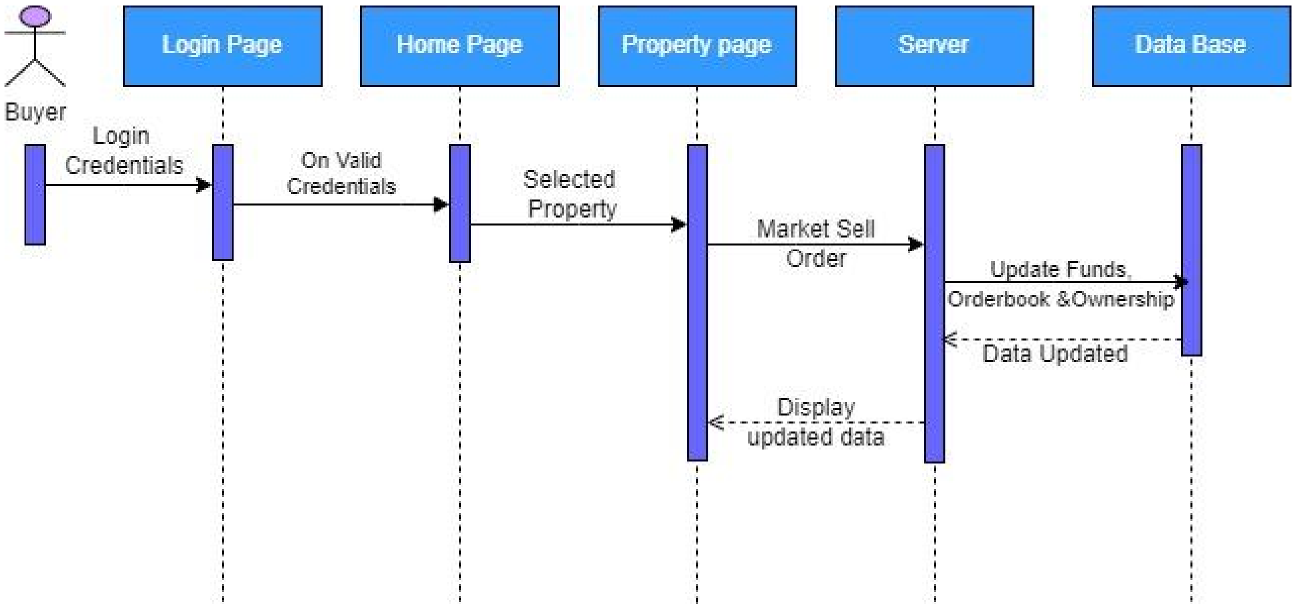
Sequence of Market order buy



Sequence of Limit order buy



Sequence of Limit order sell



Sequence of Market order sell

3.4 State Diagrams

3.4.1 Login/Register

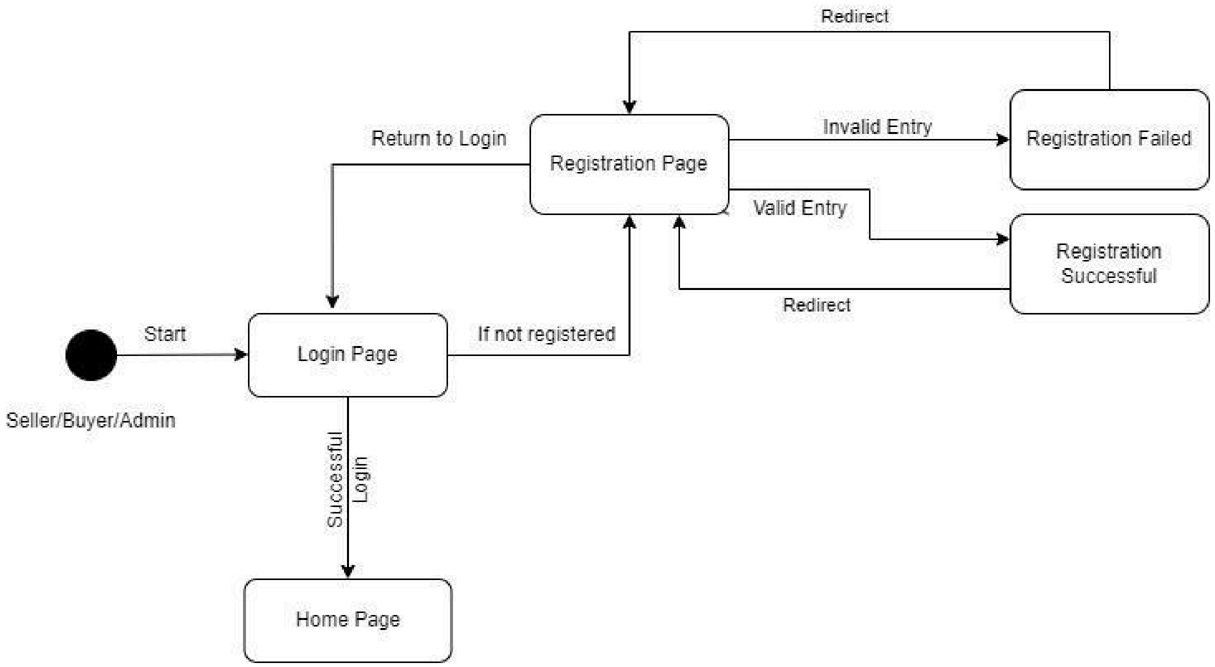


Fig 3.4.1 - Login/Register State Diagram

3.4.2 New Property Registration

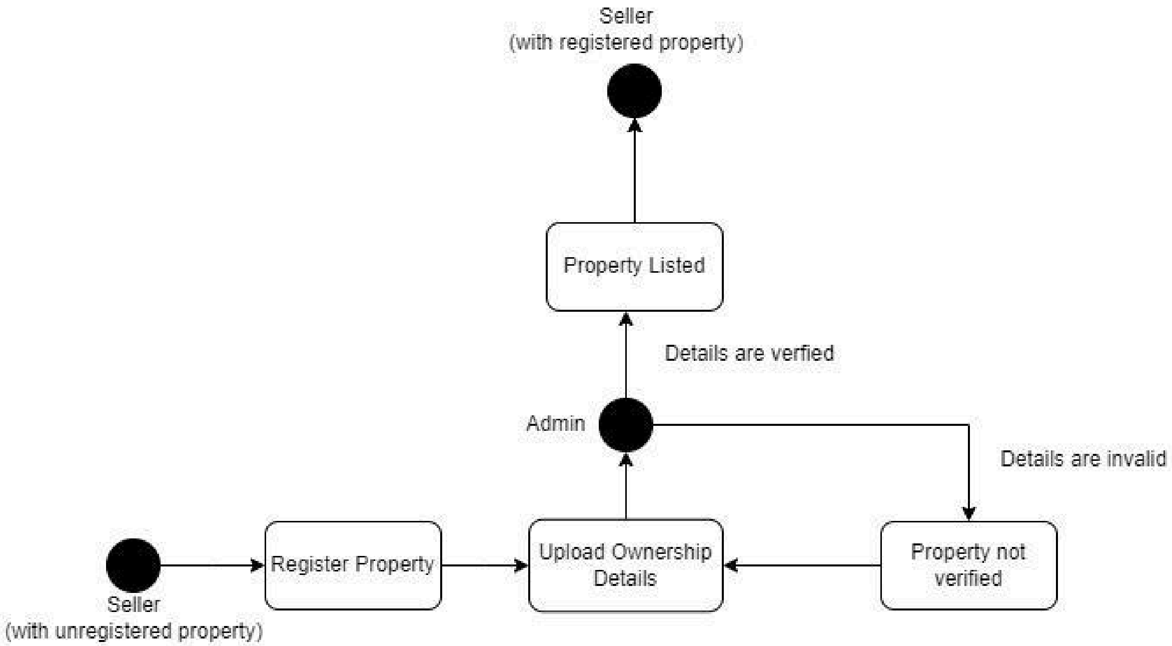


Fig 3.4.2 - Property Registration State Diagram

3.4.3 View Property Details

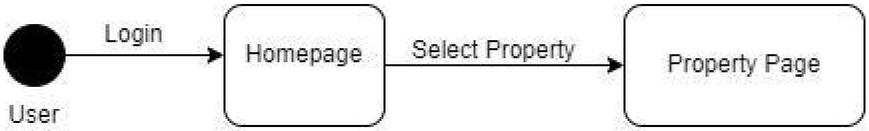


Fig 3.4.3 - Property Details State Diagram

3.4.4 Funds

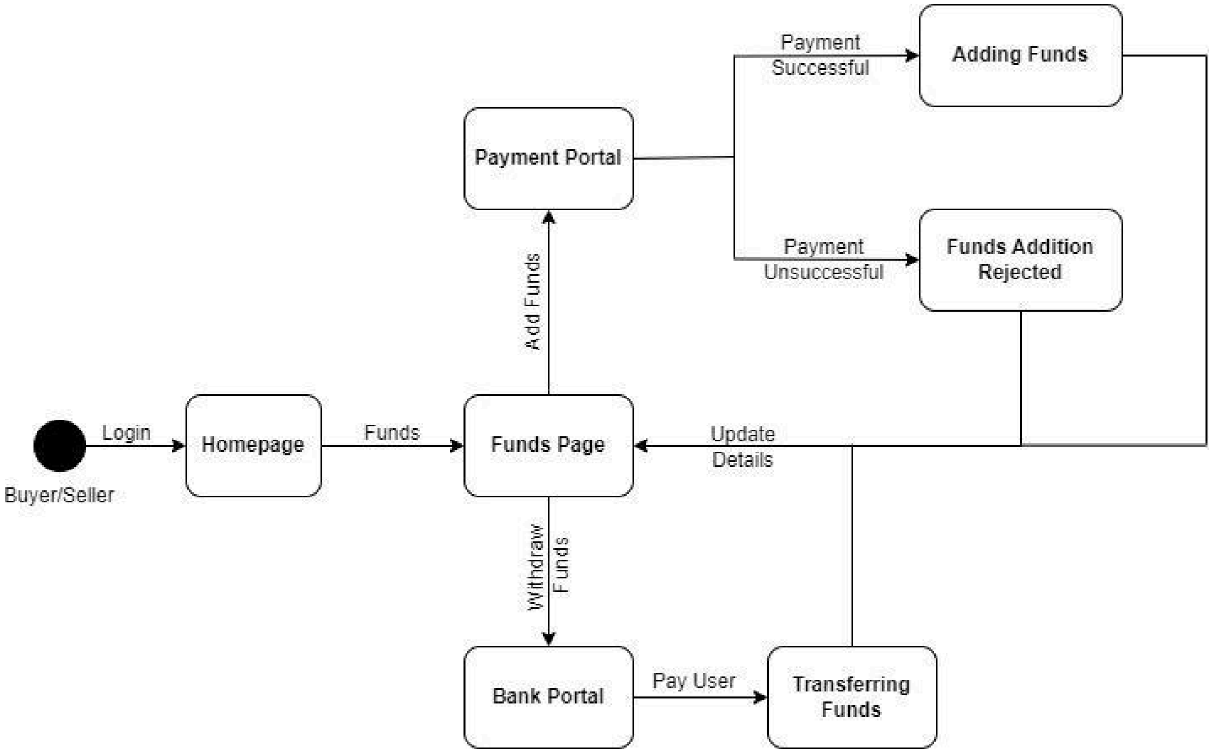


Fig 3.4.4 -Funds State Diagram

3.4.5 Market Buy Order

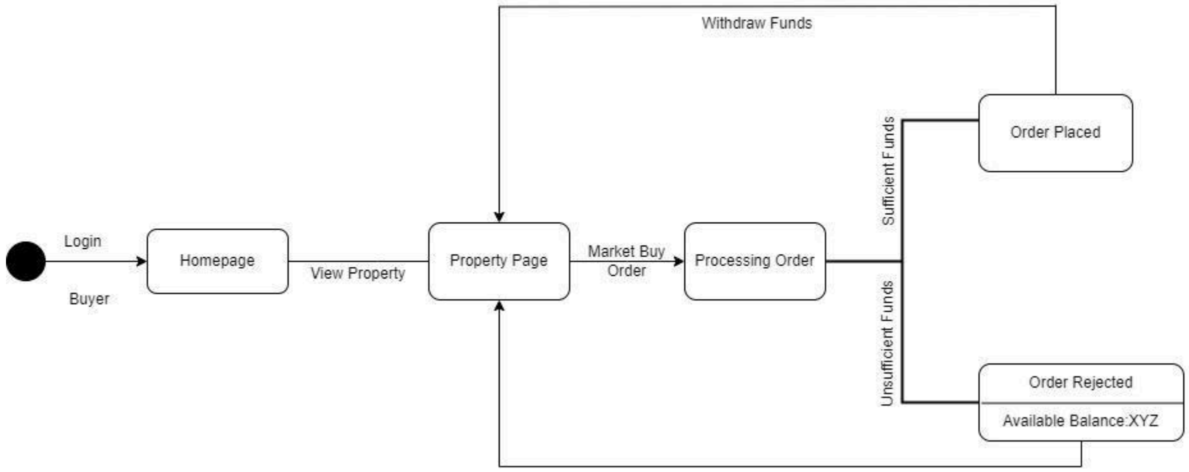


Fig 3.4.5 -Market Buy Order State Diagram

3.4.6 Market Sell Order

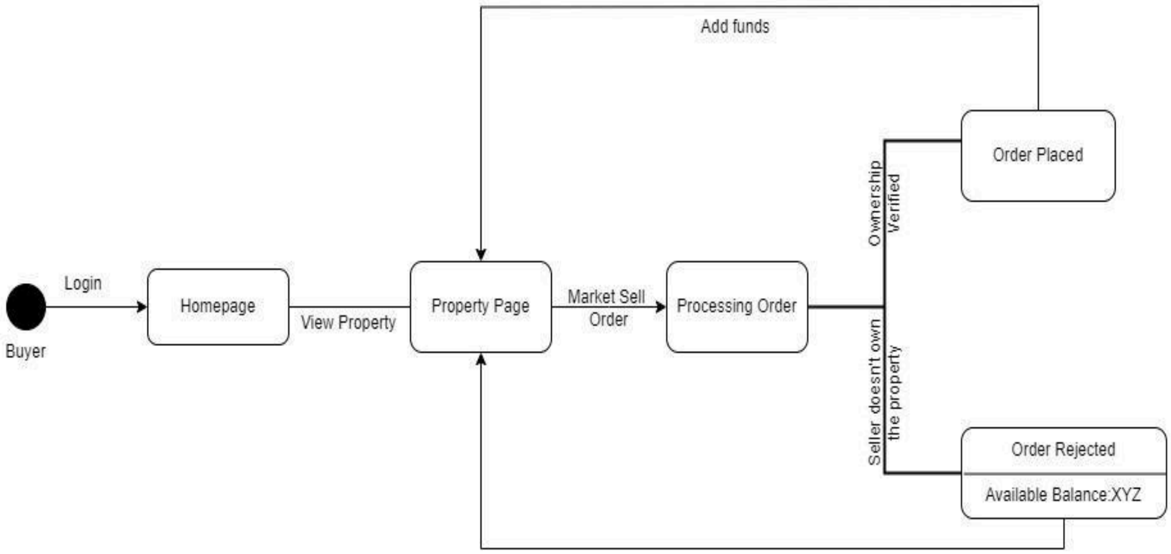


Fig 3.4.6 - Market Sell Order State Diagram

3.4.7 Limit Buy Order

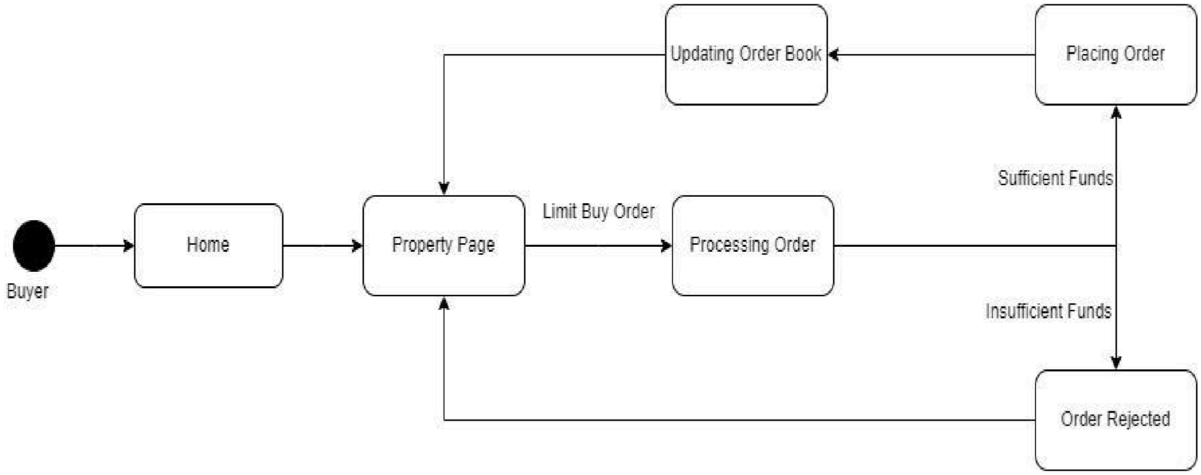


Fig 3.4.7 - Limit Buy Order State Diagram

3.4.8 Limit Sell Order

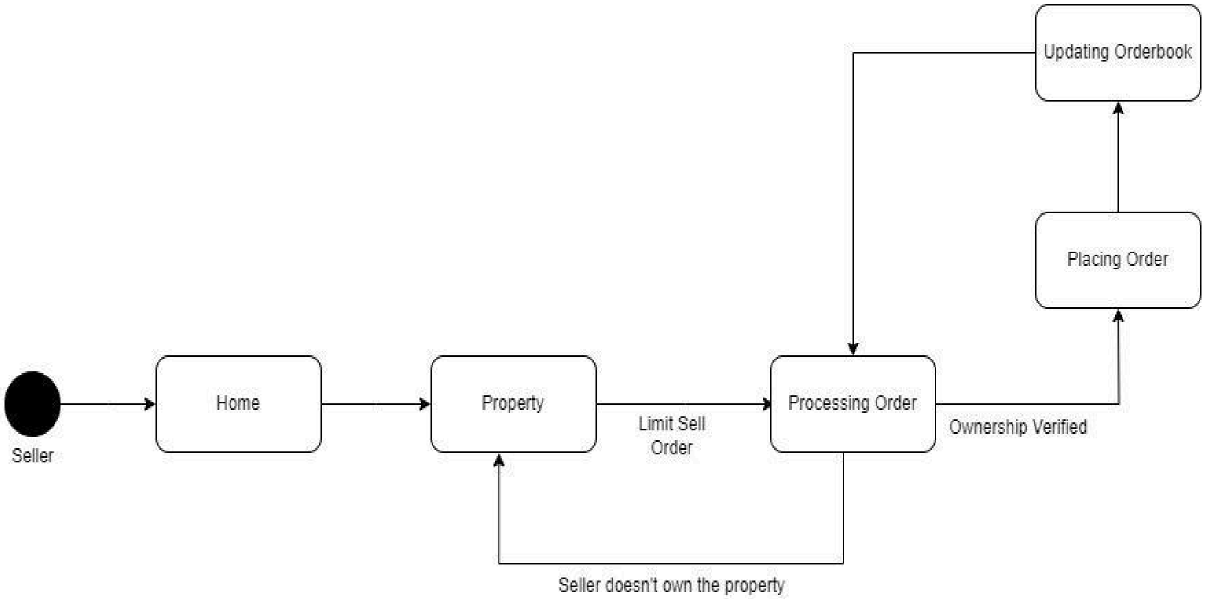
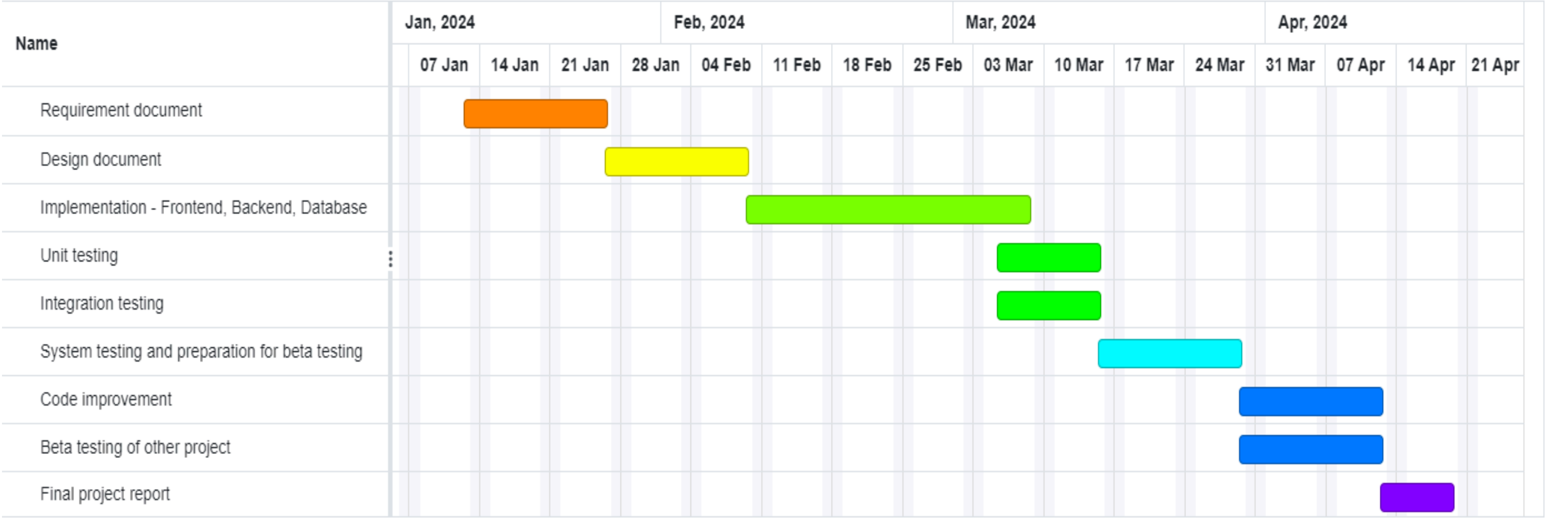


Fig 3.4.8 - Limit Sell Order State Diagram

4 Project Plan



Tasks	Person implementing it
Frontend	Chinmay, Shivang, Armeet, Utpal, Praveen, Shivaleela
Backend	Chinmay, Vaishali, Arth, Shivang, Armeet, Abhishek, Pravallika
Database	Chinmay, Armeet, Shivang
Integration	Chinmay

Tech Stack:

1. We will programme the frontend of the website in Javascript using ReactJS.
2. The server will be programmed in Python using Django.
3. We will utilise an SQL based database to store and query data.

Appendix A - Group Log

Date	Duration	Details
1 Feb	75 minutes	Discussed the implementation and the look of the user interface and the features within each webpage. Distributed the work of the design doc among the group members.
6 Feb	60 minutes	Discussed to review the first draft of individual parts to remove any doubts and errors, so that the doc is consistent and complete.
8 Feb	30 minutes	Integrated the individual parts into the final design doc and made sure there were no inconsistencies.
9 Feb	15 minutes	Final compiled document

Work done by each member:

1. *Context model* - Chinmay
2. *Human Interface design* - Vaishali, Abhishek
3. *Architecture design* - Chinmay
4. *Use case diagrams* - Armeet, Utpal
5. *Class diagrams* - Chinmay
6. *Sequence diagrams* - Shivang, Pravallika, Chinmay
7. *State diagram* - Arth, Chinmay, Praveen
8. *Project plan* - Shivang, Chinmay