South China Polytechnic university-HSBC financial science and technology innovation hackathons competition

Project Technical Documentation

Team: grassroots moving brick party

Instructor: Xu Yong

Members: Chen Ganghong, Chen Guibo, Huang Shijie, Liu boan, Peng Jiajun

2019.4.21

Catalog

[1.Abstraction 4](#_Toc6771786)

[2.Demand analysis 4](#_Toc6771787)

[2.1 Summary of Demand 4](#_Toc6771788)

[2.2 Use case design 4](#_Toc6771789)

[2.2.1 Register 4](#_Toc6771790)

[2.2.2 Login 5](#_Toc6771791)

[2.2.3 Business screening 6](#_Toc6771792)

[2.2.4 Personalized recommendation 7](#_Toc6771793)

[3.Profile design 8](#_Toc6771794)

[3.1 System overview 8](#_Toc6771795)

[3.2 System structure design 9](#_Toc6771796)

[3.2.1 System physical framework 9](#_Toc6771797)

[3.2.2 System logical framework 10](#_Toc6771798)

[3.2.2 System conceptual data model (E-R diagram) 11](#_Toc6771799)

[3.2.4 System interface design 12](#_Toc6771800)

[3.2.5 System non-functional design 12](#_Toc6771801)

[3.2.6 Glossary 13](#_Toc6771802)

[4.Detailed design 13](#_Toc6771803)

[4.1 System design objectives 13](#_Toc6771804)

[4.2 Design overview 14](#_Toc6771805)

[4.2.1 System reuse plan 14](#_Toc6771806)

[4.2.2 System interface design 14](#_Toc6771807)

[4.2.3 Object model design 14](#_Toc6771808)

[4.2.4 Database design 15](#_Toc6771809)

[4.2.5 Front-end design 20](#_Toc6771810)

[5.System development plan 22](#_Toc6771811)

[5.1 System functional development 22](#_Toc6771812)

[5.1.1 recommended hot search list 22](#_Toc6771813)

[5.1.2 Fuzzy search 22](#_Toc6771814)

[5.1.3 Online financial management course 22](#_Toc6771815)

[5.1.4 Professional investment guidance (accurate personalized recommendation) 22](#_Toc6771816)

[5.1.5 Investment and finance community 23](#_Toc6771817)

[5.2 System performance development 23](#_Toc6771818)

[5.2.1 Server performance improvement 23](#_Toc6771819)

[5.2.2 Database backup processing 24](#_Toc6771820)

# Abstraction

This document is the description document of the project, which mainly explains the technical route of our platform from the three parts of demand analysis, profile design and detailed design of the software engineering development process. The last part of the system development plan is to plan the future development of the platform.

# Demand analysis

## 2.1 Summary of Demand

This document provides an overview of the purpose and scope of the project. This part is mainly used to determine project function points and define project scope. This section can be used to introduce project function points to business people, to identify WBS, project scope management documents, and so on.

## 2.2 Use case design

### 2.2.1 Register

2.2.1.1 Use case description:

The server automatically assigns an account to the user by entering basic personal information, setting a nickname and password.

2.2.1.2 Preconditions:

No.

2.2.1.3 Postcondition:

The user successfully registered an account.

2.2.1.4 Use case event flow:

2.2.1.4.1 Basic event flow

1. Click "register" to enter the registration interface.

2. Users set their own nicknames.

3. The user enters his/her password and enters it again to check whether it is correct.

4. The system automatically assigns accounts and completes the registration process.

2.2.1.4.2 Alternative event flow

1.0 Users cannot enter the registration screen.

2.0 User-set nickname length does not conform to the specification.

3.0 User input password is too long.

3.1 The two passwords entered by the user are inconsistent.

2.2.1.5 Business data related to use cases

a. User account

b. User password

c. User nicknames

### 2.2.2 Login

2.2.2.1 Use case description:

Users log in by entering their personal account and password.

* + - 1. Preconditions:

User has registered an account.

* + - 1. Postconditions:

User successfully logged in and entered the main page.

* + - 1. Use case event flow:

2.2.2.4.1 Basic event flow

1. Click login to enter the login interface.

2. Users input their accounts.

3. Enter the user's password and click login.

4. The user successfully logged in.

2.2.2.4.2 Alternative event flow

1.0 users cannot enter the login screen.

2.0 user entered account does not exist.

2.1 the password entered by the user does not match the account.

2.2.2.5 Business data related to use cases

a. User account

b. User password

### 2.2.3 Business screening

2.2.3.1 Use case description:

Users select different businesses, enter the business they need to search, screening.

2.2.3.2 Preconditions:

The user has logged in.

2.2.3.3 Postcondition:

The user successfully obtains the required business information.

2.2.3.4 Use case event flow:

2.2.3.4.1 Basic event flow

1. User access screening function.

2. Users input keywords of the business they need to screen.

3. The user clicks the screen button.

4. The system will automatically input the screened information.

2.2.3.4.2 alternative event flow

2.0 Keywords entered by users do not conform to the specification.

4.0 System does not capture this information.

2.2.3.5 Business data related to use cases

a. Keywords entered by users.

b. Acquired business information.

### 2.2.4 Personalized recommendation

2.2.4.1 Use case description:

The system identifies the search history of the user by the search record in the next user login.

2.2.4.2 Preconditions:

The user has completed a business screening.

2.2.4.3 Postcondition:

The system successfully sent the recommendation information to the customer

2.2.4.4 Use case event flow:

2.2.4.4.1 Basic event flow

1. The user logs into the system.

2. The system automatically identifies users' historical data and automatically identifies recommendation information.

3. The system automatically sends recommendation information.

2.2.4.4.2 Alternative event flow

1.0 Users cannot access the system

3.0 System does not provide recommendation information.

2.2.4.5 Business data related to use cases

a. User history

b. System recommendation information.

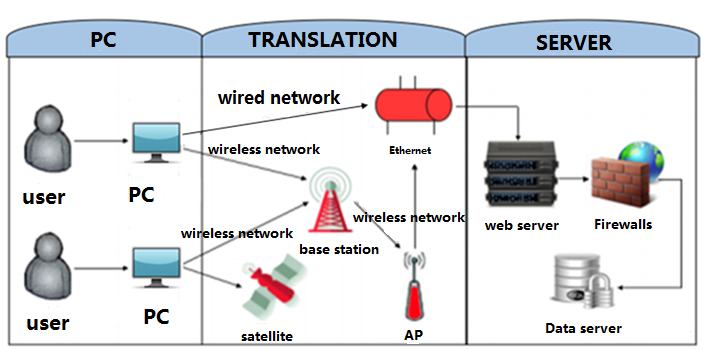
# Profile design

## 3.1 System overview

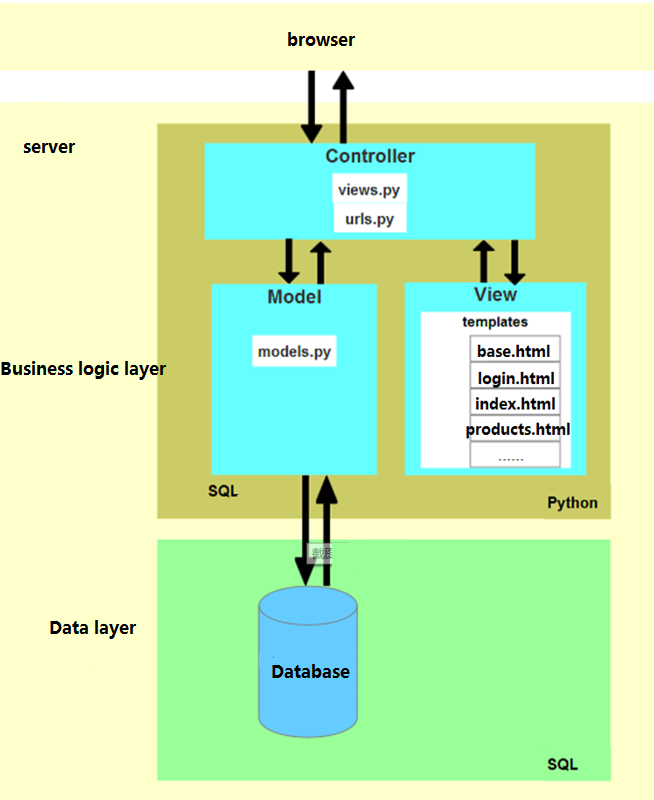
This system is aimed at Internet users. Users can enter our platform and screen financial products on our platform. The platform can display all relevant financial products, and customers can select suitable financial products. Firstly, we build the framework of "e-fi" one-stop financial product service platform, which is realized through the database server. After that, the customer enters the platform for account registration, and can screen financial products on our platform. The platform will automatically give the information of such financial products. In addition, when the customer logs in for the second time, the platform server will filter the history of the customer and give us the recommended product information.

## System structure design

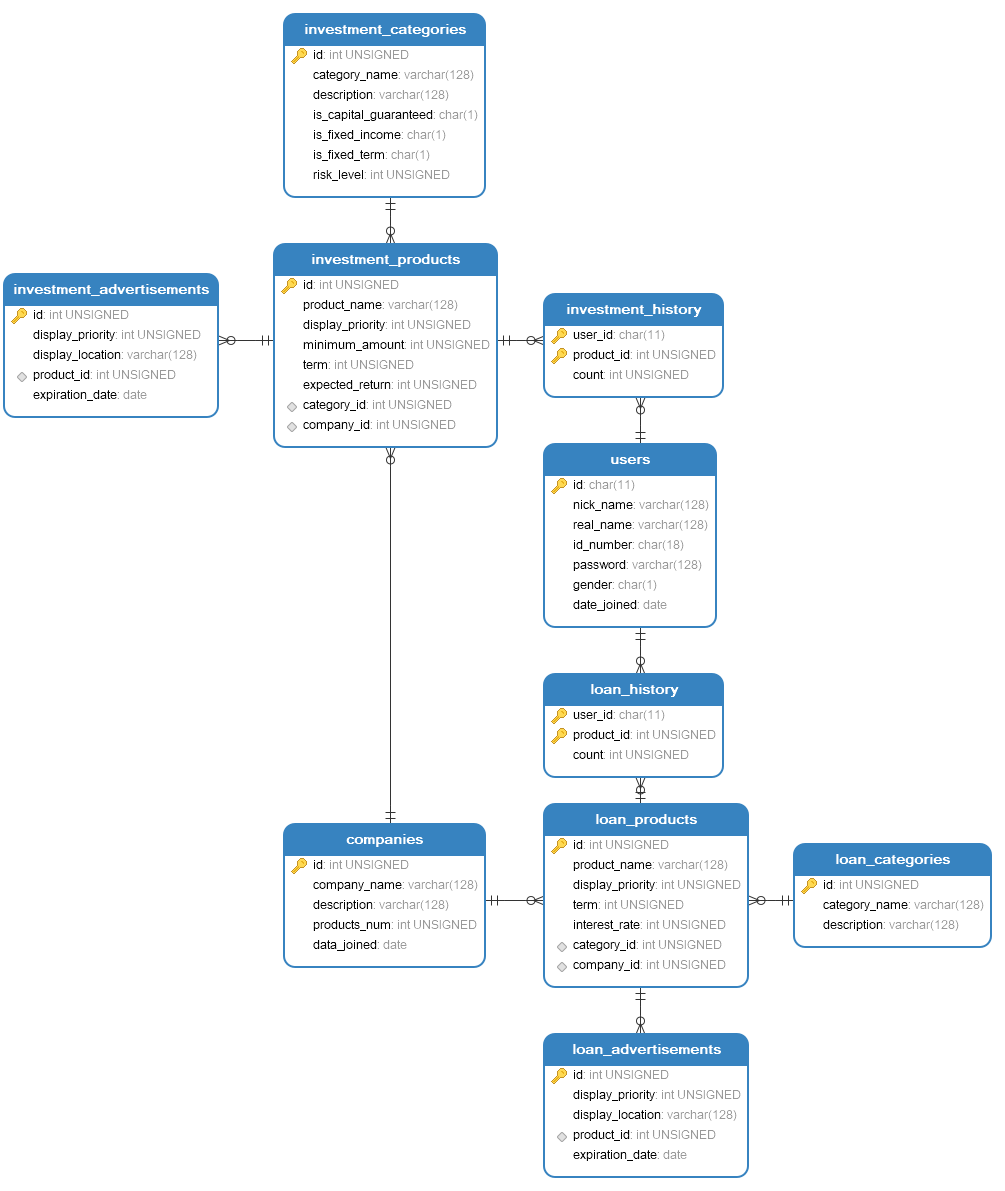
### 3.2.1 System physical framework



### 3.2.2 System logical framework



### System conceptual data model (E-R diagram)



### 3.2.4 System interface design

Man-machine interface :

(1) The system provides registration interface to project participants.

(2) The system provides login interface to project participants.

Internal interface:

Python MySQL database connection API(MySQLdb), django MySQL extension API to connect to the database.

External interface:

Not yet covered.

### 3.2.5 System non-functional design

(1) Visual requirements: the project management system interface style to be concise, and the site should look beautiful enough. Interface requirements also include the specification of the control and the scope of use of the control for a specified content. Consider using a prototype.

(2) Usability requirement: users do not have to wait too long for sending and receiving messages.

(3) Execution requirements: ensure that the system does not crash when users use the software.

(4) Operation and environmental requirements: ordinary computer.

(5) Maintainability requirements: the system should be easy to maintain and should be repaired as soon as possible when the server crashes.

(6) Security requirements: the system should achieve three features of confidentiality, reliability and integrity. The data cannot be accessed by anyone other than the authorized user, and the authorized user can access the data unimpeded. In addition, the system should be compatible with other software.

### 3.2.6 Glossary

|  |  |
| --- | --- |
| Term | Explaination |
| Nickname | The name of user’s account set by user on the platform. |
| Filter | The type of financial products selected by users for classification. |
| History | After the user logs in and searches on the platform, the search records are saved in the background database. |
| Personalized recommendation | Background server recommends financial products to users after analyzing users' historical search records. |
| Hot search list | After analyzing all users' historical search records, the background database analyzes the screening results of financial products with the highest search rate. |

# Detailed design

## 4.1 System design objectives

(1) Ensure chat software can run normally, as little as possible crash phenomenon.

(2) Secure user accounts.

(3) The web page function basic realization, and guarantees the response speed.

(4) Security: ensure that the server is not easy to receive attacks, site failure can be quickly fault processing and rapid recovery.

## 4.2 Design overview

### 4.2.1 System reuse plan

Framework reuse: B/S distributed architecture is suitable for many occasions.

### System interface design

Front-end user interface design:

homepage interface, registration page, landing page, index page, etc.

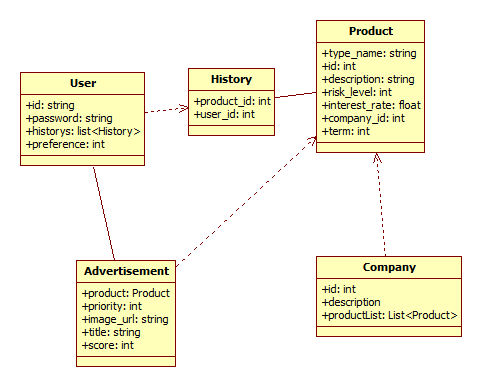
System internal interface design:

python MySQL database connection API(MySQLdb), django MySQL extension API to connect to the database.

System external interface design:

none.

### Object model design

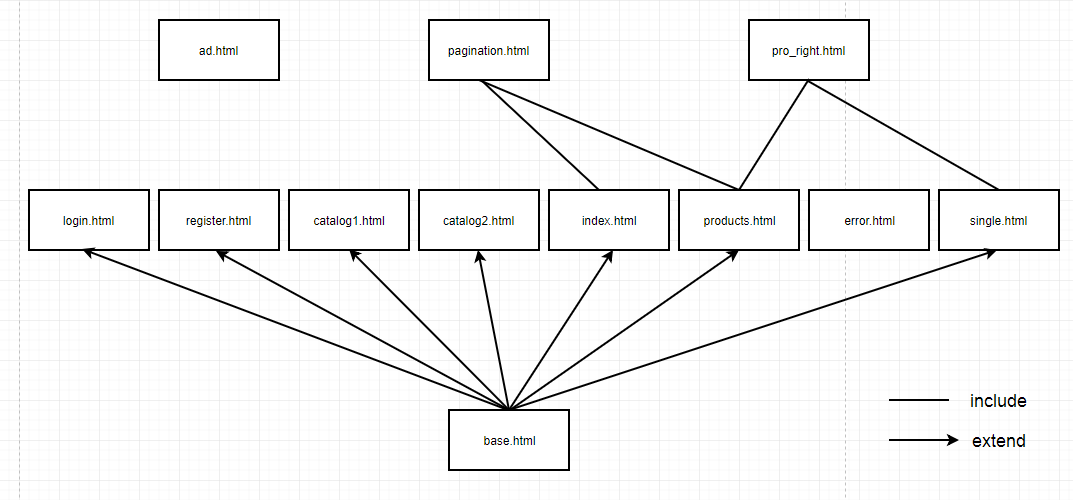


### Database design

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Data Name | Data Business Description | Business Data Type | Table | Corresponding Field | Field Type |
| user id | user’s unique id, mobile number | character string, with 11 bits | users | id | char(11) |
| user passrod | user password to login | strings of letters and numbers, with 8 to 16 bits | users | password | varchar(128) |
| user nickname | user nickname | strings less than 32 bits in length | users | nick\_name | varchar(128) |
| user name | user real name | strings less than 32 bits in length | users | read\_name | varchar(128) |
| id number | user’s id number | character string, with 18 bits | users | id\_number | char(18) |
| user gender | user gender | one character, “B” for boy while “G” for girl | users | gender | char(1) |
| user register date | user register date | date type | users | date\_joined | date |
| company ID | company’s unique id | nonnegative integer, with 5 to 10 bits | companies | id | unsigned int |
| company name | company name | strings less than 128 bits in length | companies | company\_name | varchar(128) |
| company description | company description | strings less than 128 bits in length | companies | description | varchar(256) |
| Product quantity | Number of products released by the company on the platform | nonnegative integer | companies | products\_num | unsigned int |
| company register date | date of company joining the platform | date type | companies | date\_joined | date |
| investment products type id | investment products type’s unique id | nonnegative integer | investment\_categories | id | unsigned int |
| investment products type name | investment products type name | strings less than 128 bits in length | investment\_categories | category\_name | varchar(128) |
| investment products type description | investment products type description | strings less than 128 bits in length | investment\_categories | description | varchar(128) |
| is fixed income | explain whether the product has a fixed income | one character, “T” for true while “F” for flase | investment\_categories | is\_fixed\_income | char(1) |
| is fixed term | explain whether the product has a fixed term | one character, “T” for true while “F” for flase | investment\_categories | is\_fixed\_term | char(1) |
| is capital guaranteed | explain whether the product is capital guaranteed | one character, “T” for true while “F” for flase | investment\_categories | is\_capital\_guaranteed | char(1) |
| risk level | Risk Rating of Investment Products | nonnegative integer, with value of 1 to 10 Tentatively | investment\_categories | risk\_level | unsigned int |
| investment product id | investment product’s unique id | nonnegative integer, with 5 to 10 bits | investment\_products | id | unsigned int |
| investment product name | investment product name | strings less than 128 bits in length | investment\_products | product\_name | varchar(128) |
| display prority of investment product | priority of display of investment products when searching by category and keyword | nonnegative integer, with value of 1 to 10 Tentatively | investment\_products | display\_priority | unsigned int |
| Minimum investment | minimum investment amount | nonnegative integer | investment\_products | minimum\_amount | unsigned int |
| investment term | investment term, with day as unit | nonnegative integer | investment\_products | term | unsigned int |
| expected return | Annual interest rate, with percentage as unit | nonnegative integer | investment\_products | expected\_return | unsigned int |
| type id | identify the type of investment product | nonnegative integer | investment\_products | category\_id | unsigned int |
| company | identify the company of investment product | nonnegative integer | investment\_products | company\_id | unsigned int |
| loan product type id | loan product type’s unique id | nonnegative integer | loan\_categories | id | unsigned int |
| loan product type name | loan product type name | strings less than 128 bits in length | loan\_categories | category\_name | varchar(128) |
| loan product type description | loan product type description | strings less than 128 bits in length | loan\_categories | description | varchar(128) |
| loan product id | loan product’s unique id | nonnegative integer | loan\_products | id | unsigned int |
| loan product name | loan product name | strings less than 128 bits in length | loan\_products | product\_name | varchar(128) |
| display prority of loan product | priority of display of loan products when searching by category and keyword | nonnegative integer, with value of 1 to 10 Tentatively | loan\_products | display\_priority | unsigned int |
| Repayment period | Repayment period, with day as unit | nonnegative integer | loan\_products | term | unsigned int |
| loan interest rate | loan interest rate, with percentage as unit | nonnegative integer | loan\_products | interest\_rate | unsigned int |
| advertisement id | advertisement’s id | nonnegative integer | investment\_advertisements，loan\_advertisement | id | unsigned int |
| display prority of advertisement | display prority of advertisement | nonnegative integer, with value of 1 to 10 Tentatively | investment\_advertisements，loan\_advertisement | display\_priority | unsigned int |
| display location of advertisement | display location of advertisement, such as main, sidebar | strings less than 128 bits in length | investment\_advertisements，loan\_advertisement | display\_location | varchar(128) |
| product id of advertisment | identify the product id of advertisment | nonnegative integer | investment\_advertisements，loan\_advertisement | product\_id | unsigned int |
| expiration date of advertisement | expiration date of advertisement | date type | investment\_advertisements，loan\_advertisement | expiration\_date | date |
| user id | identify the user to which the history belongs | character string, with 11 bits | investment\_history，loan\_history | user\_id | varchar(16) |
| product id | identify the product to which the history belongs | nonnegative integer | investment\_history，loan\_history | product\_id | unsigned int |
| browse times | record the number of times users browse the same product | nonnegative integer | investment\_history，loan\_history | count | unsigned int |

### 4.2.5 Front-end design

Front-end framework:



Front-end component functions:

|  |  |  |
| --- | --- | --- |
| Page’s name | Explaination | Whether consolidation |
| ad.htlm | The pages used to display ads will be used to expand the monetization model available in the future. | No |
| base.html | Basic pages. Many front-end pages extend from this base page. | Yes |
| catalog1.html | It is extended from base. html to do the classification of financial management. | Yes |
| catalog2.html | Extends base. html to do the classification of loan classes. | Yes |
| error.html | To handle error messages. | No |
| index.html | Extends to base.html to display product information that the server has processed and recommended to the customer. | Yes |
| login.html | Extends to base.html, which allows users to log in. | Yes |
| pagination.html | Extend to base. html, achieve pagination function. | Yes |
| pro\_right.html | Extends to base. html to place product recommendations for hot search lists. | Yes |
| Products.html | Extends base.html to display a list of products that the user has searched. | Yes |
| register.html | Extension to base. html for user account registration. | Yes |
| single.html | Extends to base.html, where a selected product can be displayed on this page. | Yes |

# System development plan

## 5.1 System functional development

### 5.1.1 recommended hot search list

Through analyzing the historical search records of all customers, the "e-+" platform can calculate the financial products with the largest number of searches by big data algorithm and display them."Hot search list."."This kind of" hot search list "can be achieved after the platform scale has reached a certain level of business," he said.

### 5.1.2 Fuzzy search

At present, due to the limited ability and time of our project team, we can only implement a precise search for keyword clicks, which is certainly not enough.In the future, a search based on keywords can be implemented on the platform, and users can enter keywords for search, and the system can use the mature third party library for fuzzy search of keywords.To achieve more convenient and more appropriate customer search.

### 5.1.3 Online financial management course

This is the platform's medium-term development plan.Because every user who logs on to the platform to register and search for the platform does not necessarily have a certain financial knowledge, and depending on the company background behind the platform, the platform can carry out online financial management courses."We will continue to improve the quality of our financial services," he said.This is also a part of the platform's medium-term profit model.

### 5.1.4 Professional investment guidance (accurate personalized recommendation)

At present, the personalization recommendation of the platform is limited to the historical data of the client, but the search of the client often has certain difficulty and uncertainty.Personalized recommendation does not reflect customer preferences well.After the platform is developed to a certain scale, it can rely on the company background behind the platform and set up a professional investment guidance team on the platform to recommend a one-to-one accurate and personalized investment to prospective customers.This is also a part of the platform's medium-term profit model.

### 5.1.5 Investment and finance community

The customer's profile is one of the criteria for measuring the number of active users on a platform.It is one of the aspects that our platform needs to consider whether we can keep the old customers well so that the active users of the platform can survive as much as possible.Therefore, we can set up an investment and financial community on the platform, on the one hand, so that all users registered on our platform can learn from each other.On the other hand, it is possible to publish some financial advertisements in the community or to popularize some knowledge of financial products to increase the retention of customers.

## System performance development

### 5.2.1 Server performance improvement

Customer's investment and financial needs will change with the degree of cold and hot financial markets.Therefore, we cannot rule out the possibility of a surge in the number of platform customers once the financial market is overheated after our platform has developed to a certain scale.This situation may lead to overloading of the platform server or a slide projector.Therefore, in the development of the platform, how to maintain the server well and improve server performance is a must to consider.On the one hand, the platform can buy servers with higher performance and greater stability, and on the other hand, it can set up standby servers to work in a timely manner in case of overloading or slide projector of conventional servers.Guarantee the normal operation of the platform.

### Database backup processing

In general, there is no possibility of background database content being damaged by human factors in a highly managed technical team.But we cannot rule out data loss due to natural force majeure or the database itself.Therefore, it is necessary to set up multiple off-site databases and synchronize them in time to ensure the consistency of data in the database.This enables the platform to function properly when a database is in trouble or data is lost.