# SUMMARY

Topic title: Real estate business

Student name: Nguyen Duc Van

Student ID: 102210195. Class: 21TCLC\_DT2

|  |  |
| --- | --- |
| DA NANG UNIVERSITY  **UNIVERSITY OF SICIENCE AND TECHNOLOGY**  FALCUTY………………………………………… | **THE SOCIALIST REPUBLIC OF VIETNAM**  Independence - Freedom - Happiness |

# GRADUATION PROJECT REQUIREMENTS

Student Name: …..…………….………….…….. Student ID :………………...

Class:…………… Faculty:..................................... Major:…………...........................

1. *Topic title:*

………………………………………………..…………………………………………

…………………………………………………………………………………………..

1. *Project topic :* ☐*has signed intellectual property agreement for final result*
2. *Initial figure and data:*

……………………………………..……………………………………………..……......……………………………………………………………………………………………..………………………………….…..………………………..………………………*Content of the explanations and calculations:*

…...………………………………………………………………………………………

…...………………………………………………………………………………………

…...………………………………………………………………………………………

…...………………………………………………………………………………………

…...………………………………………………………………………………………

1. *Drawings, charts (specify the types and sizes of drawings):*

…...………………………………………………………………………………………

…...………………………………………………………………………………………

…...………………………………………………………………………………………

…...………………………………………………………………………………………

|  |  |
| --- | --- |
| 1. *Name of instructor:* | *Content parts:* |
|  |  |
|  |  |
|  |  |

1. *Date of assignment: ……../……./201…..*
2. *Date of completion: ……../……./201…..*

|  |  |
| --- | --- |
|  | *Đà Nẵng, date month year 201* |
| **Head of Division**…………………. | **Instructor** |

# PREFACE

# ASSURRANCE

I hereby declare that:

1. The graduation project report, Topic name: Real Estate Business System, is my own research work under the direct guidance of the lecturer ................................
2. I have read, researched, translated documents and synthesized the knowledge that has made this report and ensure that it has not been copied anywhere.
3. The theories in the thesis are all used from the documents as I have referred to in the reference section in the report.

If there is any violation, I will take full responsibility.

Student Performed

Nguyen Duc Van

# TABLE OF CONTENT

[SUMMARY 3](#_Toc195694193)

[GRADUATION PROJECT REQUIREMENTS 4](#_Toc195694194)

[PREFACE i](#_Toc195694195)

[ASSURRANCE ii](#_Toc195694196)

[TABLE OF CONTENT iii](#_Toc195694197)

[LIST OF PICTURES v](#_Toc195694198)

[LIST OF TABLES vi](#_Toc195694199)

[LIST OF SYMBOL, ACRONYM vii](#_Toc195694200)

[INTRODUCTION 1](#_Toc195694201)

[1. Purpose of implementation 1](#_Toc195694202)

[2. Objective of the topic 1](#_Toc195694203)

[2.1. System Target 1](#_Toc195694204)

[2.2. Features 1](#_Toc195694205)

[3. Scope 2](#_Toc195694206)

[4. Target customers 2](#_Toc195694207)

[5. Development technology 2](#_Toc195694208)

[CHAPTER 1: THEORETICAL FOUNDATION 3](#_Toc195694209)

[CHAPTER 2: SYSTEM DESIGN AND ANALYSIS 4](#_Toc195694210)

[2.1. Business analysis 4](#_Toc195694215)

[2.1.1. Buyer's business 4](#_Toc195694216)

[2.1.2. Broker's business 4](#_Toc195694217)

[2.1.3. Admin's business 5](#_Toc195694218)

[2.2. System design 5](#_Toc195694219)

[2.2.1. Use-case diagram 5](#_Toc195694220)

[2.2.2. Activity diagram 9](#_Toc195694221)

[2.2.3. Sequence diagram 10](#_Toc195694222)

[2.3. Database design 14](#_Toc195694223)

[2.3.1. ER (entity-relationship) 14](#_Toc195694224)

[2.3.2. Converting an implementation model into a system model 14](#_Toc195694225)

[2.3.3. Database schema 21](#_Toc195694226)

[2.4. Real estate prediction system development 22](#_Toc195694227)

[2.4.1. Real estate valuation model 22](#_Toc195694228)

[2.4.2. Land forecast model 22](#_Toc195694229)

[2.5. Chapter Conclusion 23](#_Toc195694230)

[CHAPTER 3: IMPLEMENTATION AND RESULTS EVALUATION 24](#_Toc195694231)

[CHAPTER 4: CONCLUSION AND DEVELOPMENT DIRECTION 25](#_Toc195694232)

[REFERENCES 26](#_Toc195694233)

[APPENDIX 1 1](#_Toc195694234)

[APPENDIX 2 2](#_Toc195694235)

# LIST OF PICTURES

[Figure 2.1 General use case diagram 5](#_Toc195688017)

[Figure 2.2 User management use-case diagram 6](#_Toc195688018)

[Figure 2.3 Post management use case diagram 6](#_Toc195688019)

[Figure 2.4 User management use case diagram 7](#_Toc195688020)

[Figure 2.5 Account upgrade management use case diagram 7](#_Toc195688021)

[Figure 2.6 Post management use case diagram 8](#_Toc195688022)

[Figure 2.7 Favourite post management use case diagram 8](#_Toc195688023)

[Figure 2.8 Client Operation Principle 9](#_Toc195688024)

[Figure 2.9 Admin Operation Principle 9](#_Toc195688025)

[Figure 2.10 Registration/Login Flow 10](#_Toc195688026)

[Figure 2.11 Real estate article management flow as a user 12](#_Toc195688027)

[Figure 2.12 Account upgrade flow 13](#_Toc195688028)

[Figure 2.13 The ER entity relationship model 14](#_Toc195688029)

[Figure 2.14 Relational Model 20](file:///D:\VisualStudioCode\Project\RealEstate\GraduationProjectReport\MyReport\PL04_B_RuotThuyetMinh.docx#_Toc195688030)

# LIST OF TABLES

[Table 2.1 Describe role table attributes 14](#_Toc195652978)

[Table 2.2 Describe user table attributes 15](#_Toc195652979)

[Table 2.3 Describe token table attributes 15](#_Toc195652980)

[Table 2.4 Describe transaction histories table attributes 16](#_Toc195652981)

[Table 2.5 Describe favourite properties table attributes 16](#_Toc195652982)

[Table 2.6 Describe category table attributes 16](#_Toc195652983)

[Table 2.7 Describe property table attributes 17](#_Toc195652984)

[Table 2.8 Describe house table attributes 17](#_Toc195652985)

[Table 2.9 Describe furnished status table attributes 17](#_Toc195652986)

[Table 2.10 Describe type of house table attributes 18](#_Toc195652987)

[Table 2.11 Describe house characteristics table attributes 18](#_Toc195652988)

[Table 2.12 Describe house characteristic mapping table attributes 18](#_Toc195652989)

[Table 2.13 Describe land table attributes 18](#_Toc195652990)

[Table 2.14 Describe type of land table attributes 19](#_Toc195652991)

[Table 2.15 Describe land characteristics table attributes 19](#_Toc195652992)

[Table 2.16 Describe land characteristic mapping table attributes 19](#_Toc195652993)

# LIST OF SYMBOL, ACRONYM

|  |  |
| --- | --- |
| Symbol, Acronym | Detailed explanation |
| PMML | Predictive Model Markup Language |
| HTTPS | HyperText Transfer Protocol Secure |
| PKL | PKL is the extension for files created with Pickle |
| API | Application Programming Interface |

# INTRODUCTION

## Purpose of implementation

* Searching and buying real estate today is difficult, especially in determining the real value of the property, comparing suitable options and accessing transparent information. Buyers often spend a lot of time searching for properties that meet their needs in terms of location, area, amenities and finance, while sellers have difficulty reaching potential customers.
* Therefore, an online real estate business system can help display information visually, support filtering and comparing options, and integrate price prediction technology to help buyers and sellers make more accurate decisions

## Objective of the topic

### System Target

* Help users quickly post and search for suitable real estate based on filters with many characteristics.
* The real estate valuation feature provides a reference price close to the market, helping users minimize the risk of incorrect valuation, supporting the brokerage team or real estate sellers and buyers to save time in valuation and negotiation.
* The system will also support predicting real estate fluctuations to support customers' buying or selling decisions.
* Integrating VNPay, safe and convenient online payment when customers want to upgrade their account to use more features of the application.

### Features

* For guest
  + Search and filter by needs: price, area, location, property type, ...
  + View property details with images, prices and legal information, ...
  + Displays detailed information of the property along with the price determined by the property valuation system.
* For member
  + Post real estate with full information: images, descriptions, prices, legal status, ...however, the number of posts will be limited to 3.
  + You can save posts that interest you as well as upgrade your account to be able to use more features of the application.
* For brokers
  + Customers with upgraded accounts will be supported by the system to predict future real estate price compared to the market.
  + The number of posts posted by the brokerage account will also be up to 30.
* For admins:
  + Manage user accounts.
  + View transaction history.
  + View article details and moderate it.
  + Statistics and reports on business activities.

## Scope

* Property type: house, land.
* Geographical scope: Da Nang
* Language: Vietnamese

## Target customers

* Real Estate Buyers/Sellers
* Real Estate Companies or Brokers

## Development technology

* Project management and hosting: Github
* Language: Java, Python, Reactjs
* Framework: Spring boot
* Database: Mysql
* Testing tool: Selenium
* Encryption & security: Security, HTTPS Token

# THEORETICAL FOUNDATION

# SYSTEM DESIGN AND ANALYSIS



## Business analysis

### Buyer's business

* **Visiting customers**
  + View property list: can view all properties posted on the system.
  + price, area, type, etc.
  + View property details: view description, images, price, area, number of rooms, etc. and information of the poster. At the same time, the real estate price will be displayed as determined by the system.
  + Register account: if you want to save information or make transactions, visitors must register an account.
  + Search and filter properties: search by keyword, filter by location
* **Members** are those who have registered an account and can use all the features:
  + All rights of visitors.
  + Update profile and account information: Change avatar, contact information and change password when needed.
  + Upgrade account: choose an account upgrade package to use more features and pay directly through the payment gateway.
  + View transaction history: track completed or pending transactions.
  + Buyer accounts can also post properties for sale, but are limited to 3
  + Manage post: edit, update, or delete posted real estate listings.
  + Save to favorites: add properties to your favorites list for easy access later.

### Broker's business

* Brokerage account is a registered and upgraded account
* Having all rights of member
* This account will be used for additional features of the system: predict real estate trends.
* The article limits the posts that can be posted by this account to 30

### Admin's business

* User management: view list of buyers, brokers as well as add, edit and delete users.
* Post moderation: review, approve or reject real estate posts.
* System statistics and reports: track transactions, number of users, posts

## System design

### Use-case diagram

#### General use-case diagram

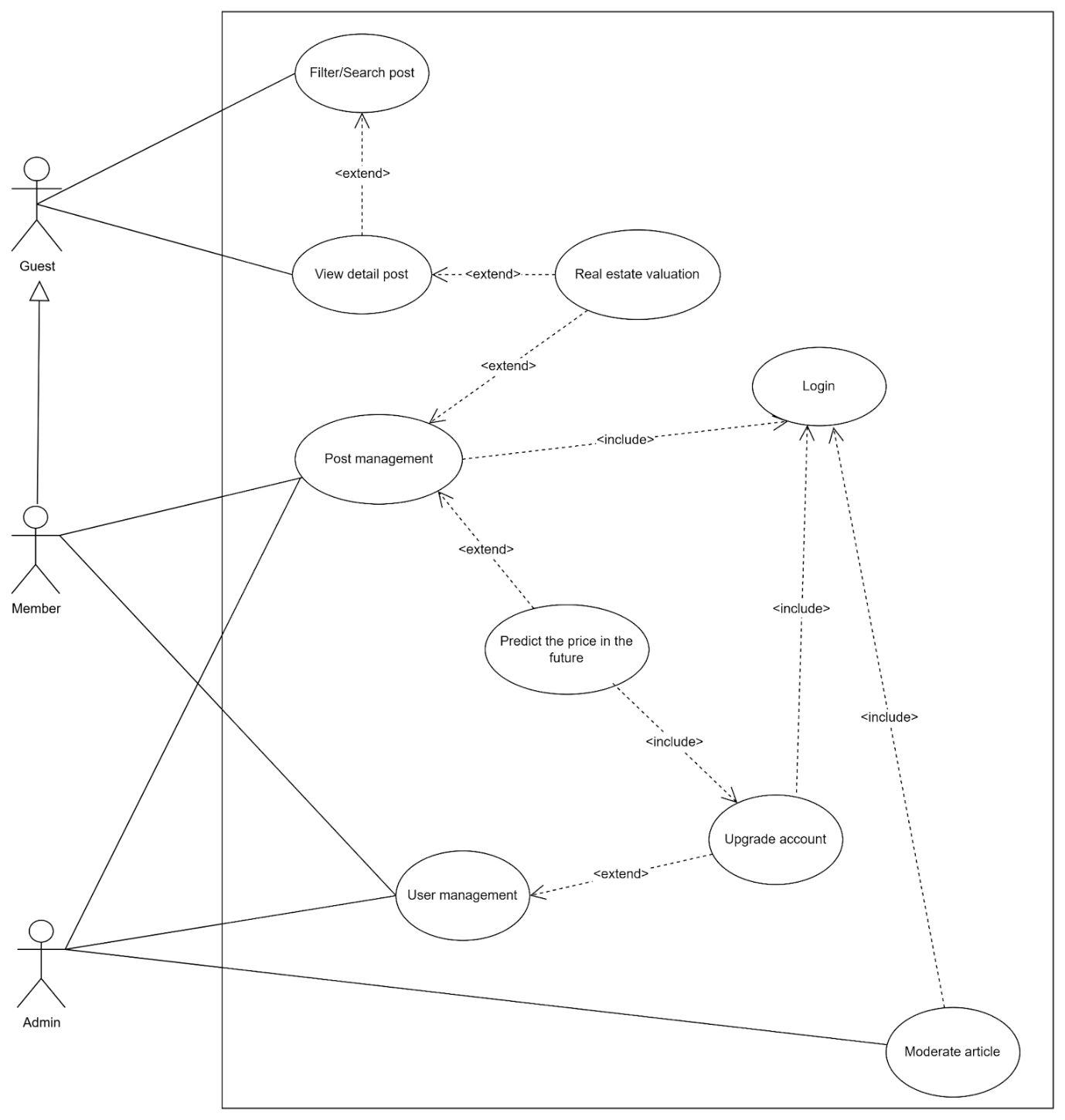


Figure . General use case diagram

#### Admin use-case diagram

* User management

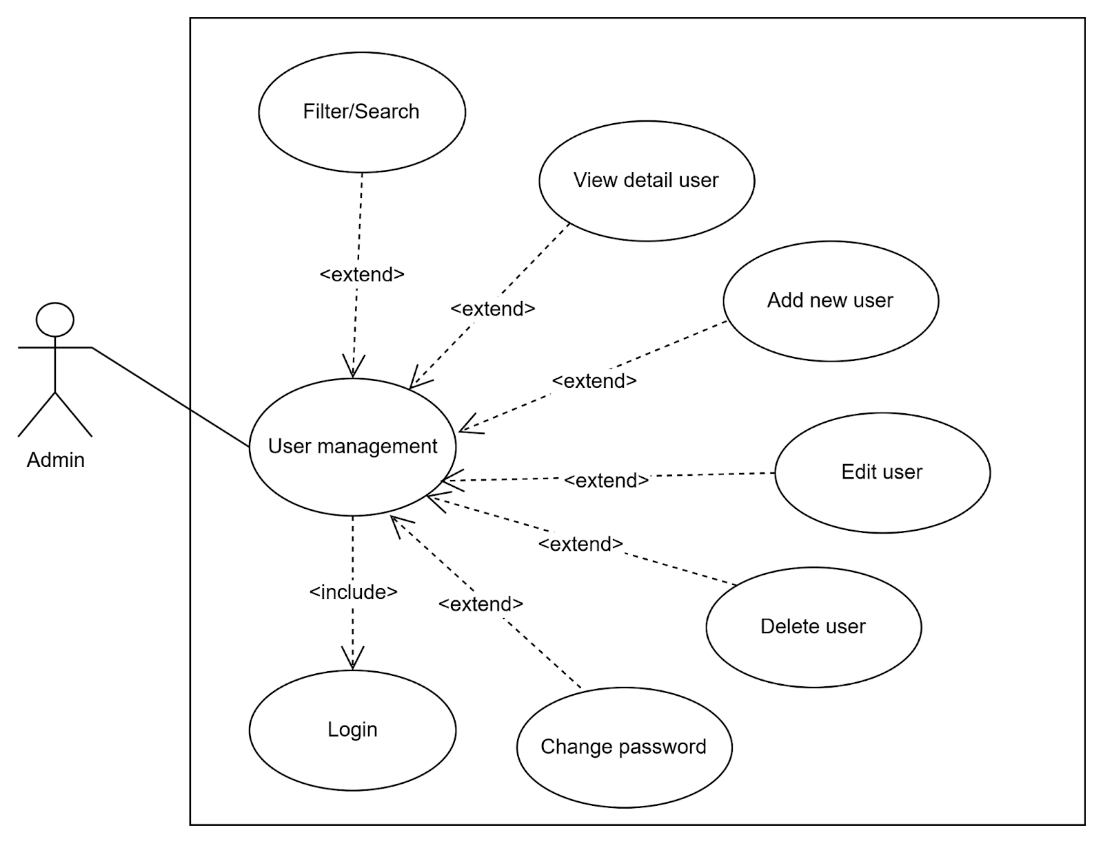


Figure . User management use-case diagram

* Post modaration

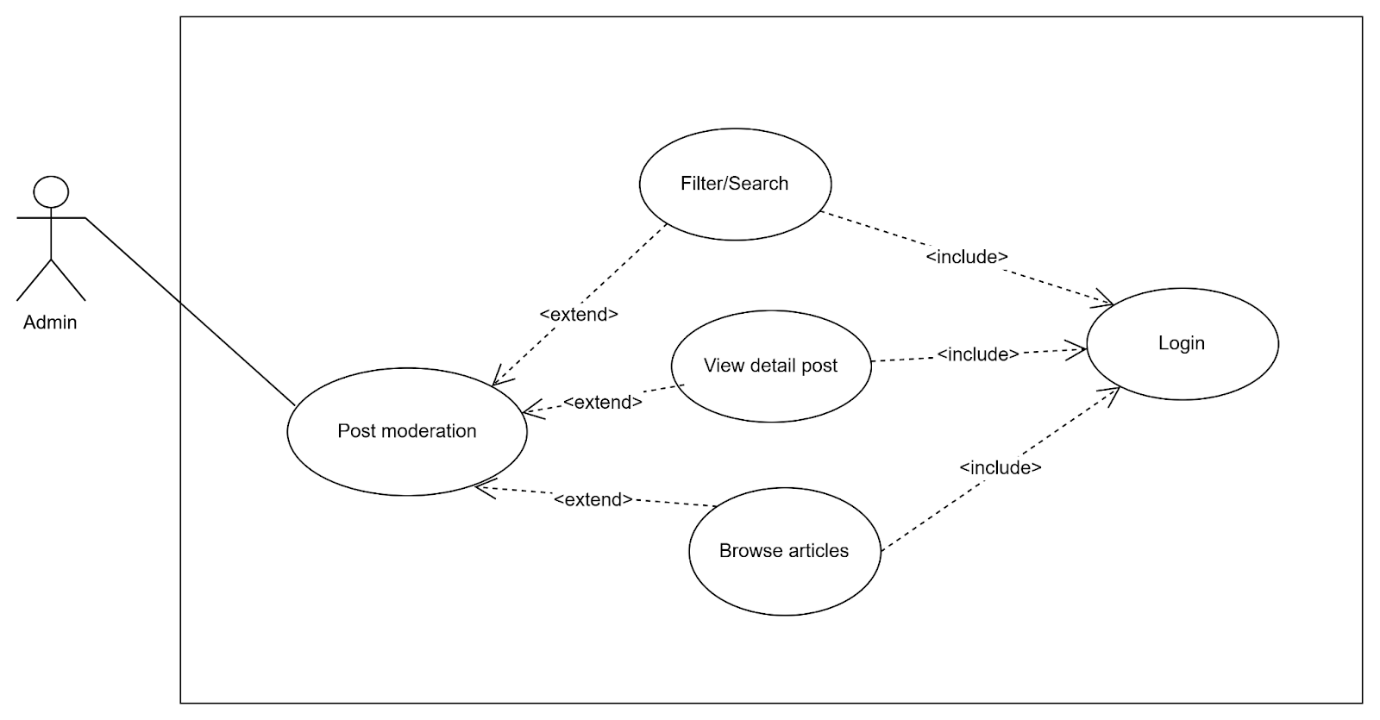


Figure . Post management use case diagram

#### User use-case diagram

* User management

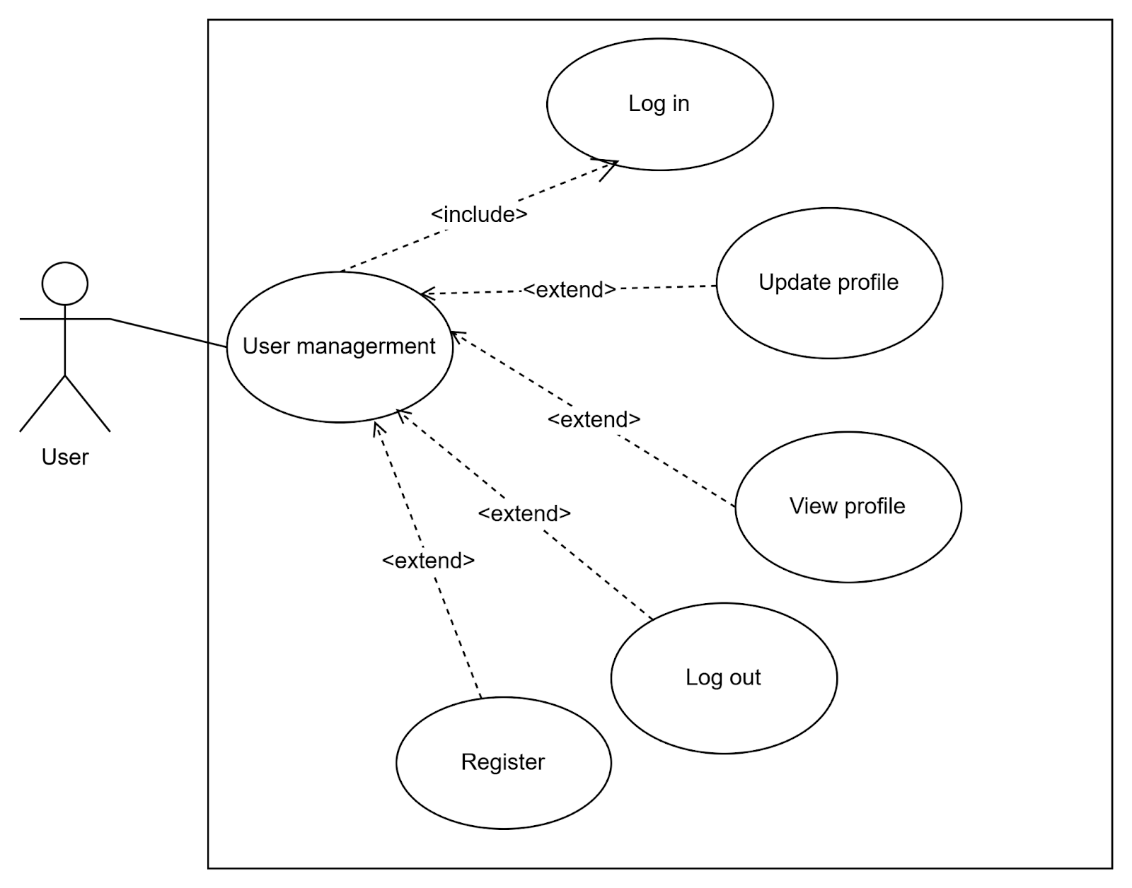


Figure . User management use case diagram

* Account upgrade management

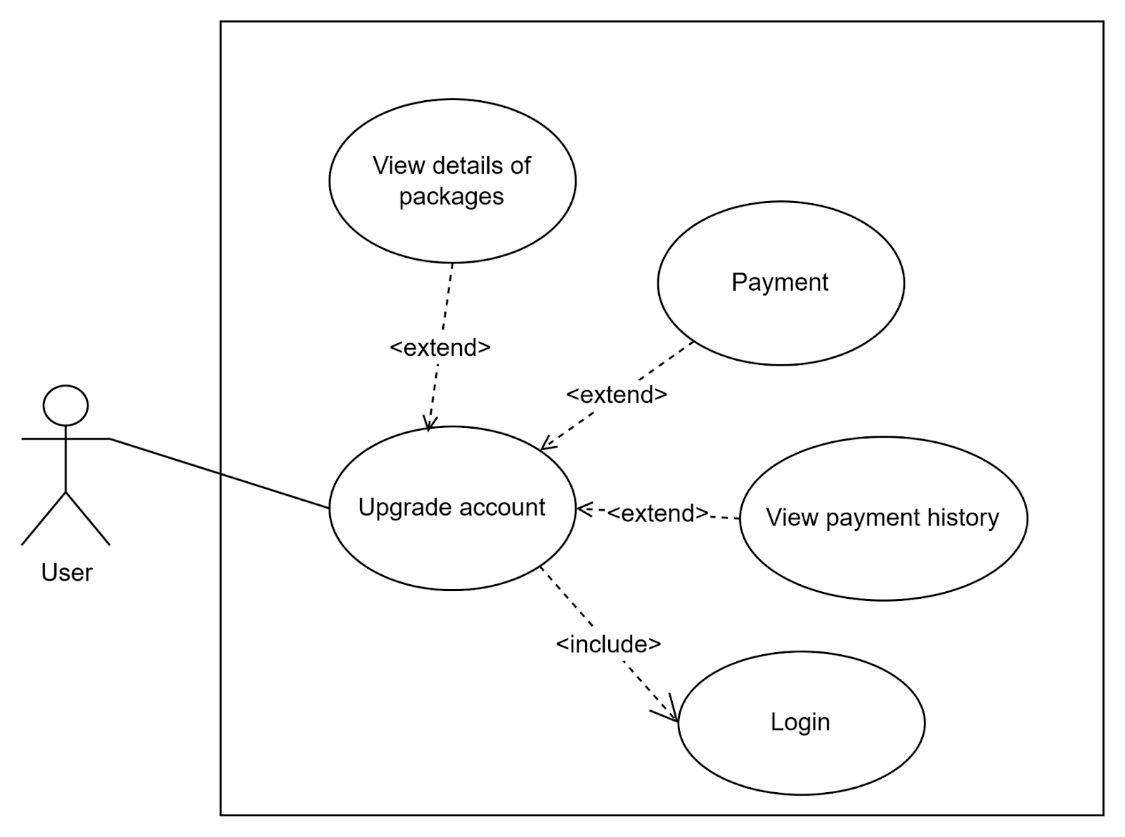


Figure . Account upgrade management use case diagram

* Post management

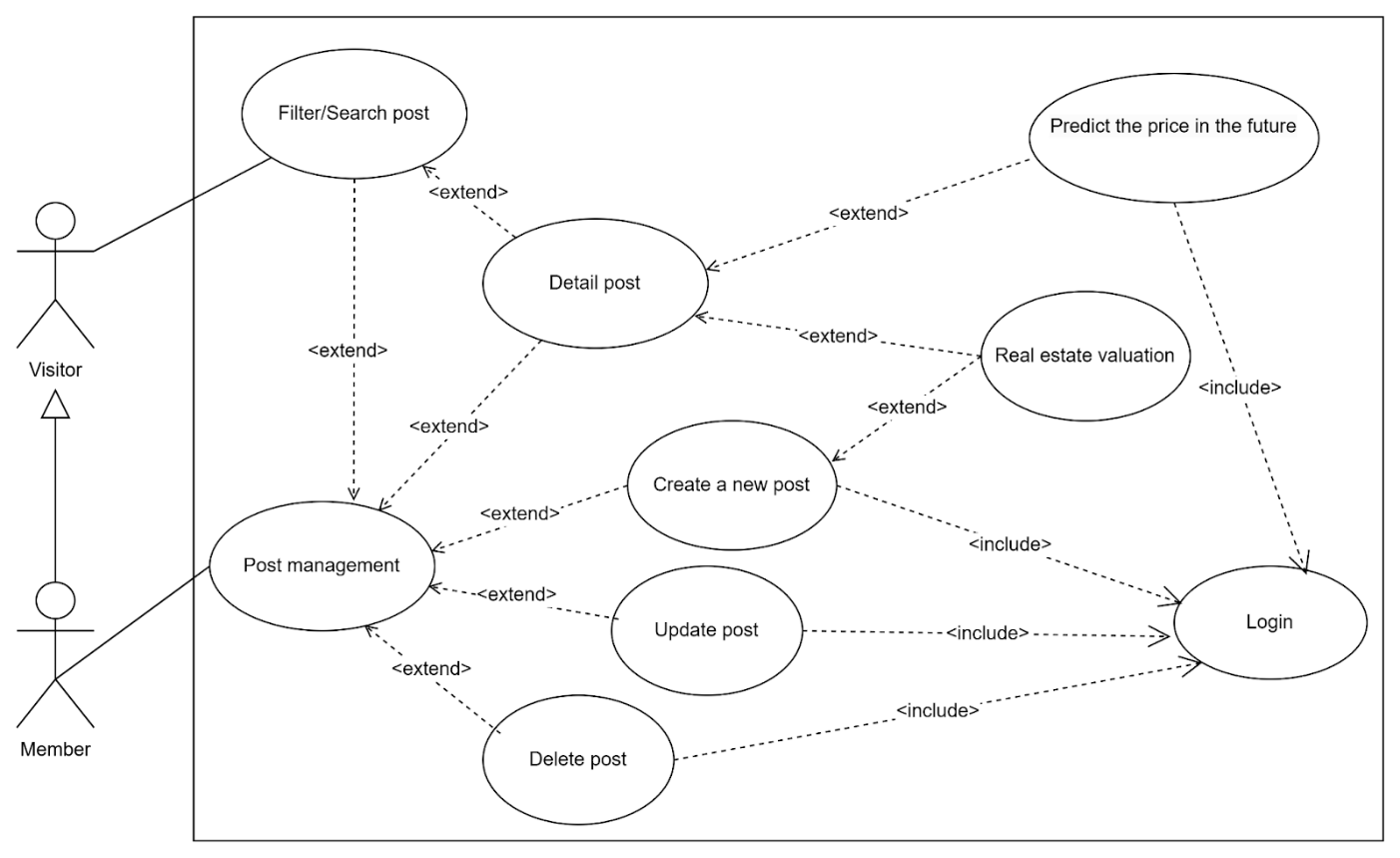


Figure . Post management use case diagram

* Favourite post management

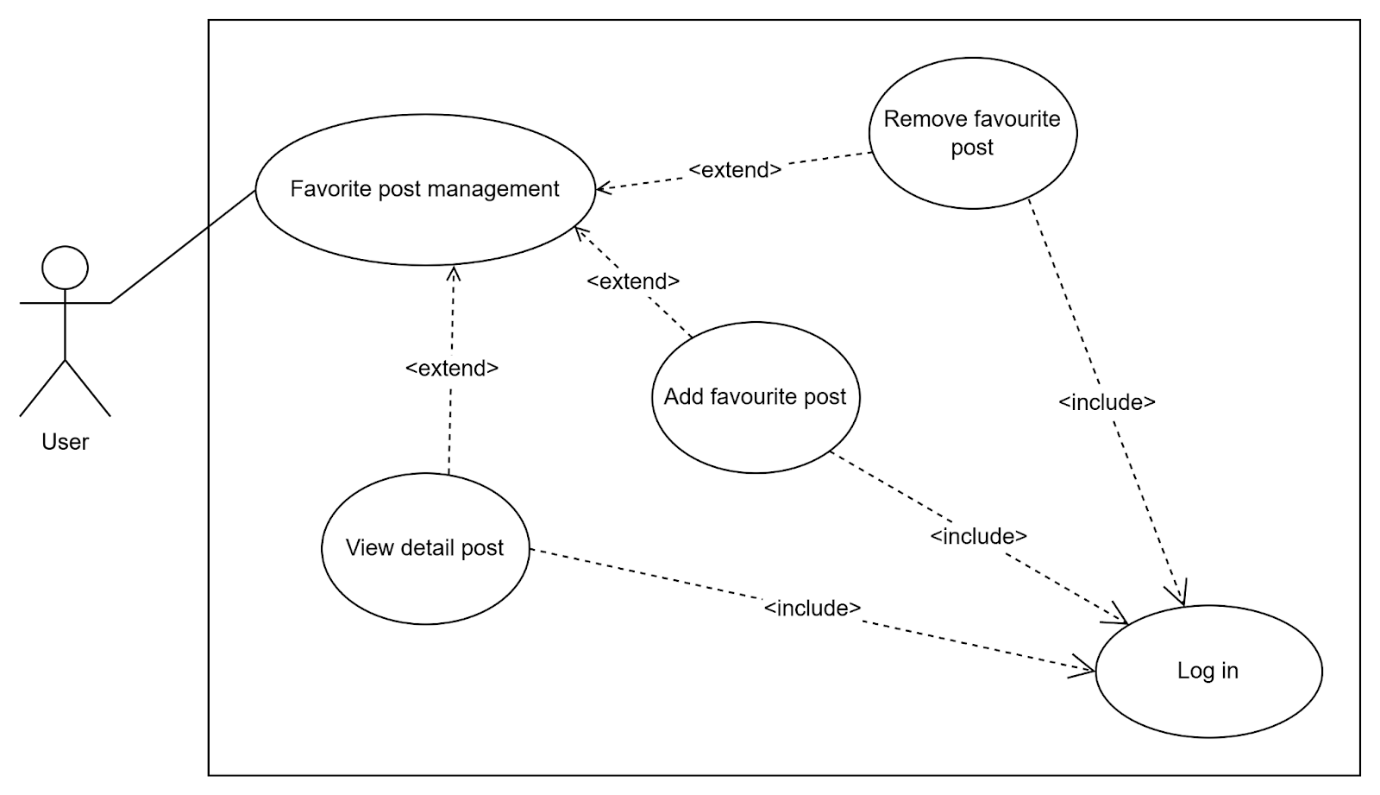


Figure . Favourite post management use case diagram

### Activity diagram

* The stream of activities access to the buyer/seller 's website

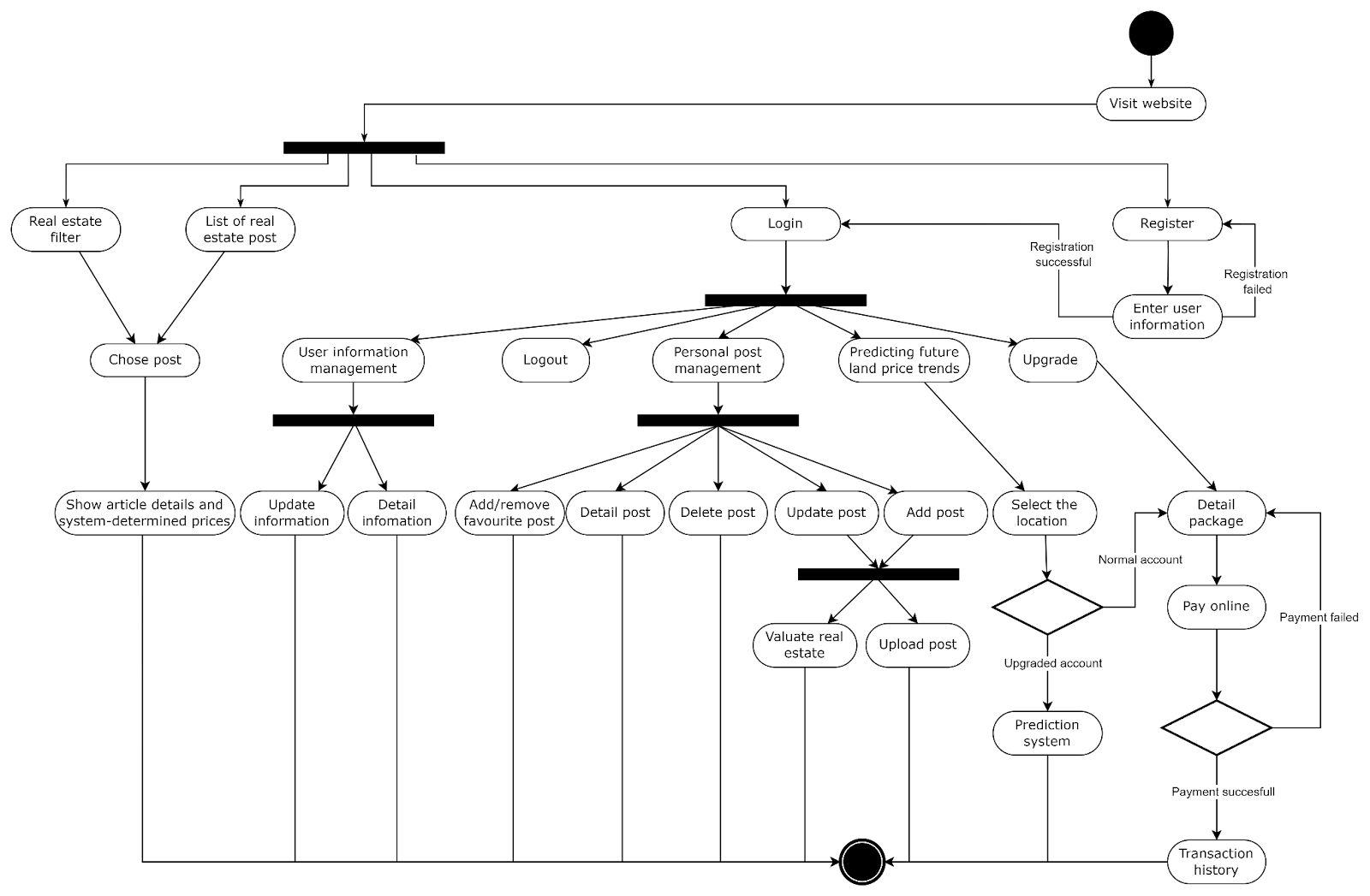


Figure . Client Operation Principle

* The stream of activities access to the admin website

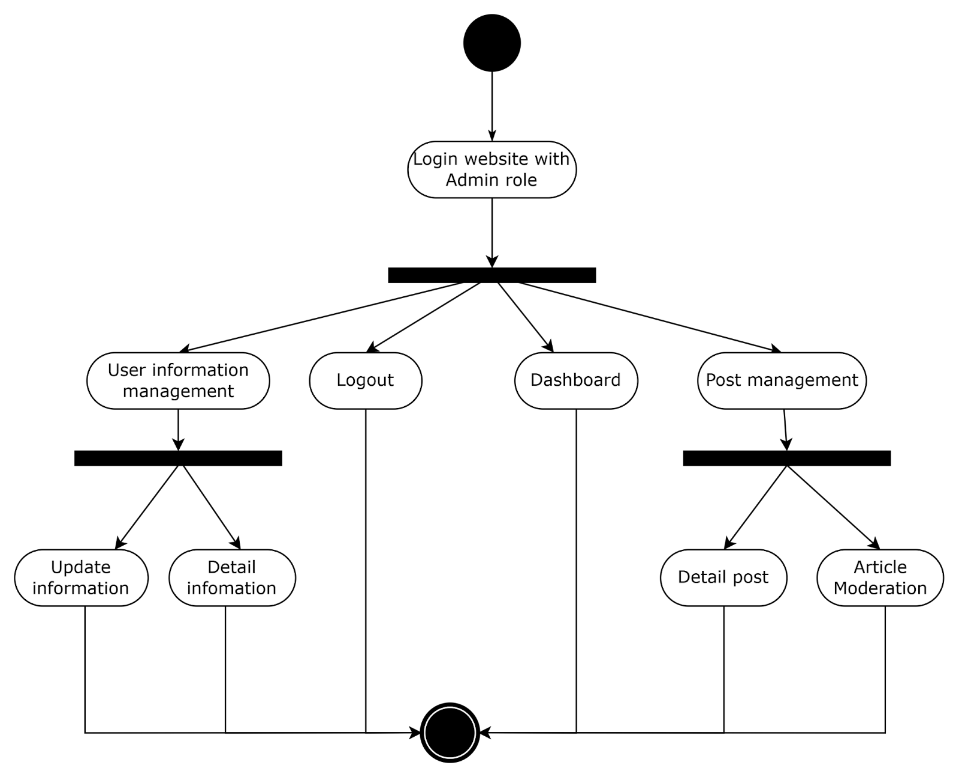


Figure . Admin Operation Principle

### Sequence diagram

* Login/Signup flow design

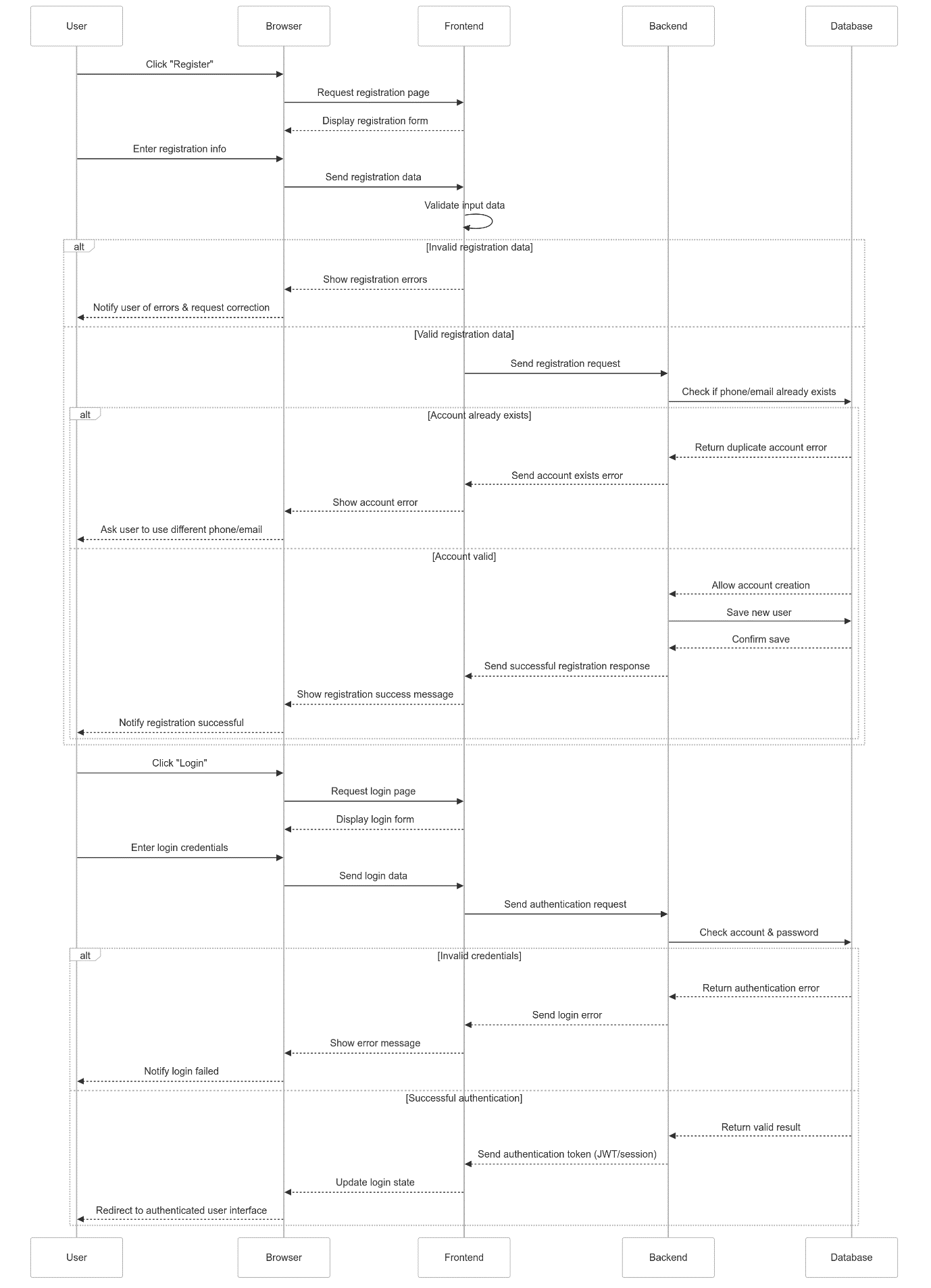
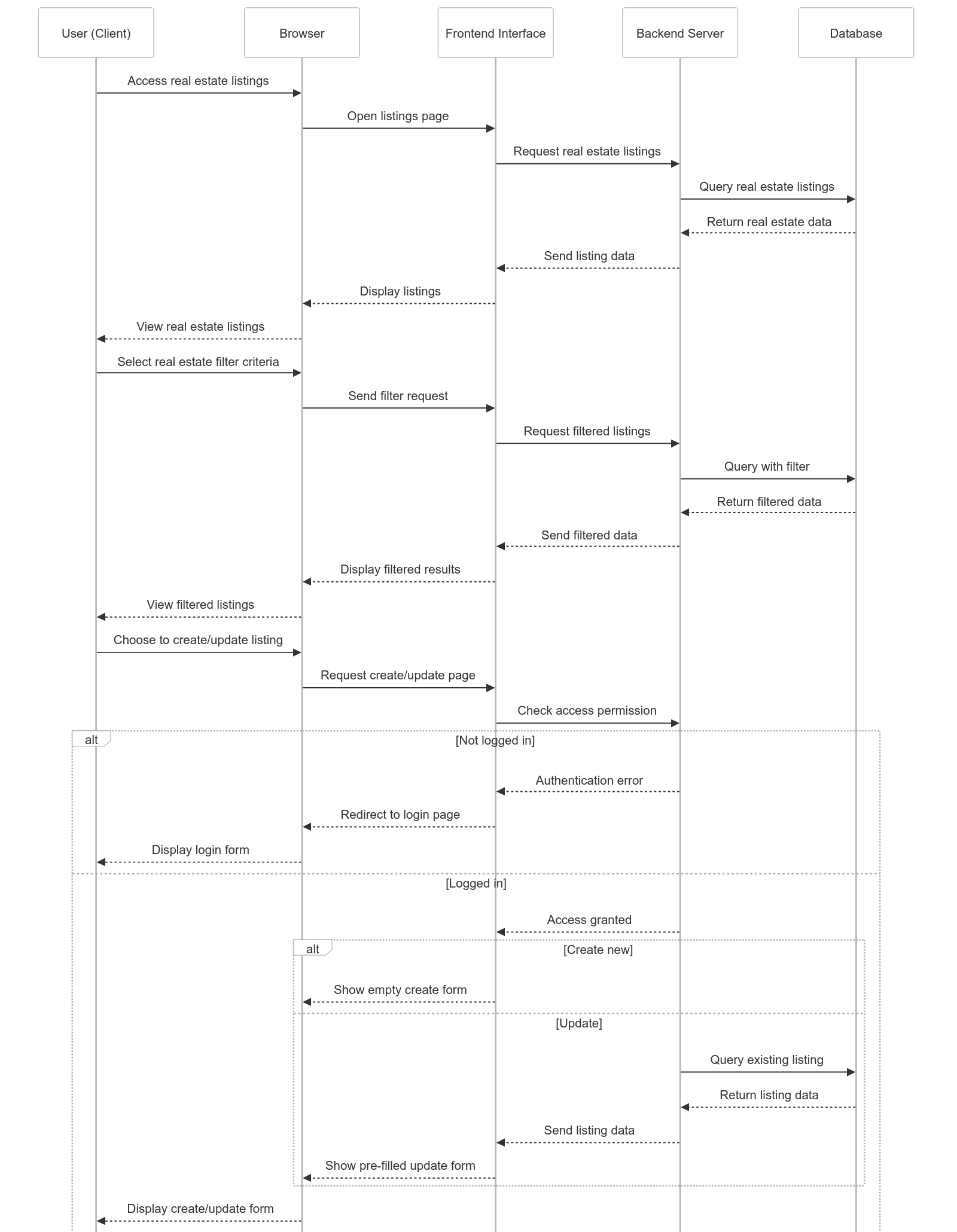


Figure . Registration/Login Flow

* Real estate article management flow design



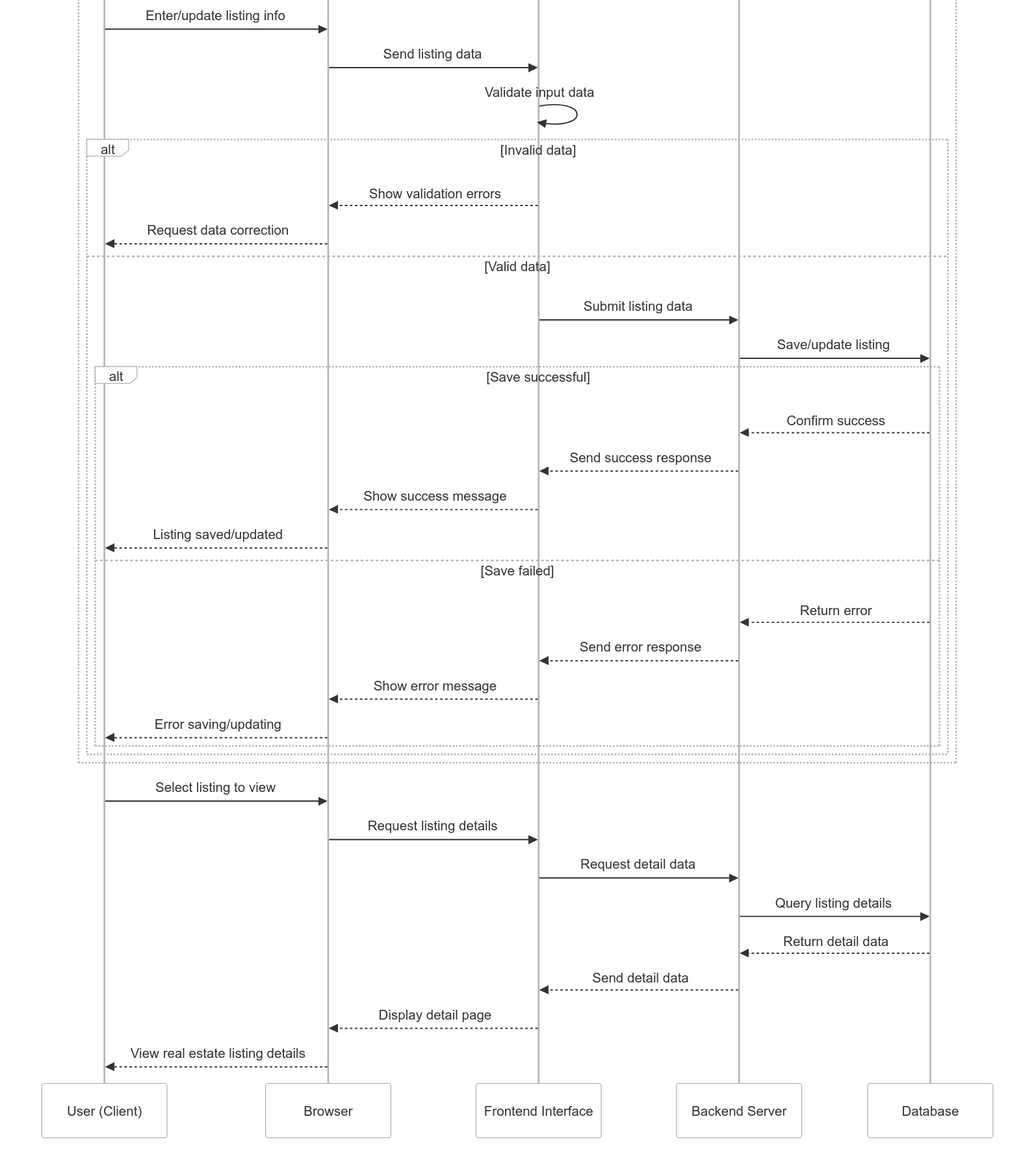


Figure . Real estate article management flow as a user

* Design of account upgrade flow integrating online payment

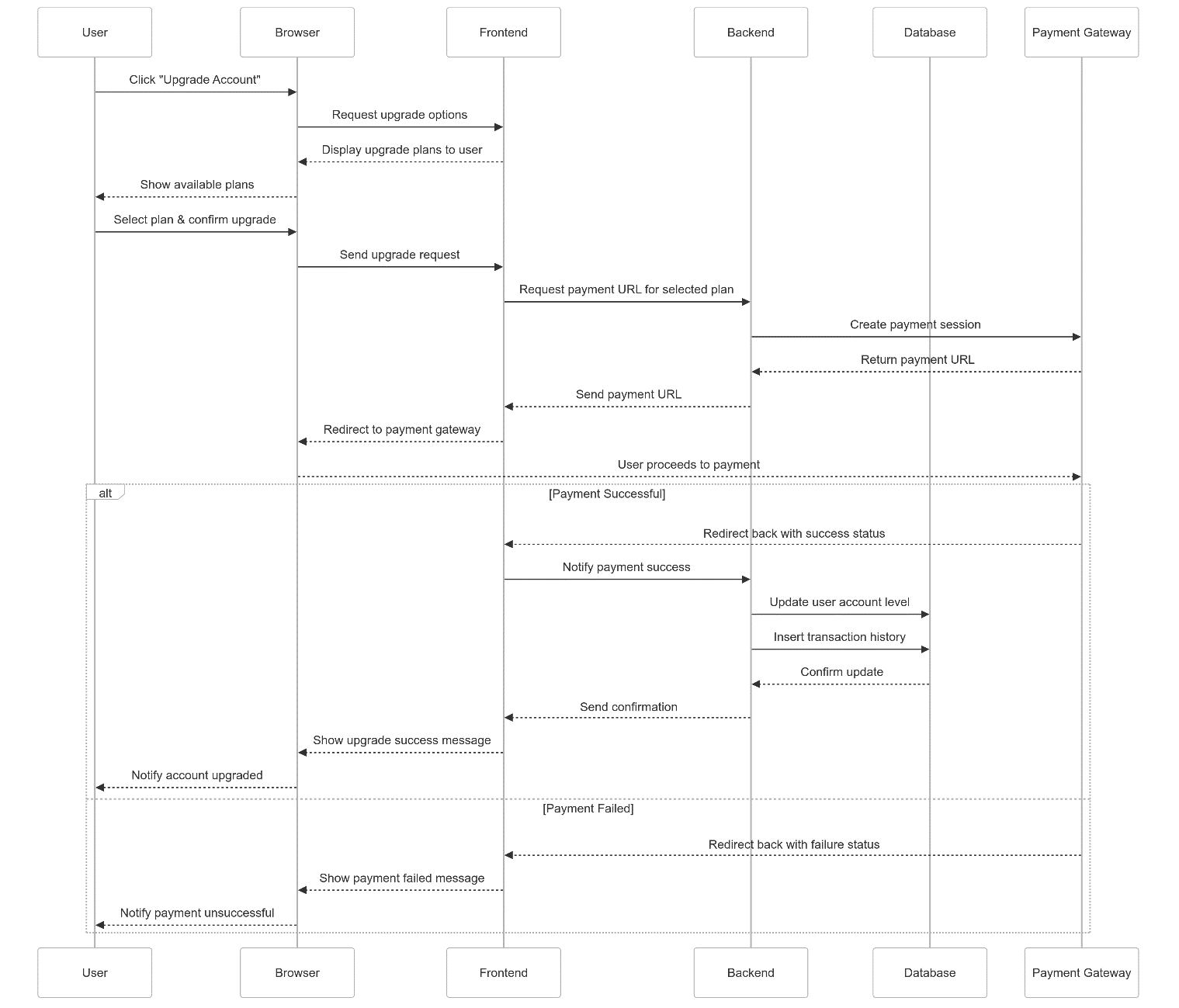


Figure . Account upgrade flow

## Database design

### ER (entity-relationship)

The ER entity relationship model is represented as

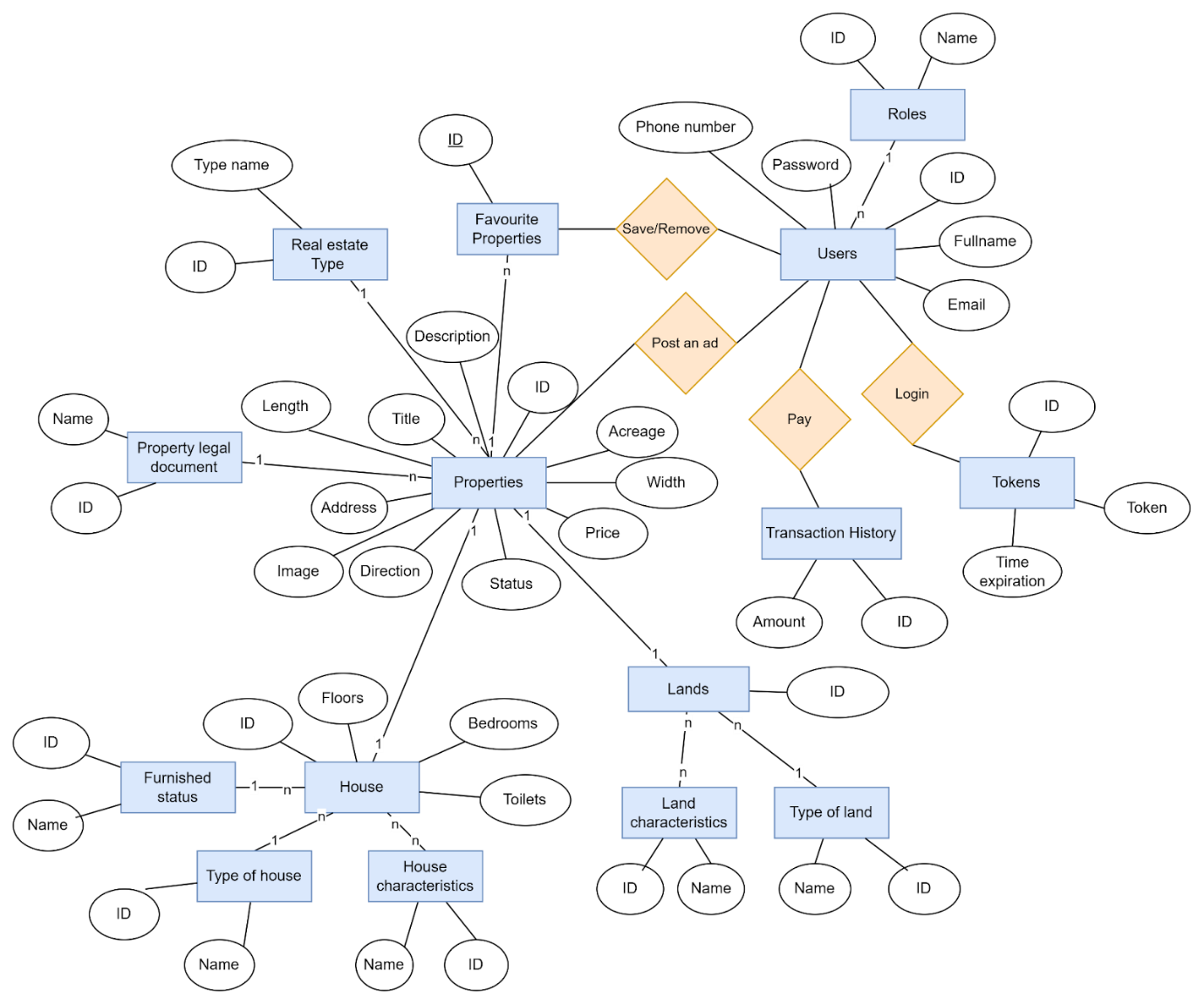


Figure . The ER entity relationship model

### Converting an implementation model into a system model

* Role table (roles)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type | Required | Constraint | Description |
| role\_id | Integer | Not null | Primary Key | Code of permission |
| name | Enum | Not null |  | Name of permission |

Table . Describe role table attributes

* User table (users)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type | Required | Constraint | Description |
| user\_id | Integer | Not null | Primary Key | Code of user |
| role\_id | Integer | Not null | Foreign Key | Permission |
| fullName | String | Not null |  | Full name of user |
| email | String | Not null |  | Email of user |
| password | String | Not null |  | Password of user(will be encrypted before entering the database) |
| phone | String | Not null |  | Phone number of user |
| created\_at | LocalDataTime |  |  |  |
| updated\_ad | LocalDataTime |  |  |  |

Table . Describe user table attributes

* Token table (tokens)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type | Required | Constraint | Description |
| token\_id | Integer | Not null | Primary Key | Code of token |
| user\_id | Integer | Not null | Foreign Key | User |
| token | String | Not null |  | A temporary or persistent string used to authenticate and authorize a user or system |
| created\_at | LocalDataTime |  |  |  |
| updated\_ad | LocalDataTime |  |  |  |

Table . Describe token table attributes

* Transaction history table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type | Required | Constraint | Description |
| transaction\_id | Integer | Not null | Primary Key | Code of token |
| user\_id | Integer | Not null | Foreign Key | User |
| amount | Number | Not null |  | Amount paid |
| created\_at | LocalDataTime |  |  |  |
| updated\_ad | LocalDataTime |  |  |  |

Table . Describe transaction histories table attributes

* Favourite property table (favourite\_properties)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type | Required | Constraint | Description |
| favourite\_ property\_id | Integer | Not null | Primary Key | Code of favourite property |
| user\_id | Integer | Not null | Foreign Key | User |
| property\_id | Integer | Not null | Foreign Key | Property |
| created\_at | LocalDataTime |  |  |  |

Table . Describe favourite properties table attributes

* Category table (categories)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type | Required | Constraint | Description |
| category\_id | Integer | Not null | Primary Key | Code of category |
| Name | String | Not null |  | Name of category |

Table . Describe category table attributes

* Property table (properties)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type | Required | Constraint | Description |
| property\_id | Integer | Not null | Primary Key | Code of property |
| user\_id | Integer | Not null | Foreign Key | User |
| category\_id | Integer | Not null | Foreign Key | Type of property |
| property\_legal \_document\_id | Integer | Not null | Foreign Key | Type of property legal document |
| status | Enum | Not null |  | Approved or canceled |
| title | String | Not null |  | Title appears at the top of the article |
| description | String | Not null |  | Detailed description of the article |
| region | String | Not null |  | Province name |
| district\_name | String | Not null |  | District name |
| ward\_name | String | Not null |  | Ward name |
| street\_name | String | Not null |  | Street name |
| longitude | Number | Not null |  | Coordinates |
| latitude | Number | Not null |  |
| direction | Number | Not null |  | Main direction of real estate |
| area | Number | Not null |  |  |
| length | Number | Not null |  |  |
| width | Number | Not null |  |  |
| images | String | Not null |  | Contains image links |
| price | Number | Not null |  |  |
| created\_at | LocalDataTime |  |  |  |
| updated\_ad | LocalDataTime |  |  |  |

Table . Describe property table attributes

* House table (houses)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type | Required | Constraint | Description |
| house\_id | Integer | Not null | Primary Key | Code of house |
| property\_id | Integer | Not null | Foreign Key | Code of property |
| furnished\_ status\_id | Integer | Not null | Foreign Key | Interior condition |
| house\_ type\_id | Integer | Not null | Foreign Key | Type of house |
| floors | Integer | Not null |  |  |
| bedrooms | Integer | Not null |  |  |
| toilets | Integer | Not null |  |  |

Table . Describe house table attributes

* Furnished status table (furnished\_status)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type | Required | Constraint | Description |
| furnished\_ status\_id | Integer | Not null | Primary Key | Code of furnished status |
| name | String | Not null |  | Describe the condition of the interior |

Table . Describe furnished status table attributes

* Type of house table (house\_type)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type | Required | Constraint | Description |
| house\_ type\_id | Integer | Not null | Primary Key | Code of house type |
| name | String | Not null |  | House type description |

Table . Describe type of house table attributes

* House characteristics table (house\_characteristics)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type | Required | Constraint | Description |
| house\_ characteristic\_id | Integer | Not null | Primary Key | Code of house  characteristics |
| name | String | Not null |  | House characteristics description |

Table . Describe house characteristics table attributes

* + Here create an additional table (house\_characteristic\_mappings) to hold the characteristic keys of the house object.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type | Required | Constraint | Description |
| house\_characteristic\_ mapping\_id | Integer | Not null | Primary Key | Code of house  characteristic mapping |
| house\_ characteristics\_id | Integer | Not null | Foreign Key | House  characteristics |
| house\_id | Integer | Not null | Foreign Key | House |

Table . Describe house characteristic mapping table attributes

* Land table (lands)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type | Required | Constraint | Description |
| land\_id | Integer | Not null | Primary Key | Code of house |
| property\_id | Integer | Not null | Foreign Key | Code of property |
| land\_type\_id | Integer | Not null | Foreign Key | Type of land |

Table . Describe land table attributes

* Type of land table (land\_types)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type | Required | Constraint | Description |
| land\_type\_id | Integer | Not null | Primary Key | Code of land type |
| name | String | Not null |  | Land type description |

Table . Describe type of land table attributes

* Land characteristics table (land\_characteristics)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type | Required | Constraint | Description |
| land \_ characteristic\_id | Integer | Not null | Primary Key | Code of land  characteristics |
| name | String | Not null |  | Lanđ characteristics description |

Table . Describe land characteristics table attributes

* + Here create an additional table (land\_characteristic\_mappings) to hold the characteristic keys of the land object.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type | Required | Constraint | Description |
| land \_characteristic\_ mapping\_id | Integer | Not null | Primary Key | Code of land  characteristic mapping |
| land \_ characteristics\_id | Integer | Not null | Foreign Key | Land  characteristics |
| house\_id | Integer | Not null | Foreign Key | Land |

Table . Describe land characteristic mapping table attributes

### Database schema

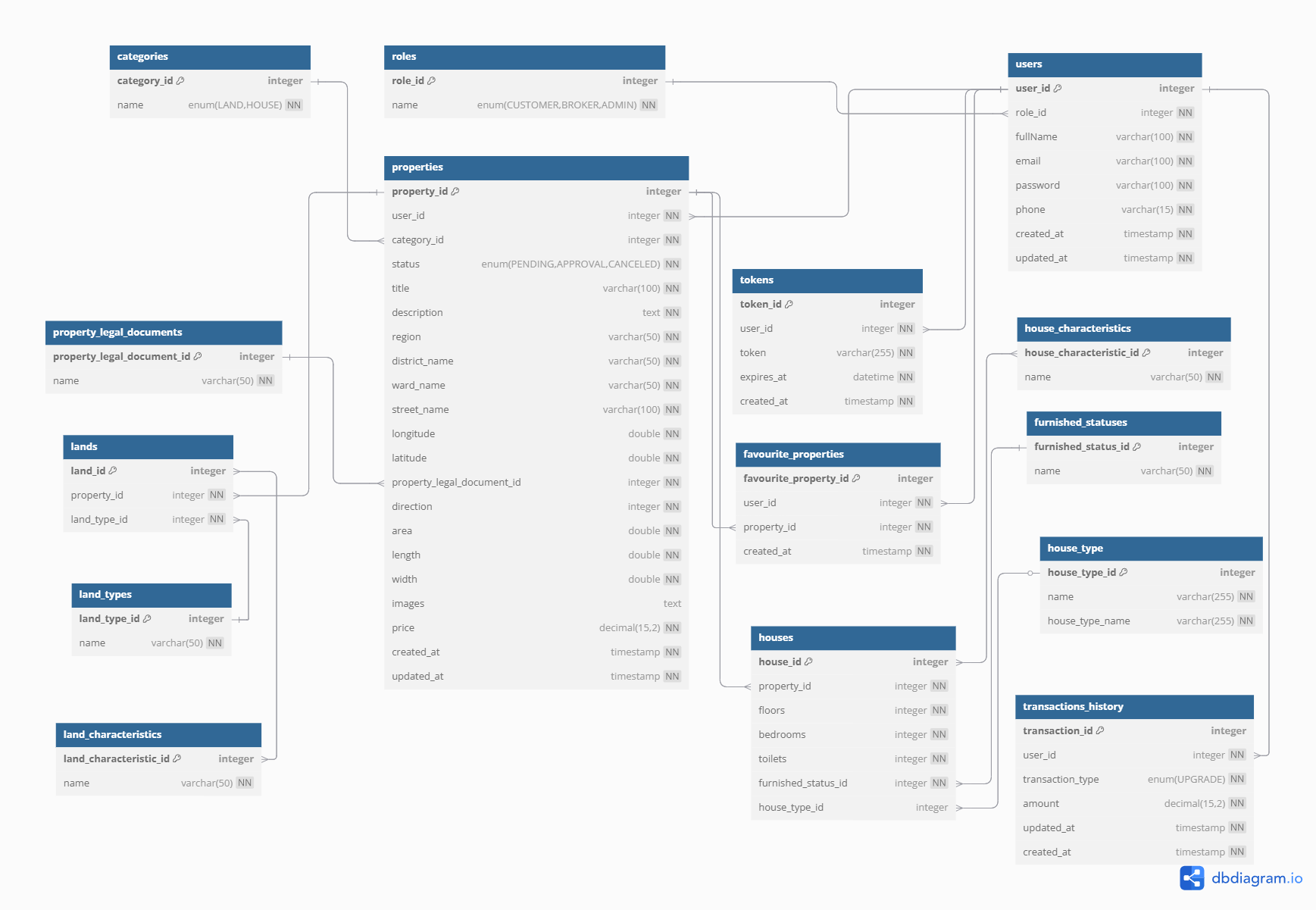


Figure . Relational Model

## Real estate prediction system development

* The problem of predicting real estate prices will be divided into two types including:
  + Real estate valuation: users will provide information related to land or the house they want to sell or buy for the system to evaluate.
  + Land price forecast: the system will predict the price in the near future of a certain area given by users at the same level.
* Common processing steps of the two models
  + Data Crawling: first, we need to prepare a good data set, collect data from the “Nhà tốt” website.
  + Data Cleaning: after having raw data from collection, we will proceed to clean the data through the following steps: handling missing values, handling outlier values.
  + Feature Engineering:
    - Feature Tranformation: following data cleaning, data visualization is conducted to examine relationships between features and evaluate suitable normalization techniques for subsequent processing.
    - Feature Selection: feature selection is performed to prepare the data for model training.

### Real estate valuation model

* After completing the data preprocessing steps—including data cleaning, visualization, normalization, and feature engineering—we proceed to the model training phase. In this phase, we choose an appropriate algorithm for the forecasting task, which in this case is Random Forest due to its robustness and accuracy for regression problems.
* Once the model is trained, we export it to a PMML file. This PMML format enables seamless integration of the trained model into Java-based systems, allowing the model to be deployed and used in production environments without needing to retrain or rewrite it in Java.

### Land forecast model

* This is a time series forecasting problem. After completing the preprocessing phase, we move on to the model training stage. In this case, the chosen model is XGBoost, known for its high performance and ability to handle complex data patterns.
* Once the model is trained, it is saved as a PKL file. However, due to the complexity of XGBoost models, converting them to PMML format is not feasible. As a result, we adopt a hybrid approach in the backend: a Python-based intermediate script is used to load the PKL model and perform the prediction.
* The backend system then exposes this functionality via an API, which acts as a bridge between the Java-based services and the Python model. When a user sends a request, the API triggers the Python script, retrieves the forecasted result, and returns it to the user in real-time.

## Chapter Conclusion

This chapter provides the reader with an overview of the real estate business operations, offering insights into the key activities and processes involved in the industry. It also includes an analysis and design of the system architecture, along with a detailed discussion of the database model. Additionally, the chapter focuses on the development of a price support system, which aims to assist buyers by offering pricing assistance and making real estate transactions more accessible.

# IMPLEMENTATION AND RESULTS EVALUATION

# CONCLUSION AND DEVELOPMENT DIRECTION

# REFERENCES

# APPENDIX 1

# APPENDIX 2