

*Software Engineering*

Smart Robo-Advisor

ARCHITECTURE

**GROUP REPORT**

1801212963 Ye Juelin (叶珏琳)

1801212938 Wang Rusi (王如思)

1801212920 Shi Yunxia (史云霞)

1801212932 Wang Bingquan (王柄荃)

1801212832 Shen Zimin (谌子民)

# CATALOG

[1. Introduction 3](#_Toc3800516)

[1.1 Purpose 3](#_Toc3800517)

[1.2 Scope 3](#_Toc3800518)

[1.3 Definitions, Acronyms, and Abbreviations 4](#_Toc3800519)

[2 Design guidelines 6](#_Toc3800520)

[2.1 Assumptions / Constraints / Standards 6](#_Toc3800521)

[3. Algorithms and Use Cases 11](#_Toc3800522)

[3.1 Algorithms 11](#_Toc3800523)

[3.1.1 Multi-factor Models Strategy 11](#_Toc3800524)

[3.1.2 Category Asset Allocation 17](#_Toc3800525)

[3.1.3 Timing Strategy 18](#_Toc3800526)

[3.1.4 CTA Strategy 18](#_Toc3800527)

[3.1.5 Statistic Arbitrage 20](#_Toc3800528)

[3.1.6 Industry Rotation Strategy 20](#_Toc3800529)

[3.2 Use Cases 21](#_Toc3800530)

[4. Design Overview 21](#_Toc3800531)

[4.1 Introduction 21](#_Toc3800532)

[4.2 Software Architecture 22](#_Toc3800533)

[4.3 System interfaces 24](#_Toc3800534)

[4.4 Performance 25](#_Toc3800535)

[5. System Design 26](#_Toc3800536)

[5.1 Introduction 26](#_Toc3800537)

[5.2 Data and database design 26](#_Toc3800538)

[5.3 Dynamic model 27](#_Toc3800539)

[5.4 Application program interfaces 28](#_Toc3800540)

[5.5 User Interface Design 29](#_Toc3800541)

[5.6 Performance 35](#_Toc3800542)

[5.7 Compliance 36](#_Toc3800543)

[6. Product Design Specification Approval 36](#_Toc3800544)

[APPENDIX A: REFERENCES 37](#_Toc3800545)

[APPENDIX B: GLOSSARY 38](#_Toc3800546)

# 1. Introduction

## 1.1 Purpose

Architectural design is the critical link between design and requirements engineering, as it identifies the main structural components in a system and the relationships between them. The output of the architectural design process is an architectural model that describes how the system is organized as a set of communicating components.

Software architecture is important because it affects the performance, robustness, distributability, and maintainability of a system. We need to design a system organization that will satisfy the functional and non-functional requirements of the system.

## 1.2 Scope

This document will describe the structure and nature of the system's constituent units, the relationships between them, their synthesis methods, and the synthesis constraints. The table 1.2 shows the basic units and the interaction they support.

|  |  |
| --- | --- |
| Units | Interaction |
| Module | Procedure call and data sharing |
| Object | Method call |
| Filter | Data flow |
| Process | Message passing, remote procedure call and communication protocol |
| Data file | Read-write |
| Database | Mode and query language |

**Table 1.2 – Units & Interaction**

Our architectural style is **layered architecture.** It organizes the system into layers with related functionality associated with each layer. This layered approach supports the incremental development of systems, and the architecture is also changeable and portable. It will mainly include three layers as shown in Picture 1.1.

User Interface

Database

Middle Engine

**Picture 1.1 – Layers**

* User Interface: implement interaction with the user.
* Middle Engine: the core of our project, which mainly contains supporting algorithms to realize the functions.
* Database: get data from Wind or other finance data sources through API connectors and do some data cleaning for the use of middle engine system's strategies.

## 1.3 Definitions, Acronyms, and Abbreviations

This part defines the important glossaries and acronyms in the document in Table 1.1, so that everyone who looks at the document can have a basic understanding of each term in the document.

|  |  |
| --- | --- |
| Glossary | Definition |
| Quantitative Trading | Use computer technology to select strategies from a large historical data that can generate excess returns with a variety of ‘high probability’ events. |
| Artificial Intelligence | [Intelligence](https://en.wikipedia.org/wiki/Intelligence) demonstrated by [machines](https://en.wikipedia.org/wiki/Machine). |
| Asset Allocation | Allocate funds in different asset classes based on investment demand. |
| Back-test | Test the strategy with historical data. |
| Multi-factor | Use a series of factors as the stock selection criteria and select stocks with higher scores to build a portfolio. |
| CTA | Timing strategy in the futures market. |
| Statistical Arbitrage | Find out the two assets with similar trend, buy low and sell high. |

**Table 1.1 - Definitions**

# 2 Design guidelines

## 2.1 Assumptions / Constraints / Standards

* The architectural design should be modularized, easy to maintenance and upgrade.
* Use unit design techniques based on operations and data encapsulation.
* Use special mechanisms to reliably address concurrency control and distribution system issues.
* Strictly based on layered architecture and its principle to develop our software.
* Interfaces are important, which makes integration simple and assists testers in functional testing of components.

The software architecture is the skeleton of the system, and the design should start from the architecture and then consider the details.

**(1) Performance standards:**

① Response time:

* In 95% of cases, the general time response time does not exceed 1 second, and during the peak period the response time does not exceed 3 seconds.
* The time required from the click to the next screen should not exceed 300 milliseconds.
* When the network is unblocked, the time required to dial-up to the GPRS network must not exceed 5 seconds.
* In the recommended configuration environment: the login response time is within 2 seconds, the refresh response time of the column is within 2 seconds, the response time of the entry page list is refreshed within 2 seconds, and the response time of the information entry is turned on within 1 second.

② Business volume:

* The system allows 50,000 users to be online at the same time.
* The system can simultaneously satisfy 10,000 user requests and provide browsing capabilities for 25,000 concurrent users.

③ System capacity:

* The database can hold at least 100GB of data.

④ Accuracy:

* The backtracking tracking error rate does not exceed 1%.
* The accuracy of the calculation is 4 digits after the decimal point.

⑤ Resource usage rate:

* CPU usage <=50%.
* Memory usage <=50%.

**(2) Security standards:**

* Strict access control system. After the user is authenticated, he can only access the data within its scope of authority, and can only perform operations within its scope of authority.
* Different users have different identities and rights. They need to provide trusted authorization management services under the premise of authenticity of users, to protect data from illegal/over-authorized access and tampering, and to ensure the confidentiality and integrity of data.
* Provides operational log management and security auditing to track historical usage of the system.
* General can withstand malicious attacks from the Internet. Such as viruses (including Trojans) attacks, password guessing attacks, hacking, etc.
* At least 99% of attacks need to be detected within 10 seconds.

**(3) Reliability standards:**

* There is a prompt for the input, and the data has an inspection mechanism to prevent data anomalies.
* The system is robust and should be able to handle various abnormal situations that occur during system operation, such as human error, illegal data input, hardware device failure, etc. The system should be able to handle it correctly and avoid it properly.
* The probability that the service cannot be completed due to the failure of the software system is less than 5‰.
* The system is required to run 24 hours a day, and the total number of outages for continuous operation throughout the year cannot exceed 10 hours.
* The system defect rate has a maximum of 1 failure every 1,000 hours.
* In 1,000,000 operations, there is a maximum of one time when the system needs to be restarted.

**(4) Compatibility standards:**

* The system should support IOS, Android, Windows operating system;
* The average time to replace the system database is no more than 2 hours and no data loss is guaranteed.

**(5) Data security needs:**

* Network delivery data should be encrypted. It is necessary to ensure that data is not sneaked, stolen or tampered with during collection, transmission and processing. Business data needs to be encrypted at the time of storage to ensure that it is not hackable.

**(6) Ease of use needs:**

* Within 3 months of using the product, 60% of users should be able to use it to complete the portfolio configuration function within 1 minute, and the failure rate is controlled within one ten thousandth.
* 60% of users will realize that this is a smart investment system within 5 seconds of seeing the product for the first time.
* 80% of users receive asset allocation advice within 5 minutes after receiving a 2-hour system introduction training.

**(7) Availability standards:**

* Provides data backup and recovery functions to enable timely recovery and restore of data (provided by hardware and third-party software) when system data is lost due to system errors or other reasons or system data is corrupted.

**(8) Maintainability standards:**

* 90% of the BUG modification time does not exceed 1 working day, and the other does not exceed 2 working days.
* Any method of any object does not allow more than 200 lines of code.
* Installing the new version must keep all database content and all personal settings unchanged.
* The product must provide a tool that can track any database field.

D

# 3. Algorithms and Use Cases

## 3.1 Algorithms

### 3.1.1 Multi-factor Models Strategy

The multi-factor strategy is a widely-used stock picking strategy. The basic idea is to find some indicators that are most relevant to the rate of return, and build a stock portfolio based on the indicator, expecting the portfolio to outperform in the future. The key to a multifactor model is to find the correlation between factors and yields.

**(1) Data Preprocessing**

* Basic Data Acquisition

As the first step in building a model, it is important to ensure the comprehensiveness and rationality of the data used. We first summarize the types of factors of different styles, then subdivide the relevant factors under different styles, and determine the calculation method of factors by synthesizing the economic meaning and relevant parameters.

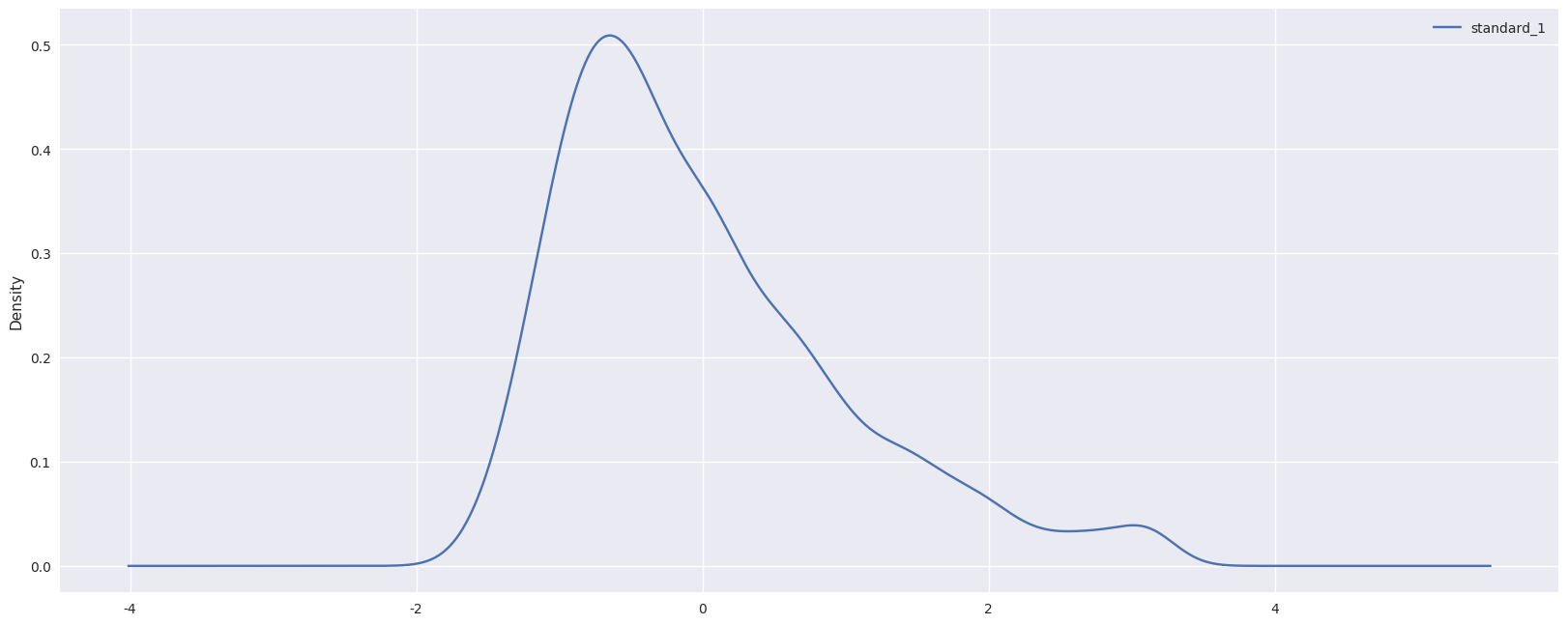
Style factor refers to the unique overall performance of this kind of factor. According to Barra, it can be divided into nine categories: Beta, momentum, scale, profitability, volatility, growth, value, leverage and liquidity. There are also subdivisions in each large class of factors. In addition, there are various new factors that have been explored in order to better analyze the characteristics of different market periods.

* Extreme Value Processing

Before standardizing data, we process outliers first. The method of outlier processing is to adjust it to the upper and lower limits, where the upper and lower limits are given by the criteria for judging outlier values. There are three criteria for judging outliers: MAD, 3σ and percentile method. The main idea is to define the upper and lower limits first, and then adjust the outliers beyond the limits to the upper and lower limits.

* Data standardization

In order to better compare and regress the factors, we need to standardize the factors. Standardization has a series of meanings in statistics, generally using Z-score method. The processed data are transformed from dimensionless to dimensionless, which makes the data more centralized, or enables different indicators to be compared and regressed.



Picture 3.1.1- Standardized post-processing factor distribution

**(2) Single factor test**

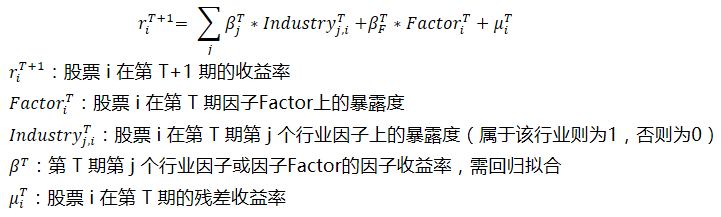
The factor pool we initially constructed when collecting data is logically related to the return rate in a certain economic sense. Next, we make an empirical analysis of them and screen out the factors that are not highly relevant to the return rate, so as to get a truly effective factor pool.

* **Characteristic Analysis**

The correlation between the factors is preliminarily analyzed to determine whether the performance of the factors is roughly similar. Also, Pearson or Spearman method is used to calculate the self-correlation coefficient of the factors and to observe whether there is a significant difference in the decay rate of the factors.

* **Neutralization Processing**
* **Regression Analysis**

The exposures of T-period factors and the stock return rate of T+1 are regressed, and the regression coefficient is the factor return rate of T-period. The model is as follows:



Where: riT+1 : Return rate of stock i at time T+1

FactoriT : Factor exposure of stock I at time T

Industryj,iT : Belong to industry j =1, else =0

βT : Return rate of factor, needs regression fit

μiT : Residual Return Rate

At the same time, because of the influence of small-cap stocks and the heteroscedasticity of regression, we adopt weighted least squares regression (WLS), which weights the square root of the market value of individual stocks.

By analyzing the t value, we can judge the significance of the corresponding regression coefficient, and then we can conclude whether the factor really has an explanatory effect on the next stock return.

Evaluation methods:

* Mean absolute value of t: judging significance
* Ratio of factor return > 0: judging whether the positive impact of this factor on stock return is obvious
* The proportion of absolute value of T >2: judging whether the significance is stable
* The t value of the hypothesis of zero factor return rate: judging whether the return series of this factor is significan ≠0.
* **IC Assisted Analysis**

Factor effectiveness refers to whether a factor can achieve sustained and stable alpha returns. This part mainly uses IC analysis and its derived indicators to evaluate the effectiveness of factors.

IC (information coefficient) is defined as the Pearson or Spearman correlation coefficient of the exposures of each stock at each time cut-off point and the earnings of the next period. The higher the IC value, the more obvious the correlation between the exposures of the factor and the future returns.

* **Layered back-test**

The stock pool is divided into N portfolios according to the size of the factors, or each industry is divided equally. The weight of each share is equal to that of the benchmark (e.g. Shanghai and Shenzhen 300). At this time, the combination is industry neutral.

By grouping the cumulative income graph, we can simply know whether the factor has a monotonous increasing or decreasing relationship with the return rate. There are many evaluation criteria for the results, such as annual return rate, Sharp ratio, information ratio, maximum withdrawal and so on.

**(3) Synthesis of factors**

After the previous analysis, we have screened out the factor pool which has a significant relationship with the yield. However, the factors at this time are still defined by our subjective, they may have a strong correlation with each other. If left untreated, the portfolio will expose too much risk to the same factors, and multiple collinearity will lead to the deviation of the results of multiple linear regression.

**(4) Constructive Model**

* **Determining Factor Weight**

There are four ways to determine weights:

* Equal weight treatment of each factor.
* Factor IC mean weighting. This method takes into account the difference of factor validity and assigns better weights to the factors with more significant performance.
* IR\_IC weighting. According to the basic criterion of return-risk, this method takes into account the factor validity and stability.
* Maximizing compound factor IR. The optimal weight of each factor is obtained by maximizing the IR of the multifactor model, and the optimal multifactor model is constructed by solving the problem. The covariance matrix can be obtained by using common covariance matrix or Ledoit-Wolf compression method.
* **Stocks scored and screened**

By simply assigning weights to complete the construction of multi-factor model, the weight distribution among stocks is generally equal weight, or weighted according to the size of market value.

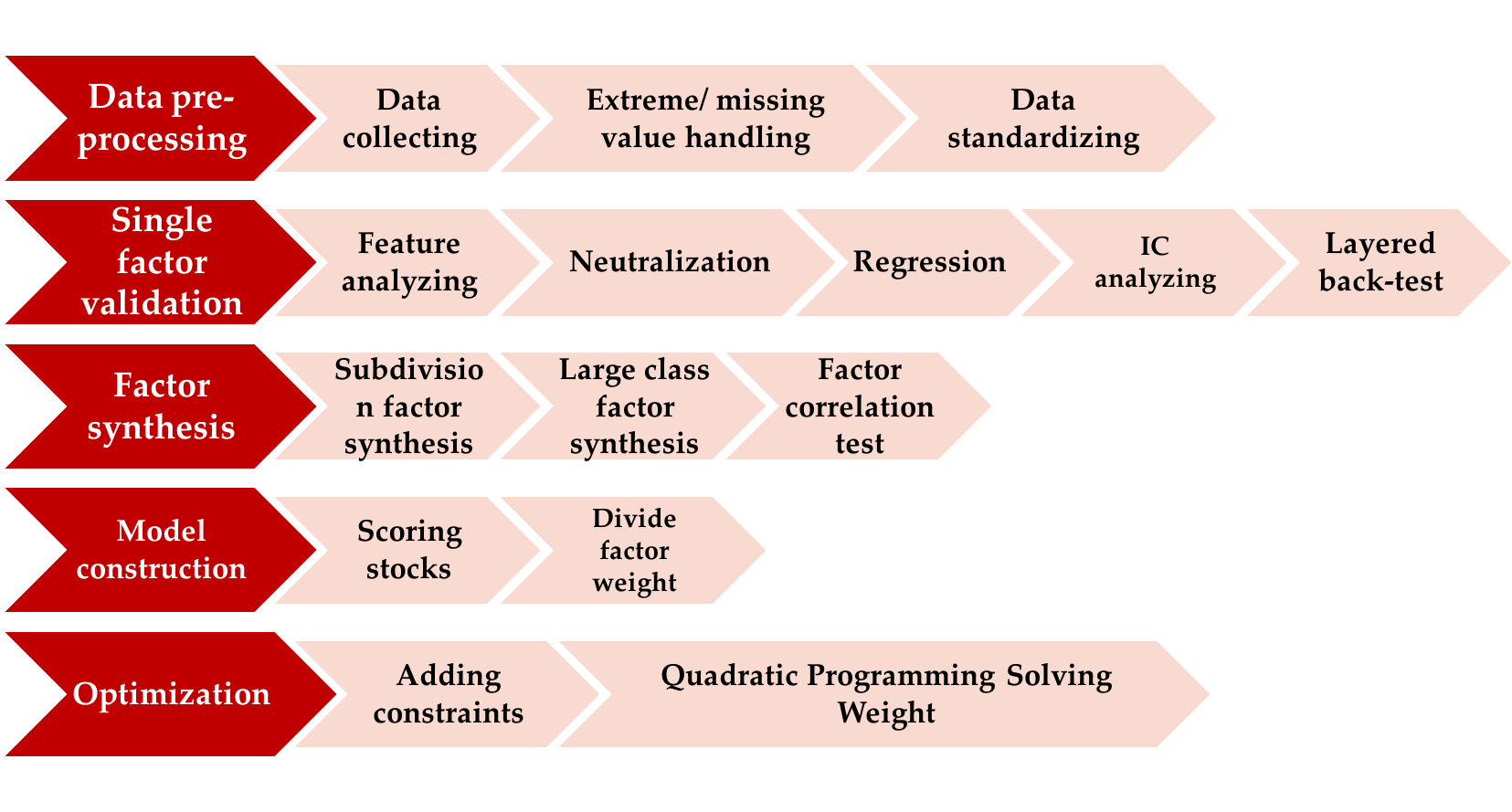
**(5) Combination optimization**

In order to solve the problem of excessive risk exposure in a certain industry, it is necessary to optimize the model buy setting Mean-variance optimization objective.

Also, we can add constraints including:

* Industry weight constraints
* Factor exposure constraints
* Upper and Lower Limits of Stocks
* Revenue target
* Risk objectives

Picture 3.1.2 shows the whole multi-factor model framework.

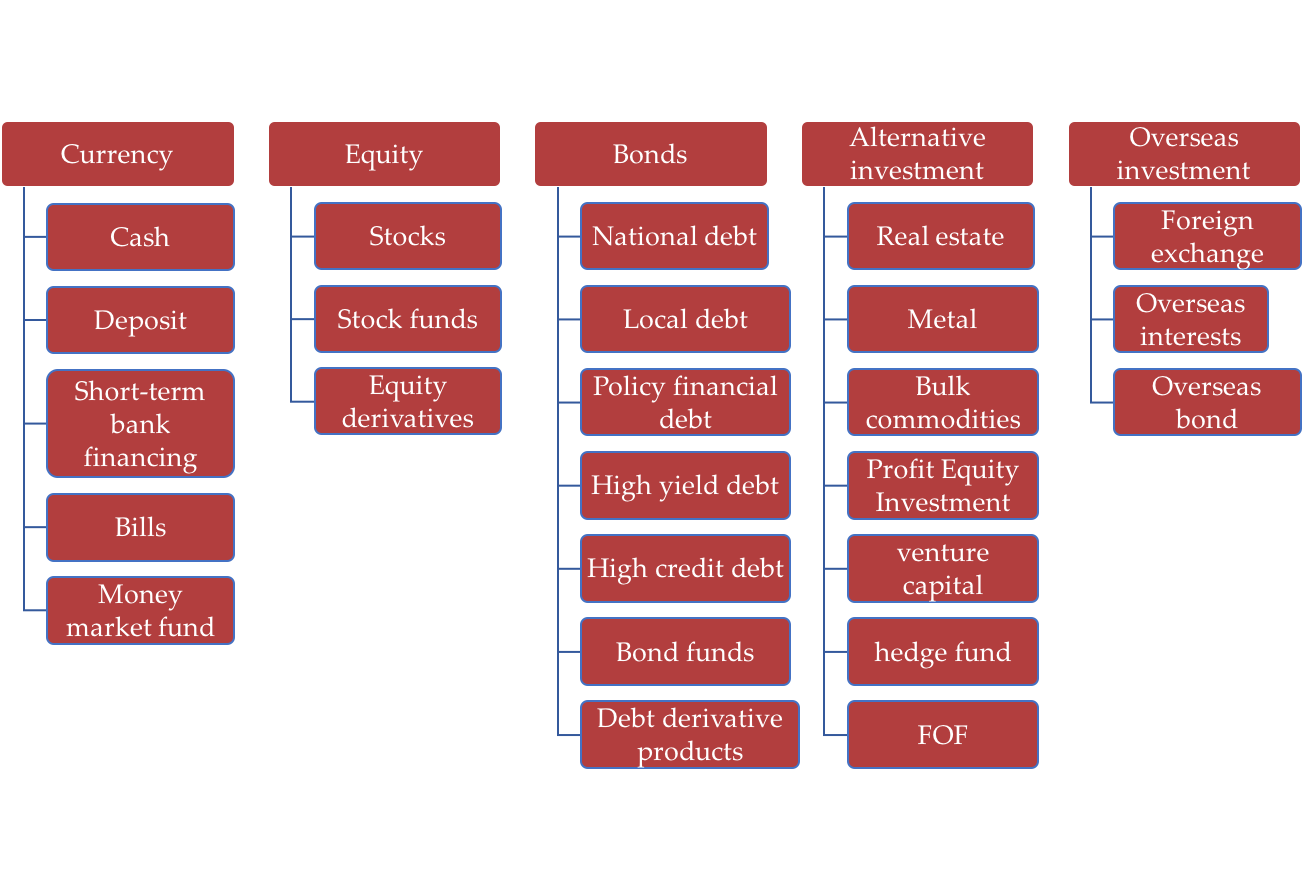
* 
* **Picture 3.1.2- Multi-factor model framework**

**(6) Stock index hedge**

Multi-factor strategy can also be used to hedge with short-selling stock index futures. Share Price Index Futures (SPIF) is a standardized futures contract with the stock price index as the subject matter. The two parties agree that the trading of the underlying index can be carried out according to the size of the stock index that has been determined in advance in a certain future date, and the difference will be settled by cash settlement after the expiration. We utilize the characteristics of futures, including low transaction cost, high liquidity, leverageable trading, and short-selling to achieve Alpha trading strategy for hedging risks.

### 3.1.2 Category Asset Allocation

The major asset allocation is to allocate different types of assets in the portfolio, and adjust them dynamically according to the situation of investors and the market. One of the common methods is to use the mean-variance model to analyze and forecast the return, volatility and correlation of various assets, and to get the optimal portfolio by combining utility function.



**Picture 3.1.3- Classification of Large Categories of Assets**

### 3.1.3 Timing Strategy

Time-based trading refers to the use of a method to judge the trend of the trend, whether it is rising or falling or consolidating. If the judgment is up, then buy and hold; if the judgment is down, sell the clearance; if the judgment is shock, then carry out the high-selling and low-slow, so that the yield can be far beyond the simple buy-holding strategy.

### 3.1.4 CTA Strategy

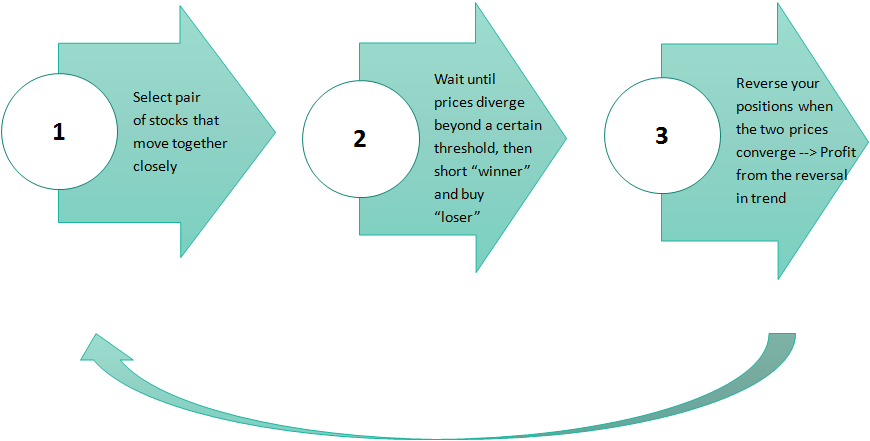
This system uses the classic Dual Thrust strategy.

The Dual Thrust strategy was developed by Michael Chalek in the 1980s, and the Dual Thrust strategy is easy to use and versatile. In this strategy, the definition of the oscillating interval is critical, which is the core of the trading system. The specific steps of the strategy are:

* The first step is to calculate the price volatility interval of the previous N days, which is the Range value. In the specific calculation, the four price points of the previous N days are introduced: HH, HC, LC, LL, which represent the maximum value of the highest price of the previous N days, the maximum value of the closing price, the minimum value of the closing price, and the minimum value of the lowest price. And calculate the Range value by the formula Range = Max(HH-LC, HC-LL).
* In the second step, parameters K1 and K2 are introduced (this parameter can be determined based on historical data and dynamically adjusted) to determine the price of the upper and lower rails that trigger the transaction. Among them: upper rail = opening price + K1 \* Range; lower rail = opening price - K2 \* Range.
* The third step, the trading rules are: (1) When the price breaks above the upper track, if the position is open at the time, the position is closed first, then the multiple positions are opened; if there is no position, the multiple positions are opened directly; (2) when the price When you break down the lower track, if you hold multiple positions at that time, you will close the position first and then open the position; if there is no position, open the position directly.

### 3.1.5 Statistic Arbitrage

The system uses a pair-trading strategy to match stocks based on the similarity of market capital. When one stock outperforms another, the underperforming stock will climb its value as the market expects, resulting in an excellent return on investment. This process is to hedge against market changes/trends. Since a large number of stocks participate in the statistical arbitrage strategy, the capital turnover rate of the portfolio is very high, and the scale of the dissemination is relatively small, so this strategy is usually implemented in an automated manner, and it is highly valued to reduce transaction costs.



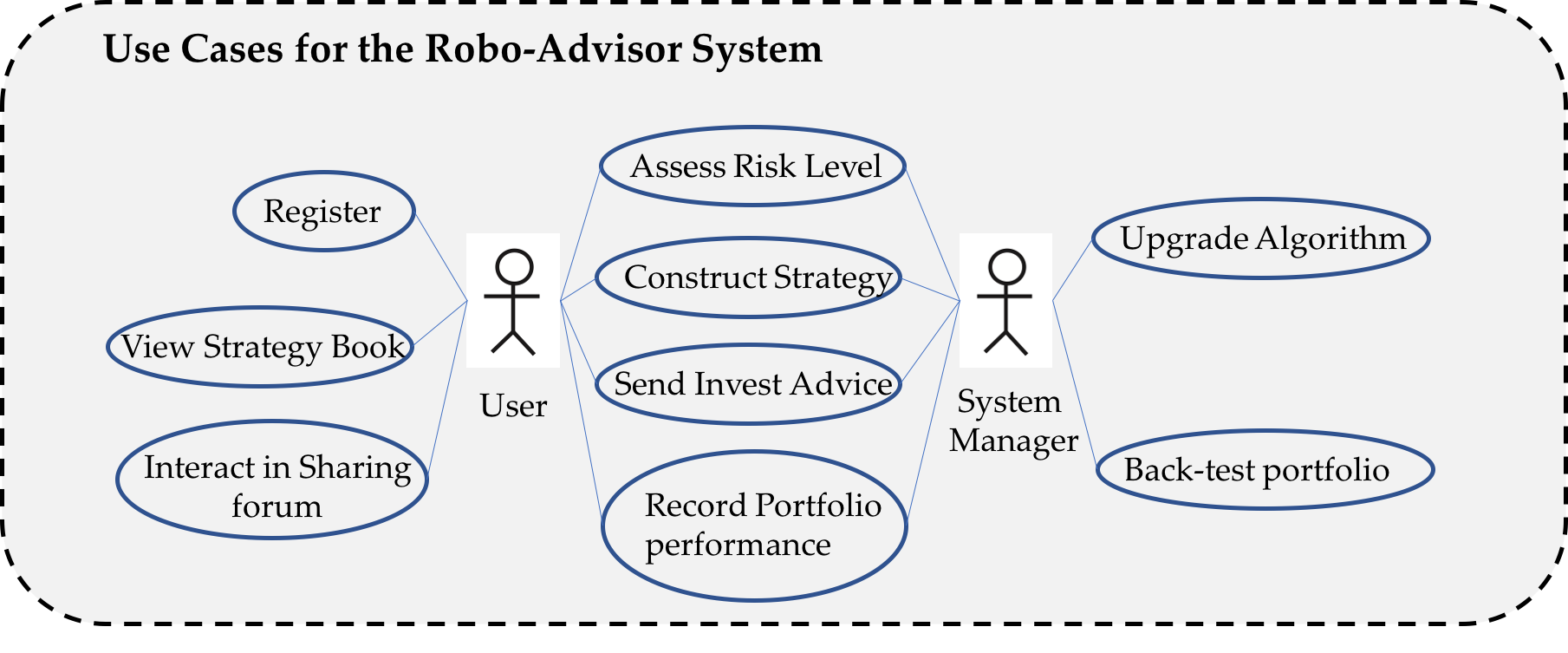
**Picture 3.1.5- Pair-trading Strategy**

### 3.1.6 Industry Rotation Strategy

The industry rotation adopted by this system is an active trading strategy that utilizes market trends to profit. The essence is to use the mismatch of different investment types to change the industry varieties to maximize the return on investment. According to the difference in the performance of different industries, the rotation configuration is to strive to capture the industries with better performance in the interval and eliminate the underperforming industries. When judging that the market is not prosperous, the equity positions are lowered, and the bonds or currencies are raised. proportion.

## 3.2 Use Cases

The system use cases are shown in Picture 3.2.

****

**Picture 3.2- System Use Cases**

# 4. Design Overview

## 4.1 Introduction

Software architecture is a collection of structured elements, that is, a collection of components, including processing components, data components, and connection components. The processing component is responsible for processing the data, the data component is the processed information, and the connecting component connects the different parts of the architecture.

Software architecture provides a high-level abstraction of structures, behaviors, and attributes for software systems, consisting of descriptions of the elements that make up the system, the interaction of these elements, the patterns that guide the integration of elements, and the constraints of these patterns.

Software architecture has 10 common architectural patterns and their usage: layered architecture, client-server architecture, pipe and filter architecture, container architecture, and more. Each architecture has its advantages and disadvantages. We should choose the software architecture that suits the system according to the needs of the system.

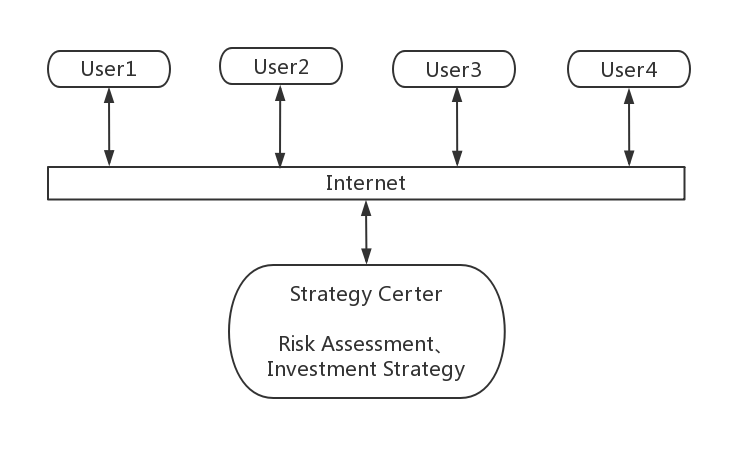
## 4.2 Software Architecture

The system mainly involves two forms of software architecture: client-server architecture and layered architecture. From the macro perspective, the entire system is the client-server architecture. Looking at the system server part alone, it is a layered architecture. Next, we will elaborate on how these two architectural forms are used in our system.

(1) Client-server architecture

This pattern consists of two parts: a server and multiple clients. The server component will provide services to multiple client components. The client requests services from the server, and the server provides related services to these clients. In normal times, the server will always listen to client requirements to provide services to customers.

Each client of the system is a user who uses the system differently, and the server side is the central management system of the system. When the user uses the system, when using each function, the service center will perform corresponding calculations according to the customer's needs, retrieve the central database data, and present the final result to the customer.

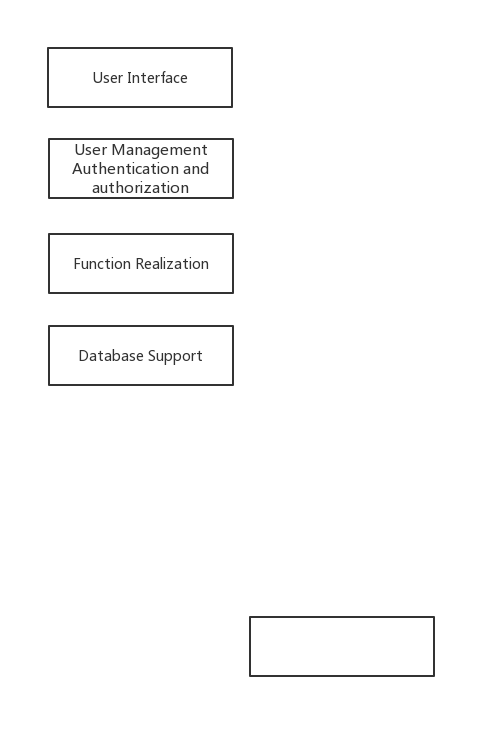


**Picture 4.2.1- Client-server architecture**

(2) Layered architecture

Layered architecture pattern is an architectural approach to achieving separation and independence. The functionality of the system is divided into several separate layers, each layer relying only on the services and facilities provided by the next layer immediately below.

In our system, a layered architecture is also used. The structure is as follows:

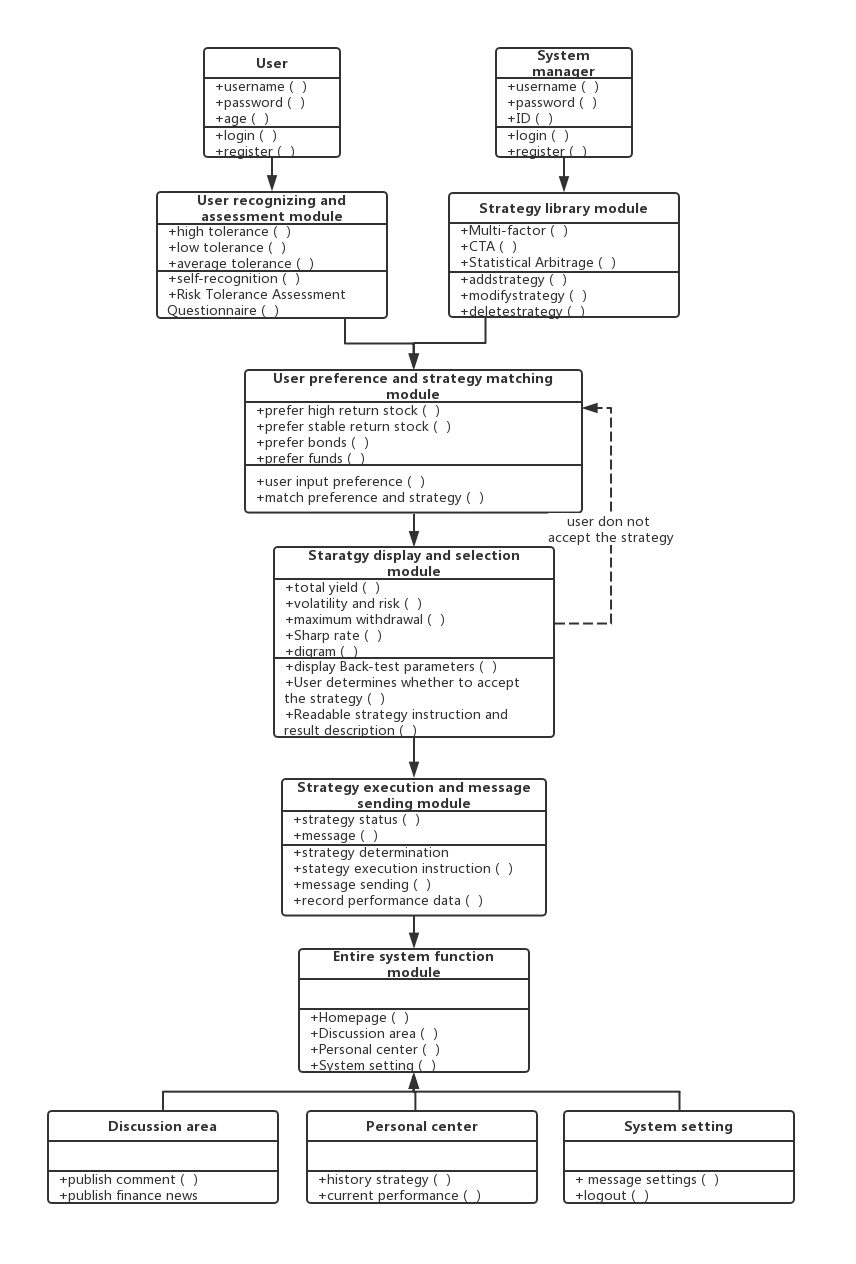


**Picture 4.2.2- Layered architecture**

The user enters user name and password in the user interface to perform a login operation. The system will verify the password, if the password is correct, user can use the corresponding various functions. The background terminal performs operations on the support of the database data and returns the result to the user, such as investment strategies, risk assessment and so on.

## 4.3 System interfaces

We designed the System interfaces according to the existing product logic. Firstly, we have four separate pages: strategy module, discussion area, personal center and system setting. And the strategy module contains the core function of the system, where internal data and logic are connected. User firstly login or register the system, then accomplish the risk preference validation. The risk parameter and expected return are sent to middle engine to match strategy. The strategy will be developed with the asset allocation advice, back-test result and investment advice in the middle, and returned to the front. If the strategy is be approved, the system will operate the strategy and timely send the trading messages to the customers. The specific process is shown in the figure below.



**Picture 4.3-System Interface**

## 4.4 Performance

For performance requirements, the system requires mainstream hardware configuration to ensure the speed of the server. The MySQL database is used to ensure stable system performance under a large amount of data exchange; the selected information system also technically guarantees system reliability and certain data security. In addition, in the actual development and application process, the corresponding password encryption technology and database backup and recovery mechanism will be adopted to ensure the confidentiality and integrity of the data, thereby ensuring the reliability and security confidentiality of the system.

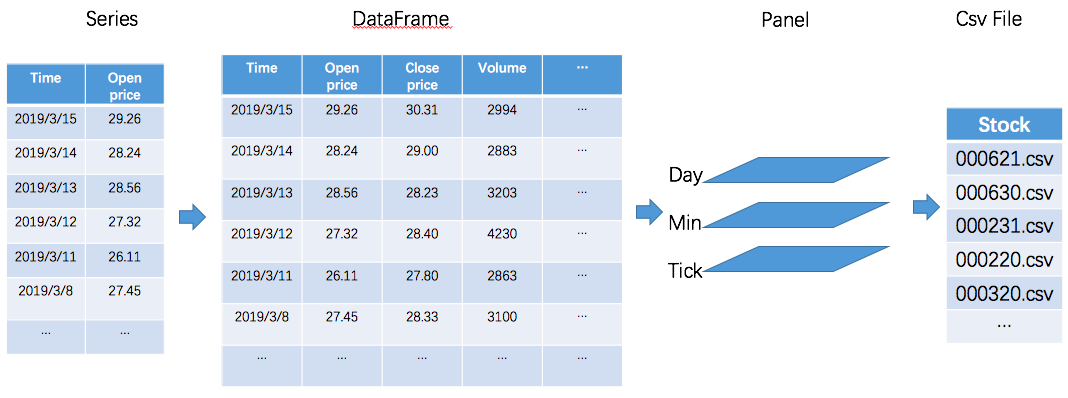
# 5. System Design

## 5.1 Introduction

Based on the system description, our system is designed as a layered information system architecture which is divided into three parts including user interface, middle engine system and underline data system. In this architecture, up layer is dependent on the below layer and the bottom layer is the base of our system which store the quota data of stocks, futures and fix income asset used by all strategies. There are large amount data for containing several frequencies and lasting for long time. In this case, it is very important to organize and store the data well.

## 5.2 Data and database design

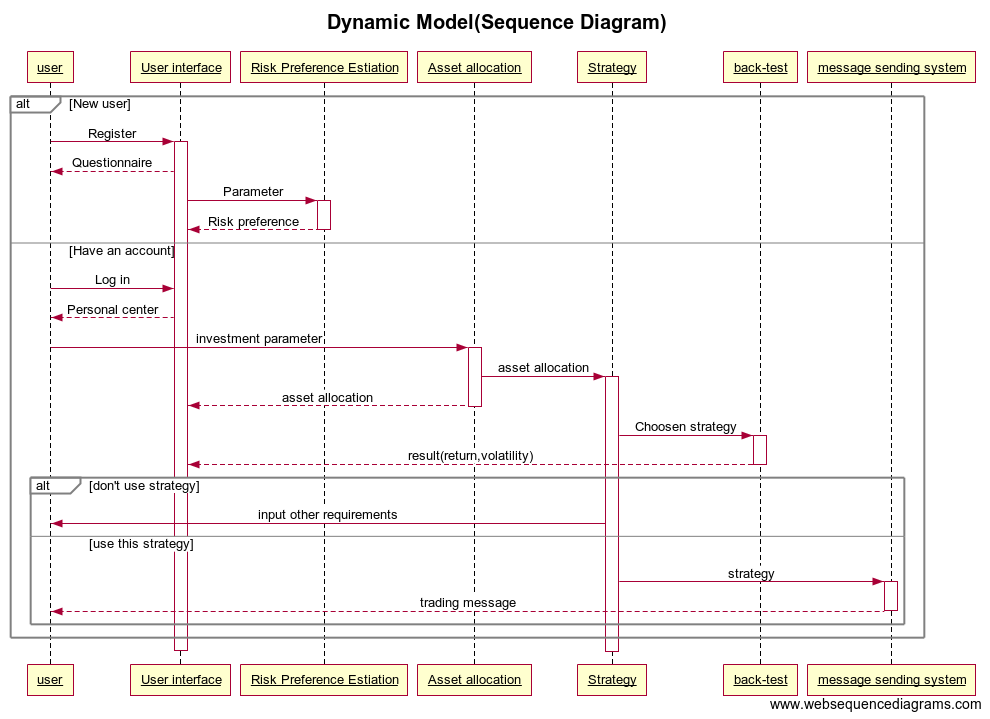
Most of our data are time series data describing the price, volume, position and other technical indicators of stocks, futures and fix income assets in different time points. Because our back-test framework often uses 'DataFrame' in 'Pandas' which is a pack of python to utilize and store the data, the data in the database are also store in 'Pandas'. We can get the quota data from the interface of Wind but we need to store them in local server for high speed calculation. To store them, we divide them according their name because every target asset has a specific code. For example, the system organizes the quota data of 'Shanghai and Shenzhen 300 stock index' whose code is '000300' in the type of 3 dimensional panel. The first dimension is time; the second is types such as open price, close price, volume, position and other technical indicators of stocks; the third is frequency such as day, minutes and ticks. Other frequency data could be generated by this frequency. Then the system stores this panel in the type of csv file in local server. There are thousands of target assets so that the system will divide them by kinds and store those csv files into several files for searching easier.



**Picture 5.2- Structure of database**

## 5.3 Dynamic model

Sequence diagram can be used to describe our dynamic model because each activity occurs in chronological order. This system has several main classes including user interface, risk preference estimation, asset allocation, strategy decision, back-test and message sending. User could control some activities through the user interface. The information between those classes are shown in the sequence diagram below.



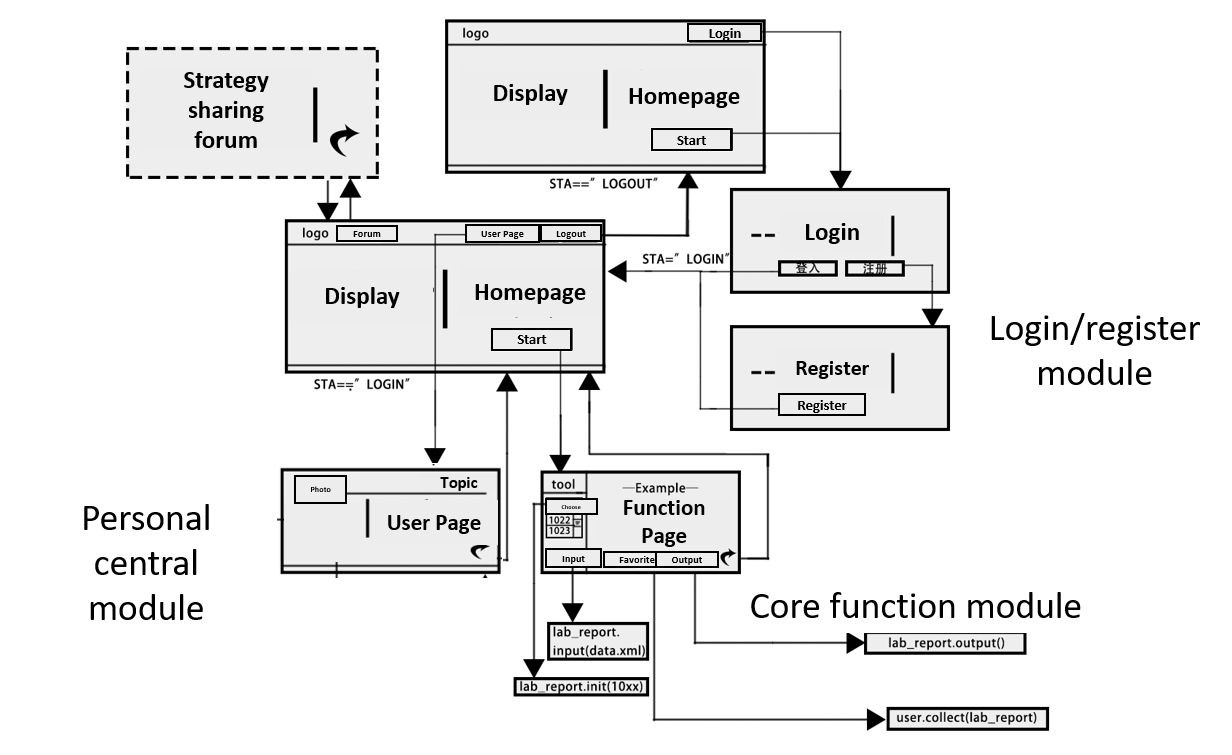
**Picture 5.3- Dynamic model (Sequence Diagram)**

## 5.4 Application program interfaces

Our system is mainly to provide trading suggestions to the consumers who don't have experience in quantitative trading or programming and we don't have to set application program interfaces(API) for users now. However, our system may evolve with development of consumers' requirements and consumers may need API to develop their own strategy when they are familiar with this area. What's more, API is very useful for big project which contains several different parts to cooperate with each other. In this case, several main parts of our system need API for others to use.

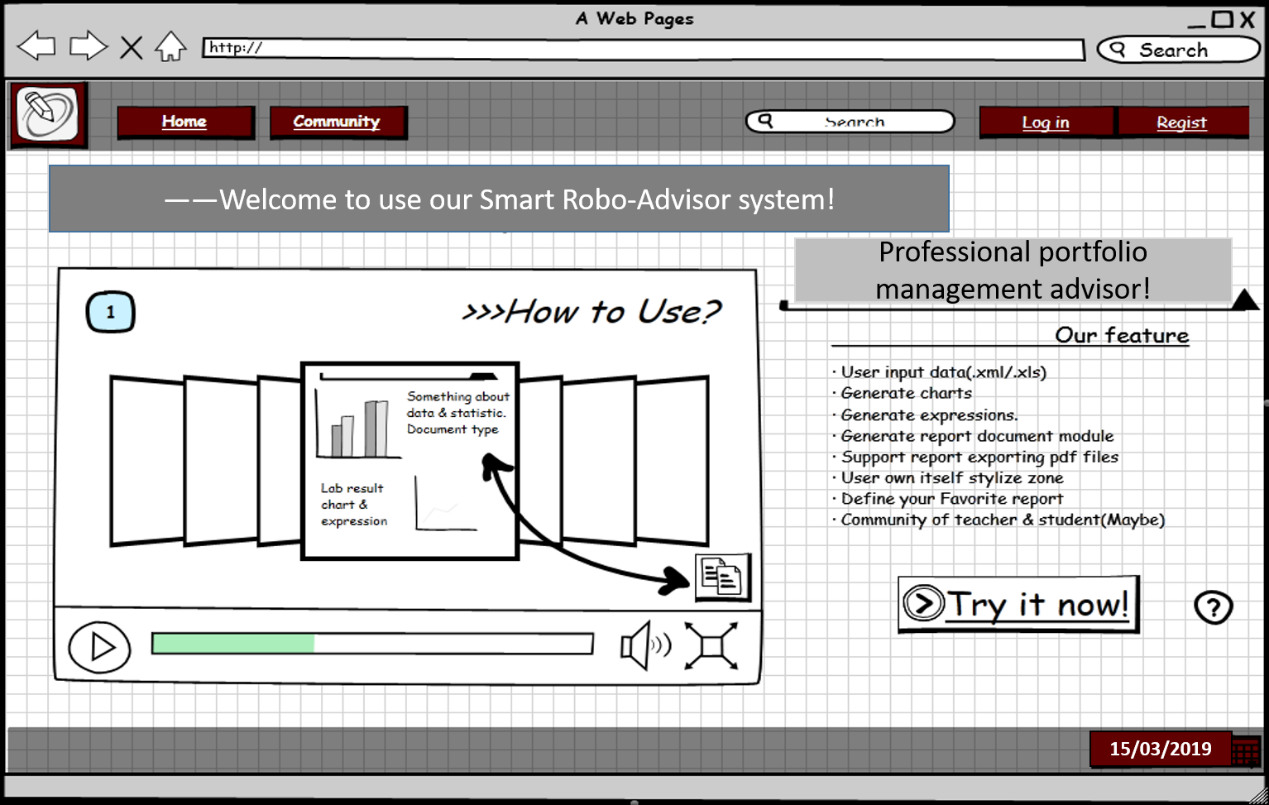
1. In the database management system, API should be set for extracting quota data of specific time period, frequency, type, and target asset because it could be widely used in back-test system.
2. In the back-test system, API should be set to get the back-test results including return, volatility, the biggest withdraw which can be used in optimizing strategy.
3. The asset allocation system should set API to return how much money are distributed to stocks, futures and fix income assets which also will be used in strategy optimization.

## 5.5 User Interface Design



**Picture 5.5.1- User Interface Design**

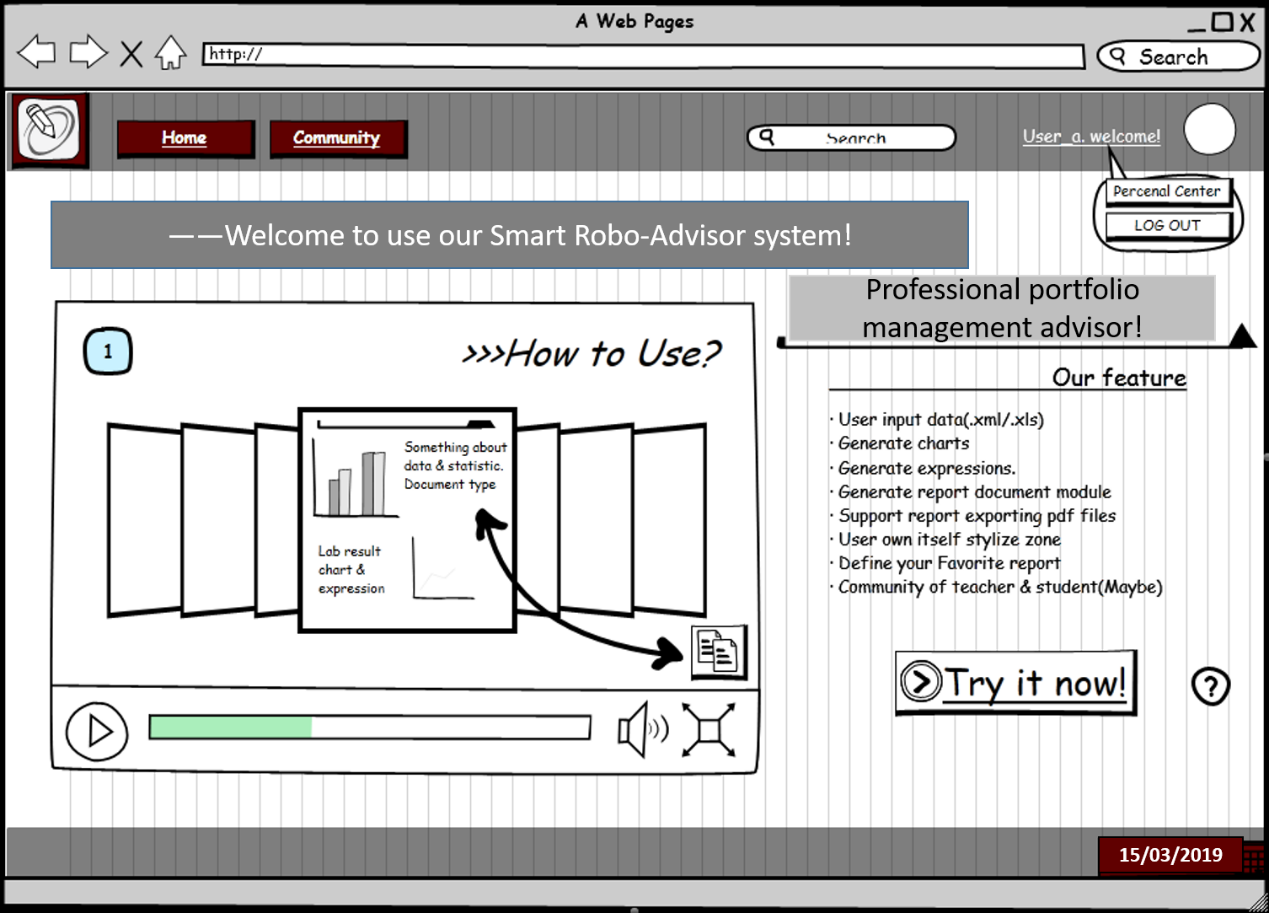
We design the whole UI framework as following:



**Picture 5.5.2- UI framework**

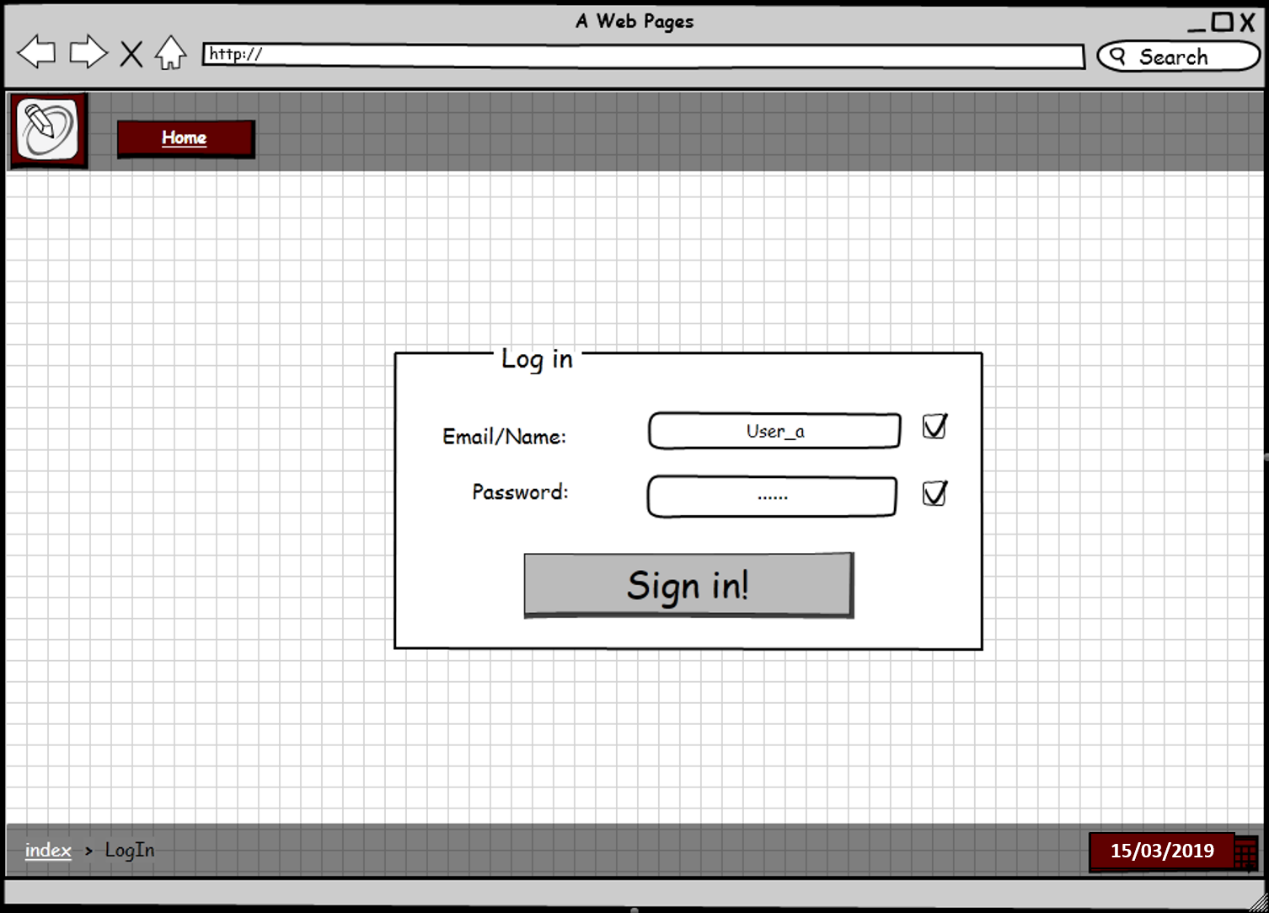
1. The above picture shows the prototype preview of the homepage preview we designed. The buttons and functions involved are:

| Button | Function |
| --- | --- |
| Home | If the user is already logged in, the home page when the user logs in is returned; if the user is not logged in, the user preview home page is returned. |
| Community | Click to enter the strategy exchange forum |



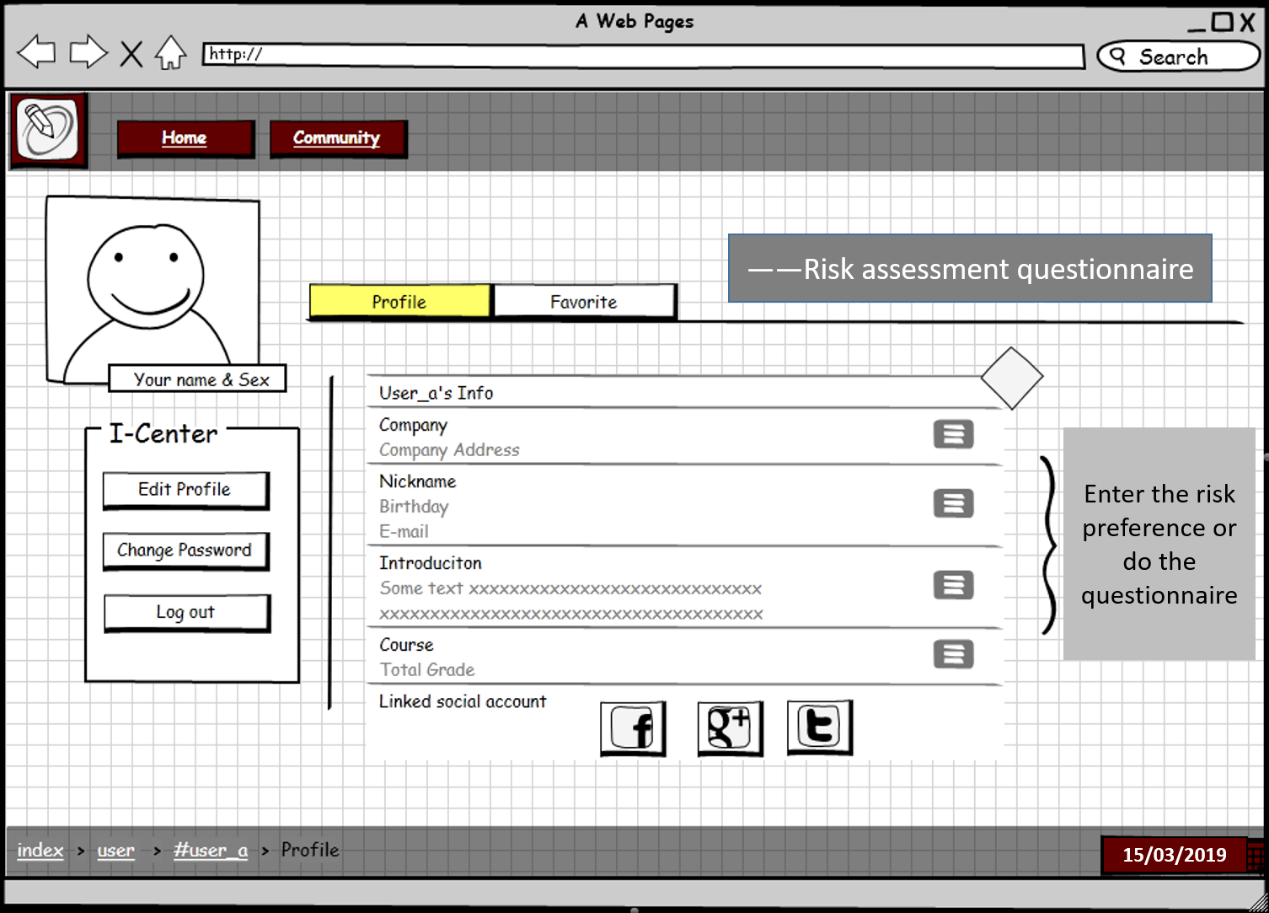
1. The added features of the home login version are: When the user clicks on User welcome in the upper right corner, a hover box pops up with two buttons. The buttons and functions involved are:

| Button | Function |
| --- | --- |
| Personal Center | Clicks to enter the user's personal center |
| LOG OUT | Click to exit the login status, and the page jumps to the home page preview interface. |

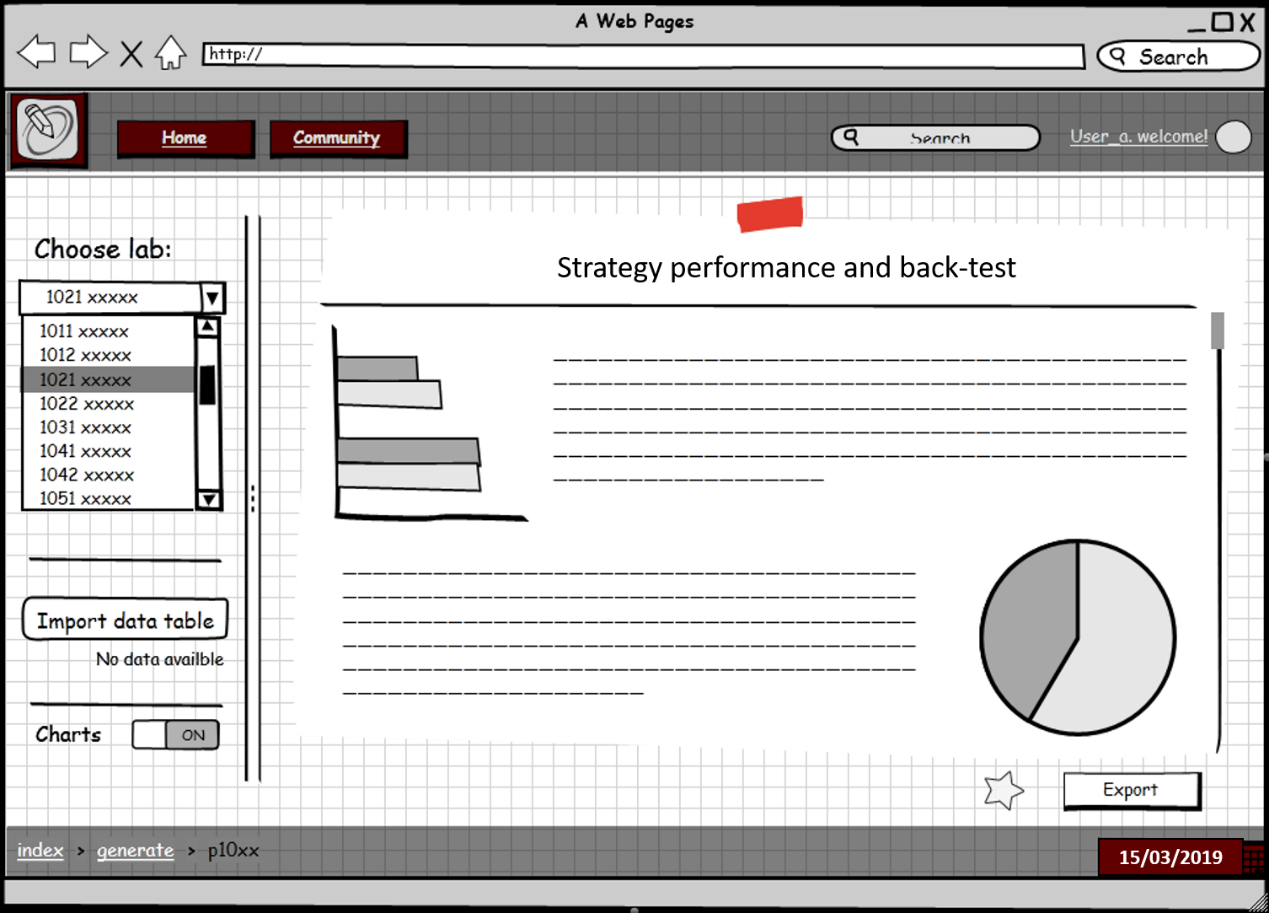


1. The above is the prototype of the login interface we designed. The controls and functions involved are:

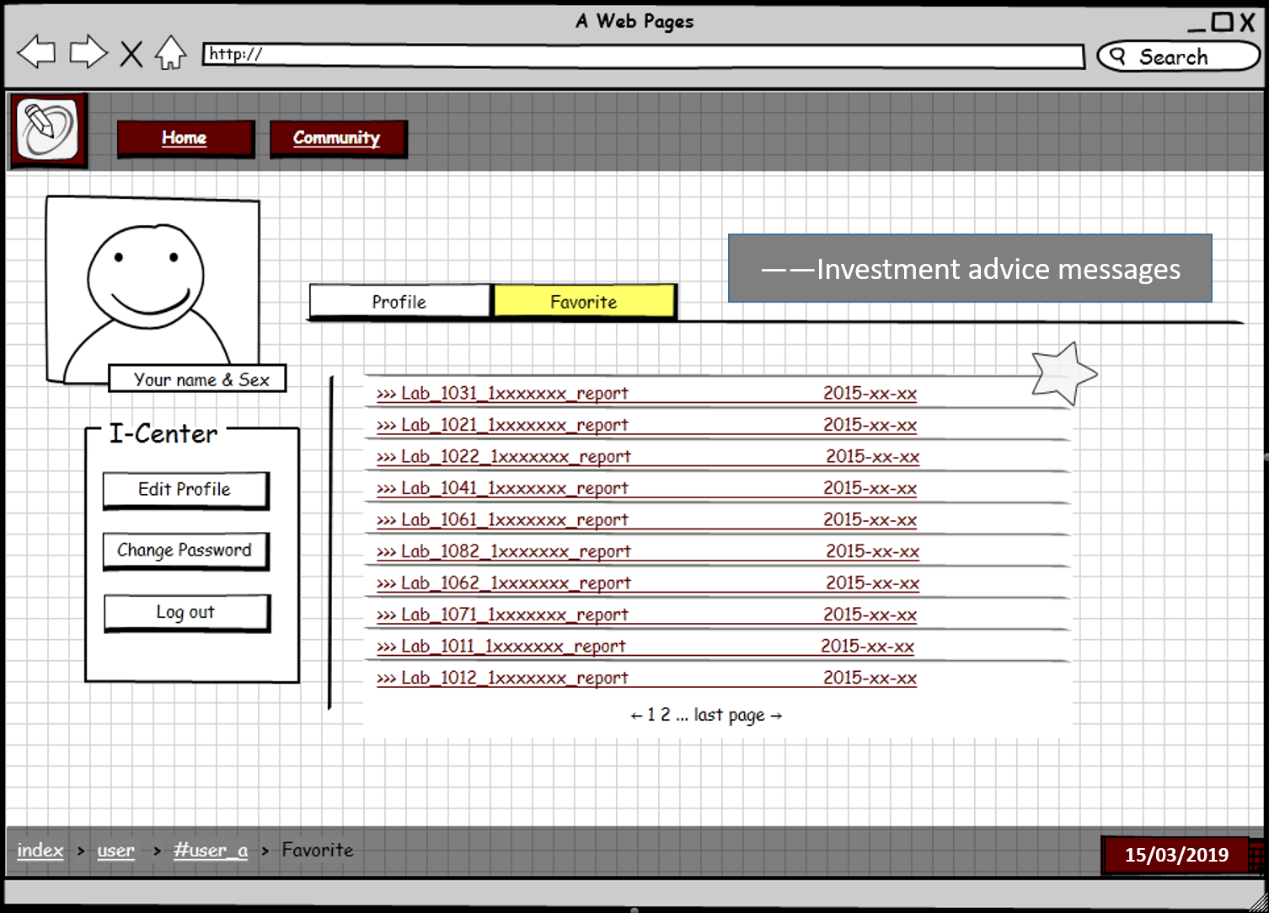
| Button | Function |
| --- | --- |
| Home | Back to Home Preview |
| Email/Name | Fill in the username or email to log in. |
| Password | Fill in the password. The entered password length is limited to 6-15 characters: If the password length does not meet the limit, the checkbox on the right becomes an error. Otherwise the checkbox is in the correct state. |
| Sign in! | Use the completed username and password to query in the database. If the record is not empty, it will automatically jump to the home login version, otherwise it will jump to the login interface. |



1. The above is the prototype of the risk assessment questionnaire. Users have two choice, one is to enter the preference by themselves. The other is doing the questionnaire in the system.



1. The above is the prototype of the strategy display page. After the user inputting various parameters, the system will return the recommended strategy, the description of the strategy and the back-test results.



1. The above is the prototype of the investment advice sending page. The system will send messages about the transaction instructions to the user on this page.

## 5.6 Performance

Performance requirements: In order to ensure long-term, safe, stable, reliable and efficient operation of the system, the system should meet the following performance requirements:

1. System ease of use

Systems are directly facing customers, and they are often familiar with computers from time to time. This requires the system to provide a good user interface and an easy-to-use human-computer interface. To achieve this, the system should use the interface familiar to the user and the Chinese information as much as possible. For the user's possible use problems, provide a complete "use documentation" to speed up the user's familiarity with the system.

1. System maintainability

Some of the data involved in the system is important to the customer. The system should provide convenient means for the system maintenance personnel to back up the data, daily security management, and data recovery when the system crashes unexpectedly.

## 5.7 Compliance

Need to meet the standards, laws and regulations, and regulations of various industries. The system involves computer software during the design and development process. All of this must be in line with mainstream international, national and industry standards. For example, operating systems, network systems, and development tools used in development must conform to common standards.

# 6. Product Design Specification Approval

****

**Picture 6- Product Design Specification**

# APPENDIX A: REFERENCES

Qian, E., & Hua, R. (2006). Active risk and information ratio. In *The World Of Risk Management* (pp. 151-167).

Konno, H., & Yamazaki, H. (1991). Mean-absolute deviation portfolio optimization model and its applications to Tokyo stock market. *Management science*, 37(5), 519-531.

Björk, T., Murgoci, A., & Zhou, X. Y. (2014). Mean–variance portfolio optimization with state‐dependent risk aversion. *Mathematical Finance: An International Journal of Mathematics, Statistics and Financial Economics*, *24*(1), 1-24.

# APPENDIX B: GLOSSARY

This part lists the definitions of the glossary used in the document in Table 3.1, so that everyone who looks at the document can have a basic understanding of each term in the document.

|  |  |
| --- | --- |
| Glossary | Definition |
| Quantitative Trading | Use computer technology to select strategies from a large historical data that can generate excess returns with a variety of ‘high probability’ events. |
| Artificial Intelligence | [Intelligence](https://en.wikipedia.org/wiki/Intelligence) demonstrated by [machines](https://en.wikipedia.org/wiki/Machine). |
| Asset Allocation | Allocate funds in different asset classes based on investment demand. |
| Back-test | Test the performance of strategy with historical data. |
| Multi-factor | Use a series of factors as the stock selection criteria and select stocks with higher scores to build a portfolio. |
| CTA | Timing strategy in the futures market. |
| Statistical Arbitrage | Find out the two assets with similar trend, buy low and sell high. |

**Appendix B: Glossary**