StockStockrs

Sprint 1 Planning Document
24th September 2016

Atul Aneja

Jalaleldeen Aref

Tarang Khanna

Wyatt Larkey

Joel Van Auken

Team 7

Index

Sprint Overview

For our initial sprint, we are going to focus on implementing the basics of each component of the web application. This will include setting up a usable front end, a barebones working RISK API server, our databases, and our machine learning module. By the end of this sprint, a user will be able to access the web application via a browser and view the basic layout of what the web application will become. The server will accept and deliver basic requests from the front end to the database and machine learning modules. Data will then be displayed on the web page from the server for users to see.

Scrum Master: Wyatt Larkey

Meeting Schedule: Tuesdays, Thursdays at 3:00pm

Risks/Challenges: This sprint we will begin implementing many different technologies and attempt to get them all to communicate so that in the future we will have a barebone functioning application to build off of. This means we may spend a lot of time researching our framework and determining what works best for our situation. Also, not every member of the team is familiar with the frameworks we are working with, so we will need to take time to learn about our framework in order to effectively implement it. Lastly our team has conflicting schedules that make physical meetings difficult to plan for. We will have to find a way to effectively communicate and make what meeting time we do have count.

Current Sprint Detail

User Story #1

As a user, I would like to access the web client as a guest.

Tasks:

1. Design home page of the web application.
   Estimated Time: 8 hours - Developer: Wyatt Larkey
2. Design information page of the web application.
   Estimated Time: 2 hours - Developer: Wyatt Larkey
3. Design browse page of the web application.
   Estimated Time: 4 hours - Developer: Wyatt Larkey
4. Allow web page to send requests to server.
   Estimated Time: 4 hours - Developer: Wyatt Larkey
5. Allow user to navigate to different web pages of the application.
   Estimated Time: 2 hours - Developer: Wyatt Larkey

Total Estimated Hours For Story: 20

Acceptance Criteria:

- Given I am a user, when I load the website then I expect to see the main page.
- Given I am a user, when I load the website then I expect to NOT be required to login.
- Given I am a user, when I am on the website then I expect to be able to navigate to all other pages of the website.
- Given I am a user, when I am on the website and I make a request for information then I expect to be able to get a response.

User Story #2

As a user, I would like to view real time stock prices in a graph

Tasks:

1. Use API calls on client side to get real time stock data on the stock requested by the user
   Estimated Time: 4 hours - Developer: Joel Van Auken
2. Graph the data points received from the API onto the user's screen
   Estimated Time: 10 hours - Developer: Jalaleldeen Aref
3. Display numerical data of certain stocks including current price and daily change.
   Estimated Time: 4 hours - Developer: Joel Van Auken
4. Display thumbnails of other stock graphs and data.
   Estimated Time: 2 hours - Developer: Joel Van Auken

Total Estimated Hours For Story: 20

Acceptance Criteria:

- Given I am on the client, when I choose a stock to view then I expect to see the real time, and a history of the current day's, price of the stock in a graphical form.
- Given I am on the client, when I look at a stock's data then I expect some numerical data, including the current price of the stock and how much it has changed in a given amount of time.

User Story #3

As a user, I would like to create an account

Tasks:

1. Design account page of the web application.
   Estimated Time: 8 hours - Developer: Wyatt Larkey
2. Create login page for the web application
   Estimated Time: 2 hours - Developer: Wyatt Larkey
3. Program server to handle new account creation requests
   Estimated Time: 8 hours - Developer: Joel Van Auken
4. Program server to store user information in a database
   Estimated Time: 2 hours - Developer: Joel Van Auken

Total Estimated Hours For Story: 20

Acceptance Criteria:

- Given I am a user, when I am not logged in then I expect there to be an option to login or create an account.
- Given I am a user, when I fill out the login form then I expect to be taken to my account page.
- Given I am a user, when I am logged in then I expect my account page to remember my basic information (name).

User Story #4

As a developer I would like to have an API server so that I can update a client easily.

Tasks:

1. Create an API endpoints in backend
   Estimated Time: 15 hours - Developer: Atul Aneja

2. Connect client to API endpoints
   Estimated Time: 10 hours - Developer: Atul Aneja

3. Develop API format and make sure the frontend and backend understand and use it correctly.
   Estimated Time: 5 hours - Developer: Atul Aneja

Total Estimated Hours For Story: 30

Acceptance Criteria:

- For this sprint, only a local call is necessary. Proper API hosting will be implemented later.
- Given I am a developer, when I call a RESTful endpoint then I expect to receive a JSON response.
- Given I am a developer, I expect the API service to update the web application.
- Given I am a developer, I expect that sending data to and from the server is easy to do and manageable.

User Story #5

As a developer, I would like easy access to database where data is stored.

Tasks:

1. Implement data tables to store the raw stock market data.
   Estimated Time: 5 hours - Developer: Jalaleldeen Aref
2. Implement data tables to store user information.
   Estimated Time: 5 hours - Developer: Jalaleldeen Aref
3. Implement data tables to store trained stock information.
   Estimated Time: 5 hours - Developer: Jalaleldeen Aref

Total Estimated Hours For Story: 15

Acceptance Criteria:

- Given I am a developer, when I need to store data for the web application, I expect to have a database table setup to handle and store it.
- Given I am a developer, when I access raw data from applicable stocks, I expect to find it stored neatly in a database where it can easily be used.
- Given I am a developer, when I access user data, I expect it to be securely stored in a database.

User Story #6

As a user, I would like to view a prediction of a stock's closing price for the current day

Tasks:

1. Doing data mining to get stock data to train machine learning on
   Estimated Time: 4 hours - Developer: Tarang Khanna
2. Ensure data and source validity, clean data for model
   Estimated Time: 4 hours - Developer: Tarang Khanna
3. Choose best features from data to train the machine learning model
   Estimated Time: 4 hours - Developer: Tarang Khanna
4. Create machine learning model to understand extracted data
   Estimated Time: 15 hours - Developer: Tarang Khanna
5. Predict given stock using machine learning model
   Estimated Time: 3 hours - Developer: Tarang Khanna
6. Client request to API for predicting stock closing price
   Estimated Time: 5 hours - Developer: Joel Van Auken
7. Send predicted data to client that requested it through the API
   Estimated Time: 5 hours - Developer: Joel Van Auken
8. Store predicted data for given stock to database
   Estimated Time: 5 hours - Developer: Jalaleldeen Aref

Total Estimated Hours For Story: 45

Acceptance Criteria:

- Given I am on the client, when I choose a stock to view then I should see expected that stocks closing stock price for the current day.
- Given I am a developer, when I train a dataset then I expect the trained data to show the predicted closing price for the day.

Remaining Backlog

* Lines with strikethroughs are stories that are either covered in the current sprint or are already completed.

Functional Requirements

As a user, I would like to ...

~~1. Create an account.~~

2. Customize my account.

~~3. Login to my account.~~

~~4. View my predicted stocks visually with a graph.~~

~~5. View my predicted stocks numerically with data.~~

6. Browse other stocks and add them to my stocks.

7. Access information describing data metrics and insight into the intricate nature of stock trading.

8. Request that new datasets be trained.

9. Contact the developers by email.

10. Adjust scope of predictions, whether it should look at short term or long term growth.

11. Practice trading stocks with virtual currency.

~~12. Be able to access application on iOS device.~~

~~13. Be able to access application on Android device.~~

14. See how much my virtual currency will grow with given stock predictions.

15. Receive notifications by email. (can be grouped with 16)

16. Receive notifications by sms or mobile notifications.

17. Add my stocks to a watch list for easy display.

18. Communicate with other users via a forum or comments section (time permitting).

19. Purchase stock shares with real money (time permitting).

As a developer I would like to ...

20. Train data sets manually.

21. Automate training data sets every day/hour/week.

22. Communicate with users via created accounts or email.

~~23. Easy access to database where trained data is stored.~~

24. Predict the general outcome, increasing or decreasing, of a trained stock.

25. Experiment with predicting the exact amount a stock will change.

26. Edit and compose information to be displayed on the site.

27. Train data sets for short term trading, long term trading, or in between.

28. Allow for predictions to accommodate non numeric variables, to allow human insight where an algorithm may fall short.

29. Manage the applications virtual currency for users.

30. Manage forum or comment posts made by users (time permitting).

Non-Functional Requirements

As developers, we would like to ...

~~31. create A front end that is responsive and elegant for all users.~~

~~32. create A database for stock data that is synchronized to the front end and updates seamlessly.~~

~~32b. create A database for user information that is secure as possible.~~

32c. create A database for virtual currency that is secure as possible.

33. create A stand-alone application for iOS (time permitting).

34. create A stand-alone application for Android (time permitting).

~~35. create Interfaces and menus for mobile users that make the application easy to use.~~

36. Interview a financial expert to gain insight on how to predict a stock.

~~37. Use machine learning algorithms efficiently.~~

38. Make predictions to a certain degree of accuracy.

~~39. Keep user data secure and private~~

40. Provide the best quality information regarding knowledge of stock trading.