What the Stock?!

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

Group #7

Aquino, Jack

Balaji, Raj

Leonberg, Gregory

Lewandowski, Jake

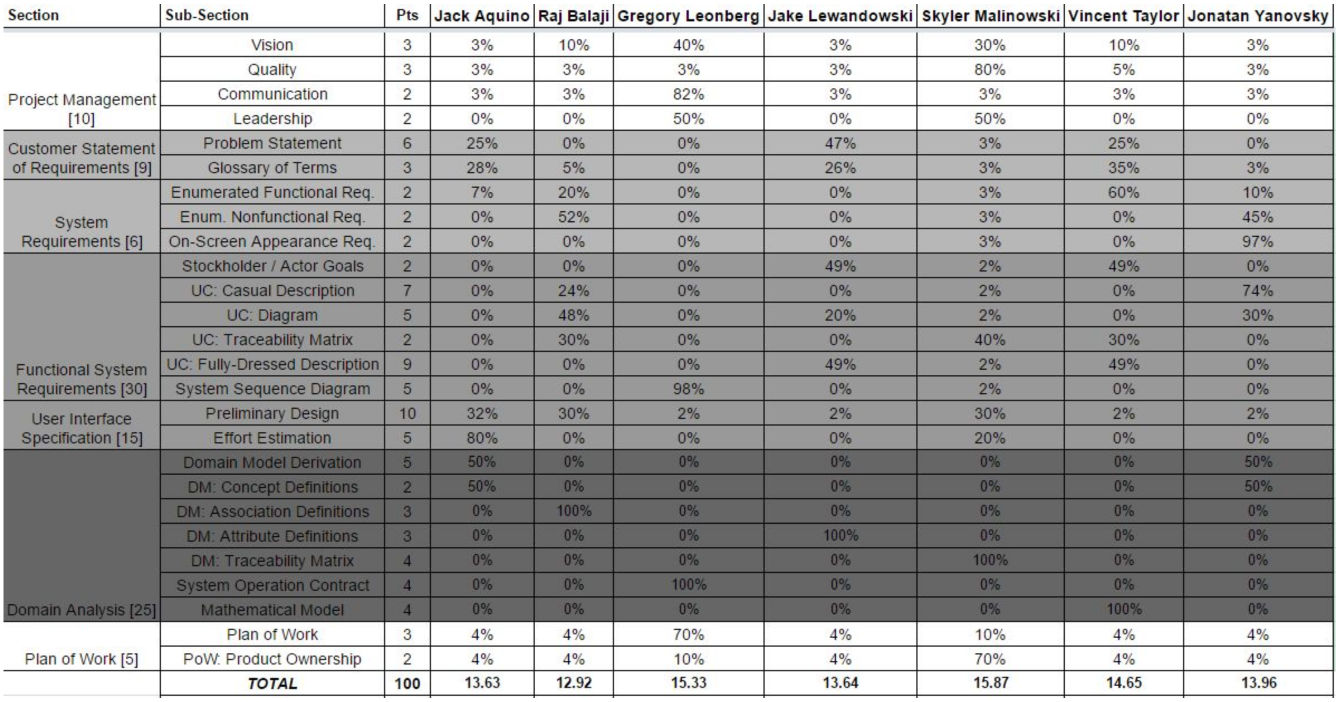
Malinowski, Skyler

Taylor, Vincent

Yanovsky, Jonatan

Team Member Contribution Breakdown

Contribution Matrix



Section	Sub-Section	Pts	Jack Aquino	Raj Balaji	Gregory Leonberg	Jake Lewandowski	Skyler Malinowski	Vincent Taylor	Jonatan Yanovsky
Project Management [10]	Vision	3	3%	10%	40%	3%	30%	10%	3%
	Quality	3	3%	3%	3%	3%	80%	5%	3%
	Communication	2	3%	3%	82%	3%	3%	3%	3%
	Leadership	2	0%	0%	50%	0%	50%	0%	0%
Customer Statement of Requirements [9]	Problem Statement	6	25%	0%	0%	47%	3%	25%	0%
	Glossary of Terms	3	28%	5%	0%	26%	3%	35%	3%
System Requirements [6]	Enumerated Functional Req.	2	7%	20%	0%	0%	3%	60%	10%
	Enum. Nonfunctional Req.	2	0%	52%	0%	0%	3%	0%	45%
	On-Screen Appearance Req.	2	0%	0%	0%	0%	3%	0%	97%
Functional System Requirements [30]	Stockholder / Actor Goals	2	0%	0%	0%	49%	2%	49%	0%
	UC: Casual Description	7	0%	24%	0%	0%	2%	0%	74%
	UC: Diagram	5	0%	48%	0%	20%	2%	0%	30%
	UC: Traceability Matrix	2	0%	30%	0%	0%	40%	30%	0%
	UC: Fully-Dressed Description	9	0%	0%	0%	49%	2%	49%	0%
	System Sequence Diagram	5	0%	0%	98%	0%	2%	0%	0%
User Interface Specification [15]	Preliminary Design	10	32%	30%	2%	2%	30%	2%	2%
	Effort Estimation	5	80%	0%	0%	0%	20%	0%	0%
Domain Analysis [25]	Domain Model Derivation	5	50%	0%	0%	0%	0%	0%	50%
	DM: Concept Definitions	2	50%	0%	0%	0%	0%	0%	50%
	DM: Association Definitions	3	0%	100%	0%	0%	0%	0%	0%
	DM: Attribute Definitions	3	0%	0%	0%	100%	0%	0%	0%
	DM: Traceability Matrix	4	0%	0%	0%	0%	100%	0%	0%
	System Operation Contract	4	0%	0%	100%	0%	0%	0%	0%
	Mathematical Model	4	0%	0%	0%	0%	0%	100%	0%
Plan of Work [5]	Plan of Work	3	4%	4%	70%	4%	10%	4%	4%
	PoW: Product Ownership	2	4%	4%	10%	4%	70%	4%	4%
	TOTAL	**100**	**13.63**	**12.92**	**15.33**	**13.64**	**15.87**	**14.65**	**13.96**

Contribution Pie-Chart

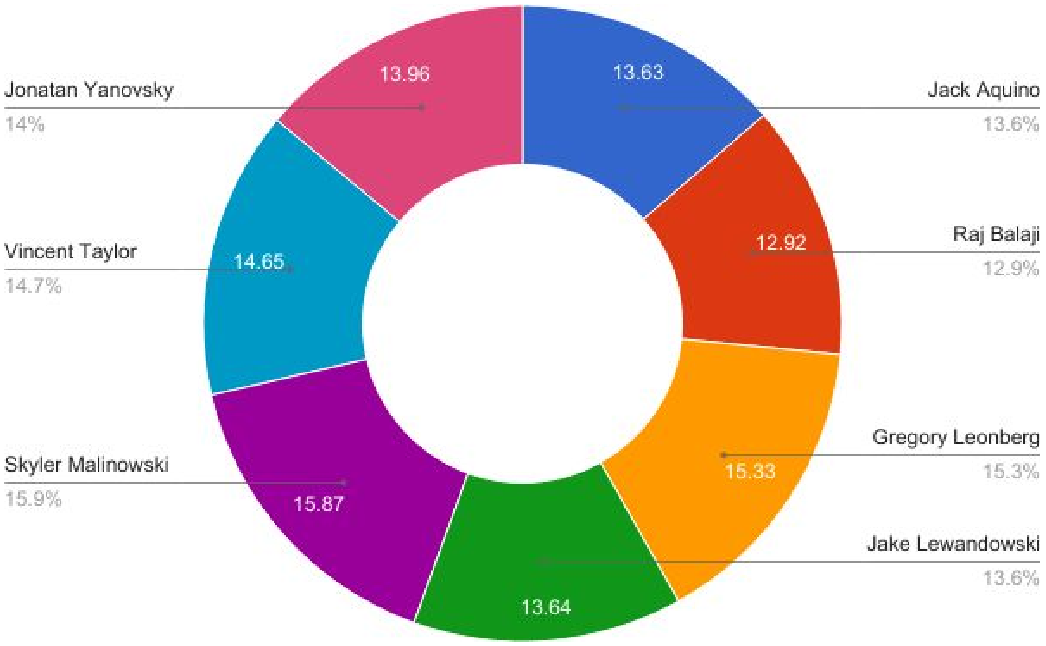


Table of Contents

Team Member Contribution Breakdown **1**

Contribution Matrix 1

Contribution Pie-Chart 1

Table of Contents **2**

Customer Statement of Requirements **4**

Problem Statement 4

Glossary of Terms 7

System Requirements **9**

Enumerated Functional Requirements 9

Enumerated Nonfunctional Requirements 10

On-Screen Appearance Requirements 12

Functional Requirements Specification **15**

Stakeholder 15

Actors And Goals 15

Use Cases 16

Casual Description 16

Use Case Diagram 17

Traceability Matrix 18

Fully-Dressed Description 19

UC-1 19

UC-2 19

UC-4 20

System Sequence Diagram 20

UC-1 20

UC-2 21

UC-4 22

User Interface Specification **23**

Preliminary Design 23

User Effort Estimation 25

Domain Analysis **26**

Domain Model 26

Concept Definitions 30

Association Definitions 32

Attribute Definitions 33
Traceability Matrix 34
System Operation Contract 35
UC-00 35
UC-1 36
UC-2 37
UC-4 38
Mathematical Model 39
Plan of Work 41
Product Ownership 44
References 46

Customer Statement of Requirements

Problem Statement

When someone thinks about the Stock Market, not many people can confidently say that they know how it works. Many people have the idea that the Stock Market like a high-risk, high-reward game where investors have the potential to make a large amount of money. While it is true that the rewards of the Stock Market are high, the risks are not as high as people believe them to be. With enough knowledge of how the stock market works, it is possible to accurately predict the market and consistently make money through stocks. Hence learning how to use the Stock Market properly is a profitable and worthwhile endeavor for future planning.

However, many people lack the necessary resources to learn about the Stock Market. Compulsory education systems merely gloss over the subject, and higher education does not explore the subject unless its for a specific degree. The goal of the “What the Stock?!” app is to provide an educational tool with which users can learn more about the stock market. The users which would benefit the most from this app are people with close to zero knowledge of the stock market, from students in high-school level economics courses to individuals who want to learn how to invest their money wisely. Our app would gradually introduce the end users to important concepts of the stock market, while consistently quizzing users on previous material in order to ensure that the user properly learned it.

When a user first logs into the application, they are greeted by a welcoming tutorial. This tutorial aims to help guide the user though our UI, while also teaching them about stocks. The tutorial will explain how to navigate through various stocks, as well as control data representation (this include graph manipulation and presentation). The tutorial will provide information on basic stock jargon, such as the definition of stocks, what a portfolio is, and the incentives for companies to sell stocks of their business, as well as a rudimentary understanding of how the stock prediction algorithms work. The tutorial is intended to bring all users — ones with close to zero knowledge through veteran investors who may have forgotten a thing or two about stocks — up to speed via proving ample background knowledge. By the time the user has completed it, they will have enough knowledge to explore various stock tickers through our app, as well as interpret the data that the graphs present.

Since the main idea of our app is to be a learning tool, we have also decided to implement a “review system” that will pop up while the user is on the app. The tutorial system is built to ensure that the user learns the material, and the review system will ensure that the user retains everything that they learned. At certain, randomly chosen intervals on the app, a review question will pop up that asks the user something that they have already learned from the tutorial. Ideally, this question will also pertain to the specific part of the app that they are looking at. For example, if the user is looking at a graph of a certain stock, the review question will ask them how to interpret the data in the graph.

Motivation

The stock market has a steep learning curve, there is wording, knowledge, intuition, and a plethora of skills that require constant practice; there lacks a tool set that can gently, and steadily guide a user into the stock market. The number of people that do not have the ability to research information or endlessly search for answers is incredibly high. People range from full-time students looking to set up a secure future, to elderly adults who want to earn extra spending money. Full-time students who are burdened with class often find a little time to heavily research secure stocks. Additionally, they cannot constantly monitor changes in the stock market, since they have classes and other activities to attend to. The elderly on the other hand may have the time, but not the skill. Searching for information may seem rudimentary, but to a retiree looking to invest in multiple places, the sheer amount of information alone acts as a deterrent. Not to mention the fact that the energy spent on compiling all of this information distracts the trader from their original goal of spending more money.

The massive amount of history and fields each stock dabbles into require research. Each stock has past figures, that to an inexperienced trader, look like Rorschach blobs. Aforementioned, the time for a range of traders is inaccessible or inconvenient. To even begin to understand the various readings on the dow jones is a trifling task; however, there are more options and more figures to command. Even if someone was to compile all of the information, required to understand basic stock trading; As their portfolio grow so does the amount of research and the need for diversification. Since most research points to diversifying a portfolio. The next logical step is to branch out and explore other areas of investment. Thus, the amount of research and time will grow, which mentioned earlier may pose impossible or inconvenient. This is purely the factor of research, after expanding a portfolio the trader then has to monitor and compile the information. This level of organization proves difficult when combining multiple sources together, and constantly switching between stocks to watch their development.

Even more, experienced traders may find the amount of information daunting. Compiling and organizing thousands of numbers, while also monitoring real time requires dedication and complete devotion. Once again the dilemma of time becomes and issue. With the required skill and determination the compilation of information is often error prone since it reigns from a multitude of different sources. The constant checking and verification prove tedious and infuriating, no matter the scale. Without a proper organization method or tools, individuals may never be able to expand and meet their goal of investing. The constant need for information as well as real-time changes proves difficult and disheartening. Without progression, diversification shrinks, inversely, the chance of an individual losing their money increases. Without the money to expand or the know how the user could find themselves in the red and losing more money than they bargained for.

The necessity for a tool that informs the user on the stock market's current state while guiding them through the entire trading process is too high to ignore. With limited time, money, and information, trading becomes difficult and stressful. The lack of organization leads to mismanagement and with enough data and human input, an error is bound to occur, such error may lead to loss, and overall deter the trader from engaging in trading. With an increasing amount of companies and data, the need for an organized, educational, quick, and straightforward system becomes necessary.

For a long time now, the stock market has been a successful tool in helping individuals earn money through investments in a variety of companies. The act of investing in stocks has made a lot of people a lot of money. Naturally, you would think that everybody would be involved in the stock market. Well, unfortunately, this is not the case.

According to CNN, only 35% households headed by someone with a high school diploma as the highest form of education actually have money in the stock market. This number goes up to 72% for households with members who have college degrees. Obviously, this is a concern since the stock market is a win-win for the investors and the companies, themselves. From the statistics, we can see that there is simply a lack of education in the act of investing and how the stock market works.

For newcomers, the act of purchasing a stock seems like a large and risky commitment. Without the proper research and basic knowledge, one would not know where to start. Most, if not all, people in this situation just turn their backs and stay away from the market as a whole. All of these people are missing out on potential big monetary benefits. A stock market is a scary place and not many people are willing to hold a newcomer's hand and teach them the ins and outs for free.

That is where we come in. We are creating a system for the average person, uneducated about the stock market. This system will be an easy way for these people to learn how the stock market works and eventually lead them to start purchasing their own stocks. This system will use an effective tutorial system in which the user is tough about the most basic aspects of the markets up to the more advanced strategies. Our vision is to be able to educate a person who has never even heard about the stock market, into one who can confidently invest and earn some easy cash.

The tutorial will not be the only feature of our system. Another attribute of the system will be real-time, effective stock prediction. We will include several stock prediction algorithms for more confidence in our predictions. Each user will be able to search any stock and view trends and predictions for said stock. The user will be able to pick stocks to track to prevent them from having to search each time. Along with our stock prediction algorithms, there will be a confidence rating for said algorithm. This will help our users be confident in their investment strategies.

As you can see, the main goal of this system will be to turn any person into an above average stock market investor and help them earn some extra cash. The tutorial combined with the post tutorial stock prediction algorithms will be effective means of bringing people up to speed on the ways of the stock market. We hope to increase the stock market literacy of the population by a sizeable amount. With this system, we believe this goal is easily achievable.

Glossary of Terms

Real-Time: Relating to a system in which input data is processed within milliseconds so that it is available virtually immediately as feedback.

Stock: A stock is an ownership share of a corporation/company.

Portfolio: A grouping of financial assets such as stocks, bonds and cash equivalents, as well as their funds counterparts, including mutual, exchange-traded and closed Funds: Are typically mainly in stock, with some amount of cash, which is generally quite small, as opposed to bonds, notes, or other securities

Dividend: A payment made by a corporation to its shareholders.

Trend: The general direction of a market or of the price of an asset.

Confidence Rating: Rating determined by the prediction on how “sure” it is of its prediction

Rise: Increase in stock price.

Fall: Fall in stock price.

YTD: (Year to Date) Refers to the period beginning the first day of the current calendar or fiscal year up to the current date.

Yield: The simple ratio of annual dividends divided by the share price.

Capital: Financial assets or the financial value of assets, such as cash (Usually initial).

Broker: A person who buys or sells an investment for you in exchange for a fee (a commission).

Margin: A margin account lets a person borrow money (take out a loan essentially) from a broker to purchase an investment. The difference between the amount of the loan, and the price of the securities, is called the margin.

Rally: A rapid increase in the general price level of the market or of the price of a stock.

Stock Symbol: A one-character to three-character, alphabetic root symbol, which represents a publically traded company on a stock exchange. Apple's stock symbol is AAPL.

Volume: The number of shares of stock traded during a particular time period, normally measured in average daily trading volume

System Requirements

Enumerated Functional Requirements

Weights (higher = greater priority)
3 = primary feature - required for success. 2 = required for proper functioning of primary features. 1 = secondary feature - for convenience

Identifier	Priority	Description
REQ-1	3	The system shall provide as accurate as it can prediction of the stocks for the selected company.
REQ-2	1	The system shall provide a response in a reasonable amount of time
REQ-3	1	The system shall be navigable with a minimal number of redirections
REQ-4	3	The system shall provide a tutorial mode in which the user can learn about the stock market and stocks in general.
REQ-5	1	The system shall remember the user’s preferences
REQ-6	2	The system shall have a security login to prevent unauthorized access
REQ-7	2	The system will provide both past and present stock history of the selected company
REQ-8	1	The system will allow users to follow certain stocks and notify them about changes to the stock.
REQ-9	1	The system will provide “feed” information from other news services
REQ-10	2	The system will display and explain confidence rating
REQ-11	2	The system will allow the user to control the graph axis and style
REQ-12	1	The system will allow for the user to replay the tutorial.
REQ-13	2	The system should allow the user to search for certain stocks
REQ-14	2	The system should allow the user to fetch stocks with a predicted rise
REQ-15	2	The system should refresh and apply the prediction algorithms periodically.

Enumerated Nonfunctional Requirements

F.U.R.P.S. =

Functionality – The F in the FURPS+ acronym represents the main product features that are familiar within the business domain of the solution being developed. The functional requirements can also be very technically oriented. Functional requirements that you may consider to be also architecturally significant system-wide functional requirements may include auditing, licensing, localization, mail, online help, printing, reporting, security, system management, or workflow

For our education tool, the main features would be

A tutorial mode about the application
Ability to fetch data from stock history websites
High-level Algorithms to predict stock data
Interactive graphs with stock and prediction data
Easy interactive UI that helps the user understand the data presented to them
Fetch specific stock information is inputted by the user

Usability – Usability includes looking at, capturing, and stating requirements based around user interface issues — things such as accessibility, interface aesthetics, and consistency within the user interface.

User Interface Accessibility – Make an online website that can be accessed at any time online as well as an app that lets the user access the program through an app.

Interface Aesthetics - Ability to change the color indexes for better viewing experience for color deficient users.

Distinctive Features - Naturally guide the user toward important information on the website with highlighted buttons/text.

Highly Customizable – Allow the user to sort and view the data at his leisure, such as viewing company information in columns/rows or in the form singular boxes across the screen.

Reliability – Reliability includes aspects such as availability, accuracy, and recoverability — for example, computations, or recoverability of the system from shutdown failure.

Availability - As our system will be run off of one of our own computers, availability will be only when the computer is on and functioning as a server for the website (in server mode). We will not be using a paid website / dedicated server provider.

Accuracy – The accuracy of the stock charts will be limited by the accuracy of our information source, which is the very accurate Yahoo Finance API. However, we cannot determine the performance of our algorithms as of now, which is why we aim for 51% accuracy at the minimum because at that point it becomes better than pure chance. Because the site is intended to be educational rather than advisory, the accuracy of our prediction algorithms is not a terribly important feature.

Recoverability – The program code is stored on Github, a Cloud-hosted Configuration Management Tool, so that it can be restored easily if our server crashes. The local server database for stock information can be dynamically reconstructed simply by running the program from a clean reset state. The local user information database will be routinely backed-up to a short-term storage medium such as flash memory by an automated script.

Performance – Performance involves things such as throughput of information through the system, system response time (which also relates to usability), recovery time, and start-up time.

Speed – Front end is basic Javascript, which will be lightweight. Back end is Python, and it will be written in such a way as to not push the server too much.

Efficiency – probably not maximum, since we are not using multithreading for the back end. Since we are not using the CPU too much multithreading is not necessary.

Resource consumption - would use a lot of system RAM, probably less CPU, but a lot of the network to retrieve data from a database.

Scalability – not gonna scale well as the backend and database program will be designed for a limited number of users and stocks to reduce complexity. Also, the program will not be well optimized in the limited amount of time we have to develop it.

Serviceability – Our code will be on Github so we can maintain, modify, adapt it easily by pushing new code to it, but then the most recent build must be downloaded onto the server and restarted, causing a crash to the existing build (not extensible well / poor installability). It is sustainable to last for a demo but probably really inefficient over a longer period of time. Testable immediately through GUIs being developed for website and algorithm views. It is modular, though, at each end of the program being relatively separate from each other. Javascript can be coded to allow localization of our program easily.

Repair Speed – We are running on GitHub so in the unfortunate case that an error does arise then we can always revert back to a state where the site worked. This allows us time to repair while decreasing the time in which the site is down. The speed of which the repair occurs also increases with the use of GitHub since we can see exactly what lines were changed between the previous version and the damaged one. This allows for immediate analysis and repair.

Installability – Since we are running off of a web-based server the installability of this software is high. Almost every modern browser supports Javascript and most preinstalled/popular browsers support the most current Javascript version. Without the need for a dedicated application like .exe, or .jar. The user should not have trouble using the service across a multitude of devices.

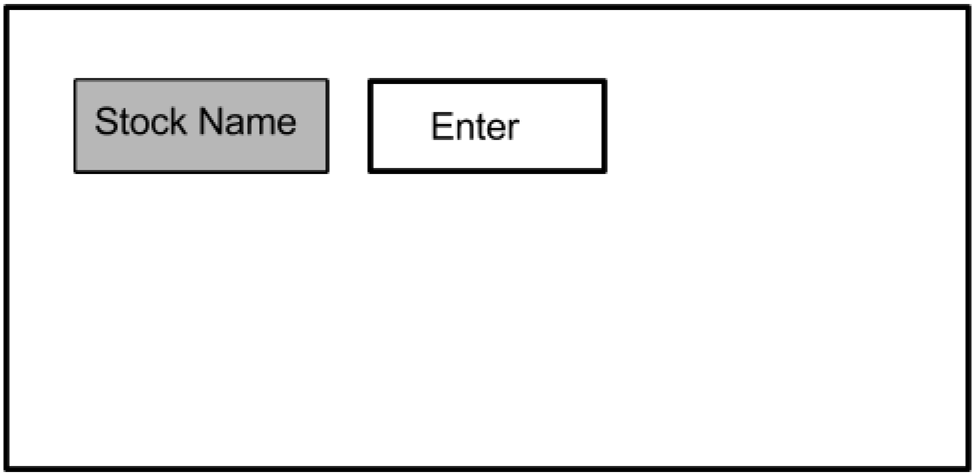
Testability- Our code does not revolve around a particular stock so as long as it works with one stock it will work for all others. Also, the Stock market is continually changing so most scenarios our program handles are tested with the use of a single stock.

Configurability- Our code is configurable since it fits around most stock scenarios. The use of graphs and multiple prediction techniques allows for the user to configure their output and presentation of data. Since there are multiple prediction techniques the system is configurable to fit different scenarios. This also reflects the system's Adaptability.

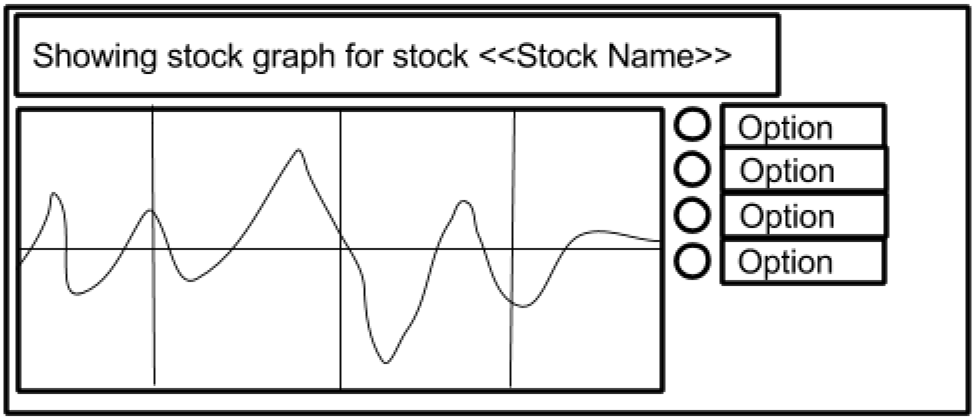
On-Screen Appearance Requirements

Our project will be composed of three vertical stacks: The tutorial website with quizzing components, the website with login, registration, news, and tickers, and the prediction and graphing site. The prediction and graphing portion will serve as a convenient way to debug our prediction algorithms as we code them. It will essentially be a very basic/empty web view of a box to enter a stock quote and an "Enter" button to get results. Doing this will cause another page to show up that will show a graph of the stock based on the gathered information from yahoo finance and our algorithms' predictions.

The webview of the enter “stock” screen:

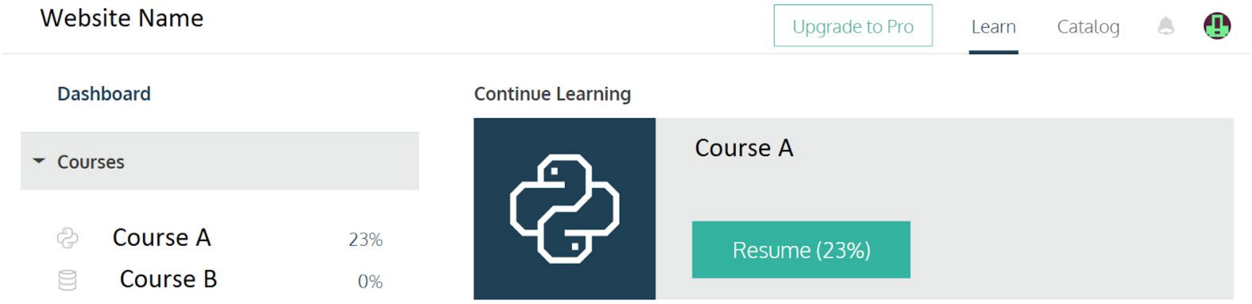


The stock graph:

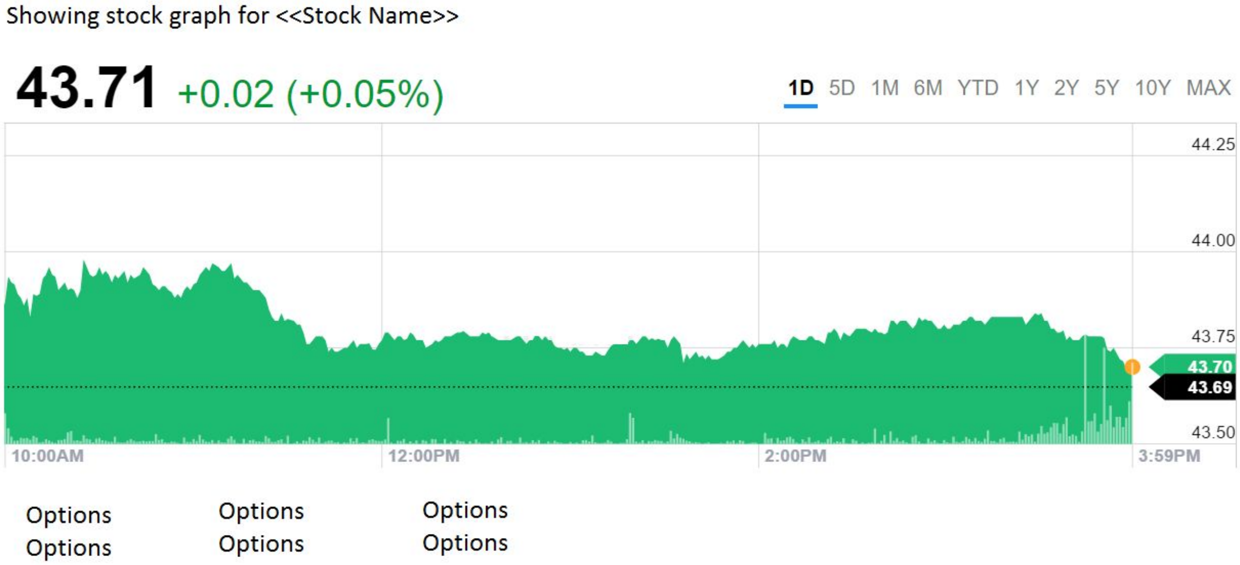


The website will be interactive and as such will have many buttons and menus, with pop-up boxes that can allow the user to enter an input to quiz questions (multiple choice, true/false). The quiz pop-up will disable other buttons on the screen and gray-out the rest of the screen that is not important to the quiz, and highlight elements that require the user’s attention. The website will allow users to see a “cleaner” version of the stock graph.

Website interface:



Website version of stock graph:



Functional Requirements Specification

Stakeholder

- Experienced investors
- Inexperienced investors
- Students
- Stock Advisors (Easily guide their clients through the process)
- Business Schools
- Business Classes
- Stock Brokers
- Stock Managers
- Teachers/Professors (Demonstration Purposes)
- International Traders
- International Finance Workers

Actors And Goals

Actor	Actor’s Goal	Use Case
Experienced Investor (initiating)	To use the stock prediction service to make effective investments.	UC-1
Inexperienced Investor (initiating)	To use the tutorial service to learn more about the ways of the stock market.	UC-2
Prediction Algorithms (participating)	To accurately predict stock trends in real time.	UC-1
Tutorial (participating)	To teach beginners about investing and the stock market.	UC-2
User Tracked Companies (participating)	To display companies chosen by the user to be displayed on the home page.	UC-3
Experienced Investor (Initiating)/News	To display all relevant information pertaining to a certain stock	UC-4

Feed (Participating)		
Prediction Algorithm /Graph Overlay (Participating)	To change the method of prediction or change the overlay of the graph from its current display	UC-5

Use Cases

Casual Description

Use Case : Casual Text Description :: Corresponding requirements.

UC-1 : A user may want to use the stock prediction service. In this case, the user may also be interested in the accuracy of our algorithms in order to secure greater profits.
:: REQ-1, REQ-2, REQ-6, REQ-10, REQ-13, REQ-14, REQ-15

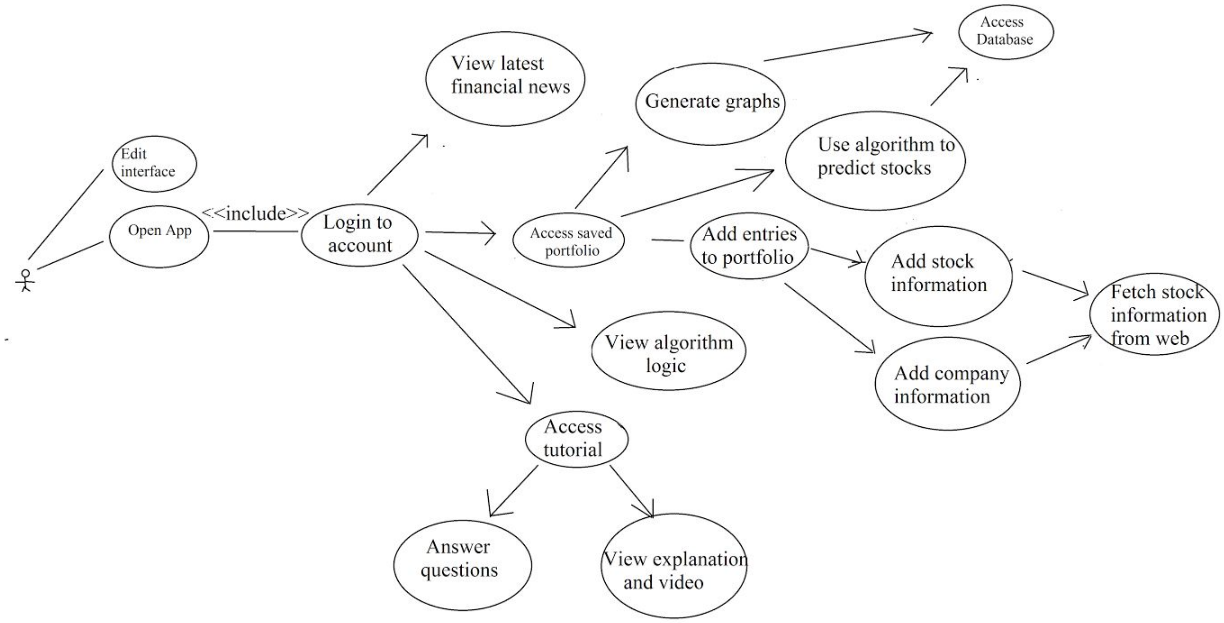
UC-2 : A user may want to use our tutorial service to learn about the stock market. They may want to replay the tutorial again to test themselves, or quiz themselves on certain topics.
:: REQ-2, REQ-3, REQ-4, REQ-12

UC-3: A user may want to have a specific list of companies kept track of so that each time they enter the site it loads that list for them.
:: REQ-2, REQ-3, REQ-5, REQ-6, REQ-8

UC-4: A user may want all of the relevant info about how the prediction model works as well as news about the company so as to make a more informed decision.
:: REQ-1, REQ-2, REQ-3, REQ-7, REQ-9, REQ-10

UC-5: A user may want his data presented in a different style, such as changing the colors or not graphing predictions, for viewing aesthetics or for colorblind users.
:: REQ-2, REQ-3, REQ-5, REQ-11

Use Case Diagram



Traceability Matrix

REQ’Ts	**Pts**	**UC-1**	**UC-2**	**UC-3**	**UC-4**	**UC-5**
REQ-1	**3**	X			X	
REQ-2	**1**	X	X	X	X	X
REQ-3	**1**		X		X	X
REQ-4	**3**		X			
REQ-5	**1**			X		X
REQ-6	**2**	X		X		
REQ-7	**2**				X	
REQ-8	**1**			X		
REQ-9	**1**				X	
REQ-10	**2**	X			X	
REQ-11	**2**					X
REQ-12	**1**		X			
REQ-13	**2**	X				
REQ-14	**2**	X				
REQ-15	**2**	X			X	
Total Pts		***14***	***6***	***5***	***10***	***5***

Fully-Dressed Description

UC-1

The user who looks to the stock prediction service to aid their stock trading decisions may find comfort or usability with the accuracy of said predictions. Users will have a confidence rating displayed on their screen. This rating reflects the "confidence" of the algorithm in use which is determined by the accuracy of the system. Periodically, the system will update and change the confidence rating. Additionally, if the user changes the type of prediction used the rating may change with it. The user will be able to apply this process to any number of stocks. For example, if the user looks for GOOG, they can apply any prediction algorithm to the stock and have a confidence value. Likewise the user's list of stocks will also refresh periodically or with algorithm changes and display/update confidence values.

1) Investor initiates the prediction service for a stock
2) Stock Fetches information and initiates the prediction algorithm
3) Prediction algorithm generates confidence value
4) Confidence value is then displayed on the graph
5) Periodically the system will refresh and reset to 2)
6) The process will reset to 2) when the user switches stocks or prediction techniques

UC-2

At the user's discretion they can activate a tutorial service that will aid the user in the system's navigation and help them understand the stock market. The tutorial will be available to replay at anytime to offer aid; However, the user may quiz themselves on the additional topics to either refresh or increase their knowledge on certain subject areas. In addition, the user will be able to re-take quizzes at anytime in order to improve their score or knowledge.

1) User initiates tutorial (2) or quiz (4)
2) The tutorial will commence and walk the user through the desired part of the interface
3) The user will follow the tutorial or cancel at anytime
4) The quiz will commence and the user will be prompted to answer questions
5) The user may end the quiz or follow it to completion
6) After the completion of the quiz the "grade" will be displayed and the system will prompt the user to retake the quiz if necessary. Otherwise, the tutorial module will be marked as completed.

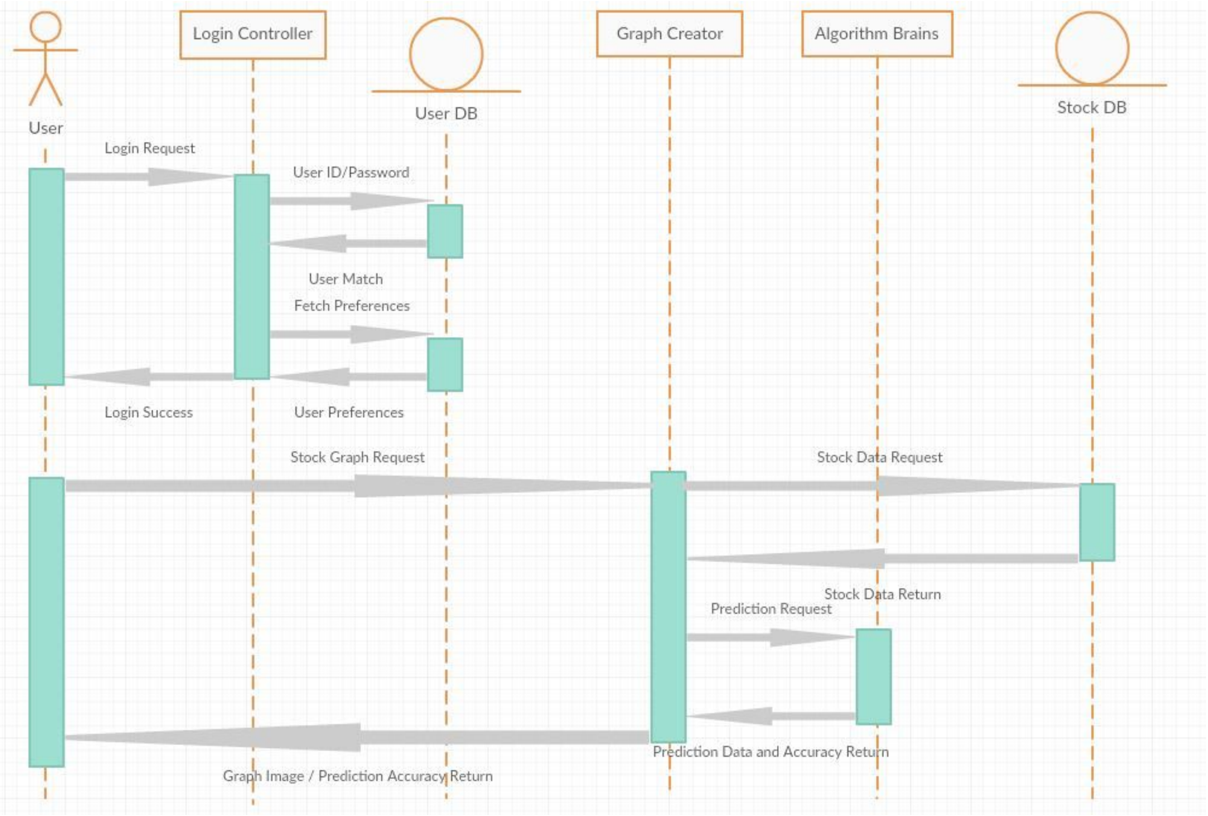
UC-4

The user is displayed information about any stock that they search. When viewing the prediction graph, the user is informed about past trends that lead to the current prediction. On top of this, the system will display relevant news on the chosen stock. The news will also play a big role in the stock prediction graph, and the system will describe why or how the given news will affect the stock. This news feed will provide outside information about the stock info. Additionally, the prediction algorithm will explain its assumption of whether or not it is a positive or a negative trend. All of this information will allow the user to have a better understanding of the stock.

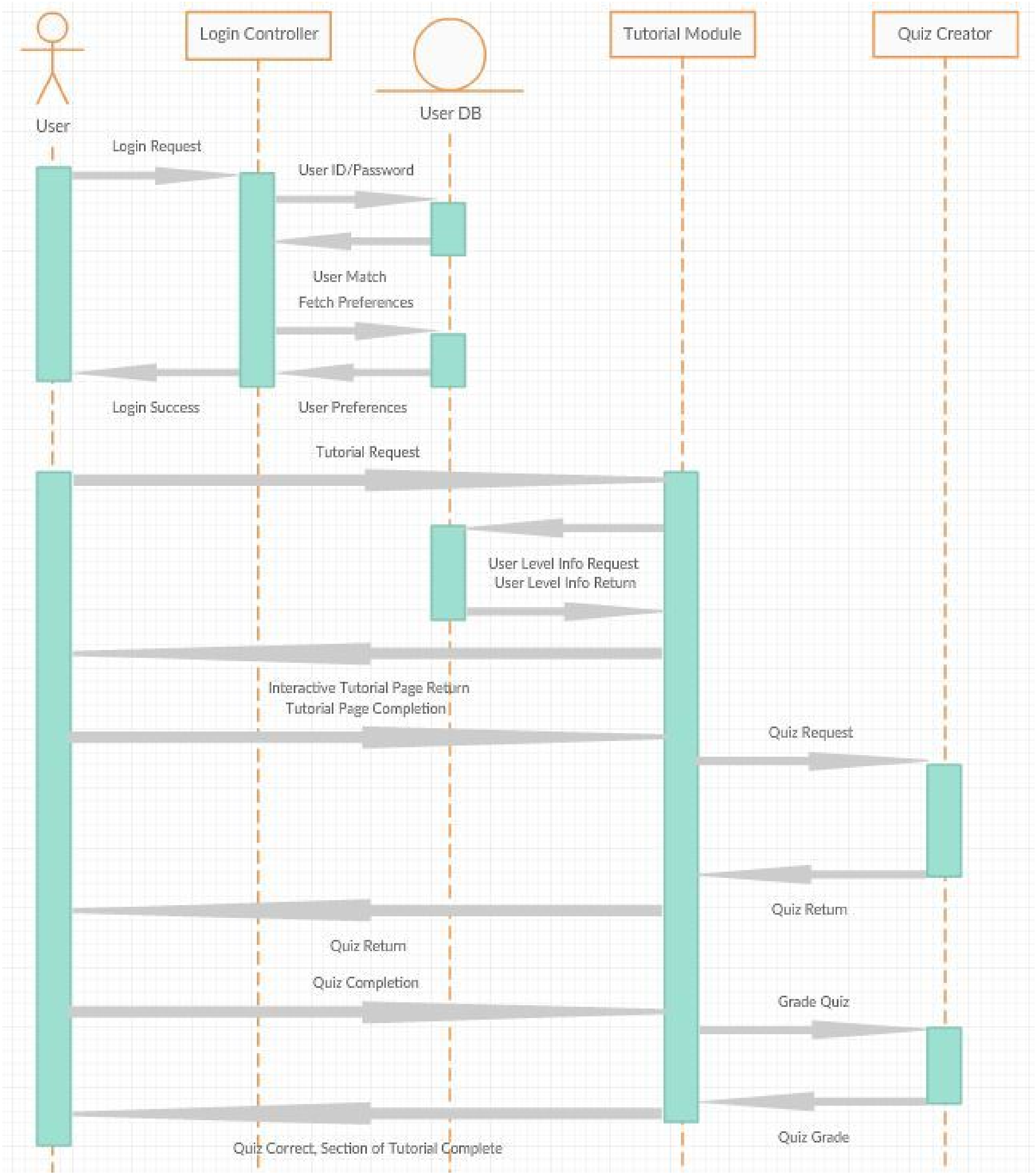
1) The user will ask for more information on a given stock
2) The system will then display a news feed correlating to the current stock
3) The system will also explain more information about the stock
4) The system will provide an explanation of the algorithm's decision
5) The system will update the prediction, explanation, and newsfeed periodically.

System Sequence Diagram

UC-1

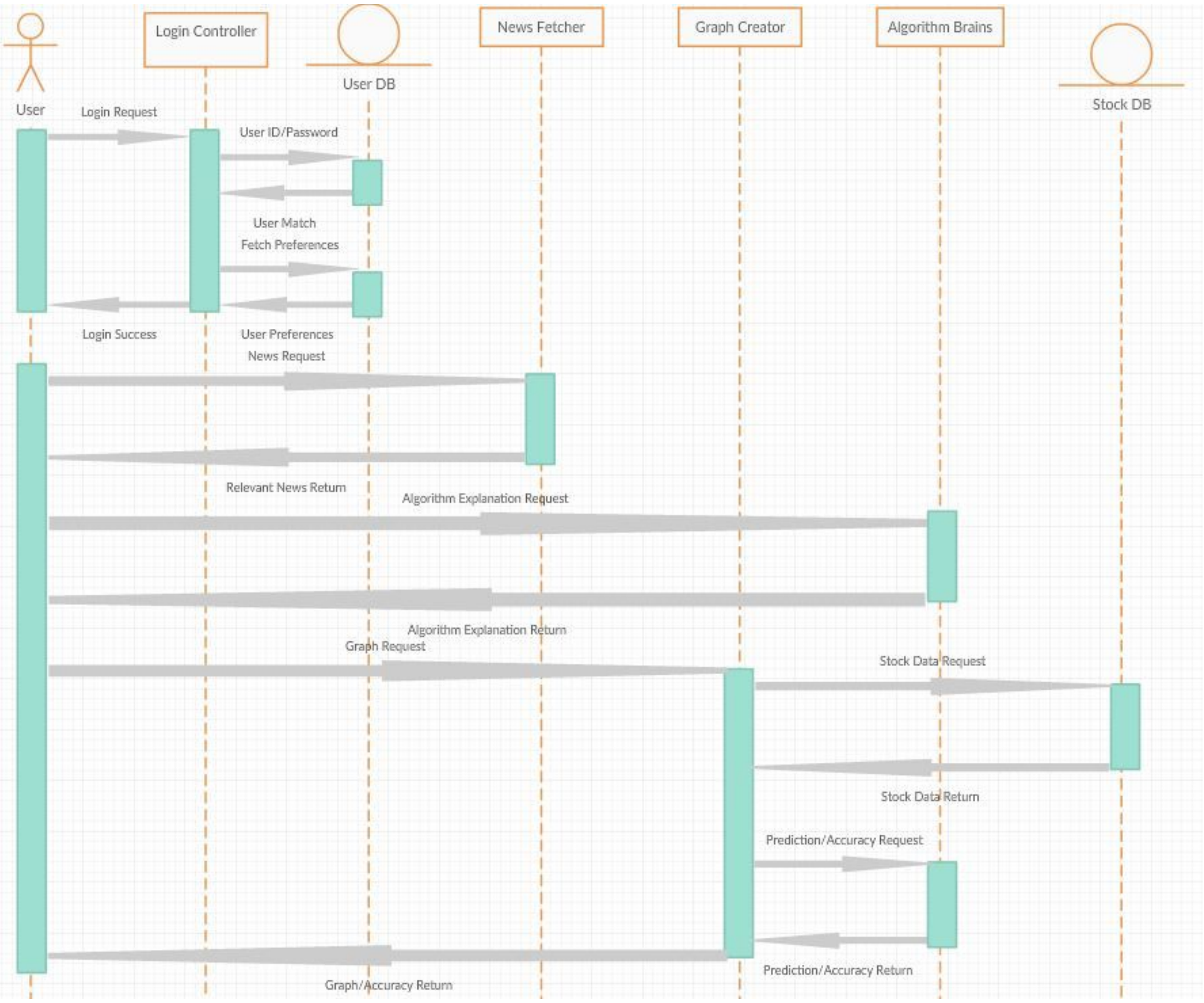


UC-2



User
Login Controller
User DB
Tutorial Module
Quiz Creator
Login Request
User ID/Password
User Match
Fetch Preferences
Login Success
User Preferences
Tutorial Request
User Level Info Request
User Level Info Return
Interactive Tutorial Page Return
Tutorial Page Completion
Quiz Request
Quiz Return
Quiz Return
Quiz Completion
Grade Quiz
Quiz Grade
Quiz Correct, Section of Tutorial Complete

UC-4

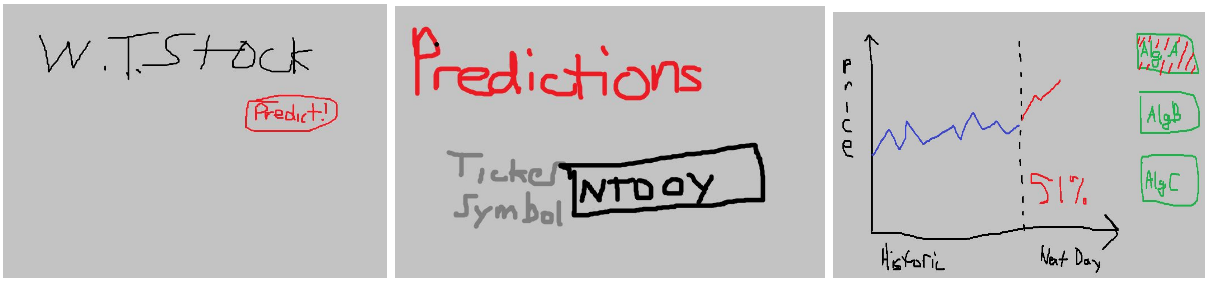

User
Login Controller
User DB
News Fetcher
Graph Creator
Algorithm Brains
Stock DB
Login Request
User ID/Password
User Match
Fetch Preferences
Login Success
User Preferences
News Request
Relevant News Return
Algorithm Explanation Request
Algorithm Explanation Return
Graph Request
Stock Data Request
Stock Data Return
Prediction/Accuracy Request
Prediction/Accuracy Return
Graph/Accuracy Return

User Interface Specification

Preliminary Design

UC-1: Stock Prediction

Using our app to obtain stock prediction data. In this UC, the user will input a stock ticker symbol, and the output will be a graph showing the historical data of that stock's price, plus a predicted value for the following day's price.



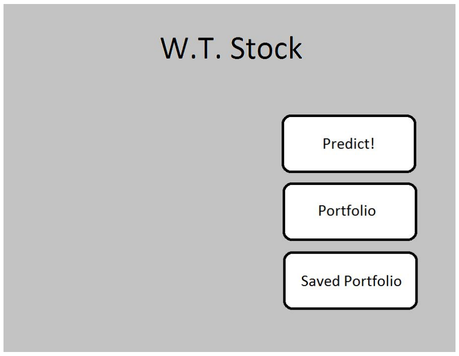
The first panel depicts our start screen. In order to initiate this use case, the user clicks on the button marked "Predict" in red on the first panel. They will be redirected to another interface, which will ask them for a ticker symbol to apply our algorithms too. In the box (marked in black in the middle panel), the user will type the ticker symbol of the stock they wish to examine. Afterwards, they will be sent to a third screen, which will show the prediction graph. The third panel has the historic price of the stock (marked in blue), as well as a predicted price (marked in red), along with a prediction confidence level, also marked in red, which is a comment on how accurate that particular algorithm has been with this stock. On the side (in green), there are our buttons for choosing the algorithm used in the predictions. The user can click on a button to change the active algorithm, which will change the red portions of the graph. The button corresponding to the currently-selected algorithm will be highlighted.

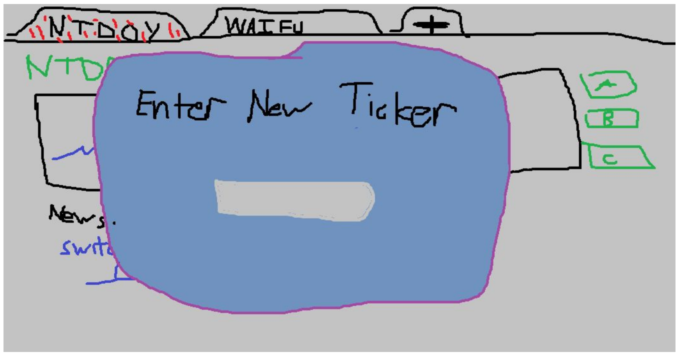
UC-2: Guided Tutorial of Stocks

In this UC, the user will be able to play through a guided tutorial on the stock market. This tutorial will take place as a lesson, explaining the specific topics, alongside multiple choice quizzes.

UC-3: Saved Portfolio

In this UC, upon re-entrance into the website after logging in (if user credentials have expired), their stock portfolio from their previous session will reappear just as they left it. Stocks can be added and removed from the saved stock portfolio.

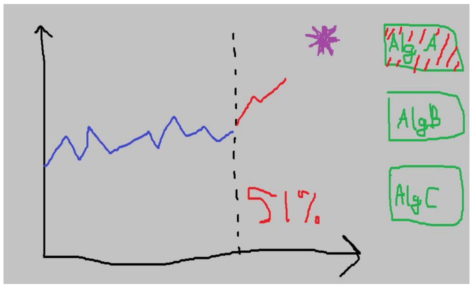


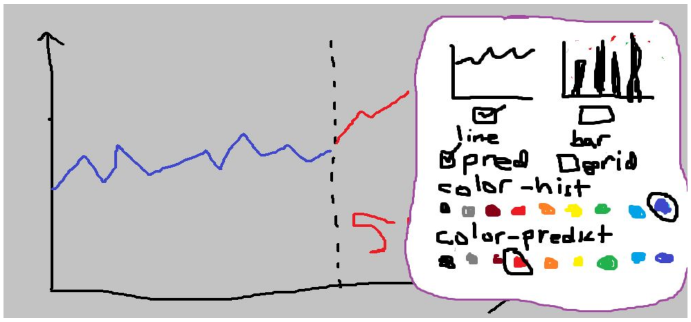


From the main menu, the users clicks on the “Saved Portfolio” button. The user is then redirected to their own saved portfolio page, which has a series of tabs at the top, one for each stock that they have in their saved portfolio.They can add stocks by clicking on the “+” tab at the top, and remove stocks by hovering over that tab and clicking on an “x” icon on it, much like opening and closing tabs in a web browser.

UC-4: Aesthetic needs

For this Use Case, the user will want to alter the way that the data is presented to them. To meet this need, we will add the options of viewing the data as a line graph or bar graph (with possibly more graph types to be determined). We will also add a tick box to toggle whether or not to show the prediction data. As a final touch, we will add features that allow the user to alter the colors used on the graph, as well as whether or not to include grid lines in the graph.





On our graphs, there will be a cog/gear icon (the symbol for system settings in most phones) that the user may click on. This will open a small popup box that shows the different options that this particular graph can be viewed with. Tick boxes will be used to show which options are currently selected, as well as for changing which options are used. Once the user has selected options that are suitable, the user can click outside of the box (or on a little X symbol that I forgot to draw) to close the option box.

User Effort Estimation

For each Use Case, the following clicks are required:

UC-1: Stock Prediction

Navigation (3 clicks) or (2 clicks) or (2 clicks 1 key) or (1 click 1 key)

- From the main menu, click on “Prediction” button.
- Click on the text entry box, to enable it (optional: have it be enabled by default, removing one click).
- (After Data Entry) click OK or press “Enter” to move onto the next page.

Data Entry (6 keys, 0 clicks) or (6 keys, 1+ clicks)

- Enter stock ticker symbol (anywhere between 2-6 characters, depending on stock. For purposes of analysis, we’ll assume 6 keys).
- Choose the algorithm used for predictions: if not using the default algorithm, +1 click.

UC-2: Tutorial Service

Navigation

- From the main menu, click on “Tutorial” button.
- Click anywhere on screen / “next” button to progress through the tutorial lesson.
- Click the “previous” button to see previous text in the tutorial.
-

Data Entry

- Click on a button to choose the section of the tutorial to play.
- Click on a button choosing which quiz to play.
- Click on an answer to a quiz.

UC-3: Saved Portfolio

Navigation (1 clicks) or (1+ clicks and/or 4+ keys)

- From the main menu, click on the “Saved Portfolio” button.
- To get Data Entry, click on “+” icon.

Data Entry

- Enter stock ticker symbol into the search box provided.

UC-4: Aesthetic needs

Navigation (1 click)

- From any graph, click on the cog/gear symbol (1 click)

Data Entry (5+ clicks, depending of if the user knows how they want the graph to look)

- Use mouse to choose line vs bar graph (1+ clicks, if the user is feeling finicky)
- Use the mouse to choose whether or not to use predictions (1+)
- Use the mouse to determine whether or not to use gridlines (1+)
- Use mouse to determine historical data color (1+)
- Use mouse to determine predicted data color (1+)

Domain Analysis

Domain Model

1. Components of domain model:

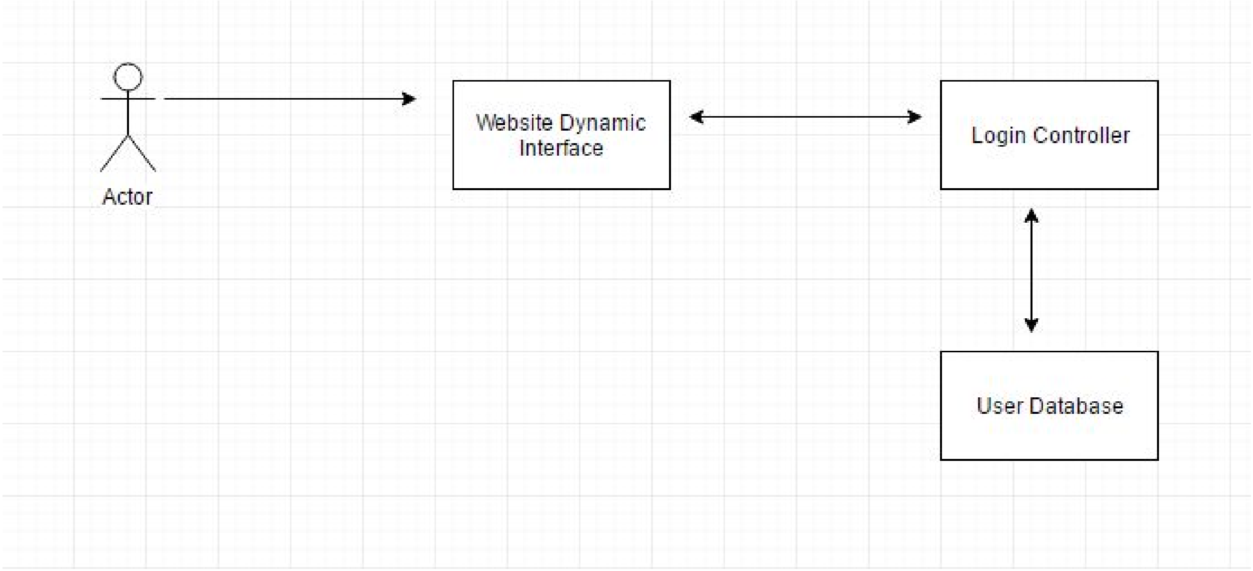
User
Login Controller
User Database
Stock Database
Graph Creator
Prediction Algorithms — “Algorithm Brains”
Tutorial Module
Quiz Creator
News Fetcher
Website Dynamic Interface
Stock Database Controller
Logout Controller
Algorithm Controller
Cookie — stored data about user

Notes: Algorithm Brains are run constantly and not triggered by a user logging in to a session. Stock Prediction would ideally be processed before a user has logged in.

UC-00: Logging In

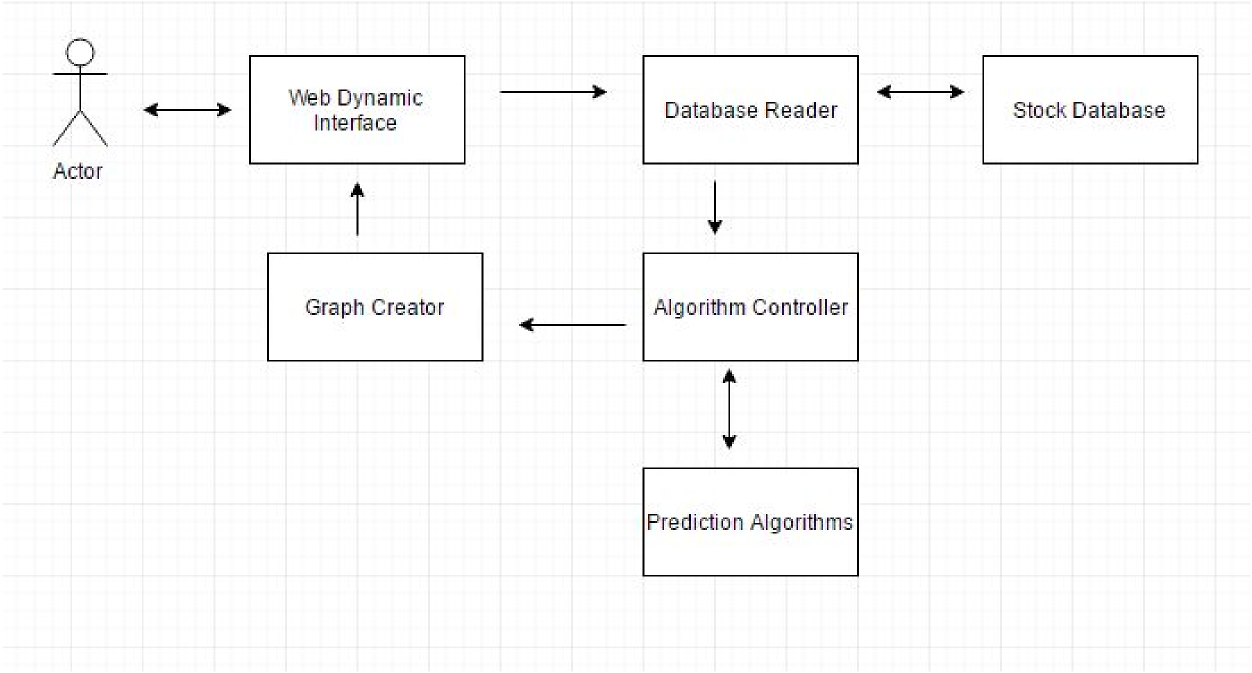
This component is common to all Use Cases (as all UCs will begin from the point where the user is not logged in yet), so it will be placed here for expediency's sake.

The User interacts with the Website Dynamic Interface in order to log into the system. The User gives the WDI its credentials (username and password), and the WDI passes it to the Login Controller. The Login Controller takes this data and references it with the User Database. If there is a match in the User Database, the Login Controller pulls any relevant data from the User Database and sends it back to the WDI, which presents it to the User in a relevant way, either through a main menu, or restoring a previous state (tabs/Stocks open) in the app.



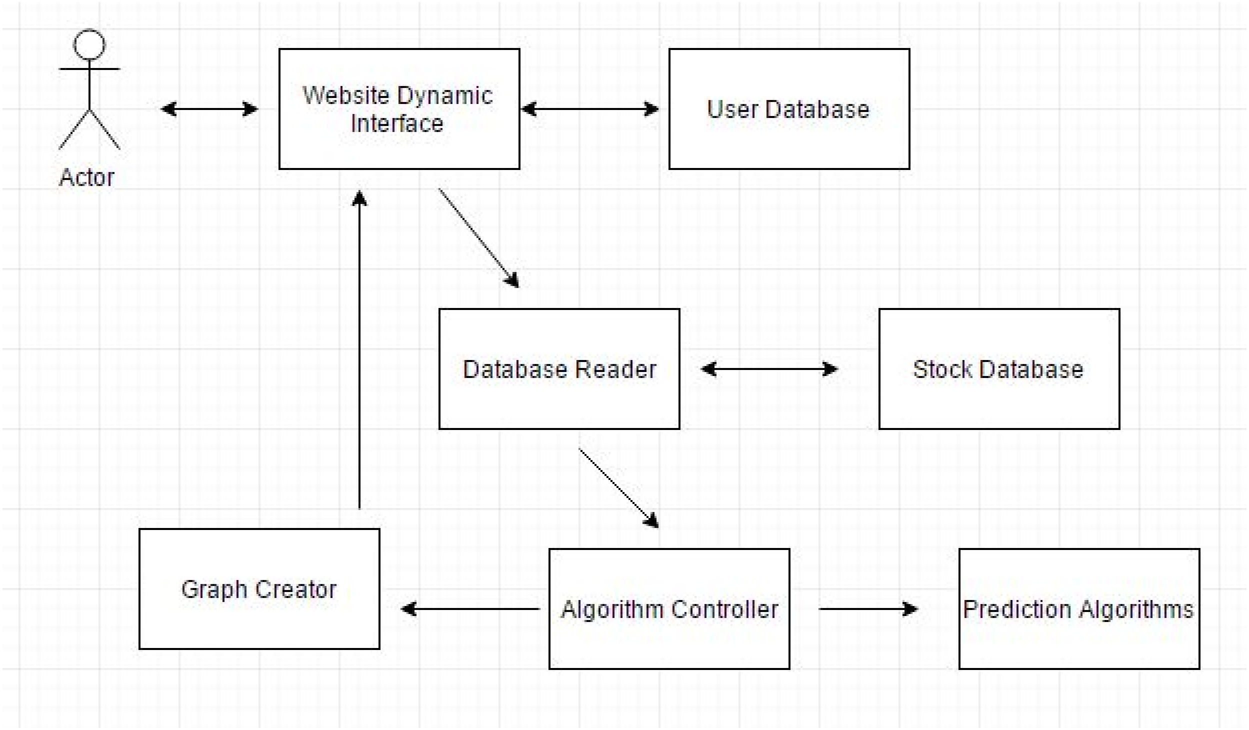
UC-1: Stock Prediction

The User starts this UC by interacting with the Website Dynamic Interface component, navigating through the interface until they input a ticker symbol to view the predictions of. The WDI takes the ticker symbol and passes it to the Stock Database, to see if the ticker entered is one of the stocks which we are allowing our algorithms to work on. If the stock is there, the ticker is then passed to the Algorithm Controller, which inputs the stock data retrieved from the Stock Database to the Prediction Algorithms. This will produce data predicting the trend that the stock will follow, which is sent to the Graph Creator, which will send a graph back to the Web Dynamic Interface for the user to view.



UC-3: Saved Portfolio

For this UC, the user will interact with the Website Dynamic Interface, telling it to view the user's portfolio of stocks. Thus, the WDI pulls from the User Database the User's portfolio of saved ticker symbols. It sends these symbols to the Database Reader, which then goes through UC-1 with each symbol, generating a graph and making it viewable to the user. The main difference here is that the symbols are saved within the User Database so that they can be retrieved easily for each User.



Concept Definitions

Note: << A / B >> where A corresponds to the module’s level of control over other modules and B corresponds to if the module does work or is an object worked upon.
Level of control over other modules: control > entity > boundary

- User - Interacts with <<boundary>>
  Components to navigate the website or app and gain access to our services

- Login Controller - <<control/worker>>
  Backend Username and Password Checker triggered by “Enter” button on our website when user login data has been entered.

- User Database - <<entity/thing>>
  Stores User data and preferences

- Stock Database - <<entity/thing>>
  Stores Stock data and algorithm information

- Graph Creator - <<entity/worker>>
  Builds a dynamic graph based on user preferences, the type of graph requested, and the stock quote passed into the module

- Prediction Algorithms - <<entity/worker>>
  Predicts stock prices for stocks passed into the module

- Tutorial Module - <<entity/worker>>
  Highlights parts of the website, along with blurring others and accepting user input, etc. Triggers when a new user registers to the website.

- Quiz Creator - <<entity/worker>>
  Builds a dynamic quiz based on previous user history and the user’s desired focus area

- News Fetcher - <<entity/worker>>
  Collects news articles from search engines based on stock and date of publication

- Website Dynamic Interface - <<boundary/worker>>

  A dynamic frontend for the website that changes website appearance based on user preferences

- Database Reader - <<entity/worker>>

  Module that can read and write to both the databases

- Logout Controller - <<control/worker>>

  Triggered by “logout” button on our website. Deletes Cookie and logs user out of website.

- Algorithm Controller - <<control/worker>>

  Backend Controller for Stock Prediction Algorithms. Passes stocks to predict and necessary parameters to Prediction Algorithms

- Cookie - <<entity/thing>>

  Stores User information so it can be used by the website interface to provide a more dynamic interface.

Association Definitions

Concept Pair	Association Description	Association Name
User ←→ Website Interface	User is prompted by the Website interface to enter his id and password to log in	Get user credentials
Website Interface ←→ Login Controller	Interface sends the data the user entered to the backend Login Controller which checks for validity	Transmit data to Login verification
Login Controller←→User Database	Checks the entered Username and password vs entries in the database. Then retrieves user information in case of a match.	Access User Database
Website Interface ←→ Database Reader	Accesses stored information such as company portfolio/stock information from previous requests	Fetch Saved data
Website Interface ←→ Tutorial Module	Opens up the tutorial app in a separate window	Access tutorial Module
Tutorial Module ←→Quiz Creator	Sends request to quiz creator to process an appropriate quiz based on user's field of interest and resend to grade the completed quiz	Request & Grade Quiz
Database Reader ←→Stock Database	Requests information from the stock database	Fetch Data from database
Website Interface ←→News Fetcher	Access the latest financial news from business websites and display it to the user	Fetch Latest news
Website Interface ←→ Graph Creator	Send request to view a graph of stocks of selected companies	Display Graph
Graph Creator ←→ Stock Database	Fetch stock information from database and process it	Fetch Stock info
Database reader ←→ Algorithm Controller	Sends Request to Algorithm controller to fetch the full	Fetch algorithm info

	Algorithm logic	
Website Interface ←→ Logout Controller	Saves the user’s changes and resets the login info	Log off
User ←→ Cookies	Dynamically record user preferences such as login info and recently accessed company information	

Attribute Definitions

Concept	**Attributes**	**Attribute Description**
Login Controller	User’s Information	Used to allow or deny access to the system by checking user’s credentials.
User Database	User Preferences	Used to hold actor’s preferred settings.
	Stock List	A list of stocks, chosen by the user, which are immediately shown after login.
Quiz Creator	Progress Checker	Keeps data on previous user history and understands where user needs help.
	Quiz Mode	Requires user to take quiz based off of history.
Tutorial Module	Stock Market Tutorial	Guides the user through the website and teaches about the stock market.
Graph Creator	User Preferences	Saves and implements user graph settings. These include: colors, tick values, etc.

	Graph Display	Displays the graph according to current stock info and user’s preferences.

Traceability Matrix

Domain Concepts	UC-0	UC-1	UC-2	UC-3	UC-4	UC-5
User	X	X	X	X	X	X
Login Controller	X					
User Database	X		X	X		
Stock Database		X		X	X	
Graph Creator		X		X		X
Prediction Algorithms		X		X		
Tutorial Module			X		X	
Quiz Creator			X			
News Fetcher					X	
Website Dynamic Interface	X	X	X	X	X	X
Database Reader		X		X	X	
Logout Controller	X					
Algorithm Controller		X		X		
Cookie	X	X	X	X	X	X

System Operation Contract

UC-00

Operation	Login
Pre-conditions	- numOfAttemtps < maxNumOfAttempts - numOfAttempts = 0 for first try - Set of users and passwords in the system established
Post-conditions	- Success flag returned to indicate pass/fail

Operation	Check Login
Pre-conditions	- numOfAttemtps < maxNumOfAttempts - numOfAttempts = 0 for first try - Set of users and passwords in the system established
Post-conditions	- numOfAttempts reset to 0 if successful - numOfAttempts increased otherwise - Success flag returned to indicate pass/fail

Operation	Get User Preferences
Pre-conditions	- Set of user preferences in the system established
Post-conditions	- Success flag returned to indicate if user for request exists - User preferences returned - User preferences in system unchanged

UC-1

Operation	Fetch Graph
Pre-conditions	- There is an established database of stock info in the system
Post-conditions	- Stock info in database unmodified

Operation	Get Stock Info
Pre-conditions	- There is an established database of stock info in the system
Post-conditions	- Entirety of stock info returned to caller - Stock info in database unmodified

Operation	Get Prediction Line
Pre-conditions	- There is an established database of stock info in the system - The graph creator has fetched the relevant stock's info
Post-conditions	- A prediction trendline is calculated and returned - Stock info in database unmodified

Operation	Get Accuracy Value
Pre-conditions	- None
Post-conditions	- Algorithm accuracy for specific stock data is calculated in comparison to historical stock info for said company - Stock info in database unmodified

UC-2

Operation	Get Tutorial
Pre-conditions	- Set of user progress info in the system established - User is currently logged in to the system
Post-conditions	- System returns tutorial section matching user’s current progress - User progress info unchanged

Operation	Get User Progress Info
Pre-conditions	- Set of users and passwords in the system established - Set of user progress info in the system established - User is currently logged in to the system
Post-conditions	- User progress info unchanged - User progress info returned

Operation	Submit Tutorial Section
Pre-conditions	- Set of user progress info in the system established - User is currently logged in to the system - User is currently in a tutorial section
Post-conditions	- User progress info updated - Flag returned to notify successful/unsuccessful completion

Operation	Get Quiz
Pre-conditions	- Set of user progress info in the system established - User is currently logged in to the system
Post-conditions	- User progress info updated - System returns quiz corresponding to most recent tutorial section

Operation	Submit Quiz
Pre-conditions	- Set of user progress info in the system established - User is currently logged in to the system
Post-conditions	- User progress info updated - Quiz completion successful/unsuccessful flag returned

UC-4

Operation	Get News
Pre-conditions	- None
Post-conditions	- Relevant news to the current company is returned

Operation	Get Algorithm Explanation
Pre-conditions	- None
Post-conditions	- An explanation of the selected algorithm's workings is returned

Operation	Fetch Graph
Pre-conditions	- There is an established database of stock info in the system
Post-conditions	- Stock info in database unmodified

Operation	Get Stock Info
Pre-conditions	- There is an established database of stock info in the system
Post-conditions	- Entirety of stock info returned to caller - Stock info in database unmodified

Operation	Get Prediction Line
Pre-conditions	- There is an established database of stock info in the system - The graph creator has fetched the relevant stock's info
Post-conditions	- A prediction trendline is calculated and returned - Stock info in database unmodified

Operation	Get Accuracy Value
Pre-conditions	- None
Post-conditions	- Algorithm accuracy for specific stock data is calculated in comparison to historical stock info for said company - Stock info in database unmodified

Mathematical Model

Our method of approach to predicting the value of a stock consists of using least square approximation to predict the incline or decline of a particular stock. The model works in the following steps:

1) Determining the form of the already plotted graph this consists of finding the shape of the current graph. Here shape consists of what degree polynomial fits best for the current graph. Most likely, the graphs will fit best about a polynomial of At^2+Bt+C=D. The way the program will determine this is to simply use the average price of a stock as a makeshift x-axis. Then based on how many times it crosses this axis (as determined by the data points) we can estimate the degree of the figure.
2) Determine our data point set. Basically, depending on the timespan chosen by our user (This comes from how far back the user wishes to see up to a certain value for computational purposes). The standard would follow along the lines of:
    a) One Day: ~96 Data Points
    b) One Week: ~672 Data Points
    c) One Month: ~2,688 Data Points
    d) One Year: ~32,256 (Depending on calculation time)
    e) >1 year: ~32,256
3) Determining the current trend or most notably recent change. This method consists of looking at how the stock has performed most recently whether it has increased or decreased since last checked. This kind of process would occur every 15 minutes and compare the current change with the previous one 15 minutes ago. This way we can crudely determine whether or not the price has increased or decreased.
    a) P_f-P_O=ΔP if ΔP >0 we have and increase if Δ<0 we have a decrease
4) Plotting the line of best fit through least square approximation. Depending on what part one observes our equation will give us the output of a matrix $[a_0;a_1;a_2]$. a_0 is c in aforementioned part one, a_1 is b, and a_2 is a. These values are computed by solving a transformation matic $(C^TC)^{-1}C^Ty$. Where y is the plotted points on the y axis that we choose to take depending on part 2. These points act as the "solution". The next step is to create the C matrix or transformation matrix. That matrix consists of three vectors. $V_1=[1;1;1;1\ldots1],V_2=[x_1,x_2,x_3,x_4\ \ldots\ldots x_n],V_3=[x_1^2,x_2^2,x_3^2,x_4^2\ \ldots\ldots x_n^2]$

5) We then find the point matching that line at the point in time in the next 15 minutes; Thus, we find an approximation that predicts the value of the stock in the next 15 minutes. This graph will be updated every 15 minutes to determine the future price with increased accuracy. This process will continue and with a

larger time span the accuracy will decrease overall since the line is more general and random outliers that are caused by unpredictable events are unpredictable.

6) Accuracy or confidence. If the stock has been showing a decline then we assume that the future will hold the same depending on the value of ΔP. The system will compare its approximation with that of the delta P and based on how two the close are in both magnitude and sign the system will reflect that as an accuracy (Preliminary).

Example of the output:

Time	Price
0	110
1	118
2	92
3	48
3.5	7

C=[1,0,0];[1,1,1];[1,2,4];[1,3,9];[1,3.5]

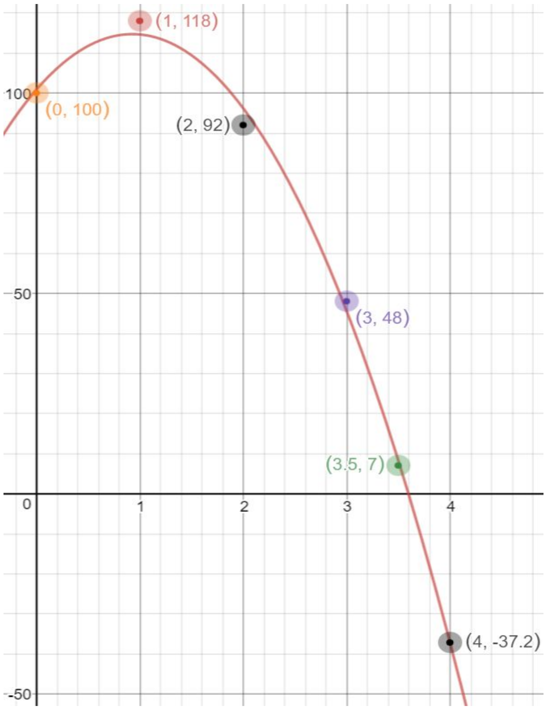
y=[100];[118];[92];[48];[7]

Solution: a_0= 101.00

a_1= 29.77

a_2=-16.11

Here the black dot represents the approximate value after time hits 4



Plan of Work

Each of the three vertical stacks (stock graphing and prediction [Team 1], website with user login and news and stock ticker [Team 2], educational tutorial and quizzes [Team 3]) operate independently of the others.

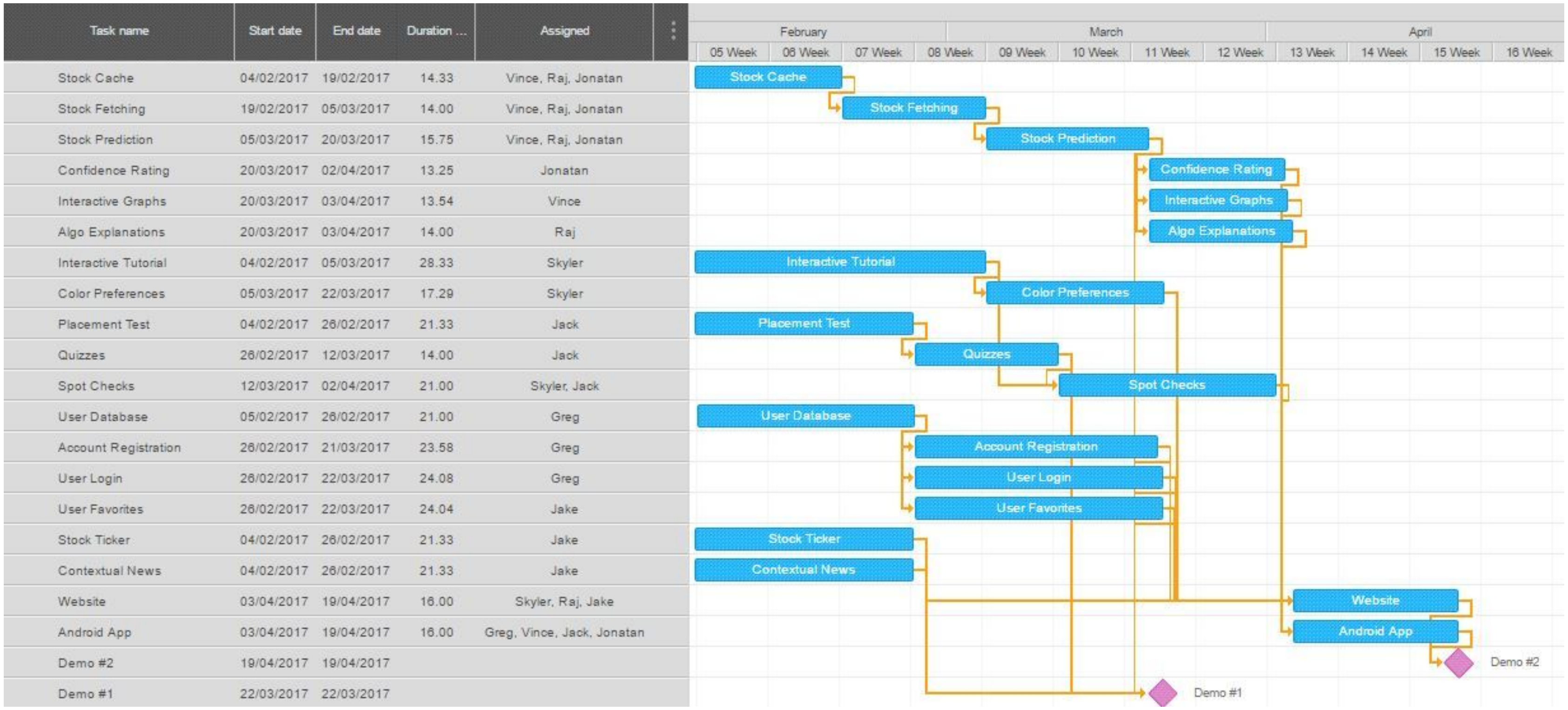
The tasks for Team 1 start out with linear dependencies, and then split off a the last stage. All 3 members will work together to complete the Stock Cache first, which has an allotted time of 2 weeks to completion. They then work on the Stock Fetching for 2 weeks. Then they work on the prediction algorithms for 16 days. After this, they each work on one of the three remaining components (Confidence Rating, Interactive Graphs, Algorithm Explanations) for 2 weeks. At this point, Team 1 finishes its tasks, dissolves, and merges into the two teams for the Website and Android app.

Team 2's tasks are more disparate, but can be separated into two sets of dependencies. The first task that must be completed is the User Database, which is a dependency for multiple components. This occurs in parallel with the Stock Ticker and Contextual News in order to maximize throughput. These are all given a timeline of 3 weeks to complete. After this stage, there remain 3 tasks (Account Registration, User Login, User Favorites) that can all be completed in parallel in a timespan allotted of 24 days. At this point, Team 2 finishes its tasks, dissolves, and merges into the two teams for the Website and Android app.

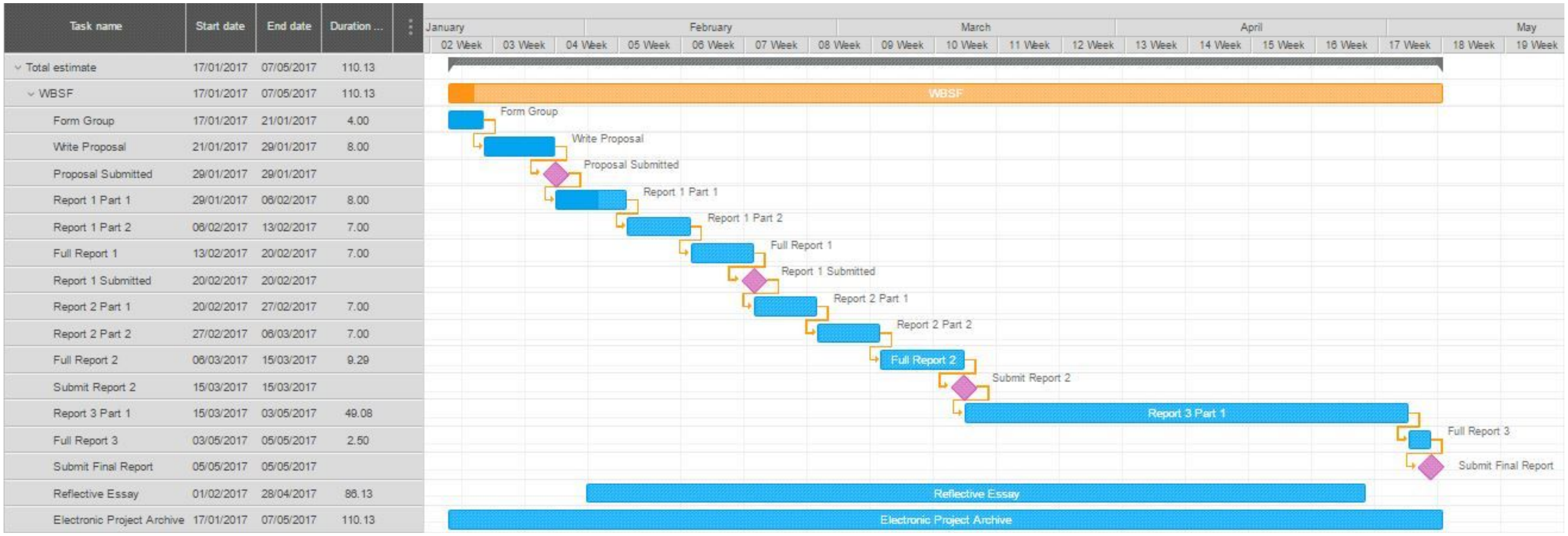
Team 3's tasks occur in two parallel timelines (Tutorial, Color Preferences) (Placement Test, Quizzes, Spot-Checks) with mostly linear dependencies for each line. The Tutorial and Placement Test are worked on simultaneously for 4 and 3 weeks respectively. Then the Color Preferences and Quizzes are worked on simultaneously for 18 and 14 days, respectively. At this time, the first timeline is complete and that member joins the second timeline to assist with Spot-Checks. In total, 3 weeks are allocated for the Spot-Check component. At this point, Team 3 finishes its tasks, dissolves, and merges into the two teams for the Website and Android app.

Demo 1 will consist of 3 separate demos. Team 1 will demonstrate their stock fetching and prediction by producing graphs on demand via a simple website. Team 2 will demonstrate their completed website that allows user to login, register, and see news and stock tickers. Team 3 will demonstrate the Placement Test, Quizzes, and Interactive Tutorial via a simple website.

The website and Android app are ongoing teams that grow as members from other teams finish their tasks. They will be the key difference between Demo 1 and Demo 2.



Task name	Start date	End date	Duration ...	Assigned
Stock Cache	04/02/2017	19/02/2017	14.33	Vince, Raj, Jonatan
Stock Fetching	19/02/2017	05/03/2017	14.00	Vince, Raj, Jonatan
Stock Prediction	05/03/2017	20/03/2017	15.75	Vince, Raj, Jonatan
Confidence Rating	20/03/2017	02/04/2017	13.25	Jonatan
Interactive Graphs	20/03/2017	03/04/2017	13.54	Vince
Algo Explanations	20/03/2017	03/04/2017	14.00	Raj
Interactive Tutorial	04/02/2017	05/03/2017	28.33	Skyler
Color Preferences	05/03/2017	22/03/2017	17.29	Skyler
Placement Test	04/02/2017	26/02/2017	21.33	Jack
Quizzes	26/02/2017	12/03/2017	14.00	Jack
Spot Checks	12/03/2017	02/04/2017	21.00	Skyler, Jack
User Database	05/02/2017	26/02/2017	21.00	Greg
Account Registration	26/02/2017	21/03/2017	23.58	Greg
User Login	26/02/2017	22/03/2017	24.08	Greg
User Favorites	26/02/2017	22/03/2017	24.04	Jake
Stock Ticker	04/02/2017	26/02/2017	21.33	Jake
Contextual News	04/02/2017	26/02/2017	21.33	Jake
Website	03/04/2017	19/04/2017	16.00	Skyler, Raj, Jake
Android App	03/04/2017	19/04/2017	16.00	Greg, Vince, Jack, Jonatan
Demo #2	19/04/2017	19/04/2017		
Demo #1	22/03/2017	22/03/2017		



Task name	Start date	End date	Duration ...
Total estimate	17/01/2017	07/05/2017	110.13
WBSF	17/01/2017	07/05/2017	110.13
Form Group	17/01/2017	21/01/2017	4.00
Write Proposal	21/01/2017	29/01/2017	8.00
Proposal Submitted	29/01/2017	29/01/2017	
Report 1 Part 1	29/01/2017	06/02/2017	8.00
Report 1 Part 2	06/02/2017	13/02/2017	7.00
Full Report 1	13/02/2017	20/02/2017	7.00
Report 1 Submitted	20/02/2017	20/02/2017	
Report 2 Part 1	20/02/2017	27/02/2017	7.00
Report 2 Part 2	27/02/2017	06/03/2017	7.00
Full Report 2	06/03/2017	15/03/2017	9.29
Submit Report 2	15/03/2017	15/03/2017	
Report 3 Part 1	15/03/2017	03/05/2017	49.08
Full Report 3	03/05/2017	05/05/2017	2.50
Submit Final Report	05/05/2017	05/05/2017	
Reflective Essay	01/02/2017	28/04/2017	86.13
Electronic Project Archive	17/01/2017	07/05/2017	110.13

Product Ownership

The project will be composed of three teams. Each team will operate completely independently of the others until it finishes its tasks, at which point it will dissolve and the members will join the two teams for stitching together the final website and producing the Android application.

	Team 1	**Team 2**	**Team 3**
Members:	Balaji, Raj Taylor, Vincent Yanovsky, Jonatan	Leonberg, Gregory Lewandowsky, Jake	Aquino, Jack Malinowski, Skyler
Features:	Stock Cache Stock Fetching Stock Prediction Interactive Graphs Algorithm Explanations Confidence Rating	Login System User Registration Account Database User Favorites Stock Ticker Contextual News Feed	Placement Test Interactive Tutorial Proficiency Quizzes Spot-Checks Color Options

Member	Completed	Working-On	Queued
Leonberg, Gregory		User Database	Account Registration User Login Android App
Lewandowski, Jake		Stock Ticker Contextual News	User Favorites Website
Malinowski, Skyler		Interactive Tutorial	Color Preferences Spot Checks Website
Aquino, Jack		Placement Test	Quizzes Spot Checks Android App
Taylor, Vincent	Stock Cache	Stock Fetching	Stock Prediction Interactive Graphs Android App
Yanovsky, Jonatan	Stock Cache	Stock Fetching	Stock Prediction Confidence Rating Android App
Balaji, Raj	Stock Cache	Stock Fetching	Stock Prediction Algo Explanations Website

References

http://www.timothysykes.com/2013/06/trading-terms-you-need-to-know/
(Used for Customer Statement of Requirements, as well as Glossary of Terms)

http://www.wikinvest.com/wiki/What_is_a_stock%3F
(Used for Customer Statement of Requirements, as well as Glossary of Terms)

https://businessanalysttraininghyderabad.wordpress.com/2014/08/05/what-is-furps/
(Used for Enumerated Non-Functional Requirements)

http://money.cnn.com/2014/09/18/investing/stock-market-investors-get-rich/
(Used for Customer Statement of Requirements, as well as Glossary of Terms)

https://www.codecademy.com/
(Used for inspiration for On-Screen Appearance Requirements and User Interface Specification)

https://finance.yahoo.com/
(Used for inspiration for On-Screen Appearance Requirements and User Interface Specification)