# CPS5301 Project Documentation

# Team 8

CONTENTS

[Requirement Engineering 3](#_Toc26532484)

[System Modeling 4](#_Toc26532485)

[Architectural Design 7](#_Toc26532486)

[Software Design and Development 8](#_Toc26532487)

[Database Design 9](#_Toc26532488)

[Project Management 10](#_Toc26532489)

[Software Testing 11](#_Toc26532490)

## Requirement Engineering

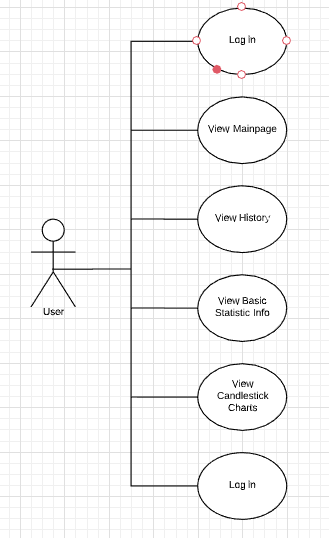
1. The system should daily update stock data from IEX.
2. The system should always keep the history stock data.
3. The stock data includes “date”, “open”, close”, “high”, “low”, “volume”.
4. The system should be able to generate statistical information automatically.
5. The statistical information includes max price, min price, average price, mean price.
6. The system should have restful APIs.
7. The system should have a graphical user interface.
8. The system should be able to generate charts based on the data automatically.

Product requirement: The system shall be available to all people 24/7.

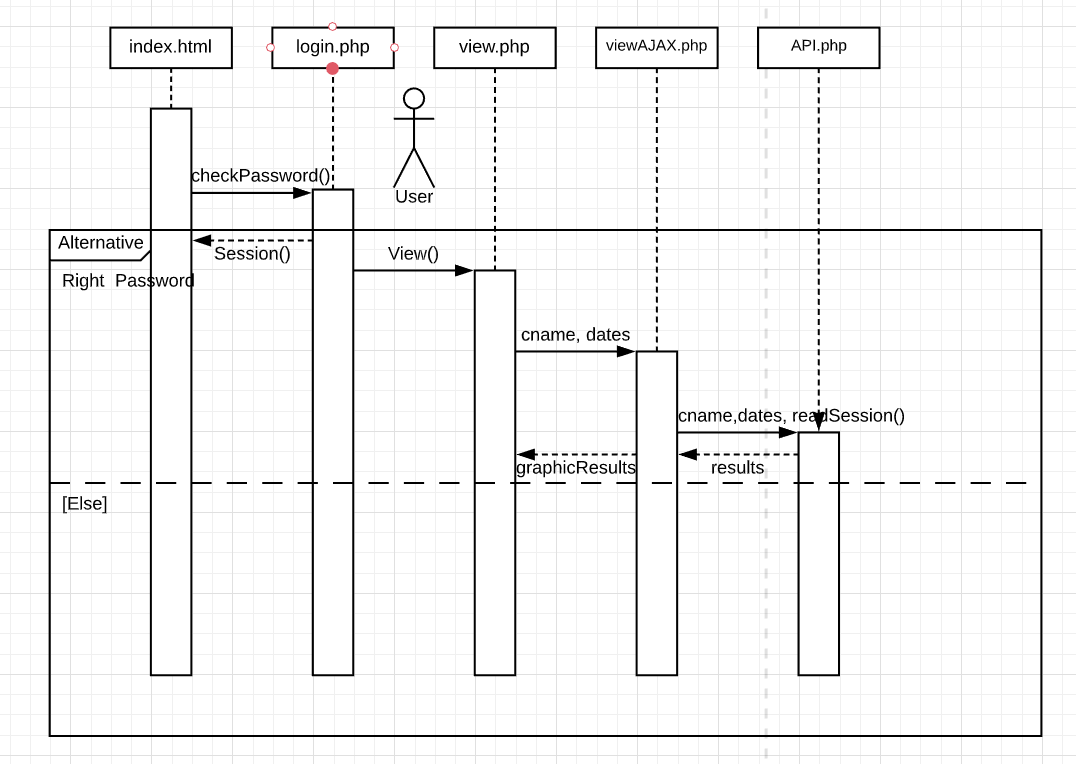
Organizational requirement: Users of the system shall authenticate themselves using their emails.

## System Modeling

|  |  |
| --- | --- |
| User | People who have privilege to access iex.yefangzhou1997.monster |
| Log in | The user should have the correct email and password to get the session for authentication. |
| View main page | The user can choose the functions they want in this page. |
| View history | The user can view the history records of the stocks. |
| View basic statistic information | The user can view the maximum, minimum, average, median of close during any period they want. |
| View candlestick charts | The user can view the candlestick charts of a stock during any period they want. |
| Log out | The user can destroy the session if they want. |

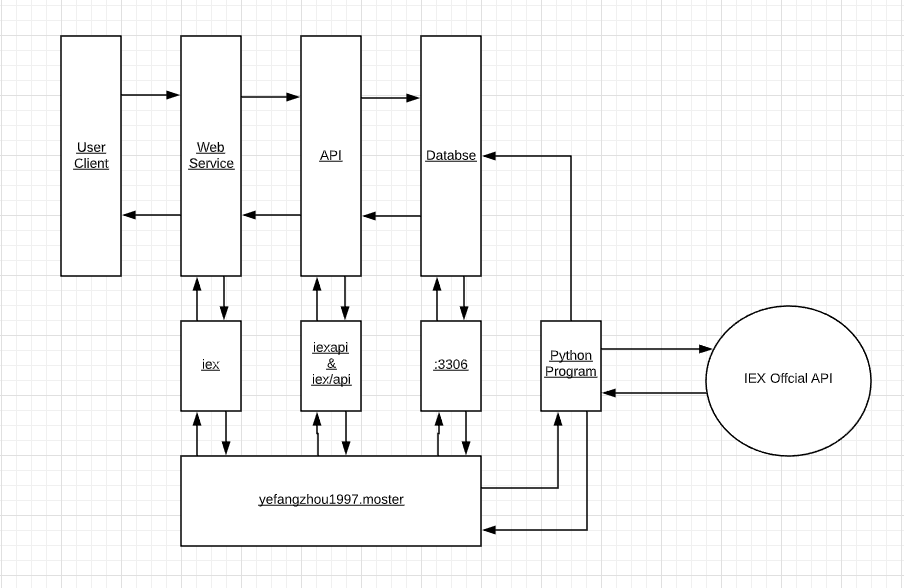


1. If the user inputs the correct password, then login.php will return a session.
2. The user can call view() to open view.php to input the query information.
3. As long as viewAJAX.php gets the info, it fetch corresponding result from API.php, which will be translated to a graph format latter by viewAJAX.php.



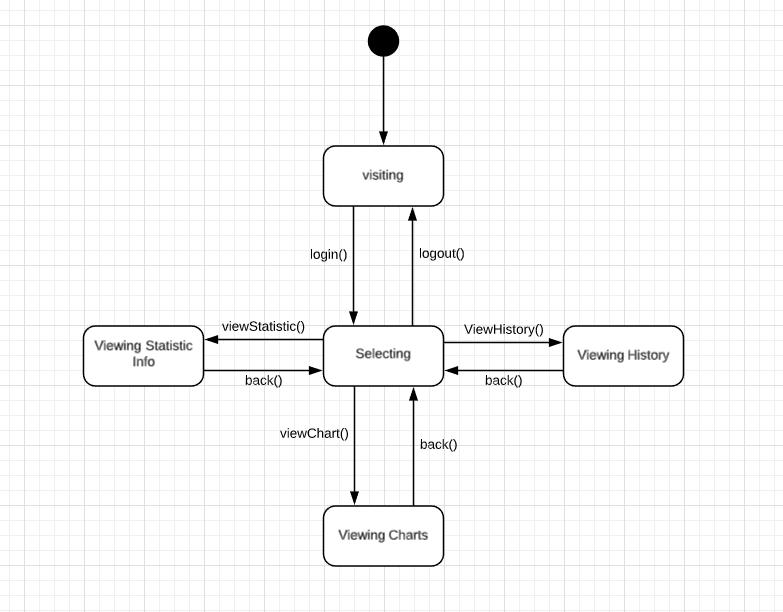
## Architectural Design

We designed a four-layer architecture to implement our project, which are User Client Layer, Web Service Layer, API Layer and Database Layer. User Client should be running on users’ personal computers, and the other three layers is running on our server whose domain name is “yefangzhou1997.monster”. Besides, Python programs are running on the server to fetch daily updated data from IEX Official API to Database Layer. Web Service Layer’s domain name is “iex.yefangzhou1997.monster”. API Layer’s domain names are “iexapi.yefangzhou1997.monster” and “iex.yefangzhou1997.monster/api”. Database Layer to is using the default port “:3306” to access from Localhost.



## Software Design and Development

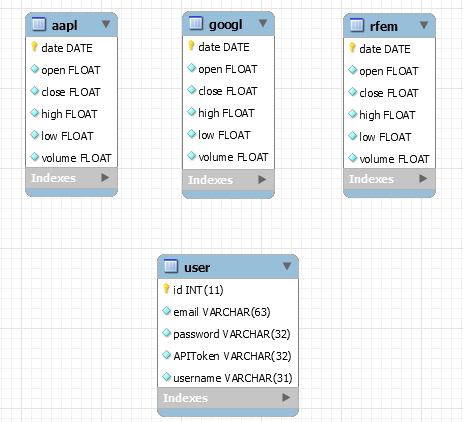
The system design is like a common website, the users need to log in to use the functions. In other words, the user should use a web browser to visit our website and log in to choose the functions he or she wanted to use on the main page. There are three main functions: viewing statistical information, viewing charts, and viewing history. As for the software development, Fangzhou would implement all the functions at the beginning and make sure his methods would work, then he would assign tasks to each team member. The benefit was that Fangzhou could solve all the problems the team members would mee and the team members could learn how to do their jobs very quickly.



## Database Design

We chose to use MySQL 5.5.62 as our database. PhpMyAdmin 4.9.0.1 and MySQL Workbench 8.0.18 are our database management tools. Totally, we have four tables. Three of them are for storing three compnaies’ data and the other one is for the using information.

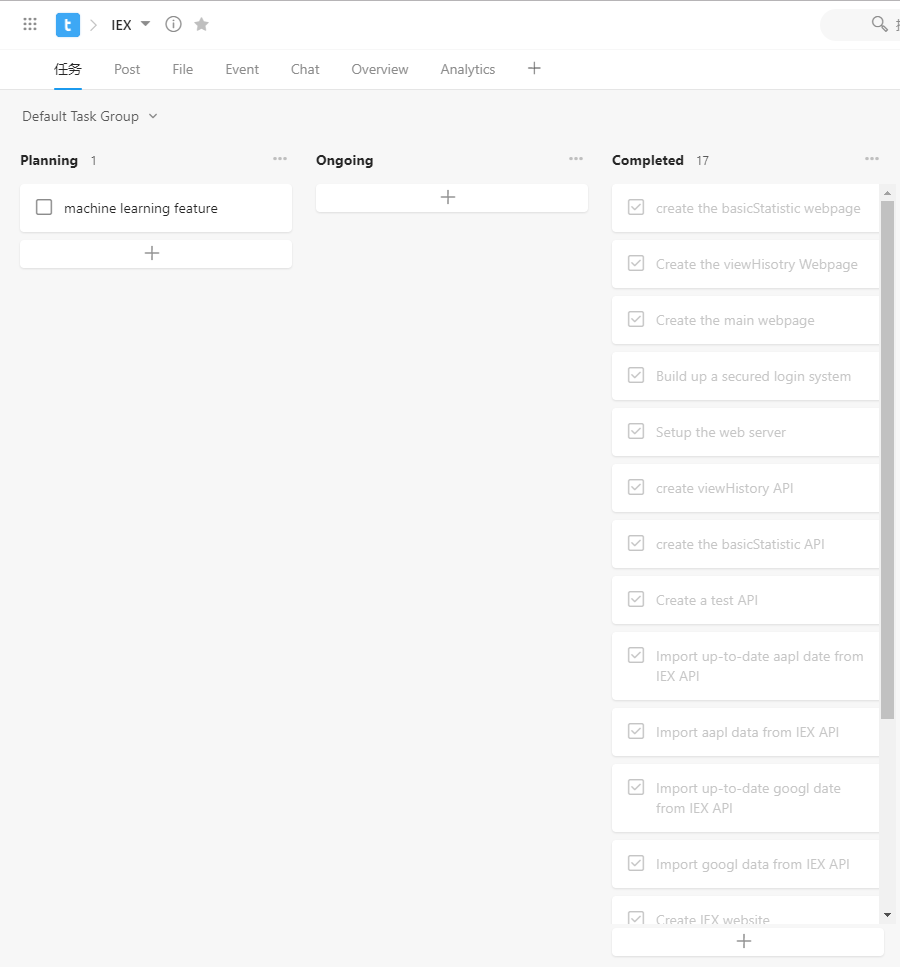
The company tables use date as their primary keys and have all the statistic information required. The user table has its own id as its primary key, which could be a better foreign key than email because matching emails would take much more time than matching integers. The passwords stored in database are encrypted by secured hash functions.

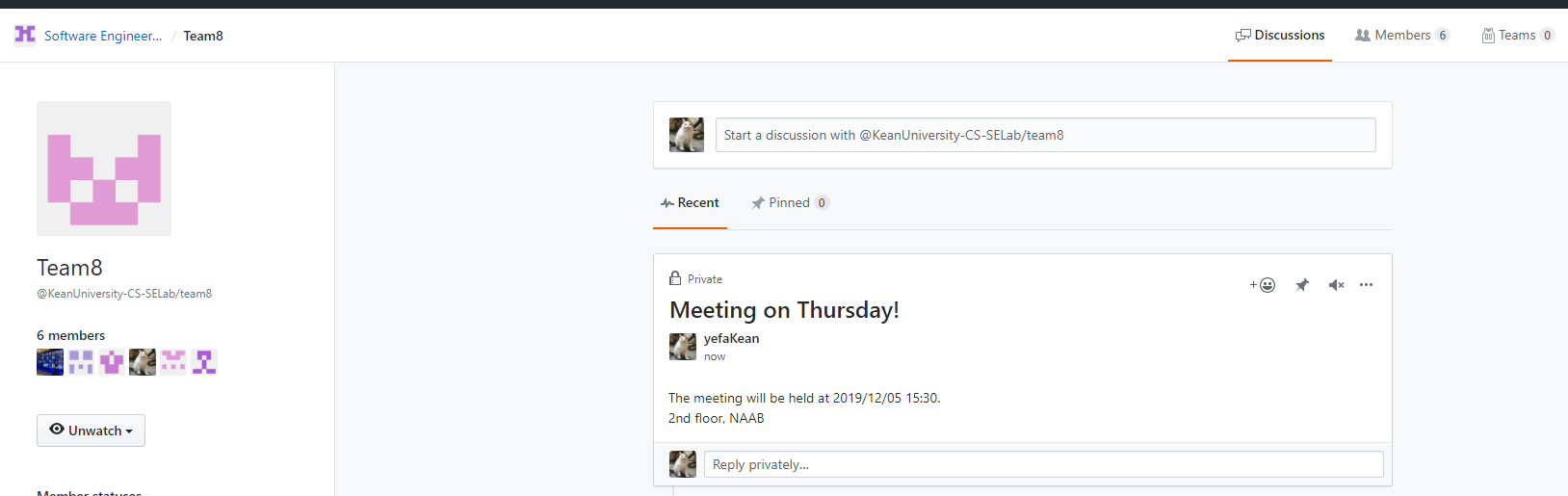


## Project Management

We use the agile methodology to manage our project. We usually have online scrums, but we did have several real face-to-face scrums.

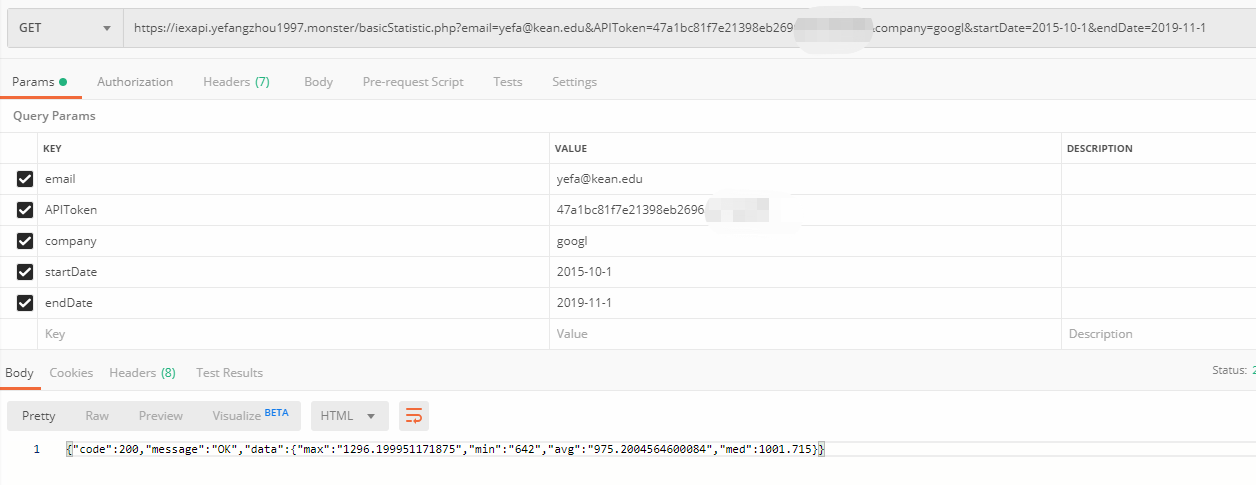
We use Teambition as our project management tool. We got the user stories from the project assignment. The leader would plan tasks for each team member according to Sprint Plan Meeting, and then the leader would update the tasks on Teambition. Team members would try to finish the tasks in time. We use GitHub Discussions to discuss and make scrum appointments. We will update our programs and documentations in GitHub Repositories.

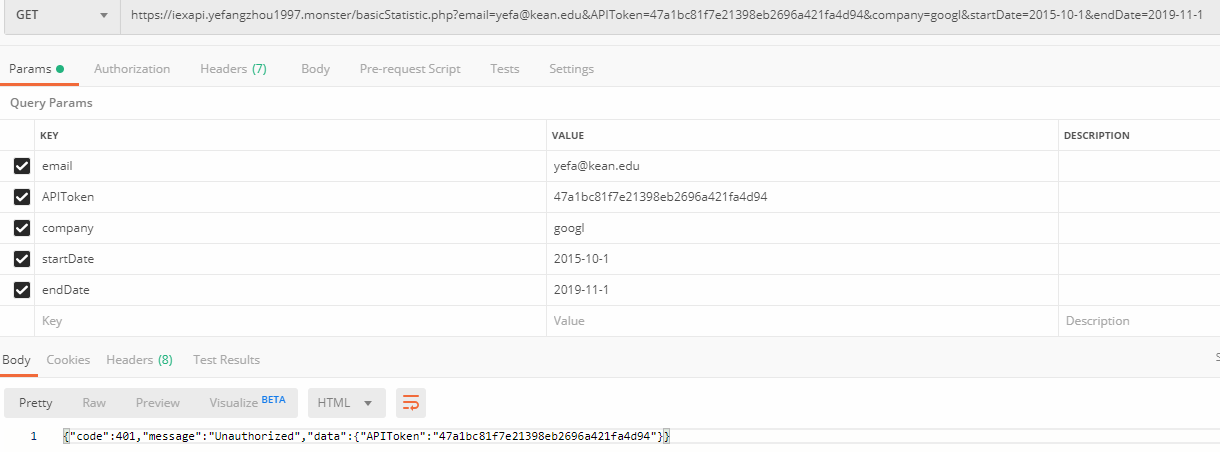




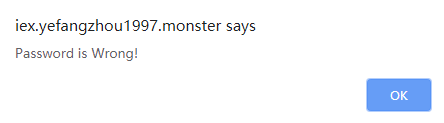
## Software Testing

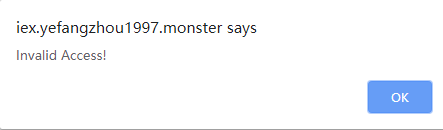
We use Postman to test our restful APIs. We prepared different cases, which include both right data sets and wrong data sets. The restful APIs pass all the tests.





For the authentication system, we use both correct password and wrong passwords to test. The tester also failed to enter the system without authentication.





The tester also prepared many date sets to test the functions. The system always return the correct results supposed to be generated.

