Project

# Progress

Tip: The progress should be based on the project proposal.

Briefly summarise what have you achieved up until now.

Milestones should be kept according to the plan.

**What does back end developers have already done?**

APIs part:

About the APIs in back end progress, we have already get the data from GDAX, Binance and Bitstamp through API request methods.

Considering about the public and private data, we have get public data from web socket, such as k-line, depth chart, current price etc.; for the private data, due to client application’s policy, we cannot become the legal clients of GDAX. So, we use sandbox as private data adding some testing trades and transactions.

Data Processing part:

The back-end has completed some charts’ data processors, real-time data acquiring, MongoDB operations APIs and project layers’ definitions. The data processors provide the mature data for the front-end view (depth chart, value and volume chart, exchange rate feed have been finished), such as search data from database, transform data format and filter the data coming from external APIs. The real-data acquiring has been finished, when the server starts to run, the new data will be updated to MongoDB. The insert, search and update functions of MongoDB operations have been completed.

**What does front end developers have already done?**

Echart and amchart:

Layout for the whole website:

We have completed “dynamic-analytic.html”, “user-login.html”, “user-register.html”, “user-forgot-password.html”, “depth.html”, “user-profile-page.html”.

The “dynamic-analytic.html” is the landing page for our program. The user can see everything in this page. Furthermore, in the landing page users could get the data about fiat-crypto currency exchange and crypto-crypto currency exchange directly. By the way, the data are real-time changing. “user-login.html”, “user-register.html” and “user-forgot-password.html” can be seen as a package of private users. They can get private account in our system which will be leveraged for store transaction data from other exchange websites. This is a better way to collect competitor’s data and attract more customers to use our website more frequently.

**Deviation to Timeline**

Tip: You need to justify any deviation to the original timeline. You should also explain the reason why there are changed to the original project planning.

|  |  |  |  |
| --- | --- | --- | --- |
| Timeline | Planning tasks | Reporting job | Deviation |
| Week 5  13-04-2018 | Definition: Proposal Report  Mission: Totally familiar with APIs in GDAX, Huobi and Binance; Can store public and private data into the database; Can run front-end demo successfully;  Proposal Report Due.  Risk: Unsuitable API invoking may delay the delivery time. | Deliverables: Proposal Report  Description: Finish two of three APIs invoking (GDAX and Binance). The frontend framework demo has been finished.  Finish the proposal report | Front end: cannot be a real-time and dynamic Candle stick chart;  The demo lack of several basic functions, but only current price and login functions can be come true.  Back-end: Huobi website cannot offer the USD exchanging method, only for CNY (Chinese Yuan); |
| Week 6  13-04-2018 | Definition: Cooperate work  Mission: Front-end and back-end program will cooperate with each other smoothly; the visualization of test data the development of main page (Navigate bar, menu, main page charts, exchange rate feed) should be completed.  Risk: None | Deliverables: An executable work line between frontend and backend  Description: Complete front-end first demo version and the development of main page template. | Front end: The home page lack of world map and the information of private data;  Navigation bar does not have two kinds of “Legal currency” and “Cryptocurrency” exchange;  Back end: Login function does not connect with back-end database, which cannot store API’s key and secret. (This function aims to provide our users private information from each official exchanging websites.) |
| Week 7  25-04-2018 | Definition: Integrate all the parts of project  Mission: Complete the integration of different components and subpages, the major functionalities of back-end and front-end should be implemented.  Risk: Some parts may not work well, that will cost time to fix it. | Deliverables: A project can do the demo  Description: Complete the demo project and start to optimize the performance. | Do not finish the whole project on schedule. We can only run some separate pages with different routes. |
| Week 8  09-05-2018 | Definition:  Mission: All the development of features and functions complete, begin the project beta V1.0 testing process  Risk: | Deliverables:  Description: Testing the website performance by a series of processes | The project has not finished yet. |
| Week 9  11-05-2018 | Definition: Progress Report  Mission: Progress Report Due. And finish the work that may be delayed.  Risk: None | Deliverables: Progress Report  Description: Finish the progress report | The project has not finished yet. |

**Why does it delay against timeline from the back end to front end?**

Normal reasons:

Layout design:

1. Through the group discussion, we picked the overlay layout as our main menu, which is a little harder than the traditional design, so we spend more time on this part. The reason why we did this is that we want to show more information about currency on home page but we need to keep the main data showing in the front of users.

|  |
| --- |
|  |
| Figure 1 |

2. According to the requirement of assignment, “bitcoin stream”, we changed what we focused before in the whole project. We will finish two parts of this website, which is bitcoin exchange with legal currencies (e.g. BIT/USD, BIT/CNY, BIT/EUR, etc.) and cryptocurrencies exchange with bitcoins (e.g. ETH/BIT, LTE/BIT, BNB/BIT, etc.). Here we replaced the original design with the form of first level and second level menus, which is more clear and convenient for users when they browse the data on our website.

Functionality design:

1. Originally, the data we intend to exhibit in the website was collected from four exchange websites, however, we made a change on this part, some data will be recollected in the form of global data.

2. About the k-line (candle-stick chart), we made a group decision that we will add some predicting algorithms in k-line chart. In the future, some special and meaningful point will be labelled in the chart, which is a better way to help our users make buy or sell decision.

Back end:

APIs:

1. Based on the requirement, “bitcoin stream”,

2. Due to comparing different official exchange websites, we pick famous exchange websites, such as Binance, Bitstamp, GDAX and OKEX or Kraken. The reason we give up Huobi.com website is that it can only provide BIT/CNY exchange, rather than BIT/USD.

3. About the new feature, trading world map, we do several research on bitnodes.com, for instance, IP, city name, latitude and longitude. (What the effects does it have?) - (1. Show the trading position. 2.?)

Data and database:

1. Real-time data for all of charts and labels;

2. The format of required data is different from what we call back;

Front end:

Layout:

1. After picking the overlay layout, we change the basic functions in the home page. At the early stage, we only want to deal with the bitcoin exchanging issues for our users, such as trading bitcoins with USD, AUD or EUR, however, for now, we add another function at the same level, trading other cryptocurrencies with bitcoin, such as LTE, ETH and BNB.

Obstacles

Obstacles in Front end:

First of all, the main obstacles in this project are that it’s hard to differentiate our web program with the other cryptocurrency exchange website. We need to study the structure of other website and think about how to make our functionalities different compared with others, even though they have already very beautiful to users. Then, when we find out some sections in other websites such as Binance and GDAX, we need to think about whether we have enough time to design and implement our designed functionalities. These two parts cost us a large volume of time.

Furthermore, we decided to make a world map to show the real-time and dynamic exchange data all over the world. Even though the bitnodes give us the source code to crawl the data seconds by seconds, the code is python programmed in the source code. Thus, if we use the same way as bitnodes did, we need to spend lots of time to figure out the running mechanism in bitnodes. We may finally quit on this job but to do it in a simpler way.

Leon: Overlay, K-line, Depth chart,

Back end:

In the process of handling with collected data, the main tasks include database operation, real-time data crawling and solve the problem of using ejs and html framework.

Data processing is to process raw data to mature data for front-end views. The major risks in this part are that we failed to get access to data source and even if we get the raw data, the data format is difficult to process. If the former situation happened, the data processor we designed would not work, which means error would come out. If latter situation happened, the data processor run into total failure. The major issue of the data process is that we cannot be promised uniformed data, because some data JSON format but the others are String format. The String format data cannot be directly stored into MongoDB, so we need to change the data format to JSON format which will influence response speed of website.

In the real-time data crawling part, it is difficult to decide how long the interval time is for catching data from data source. If the interval time is too shot, that the MongoDB will not have enough time to store the data or the front-end page cannot render the page successfully. Finally, we decide to acquire data every 5 seconds, that will leave enough time for rendering page and storing data into database.

How to code for two options of ejs and html framework is big issue in our project. At the first beginning, our program is used for ejs framework to render our front-end page but later on, with the development of other templates we need to use too many html files in the process of rendering page, so we update the view engine for supporting ‘html’ and ‘ejs’ at the same time.

Tip: In this section you need to explain any major risks, major issues, and external dependencies involved in your project. The aim is to explain how you will mitigate each risk or issue.

# Milestones and Reporting

Tip: Describe the planned timeline and any updates to the original plan here. You can show the changes to the original plan in different colour.

|  |  |  |  |
| --- | --- | --- | --- |
| Milestone | Tasks | Reporting | Date |
| Week-1 | Definition: Orientation  Mission: Participate Orientation and know members in our group  Risk: None | Deliverables: None  Team members’ express individual statements on the project. (State: Completed) | 14-03-2018 |
| Week-2 | Definition: Define framework and purpose  Mission: Figure out the purpose of web program and the necessary functionalities in this project. Decide to use which type of MVC framework and development platform.  Risk: The unsuitable purpose and framework will delay the delivery time. | Deliverables: Designed Framework  Description: Design a framework (State: Completed) | 21-03-2018 |
| Week-3 | Definition: Deploy framework and APIs  Mission: Complete the MVC framework design, and finish at least one module, for example, we can download data from third-party organisations by using RESTful API method.  Risk: If the deployed framework is unsuitable, that will delay the delivery time. | Deliverables: Framework directory and one executable work line.  Description: Develop framework and finish the draft framework. The executable work line is completed. (State: Completed) | 25-03-2018 |
| Week-4 | Definition: MongoDB creation and CRUD function development  Mission: Make sure that the database could connect with local service, and CRUD operation could be run smoothly when using RESTful API (request GDAX public data and private data successfully get response); Finish the proposal draft.  Risk: None | Deliverables: A connectable MongoDB and one executable work line.  Description: Connect the database and finish at least one completing work line. (State: Completed) | 28-03-2018 |
| Week-5 | Definition: Proposal Report  Mission: Totally familiar with APIs in GDAX, Huobi and Binance.; Could store public and private data into the database;  Could run frontend demo successfully;  Proposal Report Due.  Risk: Unsuitable API invoking may delay the delivery time. | Deliverables: Proposal Report  Description: Finish two of three APIs invoking (GDAX and Binance). The frontend framework demo has been finished.  Finish the proposal report | 13-04-2018 |
| Week-6 | Definition: Cooperate work  Mission: ~~Front-end and back-end program will cooperate with each other smoothly; the visualisation of test data the development of main page (Navigate bar, menu, main page charts, exchange rate feed) should be completed.~~  Front-end:  Depth chart, value and volume chart, exchange rate feed and overlay are finished.  Some charts of menu main page are uncompleted (candle-stick chart, bubble chart). Some pages are uncompleted (private property page, login page, register page). There parts will be moved to week 8.  Back-end:  Some parts of API are uncompleted. Some data processers which used for providing front end charts’ data are uncompleted. | Deliverables: An executable work line between frontend and backend  Description: ~~Complete front-end first demo version and the development of main page template.~~ Complete front-end first demo version and the development some parts of main page template. Finish data processer for these front end chart. | 13-04-2018 |
| Week-7 | Definition: Integrate all the parts of project  Mission: Complete the integration of different components and subpages, the major functionalities of backend and frontend should be implemented. Integrate the completed works of front-end and back-end. Develop candle-stick chart, bubble chart and relative data processer.  Risk: Some parts may not work well, that will cost time to fix it. | Deliverables: A project can do the demo  Description: Complete the demo project and start to optimize the performance. Complete some rest parts of the main page. | 25-04-2018 |
| Week-8 | Definition:  Mission: All the development of features and functions complete, begin the project beta V1.0 testing process. Move to week 10. Develop private property page, login page, register page and relative data processer.  Risk: None | Deliverables:  Description: Testing the website performance by a series of processes. Complete some rest parts of the project. | 09-05-2018 |
| Week-9 | Definition: Progress Report  Mission: Progress Report Due. And finish the work that may be delayed. And integrate all the completed work of frond-end and back-end.  Risk: None | Deliverables: Progress Report  Description: Finish the progress report. And integrate components. | 11-05-2018 |
| Week-10 | Definition: Remote host testing  Mission: Remote host testing; Multiple-platform testing (Chrome browser, Safari browser, iOS and Android browsers). All the development of features and functions complete, begin the project beta V1.0 testing process.  Risk: If the remote host testing failed, that we will use local host to do the demo. | Deliverables: The project deployed on the real webserver.  Description: Complete the remote host deployment and multi-platform test. And testing the website performance by a series of processes. | 16-05-2018 |
| Week-11 | Definition: Documentation  Mission: Complete all the documentations written. (user manual, Git readme, project report)  Risk: None | Deliverables: Documentation  Description: Finish the written of relative documentations. | 23-05-2018 |
| Week-12 | Definition: Final Report  Mission: Final Report (thesis)  Risk: None | Deliverables: Final Report  Description: Prepare and finish the final report | 08-06-2018 |
| Week-13 | Definition: Final Presentation  Mission: Final Presentation  Risk: None | Deliverables: Final Presentation  Description: Prepare and finish final presentation | 22-06-2018 |

Summary