Web-Based Stock Forecaster

REPORT 1

GROUP 9
KRISHA PAULA OLANDAY
VINAY PANJABI
VINAY SHIVAKUMAR
NEHA DESAI
JONATHAN HAAS
SIVARAMHARESH SIVA

DROPBOX: HTTPS://WWW.DROPBOX.COM/SH/58UOMDLU8CB6IPI/9FRUKZ7NXK
WEBSITE: HTTPS://BITBUCKET.ORG/VINKNEE/STOCKFORECASTER-GROUP-9

INSTRUCTOR: IVAN MARSIC

TABLE OF CONTENTS

1 CONTRIBUTION BREAKDOWN

All members contributed equally.

2 COSTUMER STATEMENT OF REQUIREMENTS

Investing money in the stock market is relatively easy but investing successfully and earning a profit can be a challenge and most non-professional investors lose money every year. There are many reasons why this can occur but the primary reason behind this is the fact that most investors just do not have the time and resources to implement the lengthy analysis that takes place by full-time investors or employees of large investment organizations. Professional investors and firms have an advantage; they have a team of investors to help with the research and spend their entire careers studying the markets. On the other hand, normal investors do not have such a luxury, many of whom work far away from the field of investments. If you've read articles about investment, many of them refer to investing as dating and long term investing is akin to marriage. Just like dating and eventually marriage, investing requires the investor to know the company they are investing in and it also requires a lot of commitment.

With this realization, our project will attempt to benefit a broad range of investors by providing calculated predictions as a tool for them to make their own decision on whether to buy, sell, or hold the stock. We will also provide them with as much information and current news about the company.

2.1 PROBLEM

In order to reduce the amount of research for investors our team has identified these problems:

2.1.1 Investors are not able to predict a stock's future performance due to various factors.

The topic of "can the stock market be predicted?" has always been a hot topic in the financial world. The stock market is constantly rising and falling on a daily bases, stocks can go up a certain amount one day and then the next day their prices may have gone down significantly. In the long term, you cannot predict stocks with precision and it's impossible to predict stocks in the short term. So the question is, what do you use to predict the constant rise and fall of the stock prices and should you even waste time trying to obtain a prediction when the behavior is deemed to be unpredictable?

There are three broad categories in stock prediction methods: fundamental analysis, technical analysis, and technological methods. These three methods rely on the analysis of past data and stock performance and are highly mathematical. However, once all the mathematical and logical research has been obtain, when can you take the volatility and emotional aspects of buying and selling stocks into consideration? Once again time is a big issue for individual investors and that same question pops up again. Should time be even spent in finding the right kind of prediction method?

Our solution to this problem is providing a stock prediction for the individual investor. They no longer have to do their own research in order to generate a forecast; we will do that for them. On the website, we will state what kind of prediction model we use to generate the prediction.

This is a method of accountability and investors do not need to question where such numbers came from. From these predictions, it is then up to the investor on whether they should buy, sell, or hold the stock.

2.1.2 Investors do not understand the company's business model or the company itself.

As mentioned before, would you date a person whom you know nothing about? It's the same for stocks; investors should not invest in a stock without knowing these three things: what the company actually does how they make a profit, and the company's fundamentals. The company's fundamentals enable you to see the how the company is doing financially; it can include its balance sheet, government filings, investor reports, etc. It's very common for individual investors to not know much about the company they are trying to invest in or they have a preconception of how well the company is doing and what they are best known for. The problem with this is, how will investors be able to know when to buy or sell a stock if they don't know exactly how the company is making money and if their assets will still be popular with consumers in the future? Investors should also know the financial stability of the company. If an investor does not know the financial stability of the company, then they may end up investing in a company that has a lot of debt.

To remedy this problem, we will include a short description of what the company does and their primary method of making money. This way, investors have a small inkling of what the company does right on the page. We will also include a link to the U.S. Securities and Exchange Commission's EDGAR database. This database contains financial statements for all U.S public companies.

2.1.3 Not having all the stock information in one area.

When doing all this research, an investor will have to look for all of this information in various places. Because of this, it's hard to keep track of all of the information. We as a team will try to provide all relevant stock information on the page and in the website. The website will also include quick links to other helpful sites such as Forbes.com and the EDGAR database.

2.1.4 Investors not being able to visualize a stock's history and how it rises and falls.

Not everyone is a visual person. Therefore, our stock data will be shown as a linear regression graph of time versus price. This way investors can view the rise and fall of the stock price. The prediction will be an extension from the current stock value on the graph and will help investors see how the stock will perform relative to the current and past values.

2.1.5 Investors not in touch with current company events, world events and the economy.

To keep up with the latest news, the website will include a "What's Trending" page. The page is where investors can view and read current news, world events, and companies are up to.

2.1.6 Investors cannot easily compare potential stocks to view which stock(s) will provide the best possible yield.

Do you notice that on many websites that sell some sort of product, they have a "previously viewed" section somewhere on the webpage? In order to help investors compare future stocks with each other, we will create a comparative list of previously search stocks with their prediction values. It will be something similar to a "previously viewed" section on a website such as Macys.com. This way as investors search for potential stocks, they will see how well it performs in the future compared to the previously searched companies.

3 GLOSSARY OF TERMS

52 Week High/Low	Highest and lowest prices of a stock that has been recorded at during the previous year.
Average Volume	The amount of stocks that traded over duration of time.
Buy and Hold	The practice of buying a good for a long term rather than trying to turn a profit quickly.
Closing Price	The price a particular stock closes at on a given trading day.
Dividend	A distribution of a portion of a company's earnings, decided by the board of directors, to a class of its shareholders. The dividend is most often quoted in terms of the dollar amount each share receives (dividends per share). It can also be quoted in terms of a percent of the current market price, referred to as dividend yield. http://www.investopedia.com/terms/d/dividend.asp
Earnings per Share (EPS)	The portion of a company's profit allocated to each outstanding share of common stock. Earnings per share serves as an indicator of a company's profitability. It is calculated as (Net Income -Dividends on Preferred Stock) / divided by the Average Outstanding Shares http://www.investopedia.com/terms/e/eps.asp
Forecast	A prediction of the future based on special knowledge
Fundamental Trading	Fundamentalists trade companies based on fundamental analysis, which examines things like corporate events such as actual or anticipated earnings reports, stock splits, reorganizations or acquisitions. http://www.investopedia.com/articles/trading/02/100102.asp
Individual	An investor who purchases relatively small lots of stocks for his or her own

Investor	portfolio.
Institutional Investor	An entity with large amounts to invest, such as investment companies, brokerages, and investment banks. Institutional investors are covered by fewer protective regulations because it is assumed that they are more knowledgeable and better able to protect themselves. Institutional investors are usually a group of people, rather than individuals.
Market Trend	The tendency of a financial market to move in a particular direction over time. Bull market refers to an upward trend, and a bear market refers to a downward trend.
Moving Average Prediction Model	A way to predict the future price of stocks based on the assumption of constant underlying mean of given prices
Opening Price	The price a stock starts off at a particular trading day.
Shares	See Stock
Stock	A type of security that signifies ownership in a corporation and represents a claim on part of the corporation's Assests and earnings.
Stock Market	The marketplace for buyers and sellers of stocks.
Stock Symbol	A unique set of symbols that represent a particular company. Ex: GOOG is the stock symbol of Google.
Ticker	See Stock Symbol
Trading Day	The duration of time the stock market is open for buying and selling stock. Ex: For the New York Stock Exchange trading day is 9:30 AM Eastern Time to 4:00 PM Eastern Time, trading days never occur on weekends.

4 SYSTEM REQUIREMENTS

4.1 ENUMERATED FUNCTIONAL REQUIREMENTS

Identifier	Priority	Requirement
REQ 1	5	The system will acquire past stock prices and data from all stocks listed on Yahoo! Finance on a daily. The stock data will then be placed into a stock database.
REQ 2	3	The program will output the stock prices as a graph of price versus date.
REQ 3	4	The system will allow for users to search for the stock based on the stock ticker or the company name
REQ 4	5	Understand and implement the moving average prediction model or the decided prediction for the closing price of the stock prices.
REQ 5	3	The stock data stored in the database will be used by the prediction algorithm to calculate the stock's prediction.
REQ 6	4	The graph generated by the program will also include the estimate of the future price.
REQ 7	3	The website should display the graph of the stock prices. The graph should depend on the input of the stock ticker.
REQ 8	3	The program should have a data structure (portfolio) that holds all the stock tickers that the user is interested in.
REQ 9	2	The website should display the current tweets or news articles from reputable sources.
REQ 10	3	The system should be password protected so only users with access can view the website.
REQ 11	5	Each stock shall have its own page in the website with all of the stock information on the page.
REQ 12	5	All stocks in the database can be searched and accessed through the search bar.

4.1.1 Anaylsis: Enumerated Functional Requirements

Our main requirement is to be able to obtain the historical data for all or most stocks (REQ-1) in the Yahoo! Finanace API and then from this data be able to generate a prediction (REQ-5). The user will then be able to search a stock on the website (REQ-3) and the system will return a webpage with all the stock's information, graph of historical data with the prediction as an overlay (REQ-11).

For REQ-1, our end goal is to have all the stocks listed in Yahoo! Finance to be stored in the database. However, for demo and test purposes, we will only have 100 or so of the top fortune 500 companies in the database. However, if a user does search for a stock that is not in the database, the system will query the information from Yahoo! Finance and the user will still be able to obtain the information.

REQ-5 and REQ-11, are straight to the point, the prediction algorithms that the Prediction Team will create uses the historical data stored in the database to generate a prediction model to be graphed alongside the stock's historical values.

The users will need to find information on a stock they are interested in. Therefore, we will have a static search bar at the top of the site (REQ-25). Where users will be able to enter a stock ticker or the company's name and search for that stock in the database (REQ-3). If the stock is not found, the database will quickly query the data from Yahoo! Fianance. This will cause a slight delay for the information to be sent to the user, because the system needs to obtain and generate the information. On the other hand, if the user enters an invalid ticker, or the system does not recognize the company name, the system will return a "Sorry, stock cannot be found' page with a small list of suggested stocks that were close to the one they entered.

4.2 ENUMERATED NONFUNCTIONAL REQUIREMENTS

	Identifier	Priority	Requirement
Functionality	REQ 13	5	Provide customers with predictive data on the stock market to aid their trading decisions.
	REQ 14	5	The prediction will use market data and indicators from the database, such as RSI and stochastics, as input to the predictive algorithm.
	REQ 15	4	Website will also display normal data on the stock such as the current price and volume.
Usability	REQ 16	2	Website will host the stock prediction as an overlay on the price graph.
	REQ 17	3	The database will be updated daily to account for the daily changes to stock values.
	REQ 18	4	Site will be simple and clean so that the customer can view everything easily.
Reliability	REQ 19	4	Earnings reports can cause unpredictable changes in a stock. The earnings date can be shown to a user as a warning about predictions surrounding that date
	REQ 20	3	Prediction models will be updated daily.
	REQ 21	3	Data backed up in the case of a site failure.

Performance	REQ 22	3	Current data transfer rates should allow for quick and easy navigation of the site and its features.
Supportability	REQ 23	2	The service will be available in the form of a website, which will allow it to be accessed by most browsers.
	REQ 24	2	Configuration options available to users.

4.2.1 Analysis: Enumerated NonFunctional Requirements

The key requirements are REQ-17 and REQ-20. In order for all the stock data to be up to date, the database will be updated daily and the prediction algorithms as well. Updating both the database and the prediction algorithm on a daily basis, is just a place holder value. Ideally, it should be updated more than once a day, since stock values change every hour, but our team has not decided on how many times a day the information will be updated (hourly, twice, three times). Since we did come to the conclusion that it will be updated at least once a day that is what we stated here.

4.3 On-Screen Appearance Requirements

Identifier	Priority	Requirement
REQ 25	5	A fixed navigation bar that includes all of the functional features, allowing the user to quickly navigate to any feature that they may choose without going through various subpages and menus.
REQ 26	5	The page must adapt to a various sizes in order to maintain functionality, consistent design, as well as ease of use on a mobile device
REQ 27	5	The website must support the latest versions of the most popular browsers, Google Chrome, Firefox, and Internet Explorer.
REQ 28	4	Function and purpose of each element on screen must be clear and direct by placing information in natural areas where natural will be defined based on other popular stock websites, as well as developer/designer intuition

4.3.1 Screen Mockup

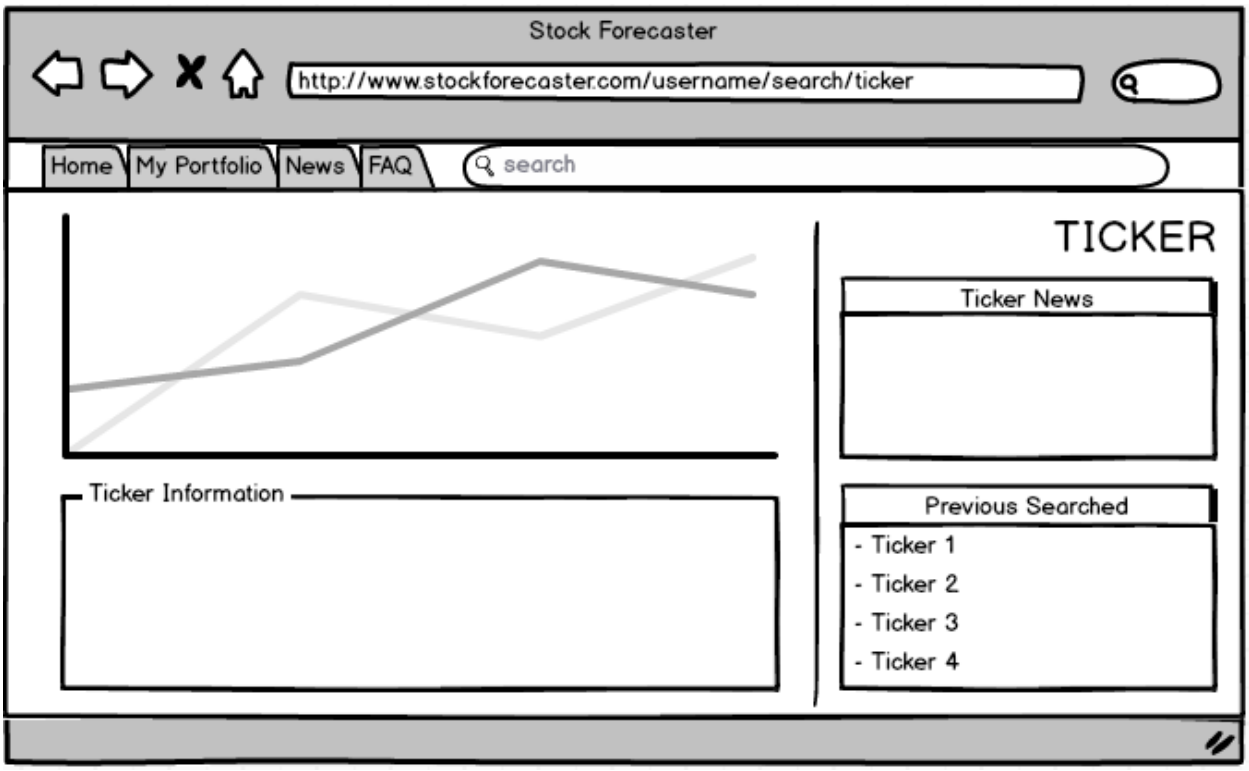


Figure 4-1. Basic outline of a Stock's subpage

4.3.1 Analysis: On-Screen Appearance Requirements

REQ 26 - This is exemplified in the top portion of figure 4-1 in section 4.3.1, screen mockups, which has the search bar, and navigation tabs to all of the site features at the top of the screen. There are no pages outside of what can be accessed from the navigation bar.

REQ 29 - This requirement is rather vague yet important at the same time. This is related to REQ 26 however that relates more toward navigation, this is referring to the actual pages that will be loaded and where the placement of information will be. An idea to meet this requirement is seen in figure 4-1.

5 FUNCTIONAL REQUIREMENTS SPECIFICATION

5.1 SPECIFYING THE DISCOVERED REQUIREMENTS:

Architectural style: Web-based – The programs will run on a Web Browser and each user's portfolio will be stored; their portfolio includes a list of all the stock tickers, information for those stocks, and corresponding graphs for those stocks. All information could be stored in a common database or on an Excel Spreadsheet. There is no form of client communication, but if we feel as if it could be an added benefit to the website, it can be implemented in the future.

5.2 STAKEHOLDERS

The stakeholders of our system are:

Individual Investors Unlike institutional investors who have a team at their disposal, individual investors don't have such a luxury and will benefit from our web application. These investors have a broad range of assets (usually smaller than institutional investors) and can either invest in the short or long term. They usually purchases small lots of stock (1-5) depending on the value of their assets. The individual investor is your standard person who does not use investing as their primary source of income.

5.3 ACTORS AND GOALS

Our system has both human and non-human actors.

User The user will take the form of an investor that registers and uses the website

Administrator The manager that is in charge or keeping the system updated and in working order

Prediction Algorithm The algorithm(s) that will calculate the prediction.

Database The database will hold all the user data and information as well as all the stock information. All the user information and their portfolios will be stored in the database.

Yahoo! Finanace API The API is where we will pool all stock data from to store into the database to generate the prediction.

Highchart Will be used to graphically plot the stock data.

5.4 USE CASE CASUAL DESCRIPTION

UC - 1 SearchForStock (REQ3, REQ7, REQ12)	Allows a registered user to search for a stock they want information on.
UC - 2 ObtainPrediction (REQ2, REQ3, REQ6 REQ13)	Obtains the input parameters for the prediction algorithm to calculate the prediction. Once the prediction is calculated, a graph will then be generated with both the stock's historical data and prediction.
UC - 3 DataAcquisition (REQ1)	In order to obtain a prediction and generate a graph, we will need to retrieve stock and market data from the Yahoo! Finance API and store it into the database. This is done in the background. If the stock is not already in the database, an algorithm will query Yahoo! Finance for the requested data.
UC - 4 CalculatePrediction (REQ4, REQ5)	From the input parameters, the prediction algorithm will calculate the stock's prediction.
UC - 5 Login (REQ10)	In order to access the website, users need to login. This why users will have their own portfolio. Also the home page of the website is customized according to the user's portfolio.
UC - 6 Register (REQ10)	Allows a visitor to fill out the registration form, allowing them access to the entire website
UC - 7 ManageUser (REQ10)	New users will be added to the database and current users will have their information updated in the data base.
UC - 8 AddUser (REQ10)	Enters new registered users' information to the database
UC - 9 EditUserInformation (REQ10)	Allows users to edit and change/update their information such as passwords, emails, and other account settings
UC - 10 ManagePortfolio (REQ8)	Allows users to add or remove stocks in their portfolio
UC - 11 News (REQ9)	Users will have access to current events and news regarding different companies directly on our site.
UC - 12 DisplayWebpage (REQ11, REQ13)	Once all the information has been obtained, the system will display the requested information on the stock webpage.

5.4.1 Use Case Analysis

UC-12 where “Users will have access to current events and news regarding different companies directly on our site” isn’t strictly limited to our website. We could potentially also provide links to the specific different companies on our websites, which could provide another source of information for the investor. Also UC-12 is meant to attract users since investors always want to stay up to date with stock news daily and updating this can attract more users.

UC-10, which states, “Allows users to edit and change/update their information such as passwords, emails, and other account settings” doesn’t necessarily derive from requirement 11. However, it is a feature that is available on all login protected accounts in case a user wants to change user information such as privacy or password. It should be provided to give the user more flexibility and personalize the experience for him or her.

From the twelve use cases above, we were able to create a use case diagram of our system. We have decided to not include UC-12 which would display the news directly on our website. This use case does not involve any actors and it is just an integral part of the website. Users can easily access the news section by clicking on the "News" tab on the navigation bar. Figure 5.1 shoes the system use case diagram.

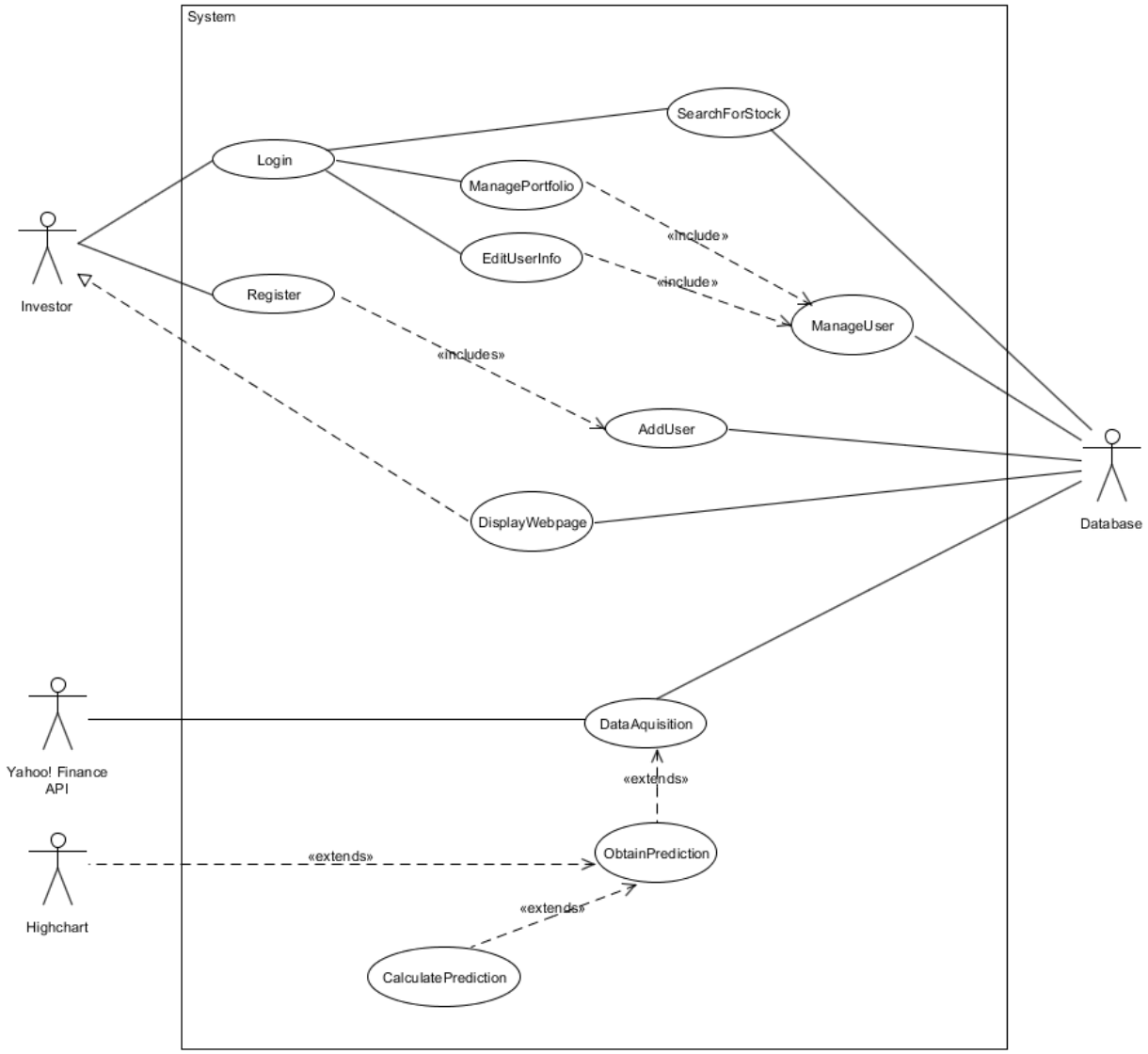


Figure 5-1: Use Case Diagram of System

5.5 TRACEABILITY MATRIX

	PW	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8	UC9	UC10	UC11	UC12
REQ1	5			X									
REQ2	3		X										
REQ3	4	X	X										
REQ4	5				X								
REQ5	3				X								
REQ6	4		X										
REQ7	5	X											
REQ8	3										X		
REQ9	2											X	
REQ10	3					X	X	X	X	X			
REQ11	5												X
REQ12	5	X											
REQ13	5		X										X
Max PW		5	5	5	5	3	3	3	3	3	3	2	5
Total PW		19	16	5	8	3	3	3	3	3	3	2	10

Table 5-1: Traceability Matrix

5.6 USE CASE FULLY-DRESSED DESCRIPTION

Out of all the use cases that we have come up with, there are four important use cases: register, login, searchStock, and ManagePortfolio

5.6.1 ObtainPrediction

This table shows a detailed summary of the ObtainPrediction use case. The interactions is defined as follows:

UC- 2: ObtainPrediction	
Related Requirements	REQ2, REQ5, REQ6, REQ7
Initiating Actors	User
Actor's Goal	Return graph of the stock's historical data and prediction should be displayed on the stock's page.
Participating Actors	Database, Prediction Algorithm, Highcharts
Preconditions	All stock data is in the database
Success End Condition	The stock's page is loaded with the graph of the historical data and prediction.
Failed End Condition	-

Flow of Events for Main Success Scenario:

1. -> **User** searches for a stock
2. <- **System** locates the stock in the database
3. -> **System** sends historical values to **Prediction Algorithm**
4. <- **Prediction Algorithm** calculates and returns the prediction
5. <- **System** generates a graph of the historical data with the prediction as an overlay.
6. <- **System** loads the graph and all stock information onto the stock's page

5.6.2 DataAquisition

This table shows a detailed summary of the DataAquisition use case. The interactions are defined as follows:

UC-3: DataAquisition	
Related Requirements	REQ1
Initiating Actors	User, Database
Actor's Goal	Obtain the stock data from the Yahoo! Finance API
Participating Actors	Yahoo! Finance API
Preconditions	Stock is currently not in the database
Success End Condition	The stock was successfully found in Yahoo! Finance and stored in the Database
Failed End Condition	Stock was not found in Yahoo! Finance

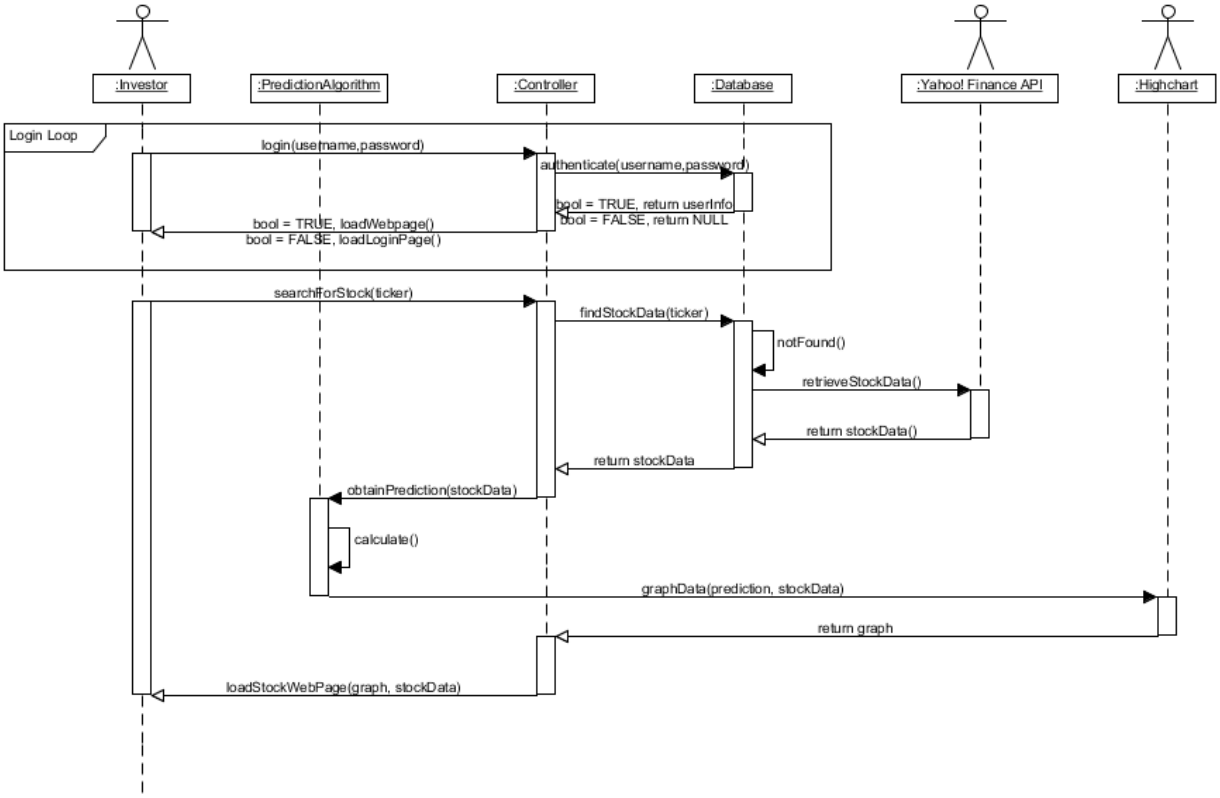
Flow of Events for Main Success Scenario:

1. **-> User searches for stock**
2. **-> System searches Database for stock**
    a. **If stock is not found**
        i. **-> Database queries Yahoo! Finance for stock data**
        ii. **<- Database stores acquired stock data**

Flow of Events for Extensions:

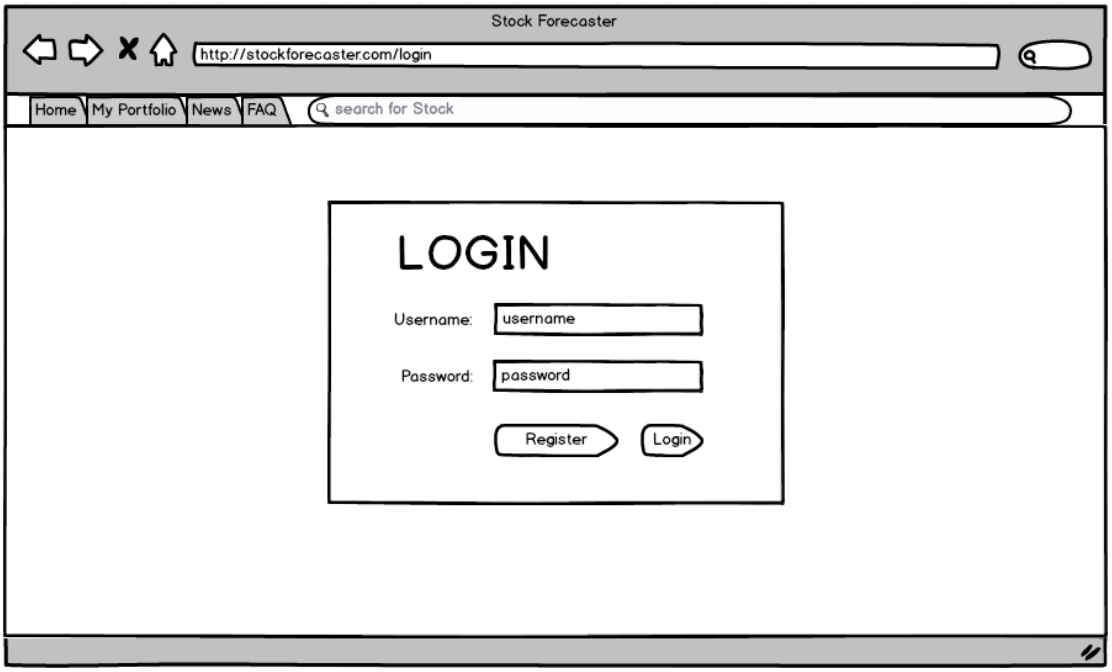
1. **Yahoo! Finance does not have the stock requested**
    a. **<- Database sends System error signal**
    b. **<- System sends user a "Stock cannot be found' page with suggested stocks that are similar to the searched stock**

5.7 SYSTEM SEQUENCE DIAGRAM

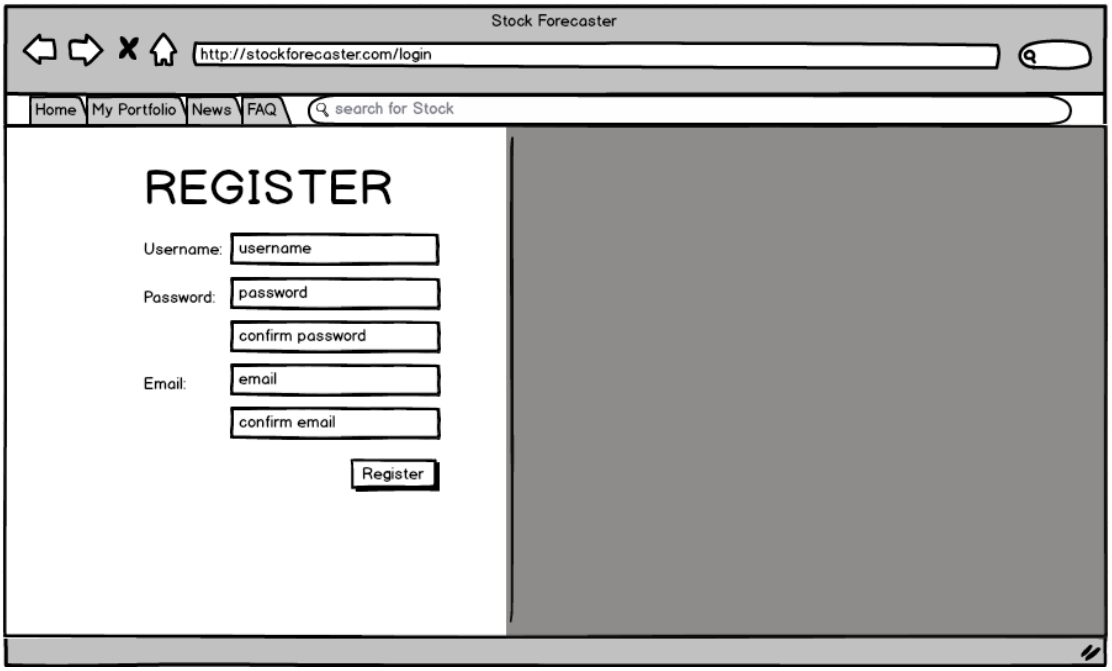


6 USER INTERFACE SPECIFICATION

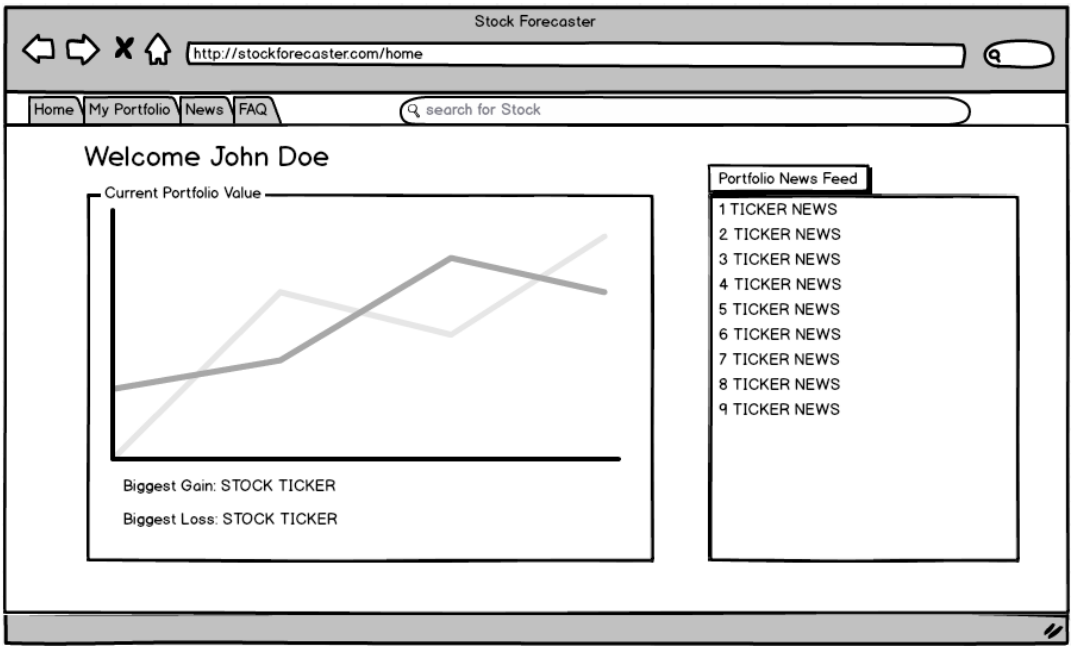
6.1 PRELIMINARY DESIGN



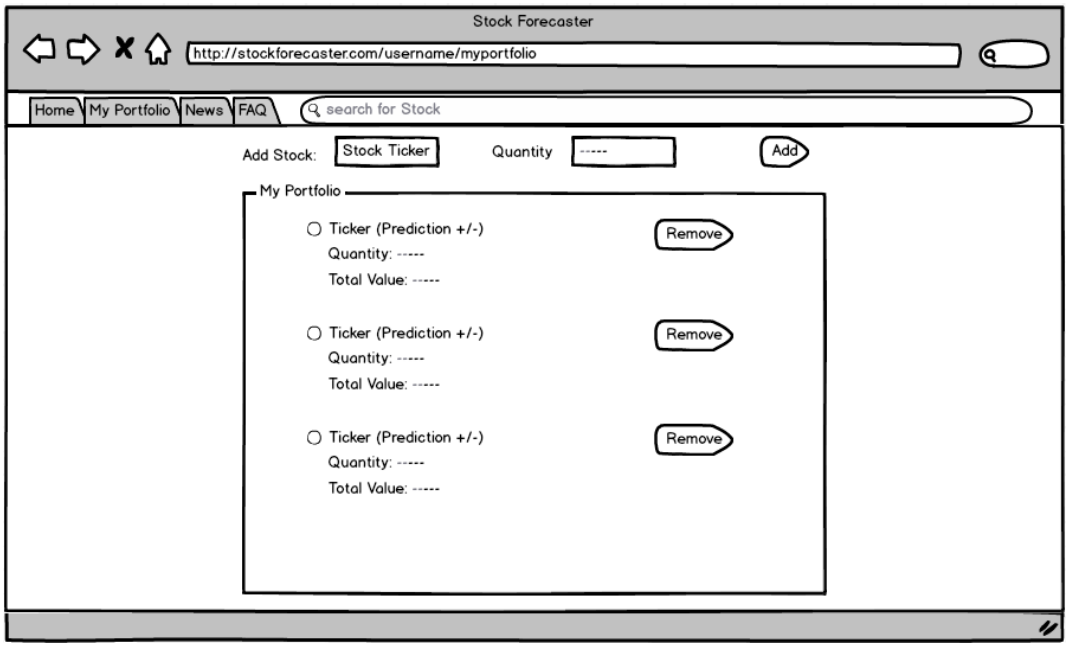
When the user loads our webpage, they do not have access to its content. Therefore, the first thing the user will see is the login screen. On here, if they are already registered, they can simply fill in their username and password in the boxes and click on login to access the site. If they are not registered, they can just click on the "Register" button.



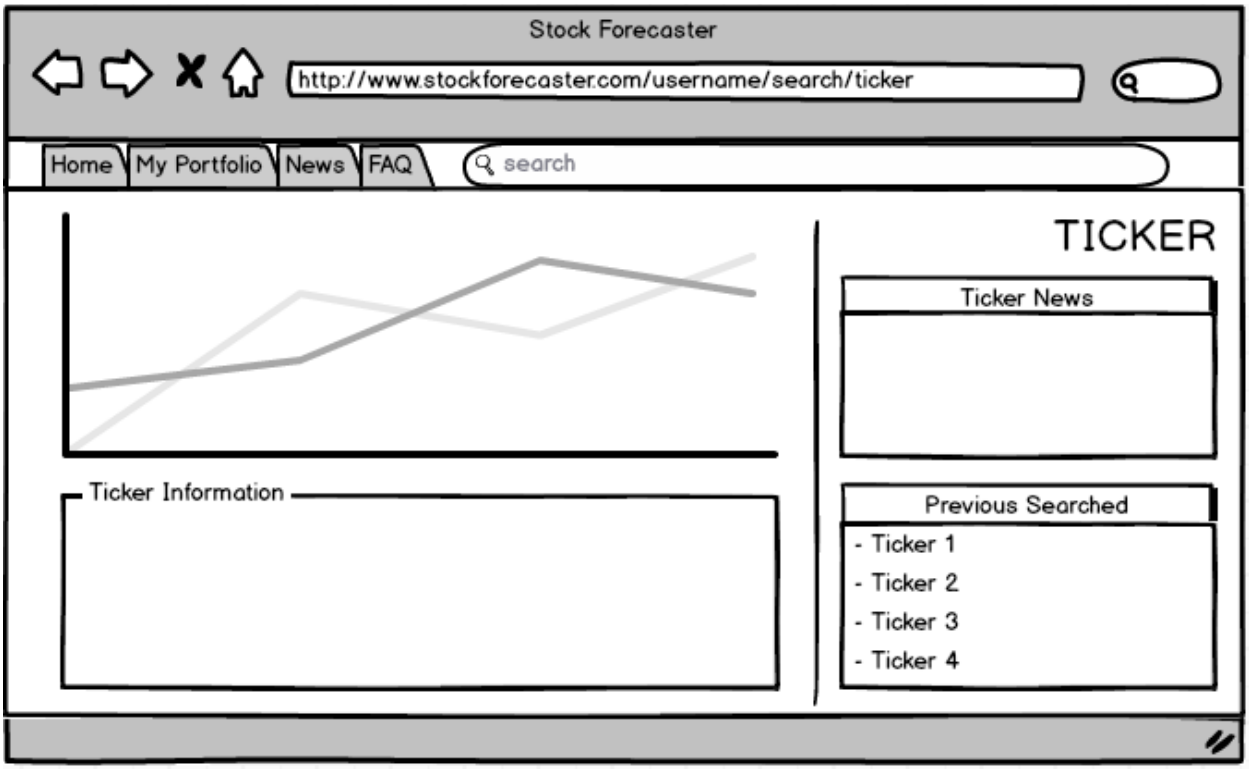
They can then register by filling in the information and hitting the "Register" button. After registration or logging in, the User will be taken to the home page. The home page itself is customized according to the User's portfolio and the current stocks they have. The home page consists of all of the User's stock's shown on one graph. It will also show the User which stock is gaining the most income and which one is the biggest loss. On the right hand side of the page, the User will see news articles and snippets of the stocks he/she has in their portfolio.



If they click on the "My Portfolio" tab, the user will be brought to their portfolio. This is where they can remove or add stocks to their portfolio. They can also add and remove by different quantities.



When a user searches for a stock, they will be shown the screen below.



The stock and its ticker will be displayed at the top right hand side of the screen and that is what will identify that this is the page for the given stock. Underneath the title is the latest news about the ticker. Underneath that will be a list of previously searched stocks. This will enable investors to quickly compare stocks. The right hand side will have a plot of the ticker's historical data. This plot will also plot the prediction as an extension of the historical plot data.

6.2 USER EFFORT ESTIMATION

The goal of designing our system was to create a clean and simple to use interface, clearly labeling each feature of the site on the homepage to avoid navigation and provide clarity. The following user effort estimations take the assumption that the user is already logged in and sitting on the home page.

1. Search for a stock prediction/information - 1 mouse click, 3 keystrokes minimum (depending on stock ticker length)

    a. 1 mouse-click on the search field on home page
    b. Enter the stock ticker (minimum 2 keystrokes, average would be 3-4)
    c. Hit enter (1 keystroke)

2. Add a stock to your portfolio - 4 mouse click, 4 keystrokes
    a. Click "My Portfolio" on the home page
    b. Click the textbox to add ticker
    c. Enter ticker info (minimum 2, up to 4 keystrokes)
    d. Clock quantity textbox
    e. Enter quantity (minimum 1 digit/keystroke)
    f. Click "add"

3. Check News/View Portfolio Information/Frequently Asked Questions (1 mouse click)
    a. Click respective link on navigation bar

4. View combined portfolio/biggest loss and gain/portfolio news feed (0 clicks)

7 DOMAIN ANALYSIS

7.1 DOMAIN MODEL

In order to build the domain model, we will need to once again analyze the primary use cases. From there we will be able to derive the important concepts of the system. We will first look at the boundary concepts that directly interact with the actors and then afterwards analyze and find the internal concepts of the system.

7.1.1 Concept Definitions

Table 7-1 will contain all the boundary and internal concept definitions that we obtained from analyzing each actor's responsibility.

7.1.1.1 Boundary Concepts

In order to analyze the boundary concepts, we must look at all the actors and how they interact with the system. The system interacts with five actors. Three of these actors handle the back end of the system and their primary function is to contain the data and information that the website application needs to output to the human actors, the investors and the administrator. We will first describe and break down responsibilities for each actor.

Let's first look at how the database interacts with the system to create a list of responsibilities and concepts. The database's primary purpose is to store user information (login information and their portfolio). Therefore looking at the UC – 7 (Login) and UC – 8 (Register), the database has the following responsibilities

- R1 – Contains all user profile information and portfolio (UserDatabase)
- R2 – Stores newly registered user information (StoreUser)
- R2 – Update user information in the database when it is changed (UpdateUser)
- R3 – Find and retrieve user information (SearchUser)
- R4 – Update changes to MyPortfolio for given user (UpdatePortfolio)
- R5 – Find and retrieve user's MyPortfolio (SearchPortfolio)

We can group these requirements into one main concept and that is **MaintainUserDatabase**. The primary use of our database is to store all of the Investor's information. Which means it will store the Investor's username, password, email, and their portfolio. It will also maintain and update all the records when they are changed by the Investors.

The Yahoo! Finance API actor is where we primarily collect the stock's historical data for calculating the prediction.

- R6 – Collect stock's historical data (DataAquisition)

The Highstock actor is what will generate the linear regression graph of historical data. Although this is a requirement for the given actor, it belongs to a concept that we will define later.

- R7 – Generate graph of historical data and prediction (CreateGraph)

Investors also interact with the system.

- R8 – Search the system for a stock. (SearchForStock)
- R9 – Change password (PWChange)
- R10 – Change Email (ChangeEmail)
- R11 – Edit user information (EditUserInfo)
- R12 – Add stocks to portfolio (AddStock)
- R13 – Remove stocks from portfolio (RemoveStock)
- R14 – Register an account (RegisterUser)
- R15 – Login and Logout (SystemAccess)

We can group these requirements into four concepts. The first will simply be **SearchForStock**; this is its own concept because it is an integral part of our system. The next three concepts are **EditStockPortfolio** {R12 and R13}, **EditUserProfile** {R9, R10, R11}, and **UserAccess** {R14 and R15}.

7.1.1.2 Internal Concepts

We can extend and break down some of the boundary concepts to create some of the internal concepts. Once again we will look at each individual actor.

Administrator:

- R16 – Authenticate user at login (AuthenticateUser)
- R17 – Check if any user information has been changed (UserUpdateCheck)
- R18 – Check if MyPortfolio has been edited (PortfolioUpdateCheck)
- R19 – Know if a user is logged in or logged out (CheckUserAccess)

The Administrator is used to manage the users in the system and to know if they are logged in or not. It primarily checks if the user belongs in the system or if any of the user's information has been updated. The administrator then alerts the database that information has been changed and sends it the updated information. Therefore these requirements are grouped together to form the **ManageUser** concept. It is also one of our use cases.

After we have obtained the data from the Yahoo! Finanace API, we will need to process this data.

- R20 – Maintain a record of collected data (StockDatabase)
- R21 – Validates that searched stock is in StockDatabase (FindStock)
- R22 – Process numerical data for a given stock (ProcessStockData)
- R23 – Calculate a prediction (CalculatePrediction)
- R25 – Maintain a record of general stock information (StockInfo)
- R26 – Maintain a record of news articles and twitter feeds (News)
- R27 – Load Stock page (StockPage)
- R28 – Load News page (NewsPage)
- R29 – Load Home page (HomePage)
- R30 – Load MyPortfolio page (PortfolioPage)
- R31 – Load FAQ/Help page (FAQPage)

These requirements fall under three concepts. One concept is **StockDatabase**; R16 and R6 (DataAquisition) belong in this concept. The data collected from Yahoo! Finance's API is then stored in a file inside the system. We will call it a database but it is not an external user like the User Database. The next concept is simply **PredictionCalculator**, and it will contain R17 and R18. The next concept is **Webpage** {R7, R23, R24, R25, R26, R27, R28, R29, and R30}. All of these requirements fall under the user interface.

Below is the summary of the concepts

Responsibility Description	Type	Concept
Contains all the user information including their portfolio	K	UserDatabase
Maintaining user information. It stores new users and updates all the information and portfolios of existing users in the system.	D	MaintainDatabase
Allows the user to change their password, email, and edit their information.	D	EditUserInfo
Allows the user to add and remove stocks from their portfolio.	D	UpdatePortfolio
Allows user to register, login, and logout of the system	D	UserAccess
Allows the Administrator to authenticate users that log into the system. It also lets the Administrator check if the user has edited their profile or portfolio.	D	ManageUser
Contains all the stock data and basic stock information	K	StockDatabase
Obtains and collects data from Yahoo! Finances API and stores them in the StockDatabase	D	requestStock
Searches and Finds the stock that the user requested	D	SearchForStock
It processes the numerical data from the StockDatabase and calculates a prediction for the stock. Afterwards it generates a graph of all the historical and predictive data.	D	ObtainPrediction
Loads all the pages and graphs that the website needs for the users to view and navigate.	D	Webpage

7.1.2 Association Definitions

Concept Pair	Association Description	Association Name
UserDatabase ↔ MaintainDatabase	Updates any user information and portfolio changes to the UserDatabase	Updates data
(EditUserProfile, EditStockPortFolio) ↔ ManageUser	Detects if changes have been made to the user information or portfolio.	Conveys updates
ManageUser ↔ MaintainDatabase	If a change has been detected, it lets the database know and provides the changed information	Provides updated data
UserAccess ↔ ManageUser	Authenticates users when they login. Keeps track of whether or not users have logged in or out of the system	Enable/Disable System Access
SearchForStock ↔ StockDatabase	Finds the requested stock in the StockDatabase	Provides requested data
DataAquisition ↔ StockDatabase	The stock database collects data from the Yahoo! Finance API	Provides and stores data
StockDatabase ↔ ObtainPrediction	Uses the data collected and stored to calculate the prediction.	Calculates Prediction

7.1.3 Attribute Definitions

Attribute Description	Attribute	Concept
String of the given stock ticker symbol or company name	tickerSymbol	SearchForStock
String that contains an error message for when the stock is not found	error	
Data Structure/pointer that contains: a username of type string a password of type string an email of type string a name of type string gender of type char portfolio of type list	user	MaintainDatabase
String of the new password	newPassword	EditUserProfile
String of the new email	newEmail	
String of the given stock ticker symbol	tickerSymbol	EditStockPortfolio
Int of the number of stocks	amount	
Bool that indicates the user is either logged in or not	access	UserAccess
An array of the stock's historical data	data	PredictionCalculator
Data Structure/linked list that contains: array of historical data String of the stock ticker symbol String of the company name	stockData	StockDatabase

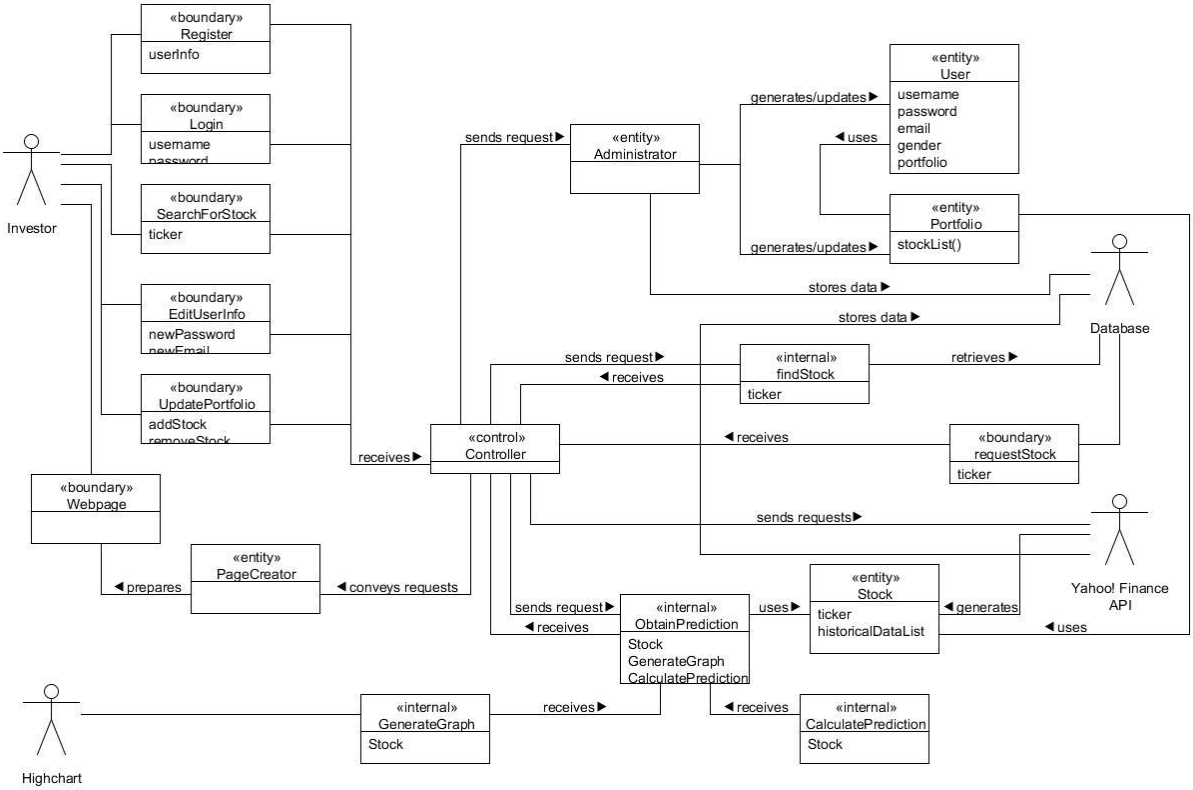


Figure 7-1: Domain Model Diagram

7.1.4 Traceability Matrix

		DOMAIN CONCEPTS										
USE CASE	PW	UserDatabase	MaintainDatabase	EditUserInfo	UpdatePortfolio	UserAcces	ManageUser	StockDatabase	RequestStock	SearchForStock	ObtainPrediction	WebPage
UC1	10									X		
UC2	10										X	X
UC3	10										X	
UC4	15							X	X	X		
UC5	4				X							
UC6	4				X							
UC7	3	X				X	X				X	
UC8	3	X				X		X	X	X	X	X
UC9	4		X				X					
UC10	1						X					
UC11	1	X				X			X			
UC12	4			X	X							
UC13	15							X	X			X
MAX PW		3	4	4	4	3	4	15	15	15	10	15
TOTAL PW		7	4	4	12	7	8	33	34	28	26	28

7.2 SYSTEM OPERATION CONTRACTS

7.2.1 ObtainPrediction

Operation: ObtainPrediction	
Preconditions	Stock has been found in the Database. The prediction has obtained the stock's historical data.
Success Postcondition	A prediction is calculated using a prediction algorithm. The prediction and the historical data is then sent to create a graph.
Failed Postcondition	Error with accessing prediction algorithm. Error accessing Highchart to generate graph.

The system should obtain the requested stock's historical data from the database. Once it is obtained, then it will send it to the PredictionAlgorithm so that a prediction can be calculated. Afterwards, both the prediction and the historical data will be sent to Highcharts to generate a graph.

7.2.2 DataAquisition

Operation: DataAquisition	
Preconditions	User requests for a stock's information that is not currently in the Database
Success Postcondition	Database queries Yahoo! Finance API for requested stock. Stock is found and all data is stored into the Database.
Failed Postcondition	Stock could not be found in Yahoo! Finance API. System loads "Stock Cannot Be Found" error and provides a suggested list of similar stocks.

When a User requests a stock but it's not found in the Database, it should be quickly retrievable. So once a stock is not located inside the Database, a query to the Yahoo! Finance API is made. Once the stock is found in Yahoo! Finance, it should be stored in the database so that the data can be used to calculate the prediction and generate the graphs. If the stock is not found in Yahoo! Finance, then an error webpage is loaded with a suggested list of stocks that are similar to the searched stock.

7.2.3 SearchForStock

Operation: SearchForStock	
Preconditions	There is a search bar at the top of every webpage
Success Postcondition	System loads page with all stock information displayed Graphs are displayed for the stock
Failed Postcondition	System returns “Could not find stock” error A list of links to assumed stocks is displayed under the error

At the top of every page there should be a search bar so that users can search for a new stock, so this is a *precondition*. The *postcondition* is that as the search is being done, if the stock exists in the Database, which it should if it is a valid entry, the stock’s information is displayed on the screen. If the stock is invalid, the “Could not find stock” error comes up and links to “assumed” stocks with similar spelling are displayed.

7.3 MATHEMATICAL MODEL

7.3.1 Moving Average Model

In order to smooth out the market fluctuations, a popular technique is to use the average of values over different subsets. This method is called the moving average model where the average of the varying prices of the stock is taken in periods. The moving average is a lagging indicator since the predication is made on past prices. The two main ways to implement is the simple moving average model and the exponential moving average. The difference is that the exponential moving average gives more weight to more recent prices. In this project we used the simple moving average and used the period as 200 days. So, the average price is taken over the past 200 trading days. For instance, the first average price is taken from day 0 to day 200 the second average is taken from day 1 to day 201. This trend is continued until the current price is taken into account. The period is always kept constant at 200 days. This

number was chosen from the research the group did which lead us to believe that 200 days will provide a good balance between placing emphasis on past prices as well as more recent prices.

7.3.1.1 Pseudo Code

```
def mean200(start_date, stop_date, table):
      # this function calculates the 200 day average
      period = 200

      # closing prices from the start date to the stop date
      # stop - start = 200
      prices = table[start_date:stop_date].Close

      sum = 0
      mean = 0

      # compute the mean of all the prices
      for val in prices:
            sum = sum + val
      mean = sum / period
      return mean
```

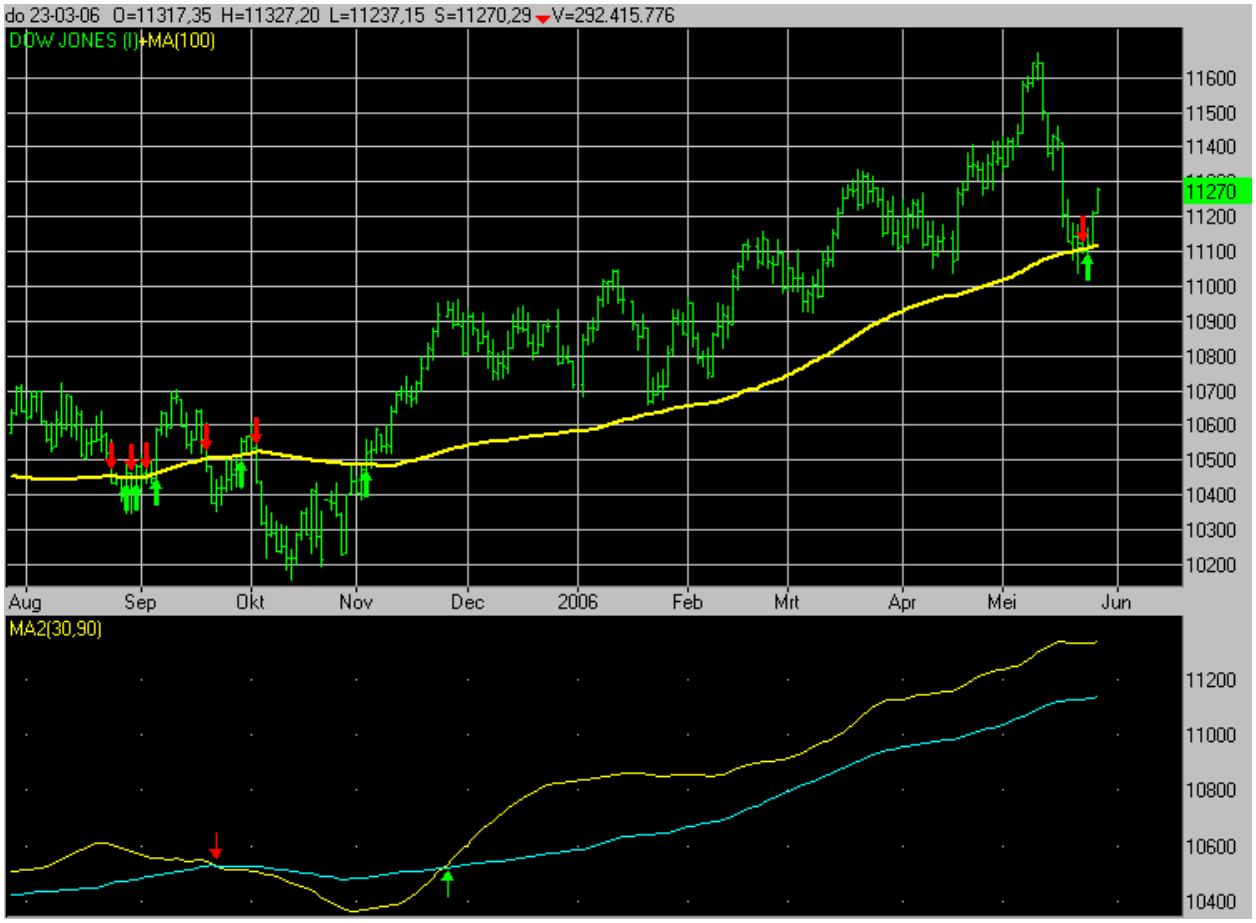


Figure 7-2: The yellow line represents the moving average of the market fluctuations.

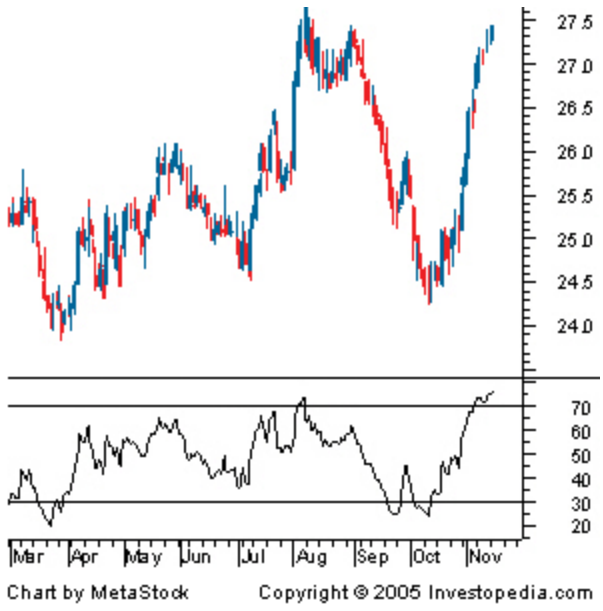
$$SMA = \frac{p_M + p_{M-1} + \cdots + p_{M-(n-1)}}{n}$$

Figure 7-3: Equation used for simple moving average. Where P_M refers to the stock price of day M. N is the period of the average in this system, n is 200.

7.3.2 Other Predictive Models

With moving average being the base of our prediction model, we hope to further expand predictions further by adding more prediction models. An example is Relative Strength Index (RSI) which is an oscillator that measures momentum providing leading predictions instead of lagging like the moving average model. We will incorporate this by having the system either sourcing these indicators or use the mathematical models used to create them to have the program calculate the predictions on its own. From there we would use the stock's current direction and momentum to determine whether the stock will continue to move in that direction or if a change in direction is going to come. The direction and momentum can be measured by an oscillator, in the case of RSI, a number is generated between 0 and 100, and if the value is approaching 30 and the stock value is depreciating and possibly undervalued based off of the average duration of a change in price and likewise, when the oscillator approaches the value of 70, the stock value may be overvalued. The following is the formula for generating the RSI where RS is average x days up closes divided by average x days down closes.

$$RSI = 100 - \frac{100}{1 + RS}$$



As we can see from the above figure we can see the actual market movement as seen in the top portion of the graph correlated with the RSI prediction model in the bottom portion of the graph.

The formula for R Pivots, support, and resistance could also be used as input as they can be important price points to the market, and it becomes less likely that a stock will move below support or above resistance the more times that resistance/support was tested and held up as a significant price point.

8 PLAN OF WORK

Weekly Group Meetings: Tuesday at 4:30pm

8.1 PROGRESS AS OF FEBRUARY 23RD, 2014

8.1.1.1 Database and Data Collection Progress (Sivaramharesh Siva and Neha Desai):

They have done some research of how to collect stock data. Siva has been playing with the Yahoo! Finance API and has managed to obtain Google's stock value information dating back to 2004. They have determined that the database will hold the opening and closing price of the Fortune 500 companies as well as their stock tickers. This information will be pulled from Yahoo! Finance. The purpose of the database is to make the process of obtaining the data faster and more efficient, than having to keep pulling from the website. They will be creating tables in an Oracle database and manipulating the information using SQL.

8.1.1.2 Front-End Web Development Progress (Vinay Panjabi and Krisha Paula Olanday):

Vinay has created the Git repository. They have come up with the overall design of the website for the on-screen requirements. All of the website design is finished. All we have to do is begin with the development of the static webpages, such as the login and registration, and FAQ page. They will also create the skeletons for the other pages that require the database.

8.1.1.3 Prediction Research and Algorithm Progress(Vinay Shivakumar and Jonathan Haas):

Siva has provided Vinay and Jon the information on the Moving Average Prediction Model. They have made the decision that the Moving Average Prediction Model is a good start. However it should not be the standalone model for prediction. They have found other models such as the MACD, RSI, and Exponential Moving Average and are currently doing more research on them. Siva has created the algorithm for the Moving Average Model. Their decision still stands that it will not be the only prediction model to consider.

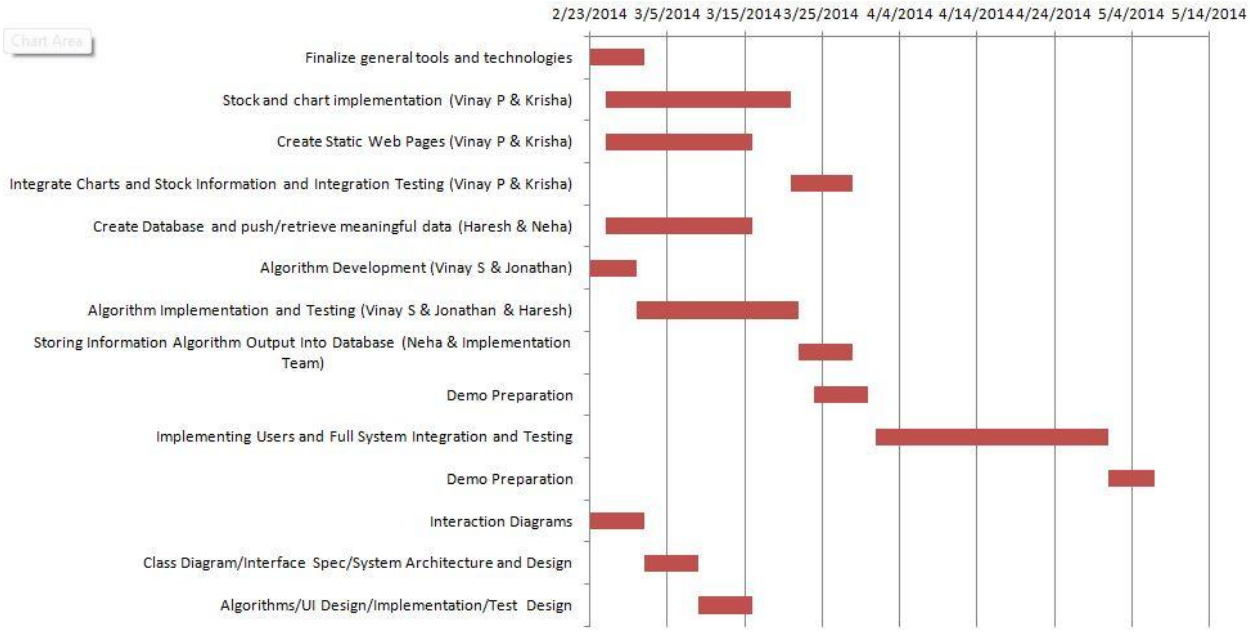


Figure 8-1: Future Plan of Work

The plan of work focuses mainly around the deliverables for this project. By demo number one we plan on having substantial progress the individual pieces developed by each sub group so the main focus for the second demo is creating users as well as full integration and testing. Part of what will make this plan of work successful is the specific roles that each sub team is being given and allowing each to master their own art by the time the first demo arrives, allowing for further development to occur thereafter as a larger group. It is important that we use the first week of finalizing general tools and techniques in order to ensure all planned technology will be able to be integrated into a working final product. This decision will be based off of research of previous projects that have been developed in the past as well as our own personal experiences and intuition.

9 REFERENCES

1. http://www.investopedia.com/terms
2. http://en.wikipedia.org/wiki/Stock_market_prediction
3. http://www.dummies.com/how-to/content/stock-investing-for-dummies-cheat-sheet.html
4. Sarkar, Asani. “Equity Market Valuation”.
5. *Master Trader – Top Strategies for Market Success.* Tigrent Learning. Print.
6. https://en.wikipedia.org/wiki/Stock