

m.o.n.e.y.b.a.g.s

Project Site: <https://github.com/SkylerMalinowski/WBSF>



Table of Contents

Team Member Contribution Breakdown Matrix	2
Problem Domain	3
Overview	3
Problem Areas of Interest	4
In Broad Strokes	4
In Depth	5
Proposed Solution	6
Target Consumers	6
Success Criteria	6
Plan of Work	7
Leadership Positions	7
Repository Keeper	7
Front End Leader	8
Back End Leader	8
Build Tree	8
Team Breakdown	10
Team Size	11
Functional Features	11
Product Ownership	12
Front End Sub-team	12
Back End Sub-team	12

Team Member Contribution Breakdown Matrix

Portion	Jack Aquino	Raj Balaji	Gregory Leonberg	Skyler Malinowski	Vincent Taylor	Jonatan Yanovsky
Overview	60%	0%	20%	0%	0%	20%
Problem Area of Interest	0%	0%	0%	30%	70%	0%
Proposed Solution	40%	0%	0%	20%	0%	40%
Plan of Work	15%	10%	20%	20%	15%	20%
Product Ownership	0%	0%	60%	10%	30%	0%
Organization & Format	0%	0%	10%	40%	0%	50%
Proofreading & Editing	0%	0%	50%	50%	0%	0%
Cover Page	0%	0%	25%	65%	0%	10%

Problem Domain

There does not exist an accurate stock prediction service available to the public (for obvious reasons). Only the richest of the rich have access to such technology, as the ability to generate money without doing any work is something that the rich wouldn't share with the public. This makes it impractical for anyone who does not have millions of dollars to invest in hiring computer scientists to build a complex system that operates on a dynamic and chaotic trading environment (e.g. stock price fluctuations) to control their income on the stock market.

Overview

We will create an application that provides users with stock data, prediction models, relevant news, and suggestions as to how to handle the stock. This will include both a website as well as an Android app. The intention is such that even a novice user will be well informed when using our website that they can participate in the stock market with relative confidence.

On the front end, the app will have two different modes of operations dependent on the type of user: generic user mode; and admin/manager mode. For generic users, the app will have a variety of features regarding viewing stock data and trends. Given one or more ticker symbols, the user will be able to view a graph showing historic stock prices along with a predicted stock price for the following day. We will have multiple algorithms available for making predictions, and the user will also be able to change which algorithm is used to predict the prices of the stocks.

We also plan on making a distinction between free users and "paid" users: free users will have access to limited features, such as the number of graphs available at once, the types of stocks available for graphing, and the types of algorithms usable. Free users will have a dummy button that they can press to upgrade to the "paid" version: however, for the scope of this project, we will not be implementing any real monetary transactions.

An administrator account is used strictly for debugging and management of the application. As such, an admin account will not be able to track and plot stocks. Instead, the admin account controls which stocks are being tracked in the application. On the back end, our prediction algorithms will be running on a set amount of stocks. Since there are thousands of stocks in existence, it would be computationally intensive to apply the algorithms to every stock in existence. The admin's job will be to choose a subset of these stocks to track in the database.

Problem Areas of Interest

In Broad Strokes

Stock prediction relies on a multitude of different informational services, numbers alone do not accurately portray the future of the stock market; therefore, a specific issue we are trying to tackle is the user's ability to interpret these predictions. Since the user is able to make decisions independent of purely numeric calculations (i.e. personal knowledge, news articles, etc), the service requires straightforwardness. We face the issue of organizing the data in a way that is easy yet effective to observe and understand what is going on.

Graph customization presents another issue. One graph with a set amount of data may not accurately convey what the clientele truly wishes to see. Therefore the graph must contain the following: Axis controls, Color Customization, and toggle/overlay (see below for specifics on each problem); however, the graph is not limited to containing only these properties for customization.

Axis Controls equates to x and y variable manipulation. X variable manipulation includes extending date ranges to go further back (up to a certain point), or as close as a day (for prediction); Y variable manipulation includes extending the exactness of values. For instance, assume we wanted to know a stock's values change, but only from changes that are \$10 or greater. The opposite applies as well, a user may request to see a stock's changes that are as small as a cent.

Color Customization equates to controlling the coloring of the graph(s) this way someone may color code specific stocks as to assist in their own organization or the organization of others.

Toggle and Overlay these two rely on each other. Overlay implies being able to compare one or more stocks over the same (x and y axis) to each other. This allows the user to track stock trends or compare similar companies. The toggle portion shows which graphs are shown and which are not, granting the user to manipulate the graph as needed.

External information presentation refers to having external information display on our site. Ideally, our service will present relevant news articles based on which stock is displayed. This allows us to include information that is not purely numerical while still maintaining an organized user interface.

In Depth

To present information in a static fashion would inhibit most users from garnering the full effects of the program. To tackle this problem, backend development and front end development must utilize customizable tools and data output as to fit the needs of the user. For example, some people respond better to visual representations. A method to tackle aforementioned representation involves using graphs or charts to display information. Likewise, these graphs also need to be customizable (see above section for graph issues). With a multitude of organizational options the user can rely more on the data within the web service; thus, cutting down the likelihood of the user leaving our service to search for more information elsewhere.

External information will also act as an information hub for stock info. Aforementioned, users may seek information that is not purely numerical, so in order to alleviate the need to search elsewhere, our service will gather news and other sources into one area. This increases organization while allowing access to non-numeric information; thus, satisfying the users need to have different sources or type of information, without the need to visit different services.

X-axis manipulation will aid the user in controlling which date ranges of stocks they would like to examine. Thus, by extending or shrinking the range the user can generate a more or less accurate prediction based on how far back they wish to view. Similarly Y-axis manipulation may also improve accuracy. A smaller y-axis scale would cause less rounding and increase overall accuracy.

Color Customization allows users control over follow trends on different graphs via color alteration. It will also accommodate individuals who are color deficient in by allowing further controls over color options geared toward the color deficient regardless of what type or type gradient. This will increase the accessibility for all users and accommodate the color deficient.

Toggle and Overlay provide organization; however, they are convenient to the consumer. If the consumer can compare stocks on the same graph they won't have to copy, or struggle to match graphs with one another. Instead, the information is displayed in one convenient location. Additionally, allowing a toggle for which graphs are displayed, aids the user in changing the graph without the need to make a whole new graph. Toggle prevents the need of repetitive work and allows the user to focus on the data rather than remaking graphs.

Proposed Solution

We are going to make a semi-accurate prediction service that will allow for paying and not paying customers to gain access to our stock prediction algorithms. We will allow users to make better investing decisions based on the accuracy of our algorithms. Paid users will be able to see the most accurate prediction for a stock based on the relative accuracy of one algorithm versus another for a certain time period — algorithm success trends over a period of time. This will enable users to make more accurate predictions when trading stocks versus traditional methods like intuition.

Target Consumers

The target audience are people who are generally interested in the stock market as a whole; said free users are interested in the trend of stocks and how multiple stocks interact with each other to maximize profit and or minimize losses. Typical paid customers will consist of people who are much like the free users but would like capabilities beyond the the free users such that extensive analysis of stocks can be done without the minor inconveniences that free users have to put up with.

Success Criteria

To be specific, we are aiming for at least a 51% accuracy of our system. This will be a 1% increase over guessing or intuition, which we will assume to be a 50% chance of being right or wrong. If we can achieve higher accuracy, we will only let paying customers access the algorithms that yield those results, while limiting non-paying customers access to our 51%-of-the-time profit-generating algorithms.

A successful product will correctly track stock data, predict with 51% or higher accuracy, and have a working user interface that allows users to access this information via textual and visual data models. This user interface will compose of both a website and an Android application to accommodate user's preferred mode of use.

Plan of Work

We will be able to achieve the proposed goal via a systematic approach and by the abilities of our group members. They are ready to work extremely diligently for long periods of time so we have the labor requirements checked off. We already know the languages required to build the front and back ends of the system and we have plenty of coding experience under our belts across all our members, so we have the knowledge required to build the system. The tools we are using are free and open-source, so we will not require much if any financial support to complete our project. Furthermore we have already generated a production map for the systematic completing of the many moving parts of the project.

We have already planned a large amount of the project, as previously stated, and have organized our team into subteams for increased productivity, so we have already demonstrated our organizational capabilities. We communicate in class, in our meetings, and through Facebook messaging, so we will not have any problems communicating module/interface requirements and features between the subteams. Additionally, a Software Configuration Management tool has been implemented in order to facilitate development. With these skills and tools, we will be able to complete the project before the deadline.

Leadership Positions

The project group will be composed of two subgroups, each consisting of three members; The “Front End” subgroup and the “Back End” subgroup. Each subgroup will have a subgroup leader. These leaders will discuss the various tasks involved with interfacing the two halves of the project, serving as liaisons for their sub-team. Each team leader will serve as the configuration manager for their half of the project. Additionally, there will be an overall configuration manager for the combined halves.

Repository Keeper

The Repository Keeper holds commit and merge control over the master repository for the project. They will be responsible for maintaining the final code base that is the combination of each sub-team's code base.

Front End Leader

The Front End Leader is responsible for overseeing the distribution of work and communication between members of the front end sub-team. They are also responsible for overseeing and checking merge requests, as well as communicating with the Back End Leader to make sure both parts of the project share a common API. They are the first in command to make executive decisions as to the implementation details of their sub-team's specific tasks.

Back End Leader

The Back End Leader is responsible for overseeing the distribution of work and communication between members of the back end sub-team. They are also responsible for overseeing and checking merge requests, as well as communicating with the Front End Leader to make sure both parts of the project share a common API. They are the first in command to make executive decisions as to the implementation details of their sub-team's specific tasks.

Build Tree

Project Component	Description	Prerequisite Component	Estimated Deadline	Member
Account Database	A database for managing user account data.	Independent	2/24	Raj
API	Library that specifies how to manipulate and get information from the core of the application.	Independent	2/24	Vincent/ Greg
UI Artistic Design	User Interface for the website and Android application.	Independent	2/24	Jack/ Skyler
Stock Fetching/Organizing Functions	Functions that will fetch and organize the stock information so that it is easily translatable for the database.	Independent	2/24	Vincent
Stock Database	A database to store stock data.	Stock Fetching/Organizing Functions	3/17	Raj
Prediction Engine	Uses stock data to create prediction models.	Stock Database	3/17	Vincent

Accuracy Tracker	Calculate the current accuracy of a given prediction algorithm for a stock.	Prediction Engine	3/31	Jonatan
Broker Suggestion	Suggest to the user whether to buy/sell/hold the stock.	Accuracy Tracker	4/14	Jonatan
User Registration	Allows new users to register accounts.	Account Database; API	3/17	Skyler
Premium Activation	Allows users to upgrade their account to premium status.	Account Database; API	3/17	Gregory
Web Login	Login Page to allow for users to access the main web page.	Account Database; API	3/17	Skyler
Admin Control Panel	Allows the application administrator to control tracked stocks.	Stock Database; Web Login	3/17	Jack
Graphical Stock Organization	Create graphs to display stock information.	Stock Database; API	4/7	Gregory
Contextual News Fetching	Tracked stocks will incite relevant news articles to appear on the website/application.	Stock Database; API	4/7	Gregory
Website HTML	A web application that allows users to interface with all of the stock, prediction, accuracy, and suggestion data	UI; Web Login	3/24 (proto) 4/21 (final)	Gregory/ Skyler/ Jack
Android Application	An android application that uses our website but displays it in a scalable form-factor with ease-of-use touch controls.	UI; Web Login	4/21	All hands on deck when done with other things

Team Breakdown

Our Team — Strengths and Expertise

Front End
<u>Gregory Leonberg</u> (Subteam Leader) Expertise: C/C++, Java, Python, Matlab Strengths: Management, Presentation, Organization
<u>Jack Aquino</u> Expertise: C/C++, Java, Python, Matlab Strengths: Learning new languages
<u>Skyler Malinowski</u> (Repository Keeper) Expertise: C/C++, Matlab, Python, E-Basic, Java Strengths: Organization, Management, Debugging, Spell-Check/Proofreading
Back End
<u>Vincent Taylor</u> (Subteam Leader) Expertise: C++, Java, Python, LaTeX Strengths: Communication, Documentation, Presentation, Analysis/Design
<u>Jonatan Yanovsky</u> Expertise: C/C++, Java Strengths: Communication, Analysis & Design, Debugging, Spell-Checking/Proofing
<u>Raj Balaji</u> Expertise: C/C++, SQL, HTML Strengths: Learning new languages, Mathematical analysis

Team Size

Our team has six members, split into two groups of three. The problem we are trying to solve can be split into two clear parts — a front end UI and website, and a back end data processing engine and database. If a member leaves any subgroup, we will still have two members in that group, allowing us to still complete that group's task even without all the original members. Any fewer than three members per subgroup, and each subgroup would depend too much on the other. If we find that we have extra development time, we will use the remaining time to implement additional features.

Functional Features

- ❖ Allow users to view a stock's price chart, on which is overlaid our "prediction line"
- ❖ Allow users to select different prediction algorithms and stocks to predict future prices
- ❖ Allow registered users to maintain a list of "tracked" stocks
- ❖ Different functionality available to registered and free users
- ❖ Sign in to the website as a registered/full user to gain access to benefits
- ❖ Admin site to allow administrator to control available stocks
- ❖ Continuously gather price data about stocks that are on the program's stock watch-list
- ❖ Overlay multiple stocks on the same stock chart
- ❖ Provide a suggestion to the user as whether to buy/sell/hold the stock
- ❖ View algorithm accuracy / algorithm confidence rating in the stock prediction
- ❖ Allow users to register and log into accounts
- ❖ Semi-continuously run various prediction algorithms on stocks tracked by the website
- ❖ Customize graph color and layout

Product Ownership

The “Front End” leader is Gregory Leonberg.

The “Back End” leader is Vincent Taylor.

The “Repository Keeper” is Skyler Malinowski.

The “Front End” sub-team consists of Gregory Leonberg, Skyler Malinowski, and Jack Aquino.

The “Back End sub-team consists of Vincent Taylor, Jonatan Yanovsky, and Raj Balaji.

All members will be responsible developing an Android application to interface with the web tools, and will begin work on that as soon as they finish all their other responsibilities.

Front End Sub-team

The Front End Sub-team will create the web applications used by both generic users (the consumers) and the administrators of the app. They are responsible for:

- Designing a graphical user interface that works on both desktop and mobile (Skyler / Jack)
- Developing a system to fetch prediction and stock data (Gregory)
- Website HTML layout (Gregory, Skyler, Jack)
- Creating an account creation system (Skyler)
- Creating a user login system (Skyler)
- Creating an admin control system (Jack)
- Implementing a news-fetching system to provide relevant data to users (Gregory)
- Creating an account upgrade system (Gregory)

Back End Sub-team

The Back End Sub-team will create the database and algorithms responsible for storing various data from stocks and users, then use the stock data to predict the trend. They will create the API through which the Front End sub-team can fetch data and call functions for the web interface. They are responsible for:

- Creating stock prediction algorithms (Vincent)
- Creating a database for user, stock, and prediction data (Raj)
- Implementing customizable graphs (Jonatan)
- Implementing an auto-fetch feature for the database to update itself (Vincent)
- Creating an accuracy tracker for the prediction data (Jonatan)
- Handle organization of database information (Raj)
- Developing a “suggestion” algorithm to suggest whether to buy/sell/hold (Jonatan)