THE UNIVERSITY OF DANANG
DANANG UNIVERSITY OF SCIENCE AND TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

GRADUATION PROJECT THESIS

MAJOR: INFORMATION TECHNOLOGY FACULTY

PROJECT TITLE:

THE REAL-ESTATE WEBSITE

Instructor: **MSc. NGUYEN THI LE QUYEN**
Student: **NGUYEN DINH QUANG**
Student ID: **102150239**
Class**:15TCLC1**

Da Nang, 12/2020

INSTRUCTOR'S COMMENTS

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

REVIEWER'S COMMENTS

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

SUMMARY

Topic title: **THE REAL-ESTATE WEBSITE**

Student name: NGUYEN DINH QUANG

Student ID: 102150239 Class: 15TCLC1

In recent years, along with the growth of the economy, come with the development of society, the accommodation needs are an urgent issue. Besides, a place to share valuable knowledge about well-being of the properties, and a service that provides properties at reasonable prices is extremely needed right now. Our application not only provides those services, but also is a site for user to easily exchange their properties's information and experiences.

DA NANG UNIVERSITY

UNIVERSITY OF SICIENCE AND TECHNOLOGY

FACULTY INFORMATION TECHNOLOGY

THE SOCIALIST REPUBLIC OF VIETNAM

Independence - Freedom - Happiness

GRADUATION PROJECT REQUIREMENTS

Student Name: NGUYEN DINH QUANG Student ID: 102150235

Class: 15TCLC1 Faculty: Information technology Major: Software Engineering

1. *Topic title:*
   The read-estate website
2. *Project topic :* ☐ *has signed intellectual property agreement for final result*
3. *Initial figure and data:*
   Data is generated by myself and search on the Internet

Content of the explanations and calculations:

- Manage users
- Manage properties
- Predict property's price
- Provide knownledge about properties

4. *Drawings, charts:*

- Use Case diagram
- Sequence diagram
- Class diagram
- Database design

5. *Instructor(s):* Nguyen Thi Le Quyen Ph.D
6. *Date of assignment:* *……/……/2020*
7. *Date of completion:* *……/……/2020*

Đà Nẵng, date month year 2020

Head of Division……. **Instructor**

ACKNOWLEDGEMENT

First of all, I would like to express my gratitude to Nguyen Thi Le Quyen, who has guided and assisted me in the graduation project performance

I would like to express my sincere thanks to my teachers who had provided me basis knowledge and learning spirit. Their help is very important to me in the graduation project performance.

I appreciate feedbacks from teachers as well as all my friends so my results have can be improved.

Once again, I sincerely thank.

Student implementation

Nguyen Dinh Quang

ASSURRANCE

I assure:

1. The content of this project is done by me under the direct guidance of MSc. Nguyen Thi Le Quyen
2. If there are invalid or infringing copies, I accept full responsibility.

Da Nang, Date 12 month 12 year 2020

Student implementation

Nguyen Dinh Quang

CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYM

Abbreviations	Explanations
Admin	Administrator
API	Application Programming Interface
JSON	JavaScript Object Notation
MSE	Mean square error
RMSE	Root mean square error
IQR	Interquartile Range

INTRODUCTION

1. Purpose of the topic

Recently, social life (agriculture, industry, culture and society ...) has developed strongly in all aspects, the population increasingly crownded. Therefore,the number of accommodation needs has grown enormously in size. As a result, the property search engine has a significant role in our life

Therefore, the Real-Estate Website is designed to :

1. Provide suitable properties for citizen.
2. Predict price for sellers.
3. Share the needed knowledge about real estate market.

The goal of topic:

- The system helps users to find information about properties, evaluation of properties and necessary market knowledge quickly.
- Website is designed to be flexible and easy for user.

2. Research objects and scope

All property owner, seller and seeker in Vietnam

Information technology components related to search, management and predict including: system design analysis, machine learning, Reactjs framework, Nodejs, MongoDB.

3. Methods of implementation

Survey real-estate market websites, collect information and data about properties.

CHAPTER 1: THEORY AND TECHNOLOGIES

Below are theories of technologies I did use in developing my project

1.1. Machine learning

1.1.1. Introduction

In this chapter, I am going to introduce about machine learning which I apply in my thesis. Today, machine learning is widely used in many fields. Typically like the self-driving Google car, cyber fraud detection, online recommendation engines like friend suggestions on Facebook, Netflix showcasing the movies and shows you might like, and more items to consider and get yourself a little something on Amazon are all examples of applied machine learning.

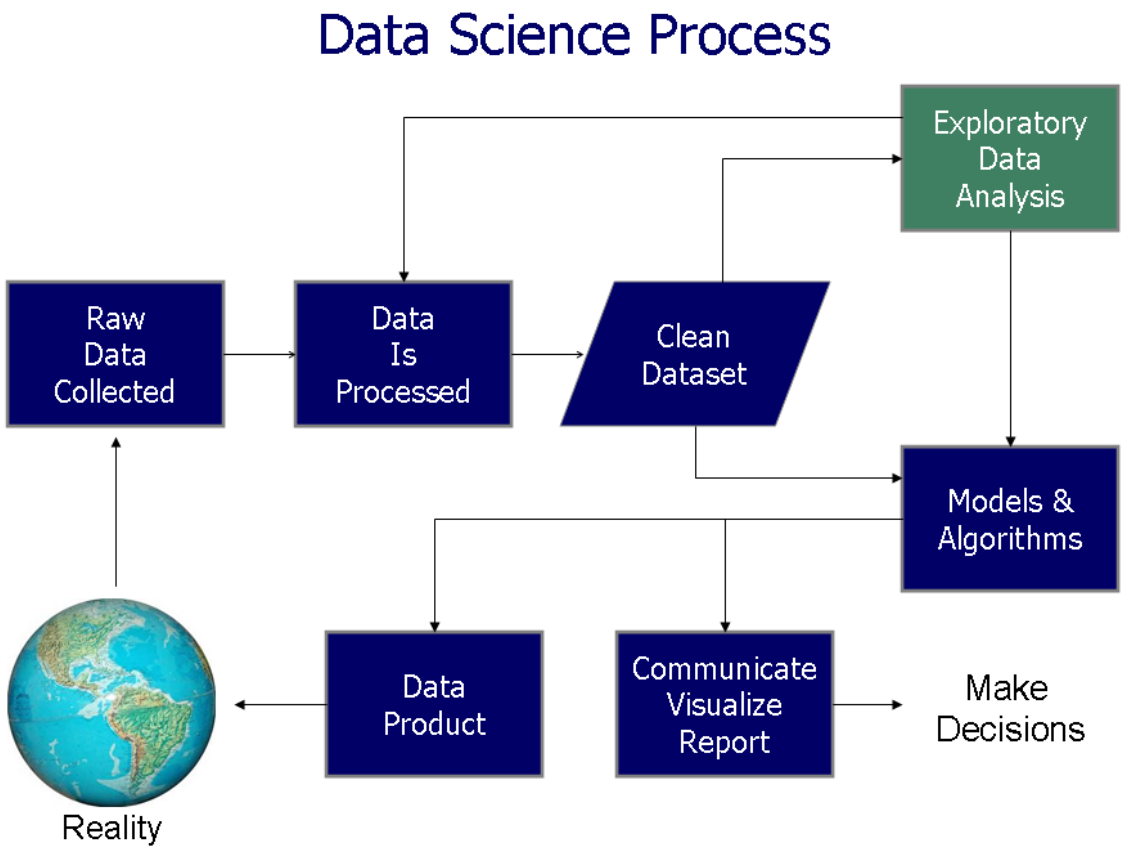


Figure 1.1 Data science solves the problems in reality

1.1.2. Machine learning model

1.1.2.1. Machine learning

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves.

Machine learning tasks are typically classified into two broad categories:

- Supervised learning: The computer is presented with example inputs and their desired outputs, given by a "teacher", and the goal is to learn a general rule that maps inputs to outputs
- Unsupervised learning: No labels are given to the learning algorithm, leaving it on its own to find structure in its input. Unsupervised learning can be a goal in itself (discovering hidden patterns in data) or a means towards an end (feature learning).

Figure 1.1 shows the General model for Machine Learning problems.

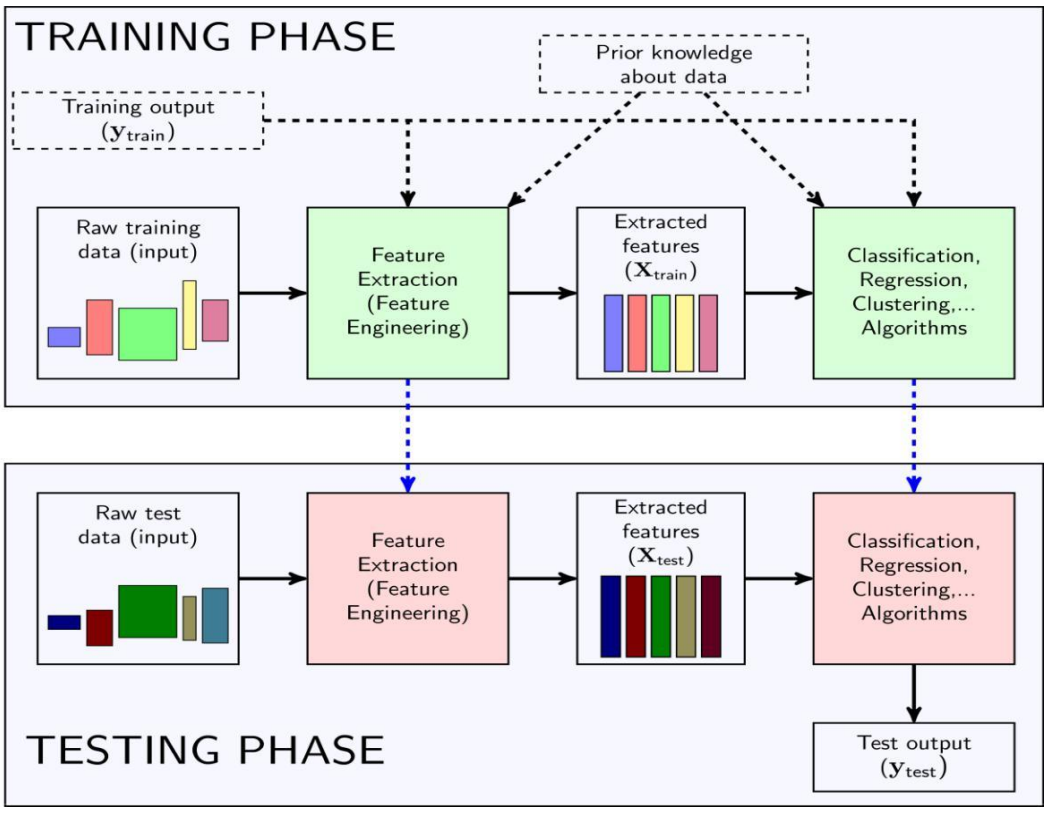


Figure 1.2 General model for Machine Learning problems

1.1.2.2. Supervise learning

Supervised learning is the machine learning task of inferring a function from labeled training data. The training data consist of a set of training examples. In supervised learning, each example is a pair consisting of an input object (typically a vector) and a desired output value (also called the supervisory signal). A supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new examples.

The main types of supervised learning algorithms include:

- Classification algorithms: These algorithms build predictive models from training data which have features and class labels. These predictive models in-turn use the features learnt from training data on new, previously unseen data to predict their class labels.
- Regression alglrithms: These algorithms are used to predict output values based on some input features obtained from the data. To do this, the algorithm builds a model based on features and output values of the training data and this model is used to predict values for new data. The output values in this case are continuous and not discrete.

1.1.2.3. Unsupervised learning

Unsupervised learning is a type of machine learning algorithm used to draw inferences from datasets consisting of input data without labeled responses.

The most common unsupervised learning method is cluster analysis, which is used for exploratory data analysis to find hidden patterns or grouping in data. The clusters are modeled using a measure of similarity which is defined upon metrics such as Euclidean or probabilistic distance.

Common clustering algorithms include:

- Hierarchical clustering: builds a multilevel hierarchy of clusters by creating a luster tree
- K-means clustering: partitions data into k distinct clusters based on distance to the centroid of a cluster
- Gaussian mixture models: models custers as a mixture of multivariate normal density components

- ❖ Self-organizing maps: uses neural networks that learn the topology and distribution of the data

1.1.3. Linear Regression

1.1.3.1. Mathematical basis

In statistics, linear regression is a linear approach to modeling the relationship between a scalar response and one or more explanatory valiables (also known as dependent and independent variables).

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

- ŷ is the predicted value.
- n is the number of features.
- θ is the model parameter (including the bias term and the feature weights).

This can be written much more concisely using a vectorized form

$$\hat{y} = h_{\boldsymbol{\theta}}(\mathbf{x}) = \boldsymbol{\theta} \cdot \mathbf{x}$$

- **θ** is the model's parameter vector, containing bias term and the feature weights
- **x** is the instance's feature vector
- **θ . x** is the dot product of the vectors **θ** and **x**
- **h()** is the hypothesis function, using the model parameters **θ**

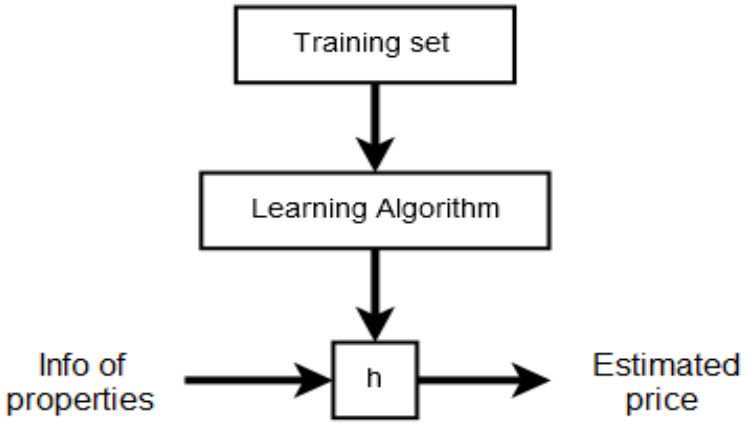


Figure 1.3 Prediction assumption model

Thus, to build the equation, it is necessary to find h, which is to find the teta parameter set. The way to find the parameter set is a trial and error method. From those results will find the best set of parameters. And from there find the teta parameter set.

But if we just do so, there will be infinitely many searches. We first need a measure of how well (or poorly) the model fits the training data, so there is a method to evaluate the good and bad model and try strategically so it finds the fastest.

The common performance measure of a regression model is the Mean Square Error (RMSE). The RMSE of a Linear Regression is calculated by using equation below:

$$\mathrm{RMSE}(\mathbf{X}, h) = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(h\left(\mathbf{x}^{(i)}\right) - y^{(i)}\right)^2}$$

RMSE is also known as the cost function of model which is a function that measures the performance of a Machine Learning model for given data. Cost function quantifies the error predicted values and expected values and presents it in the form of a single real number. To make a good model it must be minimized the cost function. This lead to a new concept Gradient Descent.

Gradient Descent is a very generic optimization algorithm capable of finding optimal solutions to a wide range of problems. The general idea of

Gradient Descent is to tweak parameters iteratively in order to minimize a cost function.

Suppose lost in the mountains in a dense fog. you can only feel the slope of the ground below your feet. A good strategy to get to the bottom of the valley quickly is to go downhill in the direction of the steepest slope. This is exactly what Gradient Descent does. It measures the local gradient of the error function with regards to the parameter vector **θ**, and it goes in the direction of descending gradient. Once the gradient is zero, it have reached a minimum.

Concretely, start by filling θ with random values (this is called random initializa-tion), and then improving it gradually, taking one baby step at a time, each step attempting to decrease the cost function (e.g., the RMSE), until the algorithm converges to a minimum.

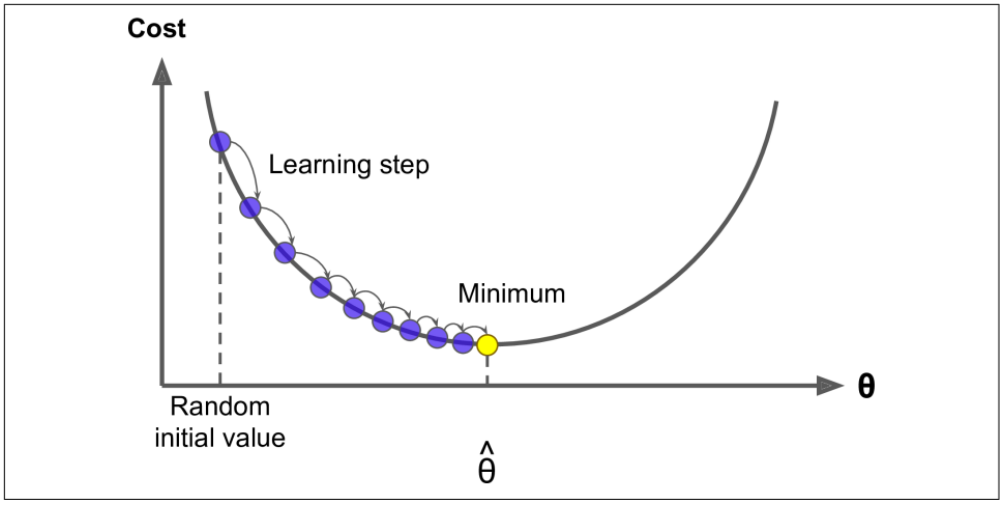


Figure 1.4 Gradient Decent

An important parameter in Gradient Descent is the size of the steps, determined by the learning rate hyperparameter. If the learning rate is too small, then the algorithm will have to go through many iterations to converge, which will take a long time.

In general, gradient-decent consist of 3 steps to minimized the cost function:

- ✓ Start with random θ values
- ✓ Change the θ parameter set to reduce cost-function
- ✓ Repeat the above step until we believe that cost-function is minimum.

1.1.3.2. Algorithm

Cost function :

$$J(\theta) = \frac{1}{2N}\sum_{i=1}^{N}(h_\theta(x^{(i)}) - y^{(i)})^2$$

Gradient decent algorithm:

Repeat until convergence{

$\theta_j := \theta_j - \alpha\frac{\partial}{\partial\theta_j}J(\theta)$ // simultaneously update for every j=0,...,n

}

$$\theta_0 = \theta_0 - \alpha\frac{1}{N}\sum_{i=1}^{N}(h_\theta(x^{(i)}) - y^{(i)})x_0^{(i)}$$ $x_0^{(i)}=1$

$$\theta_1 = \theta_1 - \alpha\frac{1}{N}\sum_{i=1}^{N}(h_\theta(x^{(i)}) - y^{(i)})x_1^{(i)}$$

$$\theta_2 = \theta_2 - \alpha\frac{1}{N}\sum_{i=1}^{N}(h_\theta(x^{(i)}) - y^{(i)})x_2^{(i)}$$

1.2. React JS

1.2.1. Introduction

React is a declarative, efficient, and flexible JavaScript library for building user interfaces. It lets you compose complex UIs from small and isolated pieces of code called “components” .[1]

React JS is an open source and component based JavaScript library used for creating dynamic and interactive user interfaces, especially for single page applications. It is highly flexible, declarative and efficient for developing scalable, simple, and fast front-end for web & mobile applications. React JS effectively handles the view layer of mobile and web app.

It allows developers to create intuitive, interactive and engaging applications with minimum coding and the best rendering performance. With the JavaScript library, developers can build large scale applications without reloading the page where data reflect in real-time

1.2.2. Feature of react

1.2.2.1. Virtual DOM

Virtual DOM (Document Object Model) is the key feature that enables React JS to build scalable and fast applications. It contains a memory reconciliation algorithm that helps React to create a representation of a web page in virtual memory. So Virtual DOM is the representation of original DOM.

Whenever changes occurred in the web application, the entire user interface is re-rendered in the Virtual DOM representation. Developers can check the difference between new DOM and earlier DOM representation. Later on, the original DOM will update only those components that have actually changed and not all components of virtual DOM. Thus, it helps in developing the app faster with the wastage of memory.

1.2.2.2. JSX

JSX stands for JavaScript XML. It is a syntax extension that describes how the web or mobile app UI should look like. The markup syntax closely resembles with HTML. It is a combination of JavaScript and XML.

With JSX, developers can build blocks of React UI, and react components easily by making the syntax of HTML. It is identical to the HTML the programmers will inject

in the web page. Thus, JSX is the best feature offered by React JS as it allows web developers to go for an easy way out.

1.2.2.3. One Way Data Binding

One way data binding does not permit developers to edit any properties of the React JS component directly. With the help of callback function, they can modify the property. This process is known as one way data binding.

React JS follows one-way data binding or unidirectional data flow that gives better control throughout the application. Additional features are required to manage the data flow that comes from another direction.

As React components are immutable, the data within them cannot be changed. Using Flux, developers can keep data unidirectional and makes the application more flexible that improves its efficiency.

1.2.2.4. Components

React components are small, reusable pieces of code that return a React element to be rendered to the page. It contains numerous components that create a user interface of web applications and each component is designed with a specific logic. The logic is written in JavaScript instead of templates, so developers can easily pass the data throughout the application and keep the state out of the DOM.

1.2.3. React advantages and disadvantages

❖ Advantages :

- Uses virtual DOM which is a JavaScript object. This will improve apps performance, since JavaScript virtual DOM is faster than the regular DOM.
- Can be used on client and server side as well as with other frameworks.
- Component and data patterns improve readability, which helps to maintain larger apps.

❖ Disadvantages :

- Covers only the view layer of the app, hence you still need to choose other technologies to get a complete tooling set for development.
- Uses inline templating and JSX, which might seem awkward to some developer.[5]

1.2.4. Lifecycle of Components

The component life cycle is shown in the figure 1.5

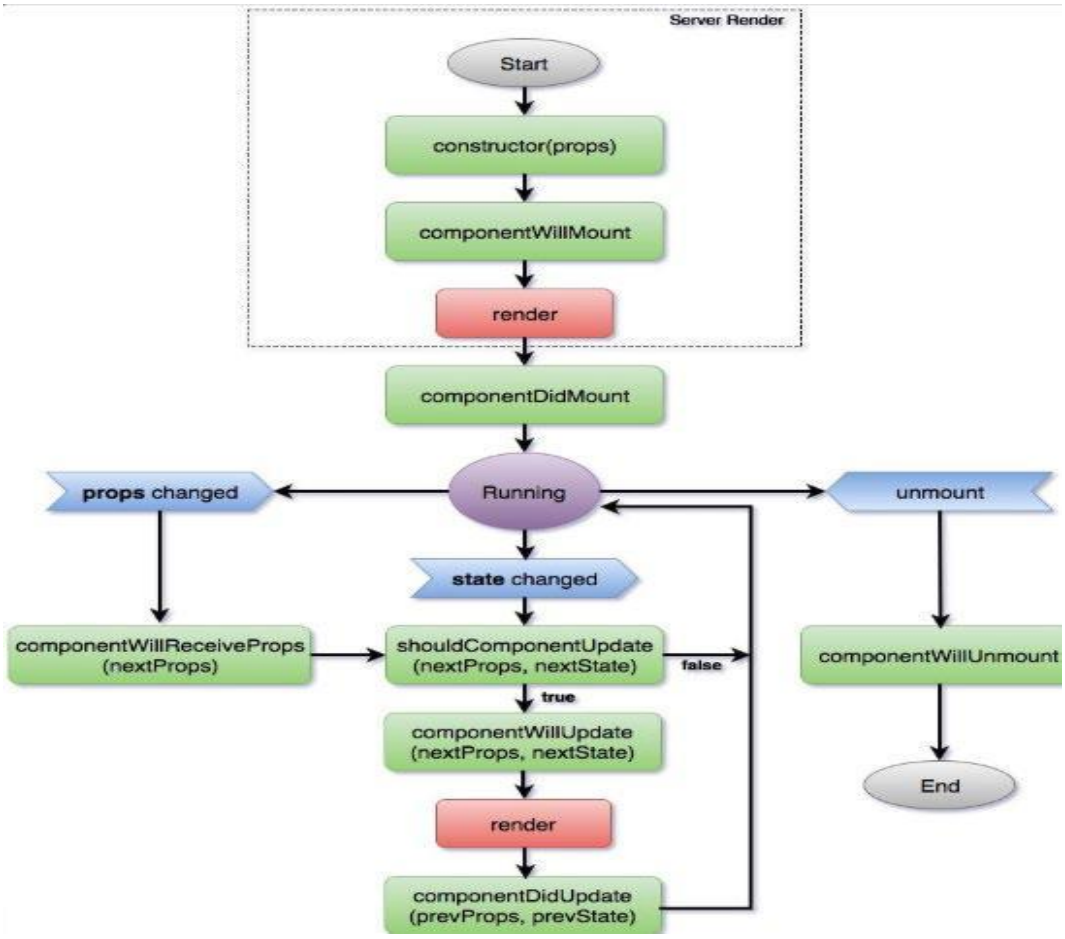


Figure 1.5 Component Life Cycle

Lifecycle Methods

- **componentWillMount** is executed before rendering, on both the server and the client side.
- **componentDidMount** is executed after the first render only on the client side. This is where AJAX requests and DOM or state updates should occur. This method is also used for integration with other JavaScript frameworks and any functions with delayed execution such as **setTimeout** or **setInterval**. We are using it to update the state so we can trigger the other lifecycle methods.

- **componentWillReceiveProps** is invoked as soon as the props are updated before another render is called. We triggered it from **setNewNumber** when we updated the state.
- **shouldComponentUpdate** should return **true** or **false** value. This will determine if the component will be updated or not. This is set to **true** by default. If you are sure that the component doesn't need to render after **state** or **props** are updated, you can return **false** value.
- **componentWillUpdate** is called just before rendering.[7]

1.2.5. Redux

1.2.5.1. Definition

Redux is a predictable state container for JavaScript apps. As the application grows, it becomes difficult to keep it organized and maintain data flow. Redux solves this problem by managing application's state with a single global object called Store. Redux fundamental principles help in maintaining consistency throughout your application, which makes debugging and testing easier.[2]

More importantly, it gives you live code editing combined with a time-travelling debugger. It is flexible to go with any view layer such as React, Angular, Vue, etc.

1.2.5.2. Principles of Redux:

Predictability of Redux is determined by three most important principles as given below:

- **Single Source of Truth:**

The state of your whole application is stored in an object tree within a single store. As whole application state is stored in a single tree, it makes debugging easy, and development faster.

- **State is Read-only:**

The only way to change the state is to emit an action, an object describing what happened. This means nobody can directly change the state of your application.

- **Changes are made with pure functions:**

To specify how the state tree is transformed by actions, you write pure reducers. A reducer is a central place where state modification takes place. Reducer is a function which takes state and action as arguments, and returns a newly updated state.

1.2.5.3. Redux Data Flow:

Redux follows the unidirectional data flow. It means that your application data will follow in one-way binding data flow. As the application grows & becomes complex, it is hard to reproduce issues and add new features if you have no control over the state of your application.

Redux reduces the complexity of the code, by enforcing the restriction on how and when state update can happen. This way, managing updated states is easy. We already know about the restrictions as the three principles of Redux(as shown in figure 1.6). Following diagram will help you understand Redux data flow better

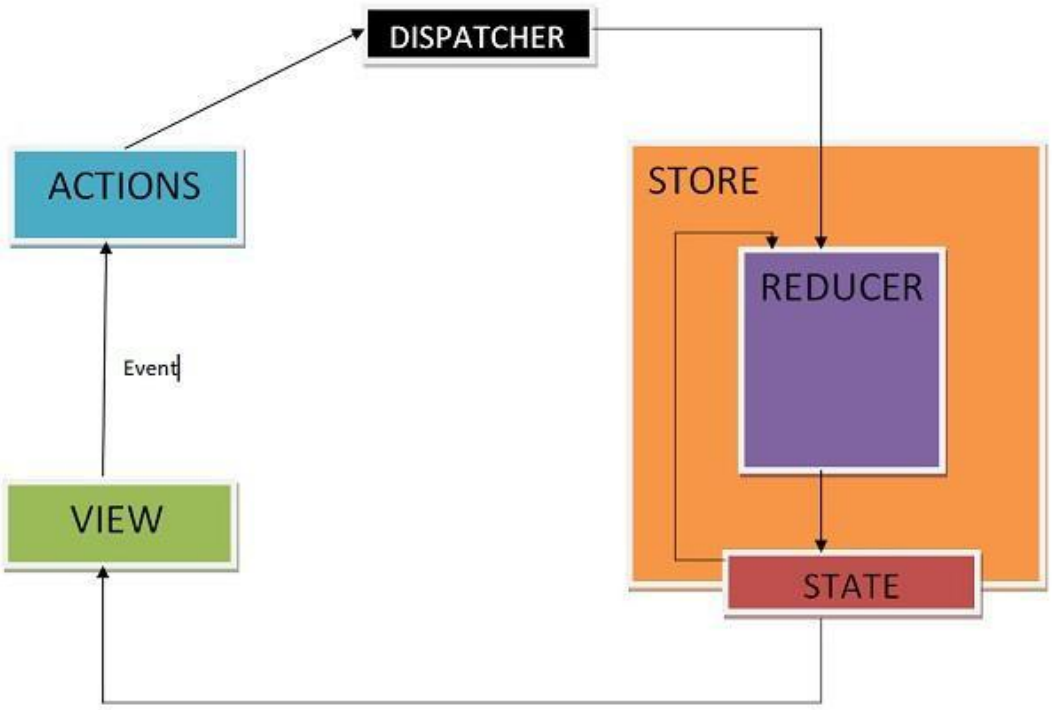


Figure 1.6 Redux data flow

- An action is dispatched when a user interacts with the application.
- The root reducer function is called with the current state and the dispatched action. The root reducer may divide the task among smaller reducer functions, which ultimately returns a new state.
- The store notifies the view by executing their callback functions.

The view can retrieve updated state and re-render again

CHAPTER 2: SYSTEM ANALYSIS AND DESIGN

2.1. Build machine learning model

2.1.1. Data collection

In this chapter, the data collected from https://alonhadat.com.vn/ .This website is one of the few websites that can crawl land data. All necessary informations are collected to train and test for model. The feature is:

- ✓ Price (VND)
- ✓ Square (m^2)
- ✓ Width (m)
- ✓ Direction
- ✓ Street width (m)
- ✓ District

I use the web data extraction tool called web scraper. Web Scraper utilizes a modular structure that is made of selectors, which instructs the scraper on how to traverse the target site and what data to extract. Thanks to this structure, Web Scraper is able to extract information from modern and dynamic websites such as Amazon, Tripadvisor, eBay, and so on, as well as from smaller, lesser-known websites.

Web Scraper runs in browser and doesn't anything require to be installed on your computer. We don't need any Python, PHP, or JavaScript coding experience to start scraping with Web Scraper. Additionally, Web Scraper offers you the ability to completely automate data extraction in Web

Because of the data limitation, I was only able to collect 7196 observation from 13 district of country. The data source is quite subjective, there are many errors and a lot of data is missing.

Figure 2.1 provides raw data with many missing values, input errors and unit differences.

price	square	width	direction	street
2,65 tỷ	75 m2	5m	Đông	5,5m
3,7 tỷ	111 m2	---	Tây Bắc	3m
3,5 tỷ	90 m2	5m	_	5,5m
1,75 tỷ	65 m2	4m	_	2,5m
2,1 tỷ	77 m2	---	_	2,5m
1,37 tỷ	53 m2	7,1m	Tây	2,5m
17,5 tỷ	2.008 m2	8m	Bắc	40m
39 triệu / m2	120 m2	---	_	15m
2,4 tỷ	56 m2	6,3m	Đông Nam	3m
4,5 tỷ	117 m2	---	_	30m
2,25 tỷ	56 m2	4m	Tây Bắc	3,5m
1,5 tỷ	52 m2	7,1m	Tây	2,5m
39 triệu / m2	180 m2	10m	_	5m

Figure 2.1 Raw data is collected from the website

Obviously, if the training data is full of errors, outliers, and noise (e.g., due to poor quality measurements), it will make it harder for the system to detect the underlying patterns, so your system is less likely to perform well. It is often well worth the effort to spend time cleaning up your training data. The truth is, most data scientists spend a significant part of their time doing just that. For example:

- If some observation are clearly outliers, it may help to simply discard them or try to fix the errors manually
- If some instances are missing a few features (e.g., 5% of properties did not specify street), I must decide whether I want to ignore this attribute altogether, ignore these instances, fill in the missing values (e.g., with the median or mean age), or train one model with the feature and one model without it, and so on.[6]

2.1.2. Data preprocessing

2.1.2.1. Handling error and missing value

Because of the limited amount of data, I don’t have a change co choose a method for removing faulty observations or features. Therefore, data must be handled very carefully to retain as much as possible while also ensuring data purity.

The price field are entered with many types such as: billion, million, million/m2, dv or ‘agreement’. The first is to convert to a single unit of measure, the billion and delete these units, leaving only the number.

The width field has a lot of missing values, normally there are 3 options to handle this: delete this feature, replace it with a mean or median value and delete the observations containing the missing values. But in my case, there are 18% observation have missing values in width. Since width depends on square, i will replace missing values based on square values.

```
18      ---
19      ---
21      ---
25      ---
33      ---
       ...
7187    ---
7189    ---
7190    ---
7194    ---
7195    ---
Name: width, Length: 1305,
```

Figure 2.2 Missing values of width

In the direction field, perhaps the lack of determining the direction when posting causes a lot of data missing in this filed, nearly 60%. Instead of deleting or replace it with median, I choose the missing value as an unknown value.

	price	square	width	direction	street	district
25	3.10	97	---	_	7.5	camle
36	2.50	90	6	_	5.5	camle
37	1.90	98	5.5	_	3.5	camle
38	2.68	100	12	_	33	camle
45	1.49	100	---	_	3	camle
...	...	...	...	...	...	...
7191	6.90	100	5.2	_	5	hbt
7192	5.98	100	6.2	_	2	hbt
7193	24.00	120	5	_	20	hbt
7194	2.00	35	---	_	2	hbt
7195	2.90	45	---	_	3	hbt

4249 rows × 6 columns

Figure 2.3 Missing values of direction

After finishing processing, checking the data again and see that there are no other missing values in the data set.

Check missing values

```
data.isnull().sum()
```

```
price        0
square       0
width        0
direction    0
street       0
district     0
dtype: int64
```

Figure 2.4 Checking missing values

2.1.2.2. Handling outlier

Sometimes a dataset can contain extreme values that are outside the range of what is expected and unlike the other data. These are called outliers and often machine learning modeling and model skill in general can be improved by understanding and even removing these outlier values.

	price	square	width	street
count	7196.000000	7196.000000	7196.000000	7196.000000
mean	6888.226153	129.035575	5.762692	11.771776
std	12370.979880	79.336749	2.894485	9.788670
min	300.000000	21.000000	2.300000	2.000000
1%	437.950000	36.000000	3.000000	2.000000
25%	1500.000000	82.000000	4.000000	6.000000
50%	3210.000000	100.000000	5.000000	8.000000
75%	6705.000000	149.250000	6.000000	14.000000
99%	65000.000000	440.000000	17.000000	55.000000
max	165000.000000	494.000000	30.000000	75.000000

Figure 2.5 Descriptive statistics

Through the statistics described above, the standard deviation is too large, as well as the difference between the data at 99% and max.

One of the ways to handle the outlier is to use the IQR method. The interquartile range (IQR) is a measure of variability, based on dividing a data set into quartiles. The

values that divide each part are called the first, second, and third quartiles; and they are denoted by Q1, Q2, and Q3, respectively.

- Q1 is the “middle” value in the first half of the rank-ordered data set.
- Q2 is the median value in the set.
- Q3 is the “middle” value in the second half of the rank-ordered data set.

The formula for inter-quartile range is given below:

$$IQR=Q3-Q1$$

Where:

IQR=Inter-quartile range

Q_1 = First quartile

Q_3 = Third quartile

This external area value may be seen is outlier.

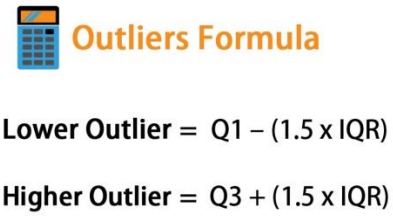


Figure 2.6 The formula calculates outlier

After applying the fomula for processing, the data was more standard and there was no big difference.

	price	square	width	street
count	5420.000000	5420.000000	5420.000000	5420.000000
mean	3547.005498	99.474354	4.793950	9.208487
std	2882.367526	35.224835	1.260874	4.881172
min	300.000000	21.000000	2.300000	2.000000
1%	430.380000	32.380000	3.000000	2.000000
25%	1400.000000	80.000000	4.000000	5.500000
50%	2700.000000	100.000000	5.000000	7.500000
75%	4600.000000	110.000000	5.000000	12.000000
99%	13500.000000	216.000000	8.627200	22.810000
max	14500.000000	250.000000	9.000000	26.000000

Figure 2.7 Data after removing outliers

2.1.3. Data analysis

Feature Price is the goal of analysis and prediction, reason for practice. Investigate the variable price and the correlation between the variables independent of the price.

From the price distribution chart below, we can see the price distribution mostly between 500 million and 7,5 billion VND. For those that have a higher price, the distribution decreases. This is long tail distribution but close to normal distribution, so the price can be used for the linear model.

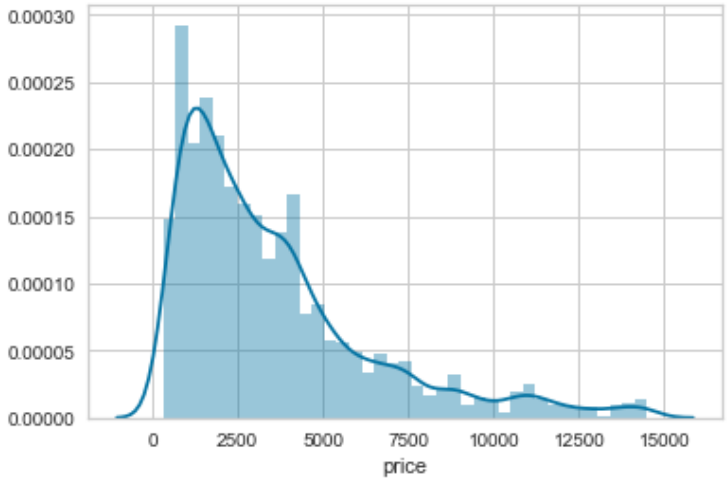


Figure 2.8 Price analysis

Price is positively correlated with square, width and street expressed through collocation map.

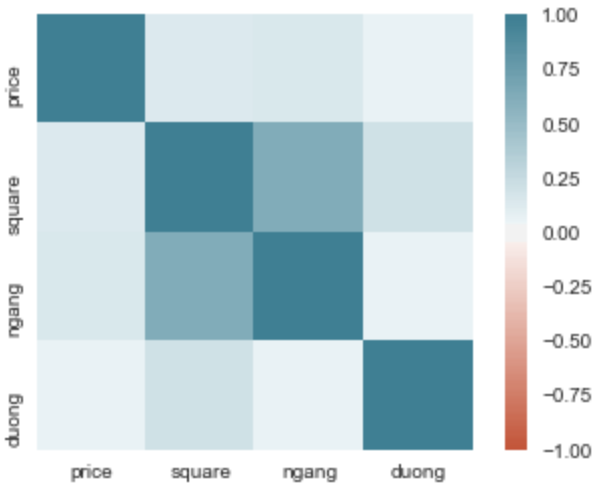


Figure 2.9 Collocation map

Price is correlated with district and direction via the sandbox chart and anova test.

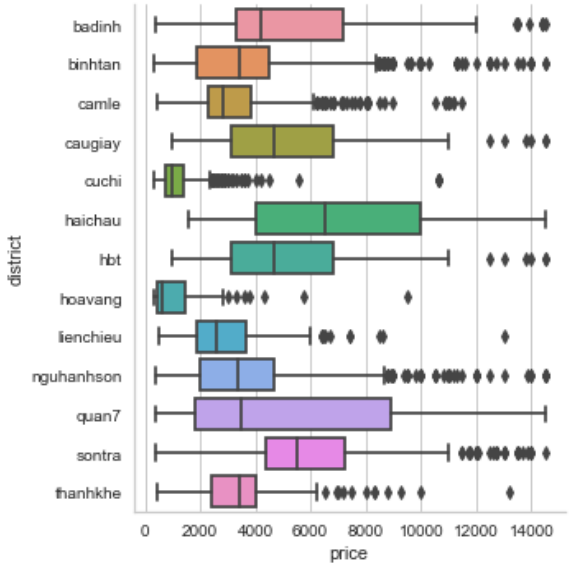


Figure 2.10 Price and district sandbox map

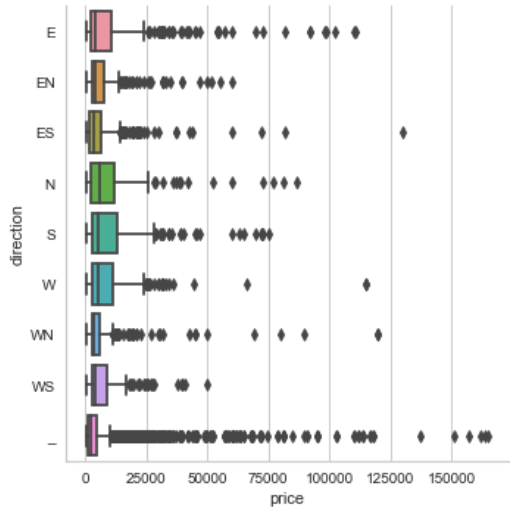


Figure 2.11 Price and direction sandbox map

2.1.4. Feature scaling and get dummy

One of the most important transformations that need to apply to data is feature scaling. With few exceptions, Machine Learning algorithms don't perform well when the input numerical attributes have very different scales. This is case the square ranges from about 30 to 250, while the width only range from 2 to 30. But scaling the target values is generally not required.

There are two common ways to get all attributes to have the same scale: min-max scaling and standardization. I choose min-max scaling to prevent negative values for independent variable.

Min-max scaling (many people call this normalization) is quite simple: values are shifted and rescaled so that they end up ranging from 0 to 1. We do this by subtracting the min value and dividing by the max minus the min. Scikit-Learn provides a transformer called MinMaxScaler for this. It has a feature range hyperparameter that lets us change the range if we don't want 0–1 for some reason.[4]

$$X_{new} = \frac{X_i - \min(X)}{\max(x) - \min(X)}$$

In machine learning or more specifically regression, algorithm will calculate the distance. One issue with this representation is that ML algorithms will assume that two nearby values are more similar than two distant values. This may be fine in some cases (e.g., for ordered categories such as “bad”, “average”, “good”, “excellent”), but it is obviously not the case for the district column. To fix this issue, a common solution is to create one binary attribute per category: one attribute equal to 1 when it is choose, another attribute equal to 0. This is called *one-hot encoding*, because only one attribute will be equal to 1 (hot), while the others will be 0 (cold). The new attributes are sometimes called *dummy* attributes. Scikit-Learn provides a OneHotEncoder class to convert categorical values into one-hot vectors

direction_EN	direction_ES	direction_N	direction_S	direction_W	direction_WN	...	district_caugiay	district_cuchi	district_haichau	district_hbt
0	0	0	0	1	0	...	0	0	0	0
1	0	0	0	0	0	...	0	0	0	0
0	0	0	0	0	0	...	0	0	0	0
0	1	0	0	0	0	...	0	0	0	0
1	0	0	0	0	0	...	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...
0	0	0	0	0	0	...	0	0	0	1
0	0	0	0	0	0	...	0	0	0	1
0	0	0	0	0	0	...	0	0	0	1
0	0	0	0	0	0	...	0	0	0	1

Figure 2.12 One hot encoding

2.1.5. Apply model

Method Hold-out divided data into 2 independent part: training set and test set. This method is suitable for small data sets. However, the samples may not represent all of the data (lack of data in the test set).

Scikit-learn support the train-test-split method that user can use to divide data random le into training and testing set at arbitrary scale. Normally, this ratio is 80:20.[3]

```
X shape: (5420, 25)
X_train shape: (4336, 25)
X_test shape: (1084, 25)
```

Figure 2.13 Holdout

2.1.6. Training and get result

At last i framed the problem, got the data and explored it, sampled a training set and a test set, clean up and prepare data for Machine Learning algorithms. Now ready to select and train a Machine Learning model.

Training models after data normalization is easy thanks to scikit-learn, the library has the models for the user to choose from and simply import them and fit the training-set. And the calculation already has the skikit processed.

Because the features and observation are low, the scikit-learn will take time to create the model very quickly. It only takes 0.03 seconds for the whole calculation process.

```
Training time: 0.032972812652587895
```

Figure 2.14 Training time

The distribution between y_test and y_predict around the line y_test = y_predict. This means our model is on right track

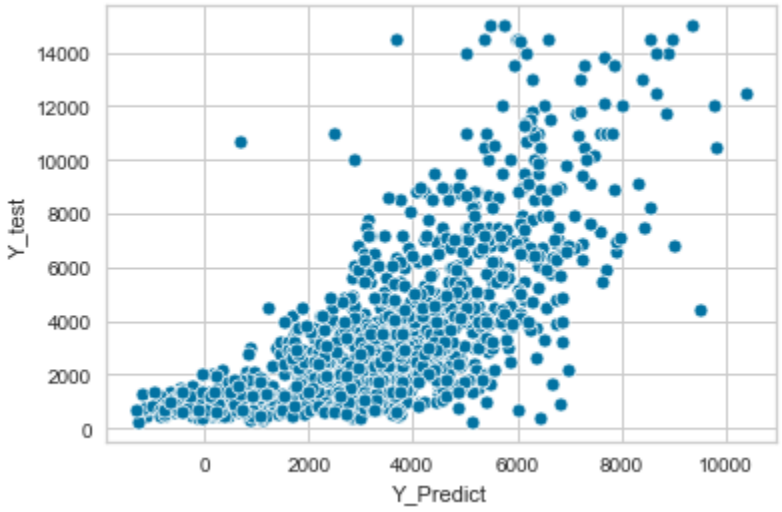


Figure 2.15 Compare y_test and y_predict

It works, although the predictions are not exactly accurate, measure this regression model's RMSE on the whole test-set using Scikit-Learn's mean_squared_error function:

```
Mean Square Error: 4319456.603440959
Root Mean Square Error: 2078.330244075989
Percentage between RMSE and Y-Mean: 58%
```

Figure 2.16 Measure model's precision

The percentage between RMSE and Y-mean is 58%, it is greater than 10% of the average value of the entire price. This means that the model is not very accurate but acceptable.

2.1.7. Try to improve the model

2.1.7.1 PCA

PCA method will represent multidimensional data on a space with orthogonal basis, if we consider each facility in the new space as a variable then the image of the original data in this new space will be represented as independent variables. Problem: if moving the original data to a new space will the interesting information of the original data be lost?

To solve this problem, the PCA method will find a new space with the criterion trying to reflect as much original information as possible, and the measure for the concept of information here is the variance.

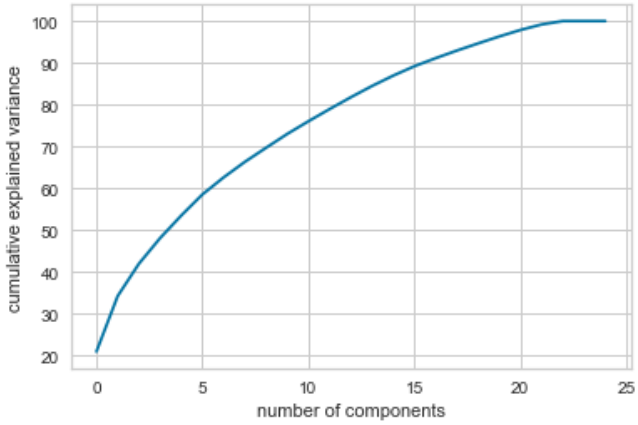


Figure 2.17 PCA

Based on the figure 2.17 that just to 20 components has been represented 95% of the data.

However, the model improves on the speed because less processing is required, but the accuracy is reduced quite a lot.

```
PCA(n_components = 20)
X_train_pca = PCA(n_components = 20).fit_transform(X_train)
X_test_pca= PCA(n_components = 20).fit_transform(X_test)
regressor_pca = LinearRegression()
regressor_pca.fit(X_train_pca,Y_train)
Y_pred_pca = regressor_pca.predict(X_test_pca)
mse = mean_squared_error(Y_test, Y_pred_pca)
rmse = np.sqrt(mse)
print('Percentage between RMSE and Y-Mean: {}%'.format(int((rmse)/(Y.mean())*100)))
```

```
Percentage between RMSE and Y-Mean: 77%
```

Figure 2.18 Precision of model applying PCA

2.1.7.2 Cross validation

A great alternative to Hold-out is to use Scikit-Learn's K-fold cross-validation feature. The following code randomly splits the training set into K distinct subsets called folds, then it trains and evaluates the model K times, picking a different fold for evaluation every time and training on the other K-1 folds. The result is an array con training the K evaluation scores:

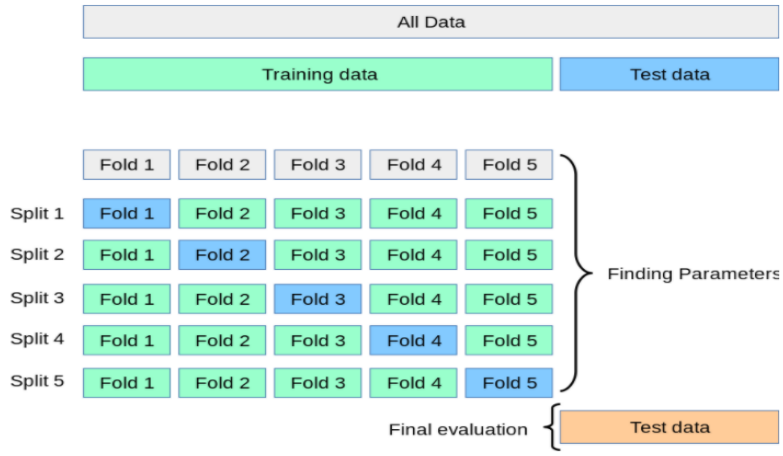


Figure 2.19 Cross-validation method

Running a model with a cross validation with k folds from 2 to 10 got the mean percentage between RMSE and Y-mean in figure 2.20. The score isn't really better than Hold-out so won't be used

```
Cross validation score with 2 fold: price    88.358087
dtype: float64
Cross validation score with 3 fold: price    88.129188
dtype: float64
Cross validation score with 4 fold: price    77.908718
dtype: float64
Cross validation score with 5 fold: price    75.261348
dtype: float64
Cross validation score with 6 fold: price    70.159086
dtype: float64
Cross validation score with 7 fold: price    72.80083
dtype: float64
Cross validation score with 8 fold: price    71.440393
dtype: float64
Cross validation score with 9 fold: price    70.569906
dtype: float64
Cross validation score with 10 fold: price    69.066612
dtype: float64
```

Figure 2.20 Cross-validation method

2.1.7.3 Grid search cv

GridSearchCV is a function that comes in Scikit-learn's model_selection package. This function helps to loop through predefined hyperparameters and fit estimator (model) on the training set. So, in the end, we can select the best parameters from the listed hyperparameters.

```
Fitting 5 folds for each of 7 candidates, totalling 35 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 4 concurrent workers.
Best Score:  2096.254538561689
Best Params:  {'alpha': 0.1}
[Parallel(n_jobs=-1)]: Done  35 out of  35 | elapsed:    3.8s finished
```

Figure 2.21 Best score and param in GridSearchCV

However, GridSearchCV's RMSE best score is greater than holdout, so it doesn't improve the model.

2.2. Specification requirements for website

The website is divided into 2 main parts, the part for users (guests and members) and administrators (admin).

Main functions for guest users include

- Watch product information.
- Search product.
- Watch blog information

Main functions for registered users include:

- Have all function of guests acount.
- Personal account management.
- Predict properties

Main functions for admins include:

- User management.
- Product management
- Blog management.

2.3. System Analysis

2.3.1. Use case diagram

2.3.1.1. Overview of guest functions

Use case diagram shown in figure 2.22 provides description of guest user functions

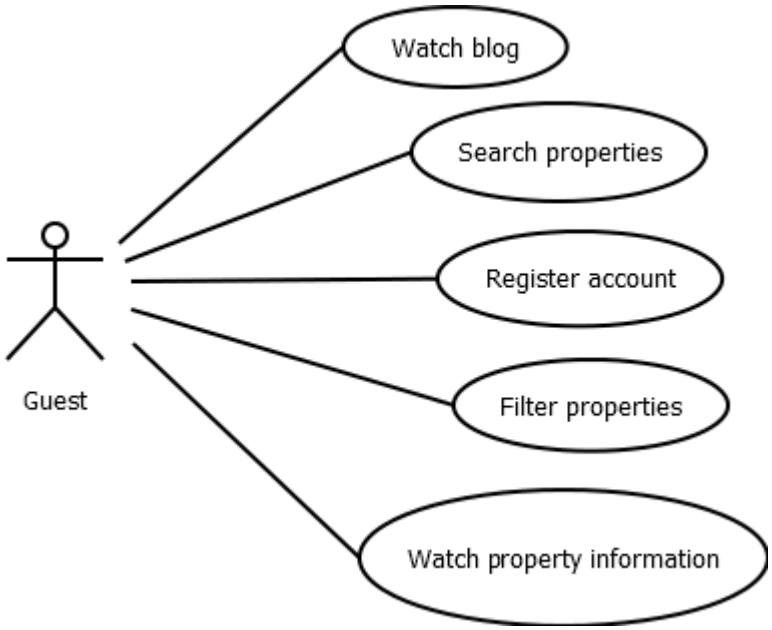


Figure 2.22. The overall use case of the guest

Use case diagram shown in figure 2.23 provides description of watch properties information.

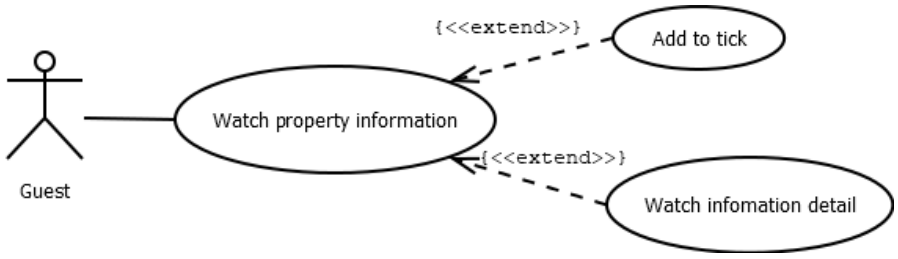


Figure 2.23. View properties information use case

2.3.1.2. Overview of registered user functions

Use case diagram shown in figure 2.24 provides description of registered user functions.

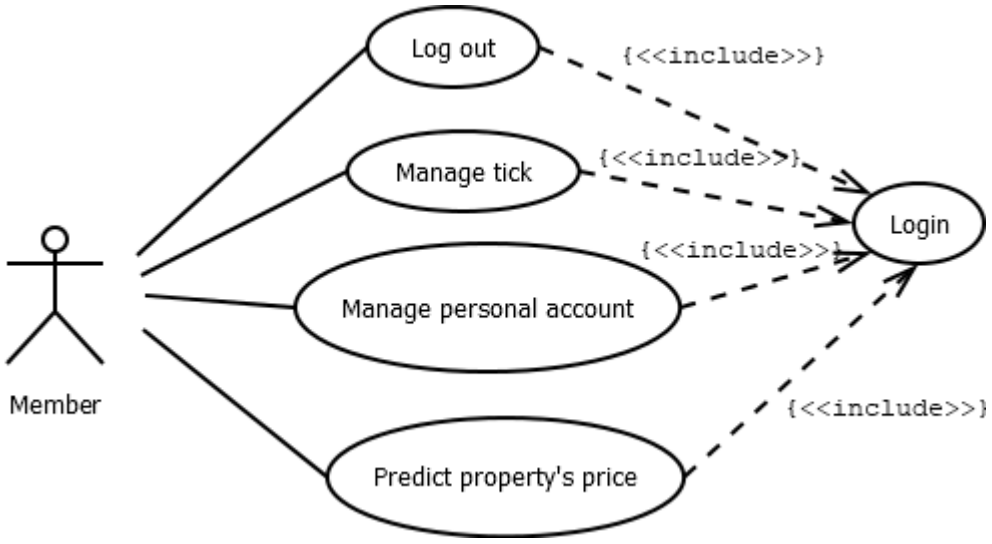


Figure 2.24. The overall use case of the registered users

Use case diagram shown in figure 2.25 provides description of manage personal account of user

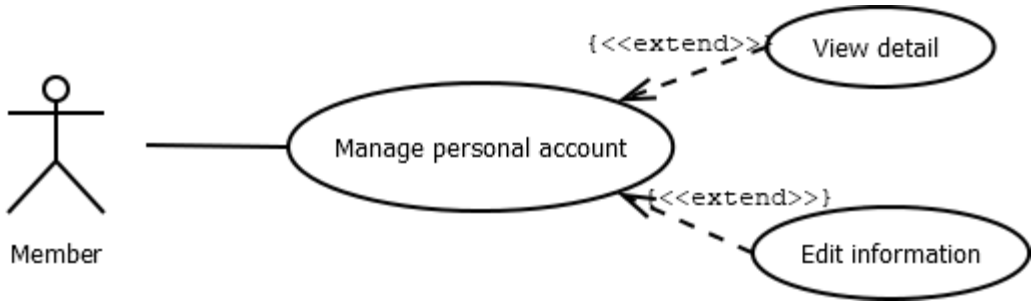


Figure 2.25. Account management use case

2.3.1.3. Overview of admin functions

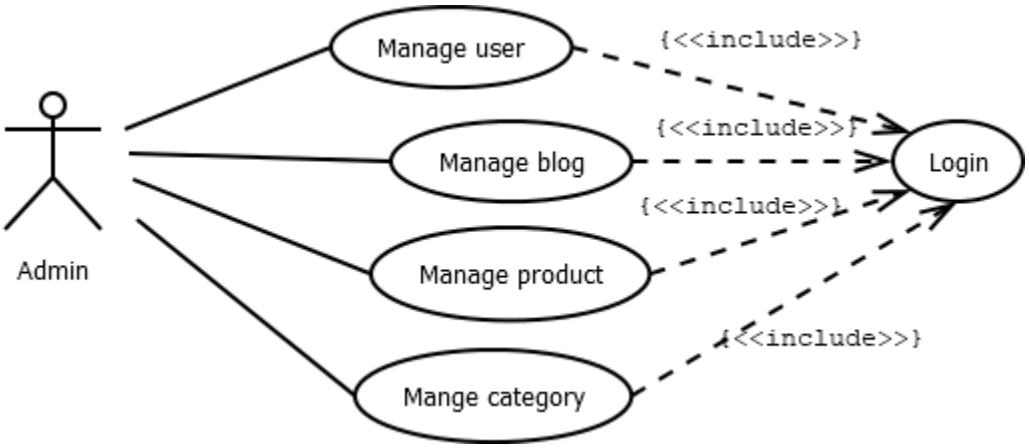


Figure 2.26. Admin management use case

Use case diagram shown in figure 2.26 provides description of admin functions.

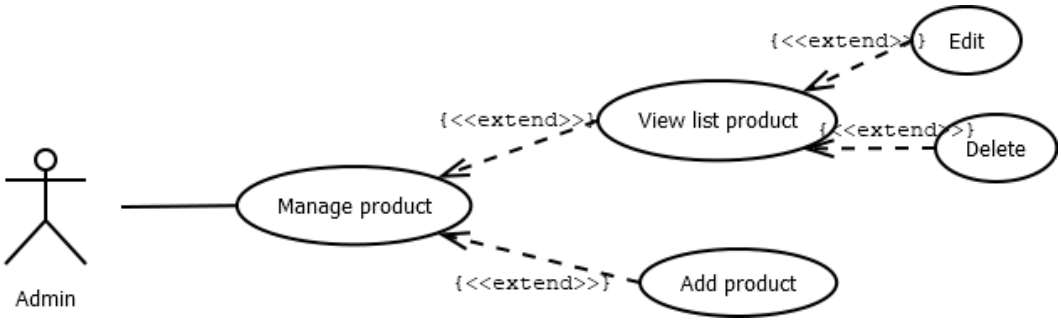
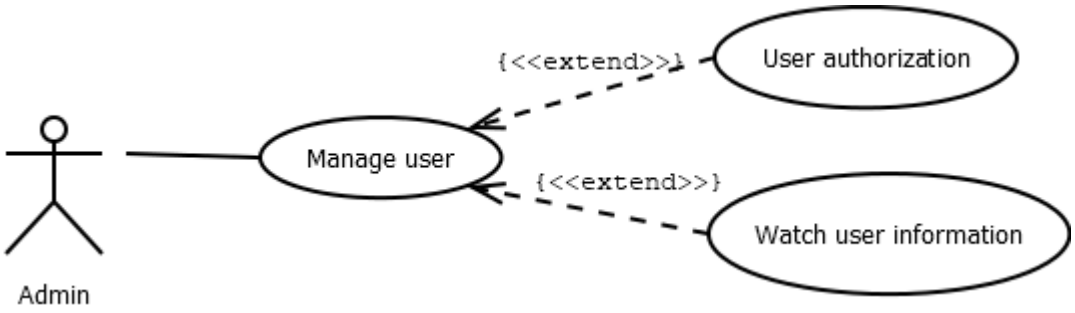


Figure 2.27. Product management use case

Use case diagram shown in figure 2.27 provides description of product management functions.

Table 2.1. Create new product

Use case name	Create new product
User	Admin
Details	User can create new product into the system
Pre- condition	User logged into the system
Main-flows	1 To add products, user click the Add product button on product management page 2 User has the ability to : Choose image of product Add more product information (price,size,details,…) Choose cause (choose item in input element)
Post-condition	If create success, the system notify of user
Alternative	If user fills form with invalid information, the system will display error message at each file



2.28. User management use case diagram

Use case diagram shown in figure 2.28 provides description of user management functions.

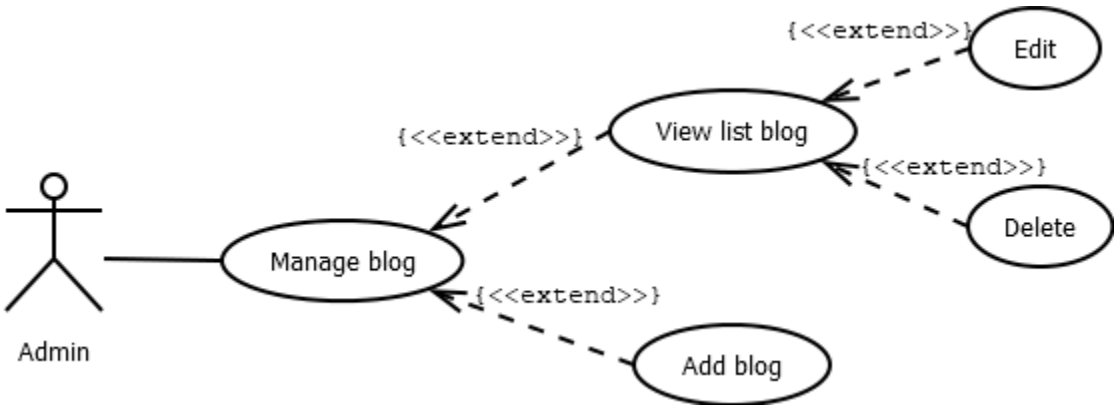


Figure 2.29. Blog management use case diagram

Use case diagram shown in figure 2.29 provides description of blog management functions.

Table 2.2. Create new blog management

Use case name	Create new blog
User	Admin
Details	Admin cal create new blog into the system
Pre- condition	Admin logged into the system
Main-flows	1 To add blog, admin click the Add blog button on blog management page 2 Admin has the ability to : Choose image of blog Add more blog information Choose cause (choose item in input element)
Post-condition	If create success, the system update blog
Alternative	If fills form with invalid information, the system will display error message at each file

2.3.2. Activity diagram

2.3.2.1. Register function

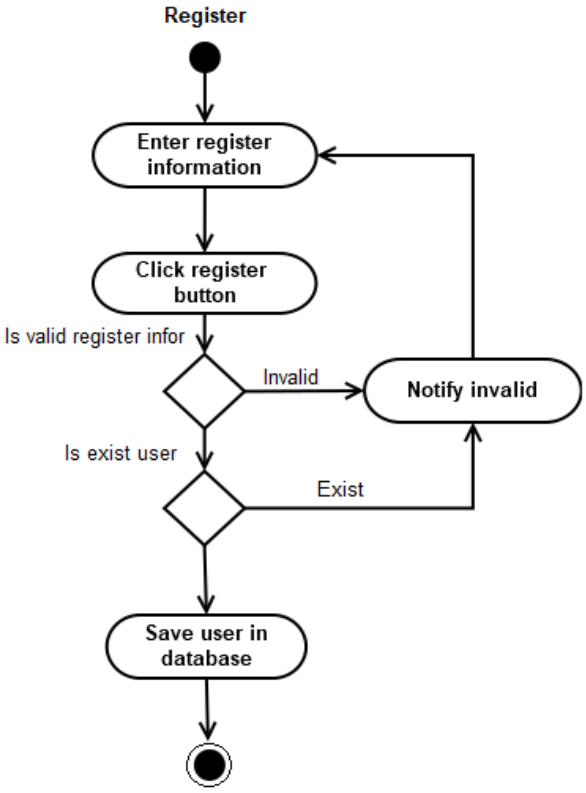


Figure 2.30. Activity register

Activity diagram shown in figure 2.30 provides description of register functions.

2.3.2.1. Register function

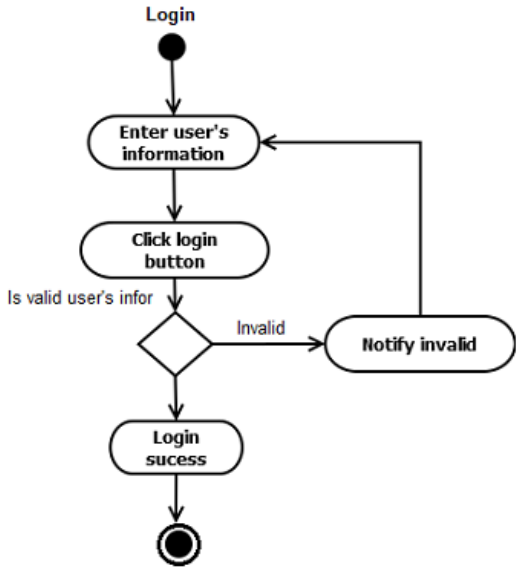


Figure 2.31. Activity login

2.3.2.2. Create product function

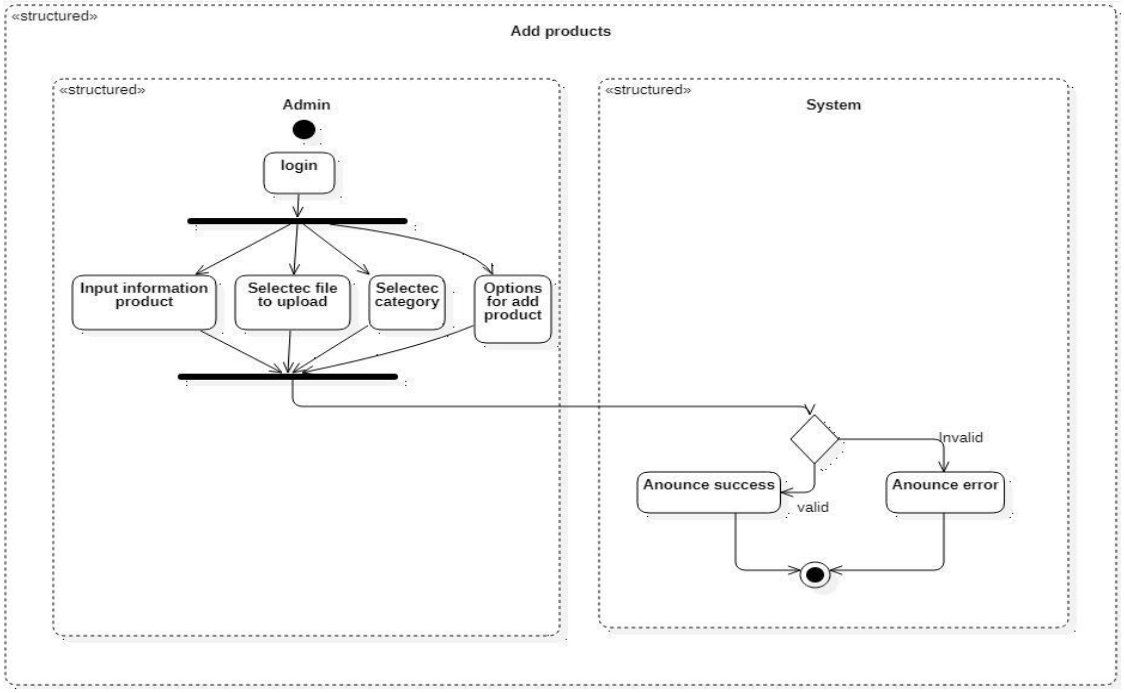


Figure 2.32. Activity create new product

CHAPTER 3: IMPLEMENTATION RESULTS

3.1. Requires Server Configuration

3.1.1. Requires Server

To deploy and put website into stable operation, the server needs to meet the following requirements:

- Python 3.8.6
- Jupyter notebook 6.0.3
- Scikit-learn 0.22.1
- Reactjs 16.10.2
- Redux 7.1.1
- React-router 7.1.1
- Node js 10.16
- MongoDB 4.2.0

3.1.2. Installation

3.1.2.1. Language

Python 3.8.6 is used in my thesis. Python is a widely used high-level programming language for general-purpose programming, created by Guido van Rossum and first released in 1991. An interpreted language, Python has a design philosophy that emphasizes code readability (notably using whitespace indentation to delimit code blocks rather than curly brackets or keywords), and a syntax that allows programmers to express concepts in fewer lines of code than might be used in languages such as C++ or Java. It provides constructs that enable clear programming on both small and large scales

3.1.2.2. Code editor Installation

In machine learning , I use Jupyter Notebook . The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

About website part, I used Visual Studio Code (VS Code) to develop both the Frontend and the Backend.

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps.

Visual Studio supports 36 different programming languages and allows the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language-specific service exists

3.2. Demonstration

The real-estate website works in a variety of ways, such as view properties, providing information, and predict the values of your properties

3.2.1. Interface of the user

3.2.1.1. Home page:

The main screen is displayed when the user accesses the website (as shown in Figure 3.1).

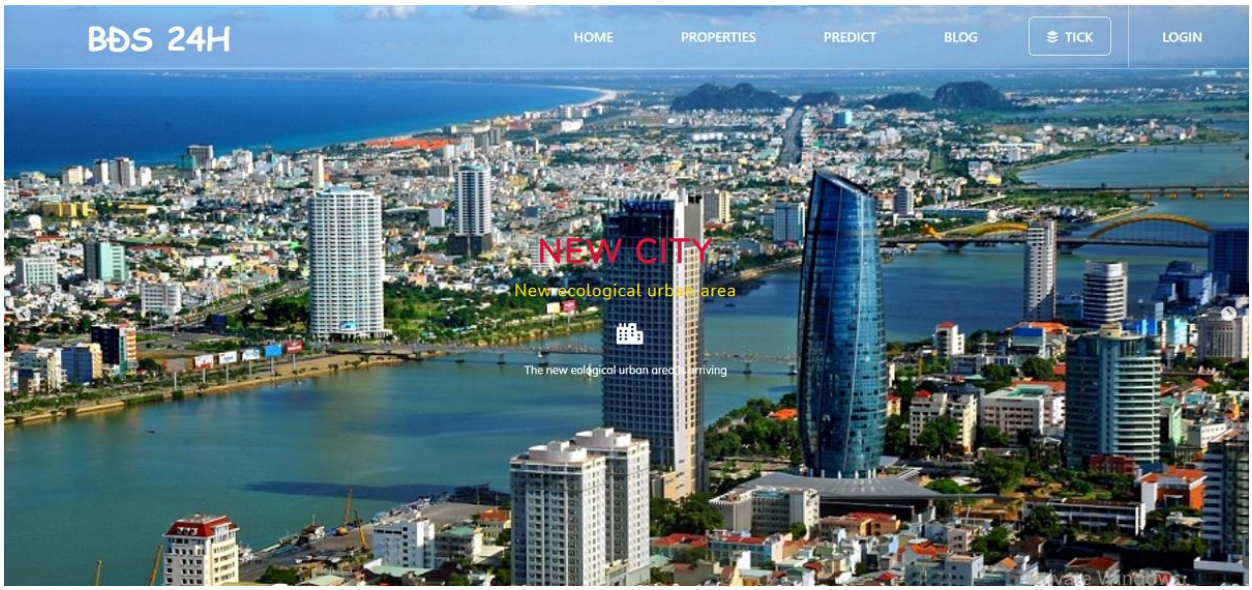


Figure 3.1. Product introduction

In addition, the main screen also introduces a number of brands with trusted names (as shown in Figure 3.2).

Figure 3.2. Brand introduction

New locations and information are displayed in the main screen (as shown in Figure 3.3)

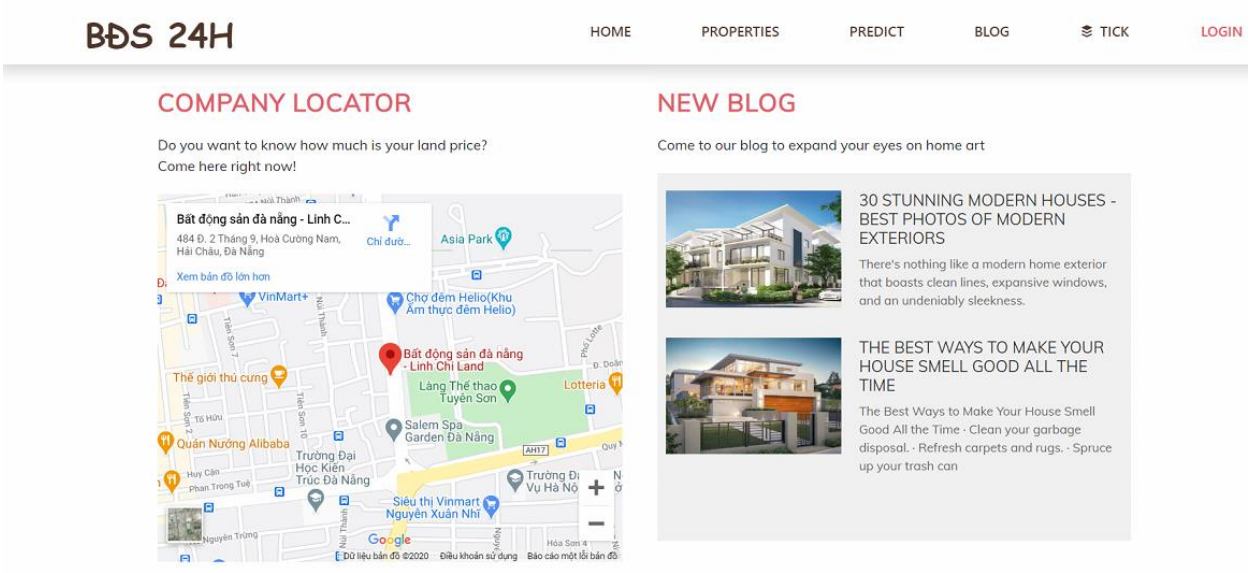


Figure 3.3. Locations and new information

3.2.1.2. Sign up page:

Account registration page (as shown in Figure 3.4).

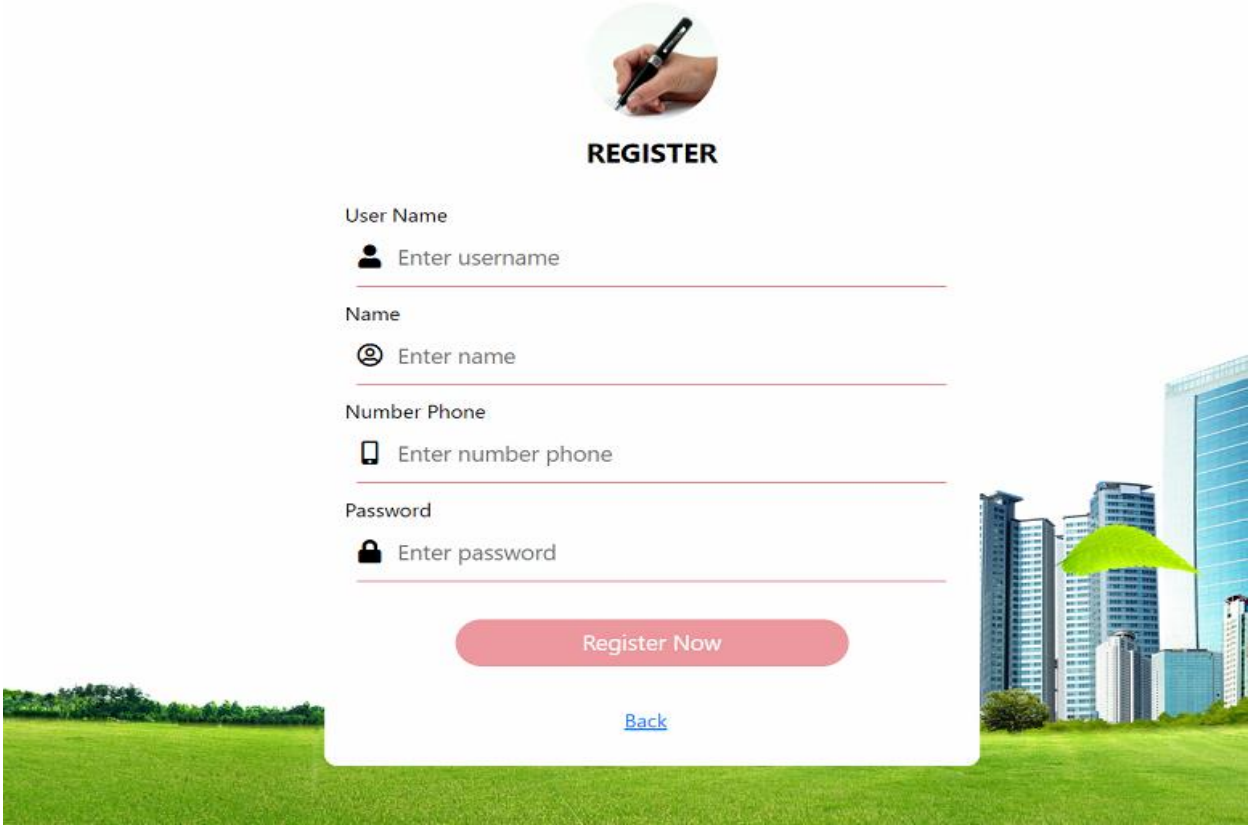


Figure 3.4. Register interface

3.2.1.3. Sign in page

Users enter the correct account information and password to log into the system (as shown in Figure 3.5).

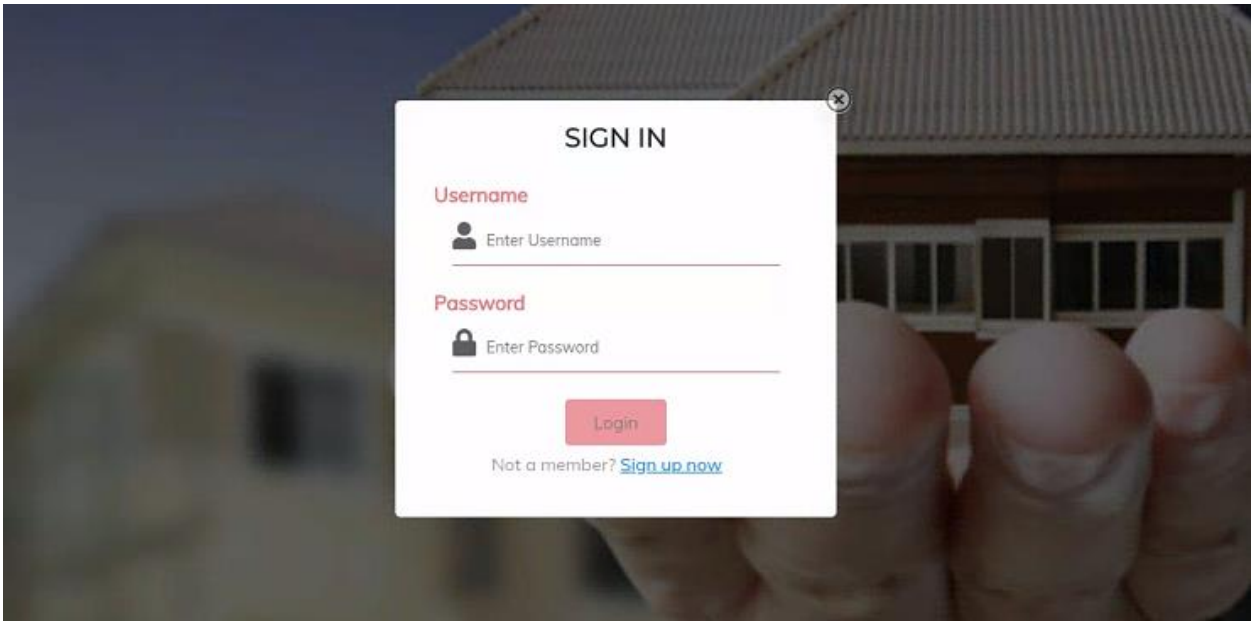


Figure 3.5. Login interface

3.2.1.4. Product page

At the initial product screen will display all products by category as slides (as shown in Figure 3.6).

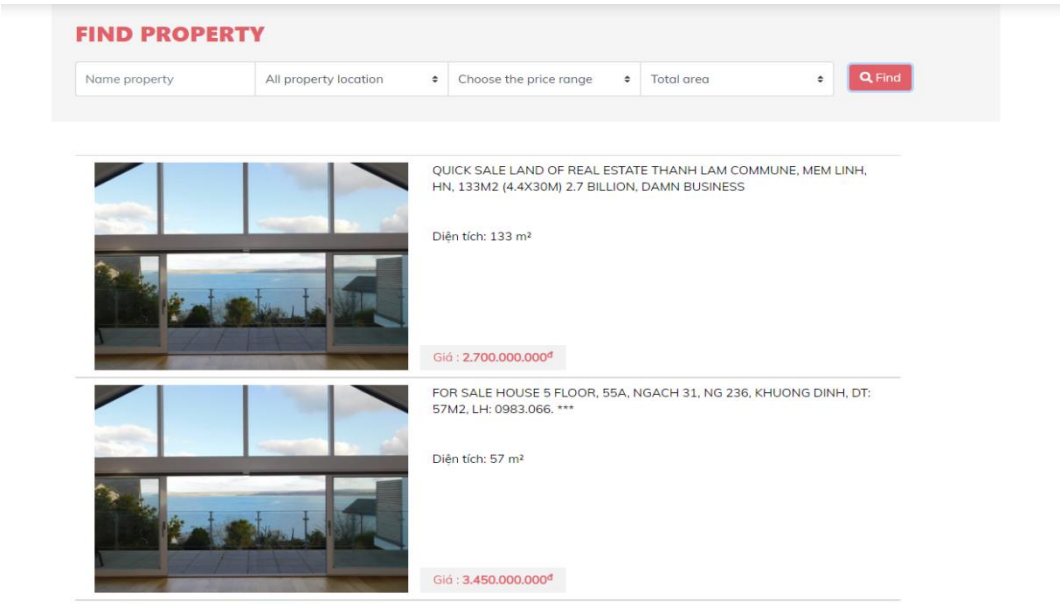


Figure 3.6. Show all products

3.2.1.5. Product by price

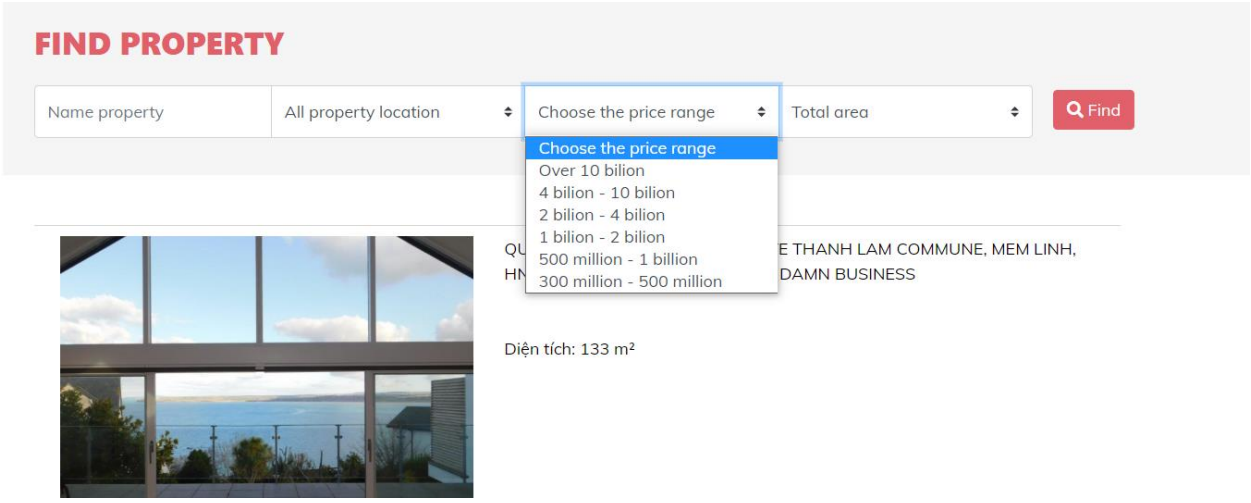


Figure 3.7 Filter product

Display the product list according to the category that the user chooses (as shown in Figure 3.7).

3.2.1.6. Product detail page:

Displays details of product information such as price, origin, quantity, product details (as shown in Figure 3.8).

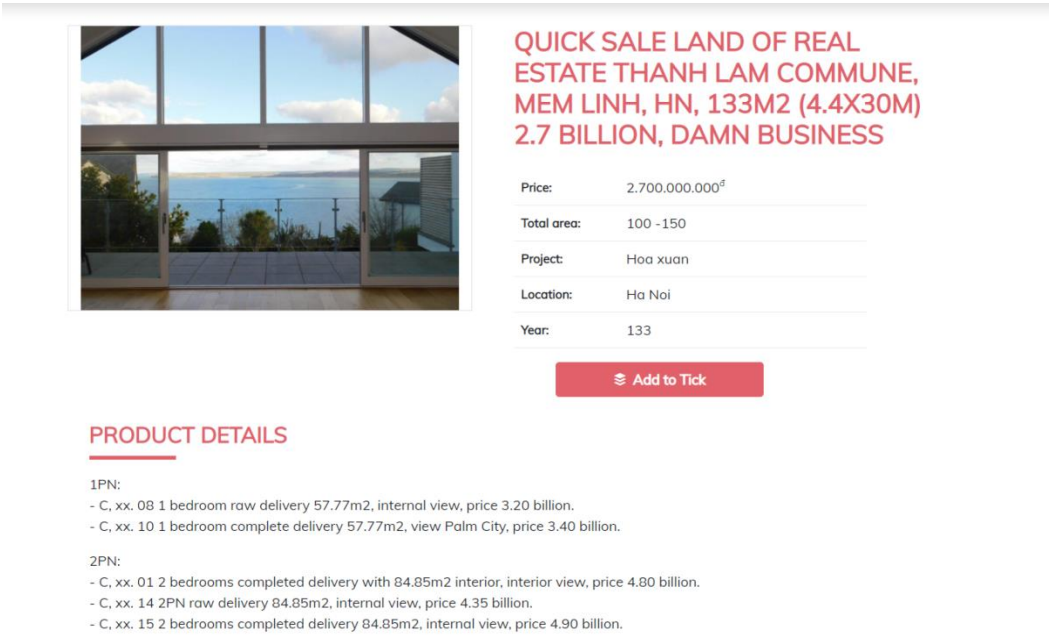


Figure 3.8. The detail information of product

3.2.1.7. Predict page

At the predict cart, users can predict their properties by themselves through informations (as shown in Figure 3.9).

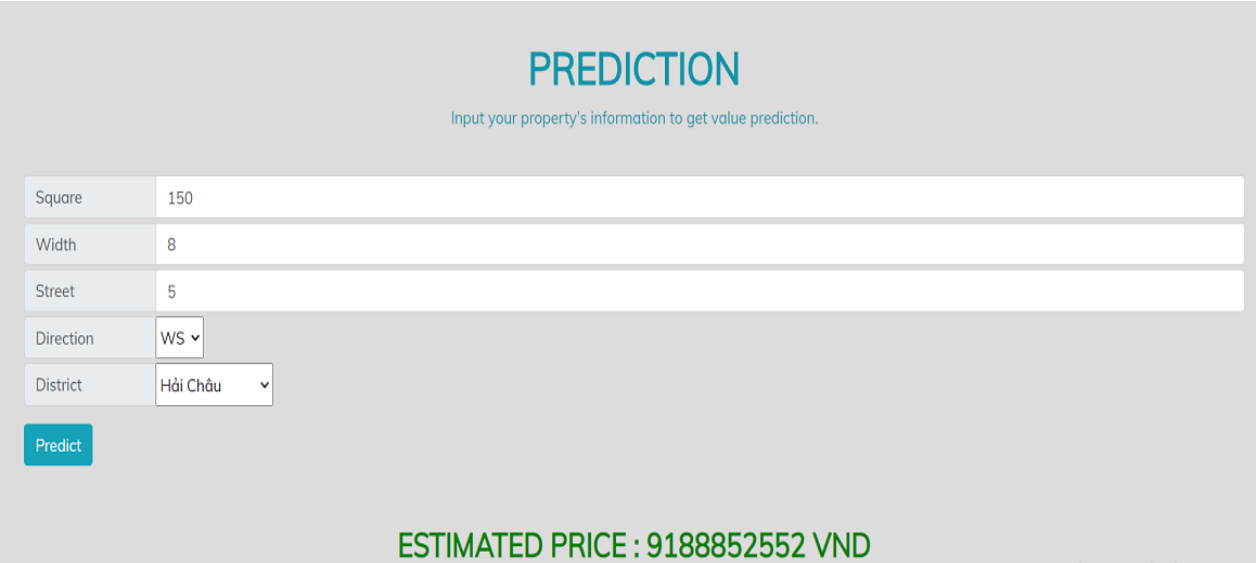


Figure 3.9. Predict page interface

In this page, users import information of their properties such as square, width, street, direction…. and get the price.

3.2.1.8. User information page

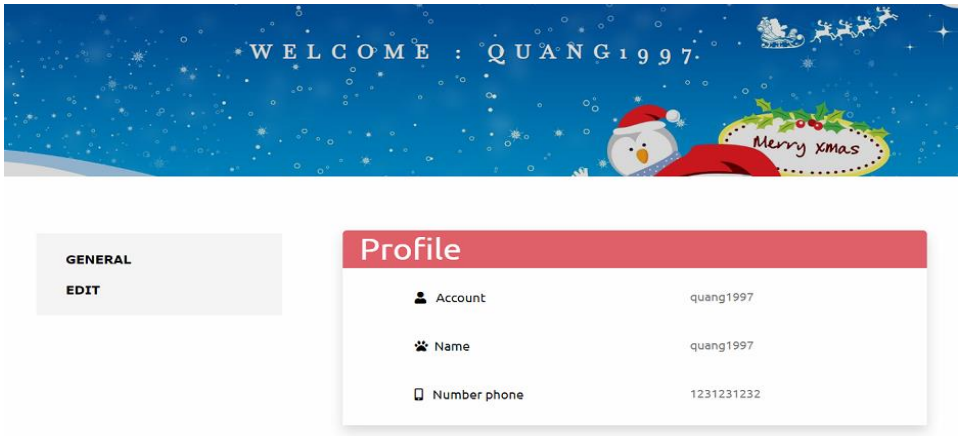


Figure 3.10. Profile information

3.2.1.9. User edit page

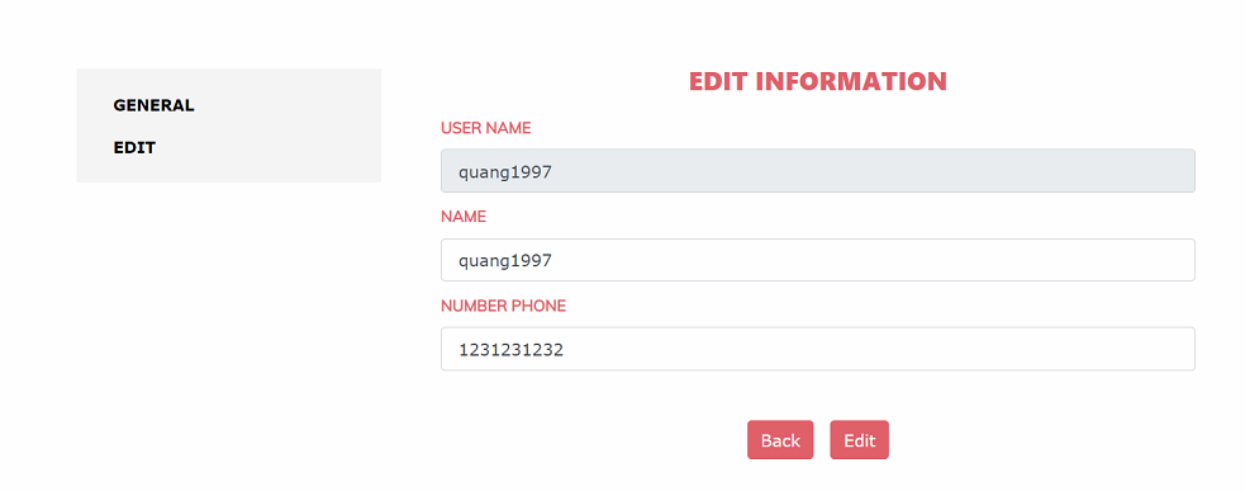


Figure 3.11. Edit user information

3.2.2. Interface of the Admin

3.2.2.1. User management page:

On the User management page, the user's accounts, including the account, name, and level of the account , are displayed as shown in Figure 3.12.

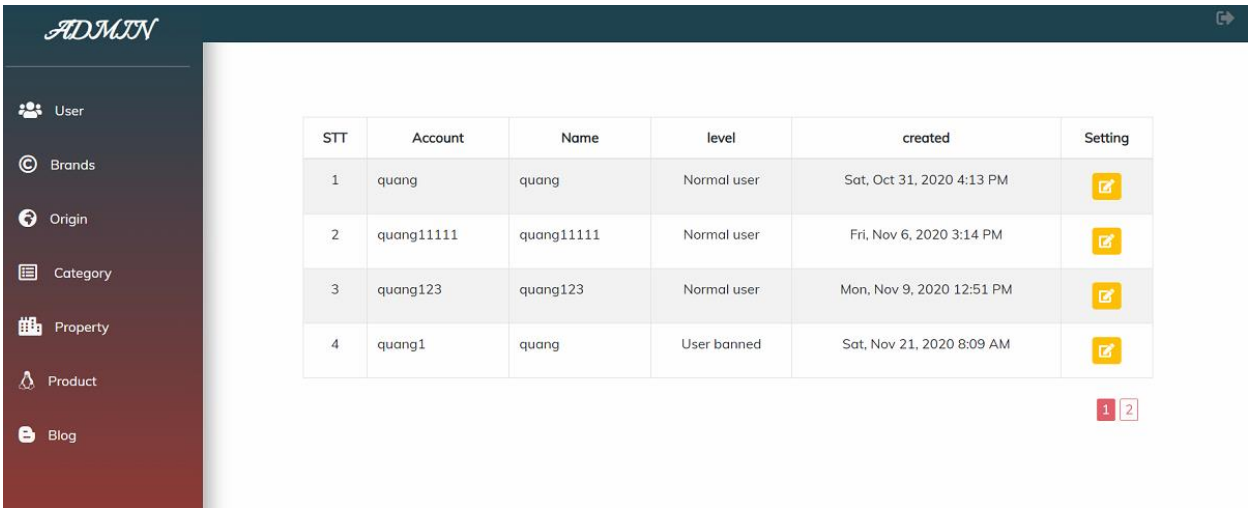


Figure 3.12. The interface for list of users

3.2.2.2. Permission for users:

Admin can assign users different roles such as Mod or Poster as shown in Figure 3.13.

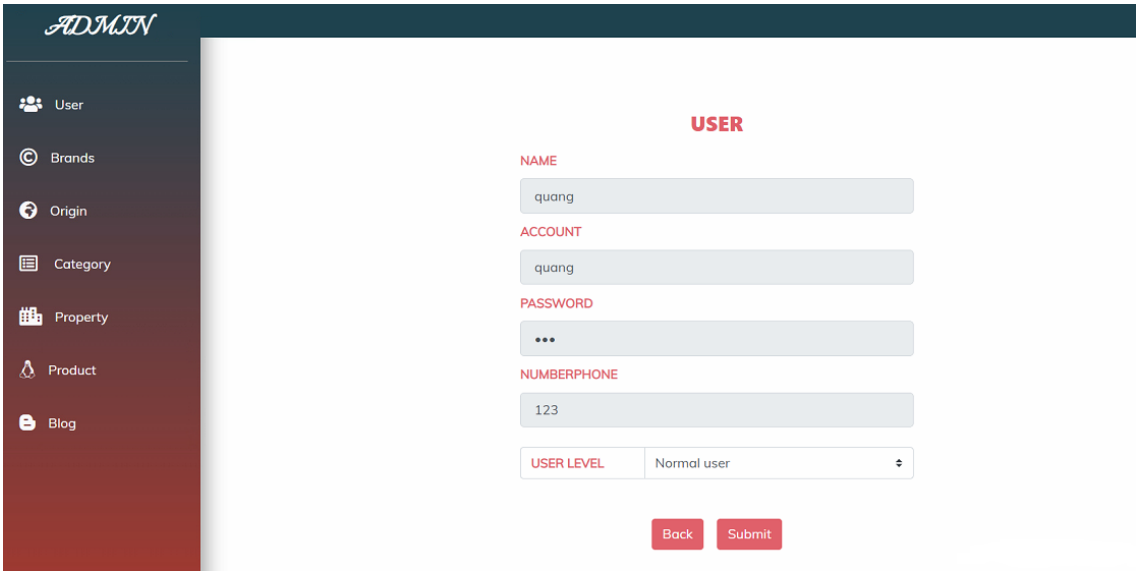


Figure 3.13. The interface for updating users

3.2.2.3. Product management page:

On the Product management page, all information of all products are displayed as shown in Figure 3.14. Besides, the manager can perform additional functions, edit, delete for products on the list.

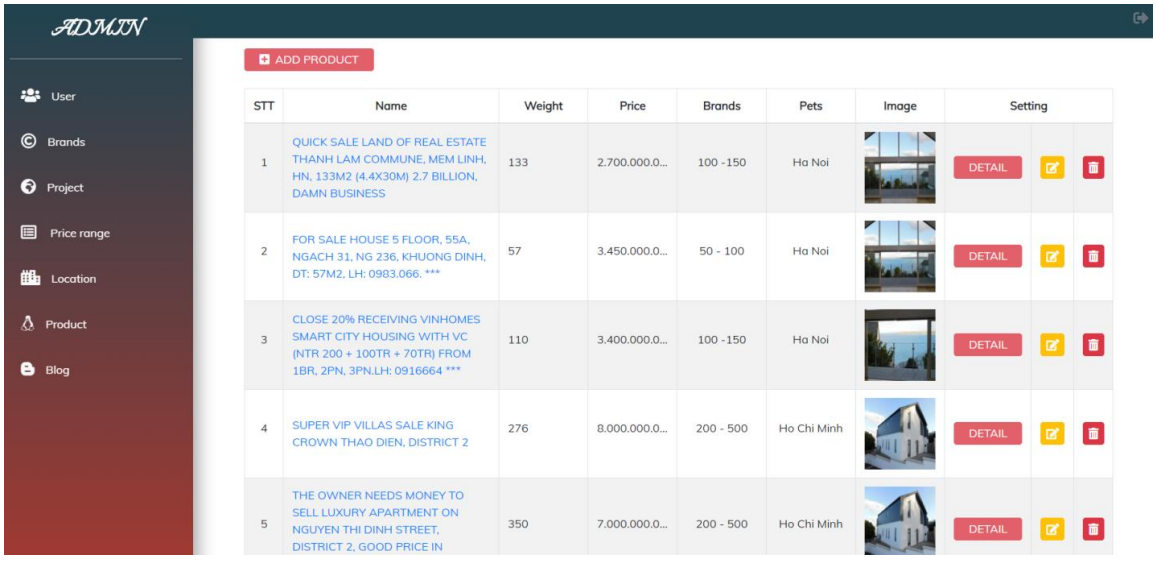


Figure 3.14. The list of products

3.2.2.4. Create Product page

The interface shown in Figure 3.15 description of adding new products to the product list.

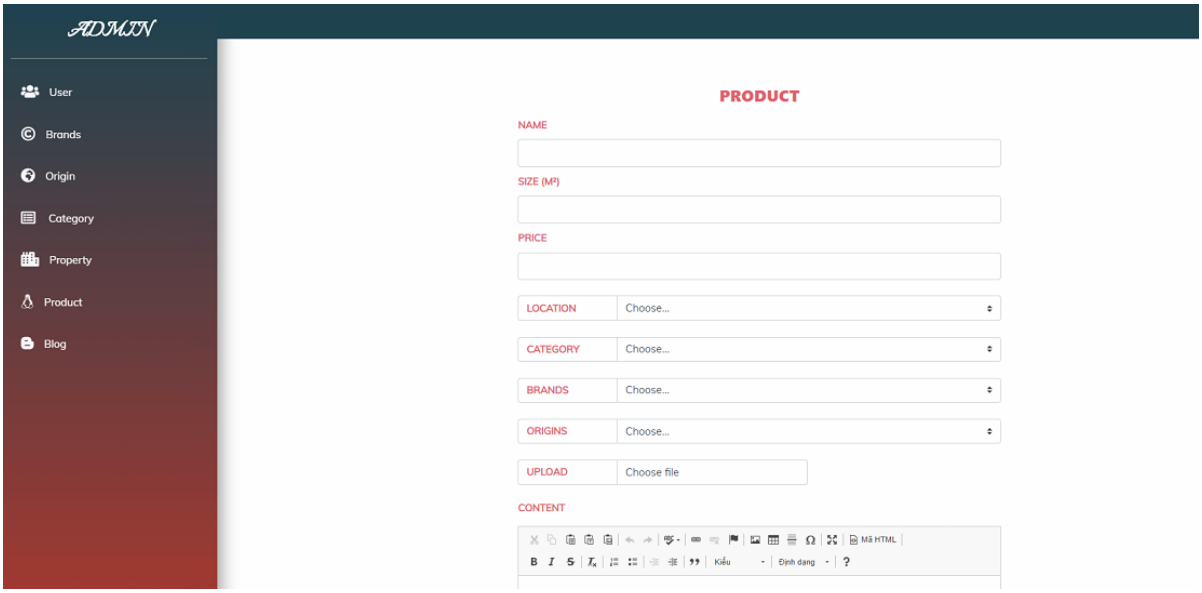


Figure 3.15. The interface for adding new products

3.2.2.5. Edit Product page

On the Edit Product page the manager can edit the product information with editing the input cell values (as shown in Figure 3.16)

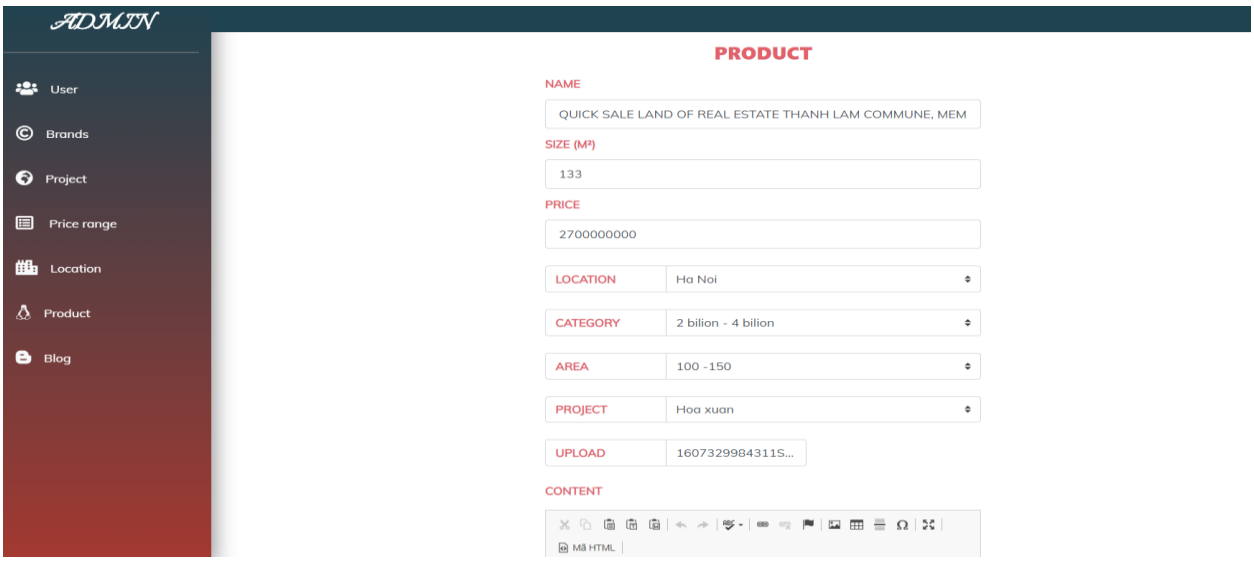


Figure 3.16 The interface for editing product

3.2.2.6. Blog management page

On the Blog management page, all information of all blog are displayed as shown in Figure 3.17.

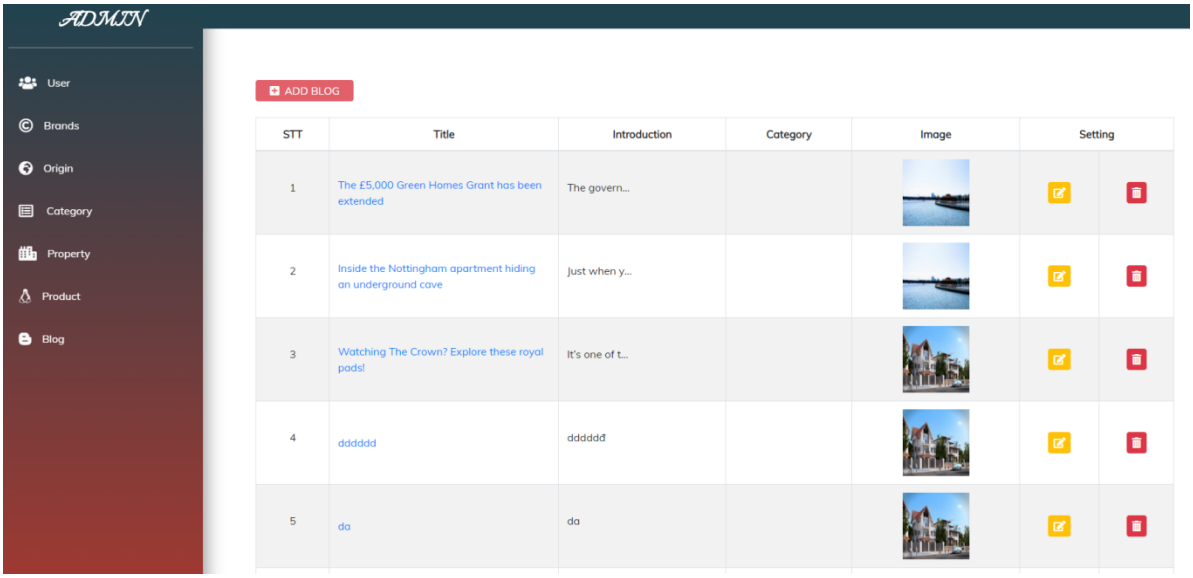


Figure 3.17. The interface for show all blog

3.2.2.7. Create Blog page

The interface shown in Figure 3.18 description of adding new blogs to the blog list.

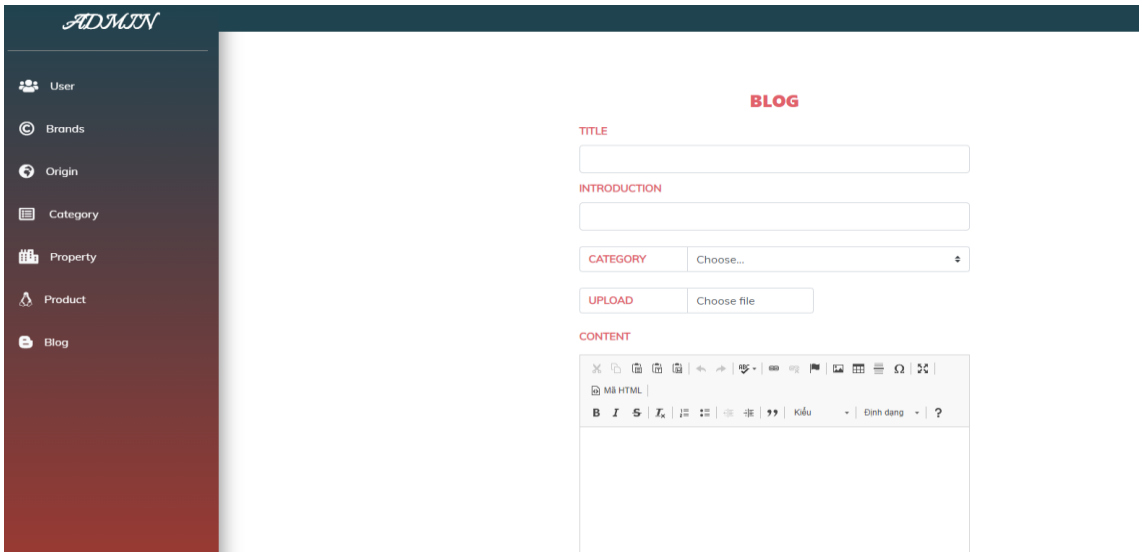


Figure 3.18. The interface for adding new blogs

3.2.2.8. Edit Blog page

On the Edit Product page the manager can edit the product information with editing the input cell values (as shown in Figure 3.19).

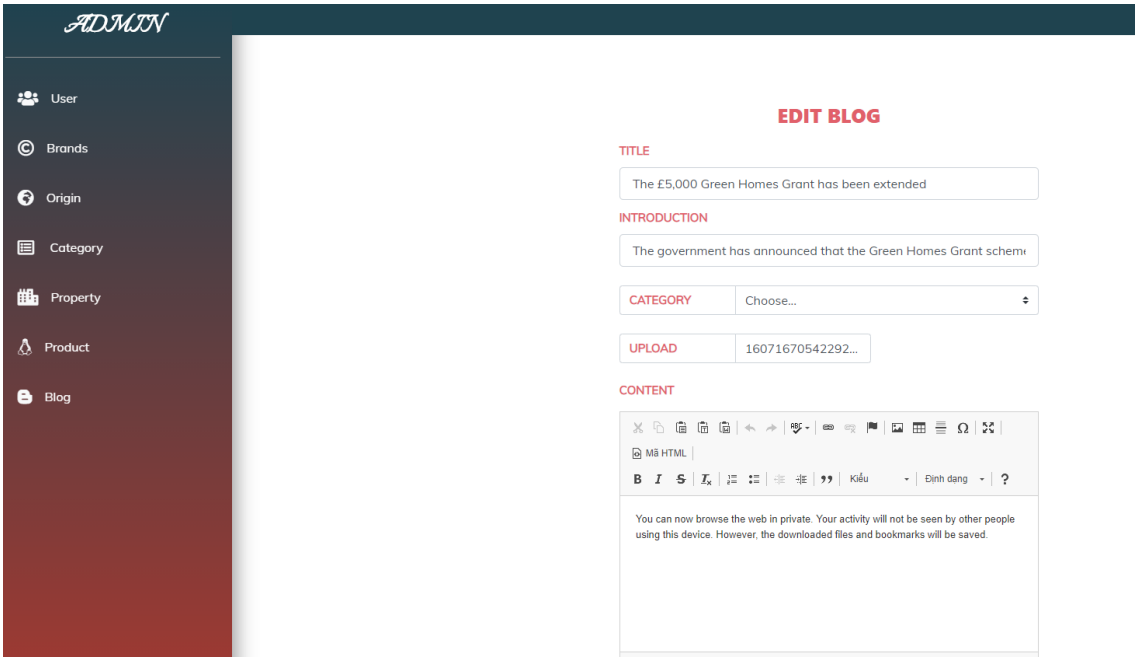


Figure 3.19. The interface for editing blog

CONCLUSION AND FUTURE WORK

1. Achieve results

In this project, I have learned about machine learning algorithms, JavaScript language and understood how to work with Jupyter Notebook, Scikit-learn, ReactJS, Nodejs and MongoDB.

I used:

- ReactJS to build the application.
- Scikit-learn to build the ML model.
- MongoDB to store database.

I have also improved all of my skills including: technical, researching and presentation since I started building this project. With them, I have built the graduation project that meets the initial requirements and have these following features:

- The system supports the user to find suitable properties.
- The system also provides users price prediction of properties
- The system supports the user to find the needed blog.

2. Future works

In the future, I will keep on developing more features such as:

- Support users to have a forum to disscuss about real-estate market.

Update more information and new properties to meet the needs of everyone

REFERENCES

[1]. https://reactjs.org/ (Data access: 23/10/2020)
[2]. https://redux.js.org/ (Data access: 23/10/2020)
[3]. https://scikit-learn.org/ (Data access: 03/11/2020)
[4].https://www.medium.com/ (Data access: 05/10/2020)
[5].https://www.credencys.com/blog/unveil-key-features-of-react-js-for-web-mobile-app-development/ (Data access: 15/10/2020)
[6] https://pandas.pydata.org (Data access: 05/10/2020)
[7] https://www.tutorialspoint.com/reactjs/reactjs_component_life_cycle.htm (Data access: 15/09/2020)