16:332:568 S15

Software Engineering
Web Application

Web Based Stock Forecaster

First Report of Group 2

Chaoran Fu, Kai Kang, Yu Sun

Yue Wu, Wuyang Zhang

March 5, 2015

“All group members contributed equally.”

Table of Contents

1. Background

The history of stock market goes back to 12th century in France where people were concerned with managing and regulating the debts of agricultural communities on behalf of the banks. Because these men also traded with debts, they could be called the first brokers[1]. Over hundreds of years this kind of market never fails to fascinate human being. Now there are now stock markets in virtually every developed and most developing economies. Not just for it is one of the most important ways for companies to raise money but also because it allows businesses to be publicly traded, and raise additional financial capital for expansion by selling shares of ownership of the company in a public market. Nowadays this magic place gathered businessmen and investors all over the world while businessmen gain capital by selling shares of the companies. On the other hand all the investors have a urge to earn money by purchase those company shares and hoping those companies could prosper. However it is human's nature to seek some effortless way of becoming rich in a very short of time. Regardless of the actual economic development, there should be an easy way to earn profit in the stock market. This is where the financial speculation come in to place. The successful prediction of a stock's future price will no doubt give investors significant payoff and continuously excite their nerves.

We will assume that the stock price is predictable and the most two common prediction methodologies are fundamental analysis and technical analysis [2]. Fundamental analysis is built on the belief that human society needs capital to make progress and if a company operates well, it should be rewarded with additional capital and result in a surge in stock price. That is why the fundamental Analysts are concerned with the company that underlies the stock itself. Technical analysts are not concerned with any company's background. They try to find a way to determine the future price of a stock based solely on the (potential) trends of the past price. Using charts search for archetypal price chart patterns, technical analysts also widely use market indicators of many sorts, some of which are mathematical transformations of price, often including up and down volume, advance/decline data and other inputs.

Huge profits will come along with risks and that same volatility in the market that can be exploited for gain, can be the cause for loss as well. No doubt stock market will always attract new and passionate traders but the risks will also keep them away at some point. They lack the appropriate tools needed for the decision making process when compared to existing professional traders. Any individuals who would consider spending their spare money trading stocks but may have second thoughts due to the lack of time, resources and knowledge in the market field.

2. Introduction

The goal of our project is to create a web-based system. The foremost purpose of the system should be to predict future stock prices. Our website should not only be able to automate the prediction process but also produce the prediction results with the high accuracy. Any results generated by the system should also come with a reasonable confidence value shown to the user the possibility of the given decision of “ask”, “bid” or “hold” is being correct. The system should use as many as possible various methods to gauge stock trends to increase the prediction accuracy because this will be concerned most by our users. The results our system displays to the user should be simple, straightforward, easy to understand and easy to remember. Our user should be able to obtain all the prediction results and illustrative diagram at a few mouse clicks away.

Some users maybe experienced with the stock market and they certainly will not be satisfied with simple prediction results. That is why the system should also have a professional mode that could access the analysis used to get to the specific result and reveal an explanation to the user. Thus, the system will serve both sides of the crowd. Users who want easy and quick results and who want more understanding of the technical analysis will both have a great experience using our website.

Most investment advisors suggest maintaining a diverse investment portfolio in order to reduce the risk of investment loss. So if a user would like to broaden the investment portfolio with multiple stocks, our system should provide a proper solution. An account based system is required. The system should be able to track and view multiple predictions simultaneously according to the user's personal interests. Also, when the system notice an impending decision, it should alert and inform the user of any decision may come immediately.

To achieve all the features discussed above, we need several steps to build every bricks of our project. The first step is to build up a highly organized database that will periodically download historical and real time stock price from open source websites. The next step is to implement prediction models and also keep improving the algorithm. Then, we build up the website and the mobile workspace as well.

As the report will show, at this point we have complete the first phase of the project. Our application runs continuously as a background process and periodically retrieves stock information, parses the received responses, and stores the extracted parameters into a local relational database.

3. System requirement

3.1 Enumerated Functional Requirement

INDENTIFIER	PW	REQUIREMENT
REQ1	5	The system shall collect current market data.
REQ2	5	The system shall collect historical market data.
REQ3	5	The system shall provide suggested profit stock based on different requirement.
REQ4	5	The system shall provide top 5 stocks picks that best fit for long term prediction.
REQ5	5	The system shall provide top 5 stocks picks that best fit for short term prediction..
REQ6	4	The system shall provide the choice to display the Simple Moving Average indicator (SMA).
REQ7	4	The system shall provide the choice to display the Exponential Moving Average indicator (EMA).
REQ8	4	The system shall provide the choice to display the Money Flow indicator(MFI).
REQ9	4	The system shall provide the choice to display the MACD indicator.
REQ10	4	The system shall provide the choice to display the Relative Strength Index(RSI) indicator.
REQ11	4	The system shall provide the choice to display the Bollinger Bands indicator.
REQ12	4	The system shall provide the choice to display the Stochastic indicator.
REQ13	3	The system shall let administrator add and remove stocks.
REQ14	3	The system shall provide the choice to compare any one stock with Nasdaq index.
REQ15	3	The system shall search any stock for current or historical data.

REQ16	2	The system shall provide login and register to users.
REQ17	2	The system shall visually track user finance via graphs and charts.
REQ18	2	The system shall provide prediction subscription to users through emails.

Table1: Enumerated functional requirement for a web based stock forecasting system

For this system, the main solution is to provide an automated process of predicting future stock prices. In order to improve the accuracy of prediction, the system must need to collect current and historical market data first. Therefore, REQ 1 and REQ 2 is given highest priority. Furthermore, the core function of the system is to help user forecast stock price so that accurate predictions can be made. Thus, REQ 3, REQ4 and REQ 5 respectively are also given highest priority.

Requirement REQ 6- REQ 12 allow users to have different prediction strategy, based on recognizing different indicators, to have a more accurate and convincing prediction. These are added features and not essential (although highly desirable) to the core functionality of the system as stated before, and thus REQ6-REQ12 were given the second highest priority.

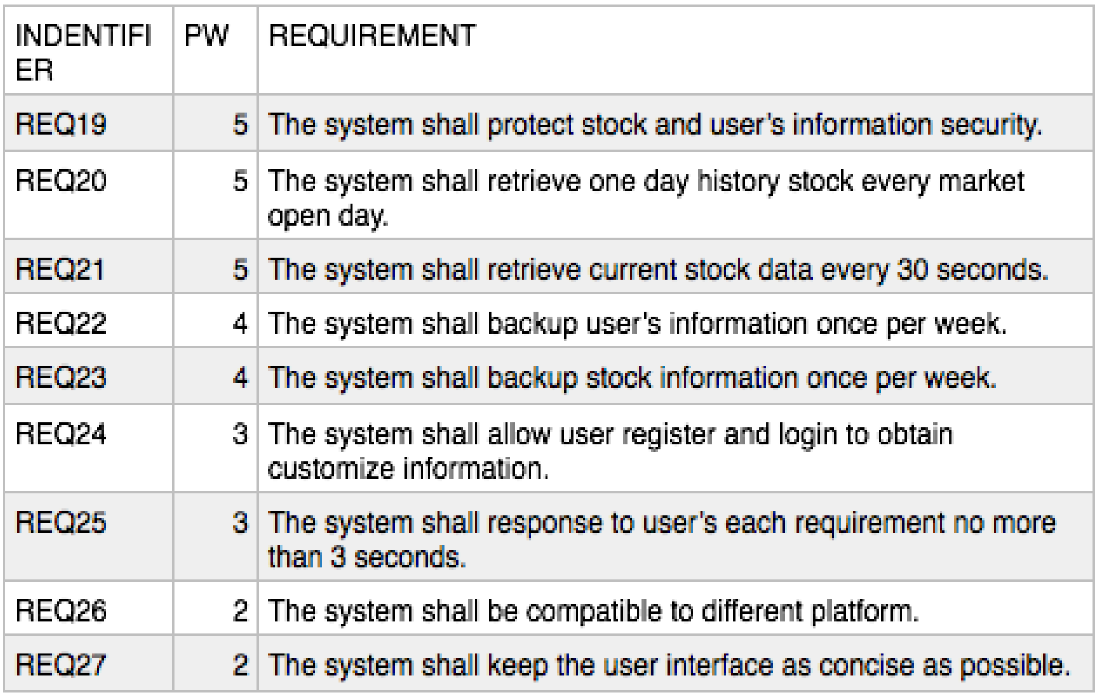
Naturally, not all listed companies can be accounted for. Thus, before predictions can even be made, the user must select the range of stocks that the system will handle. In this system we set the number as five. Therefore, the system shall let users add and remove up to five preferred stocks, which is REQ 13.

As professional financial analysts, they cannot only focus on the stocks that have the absolute profit, rather shall compare the stock price with the Nasdaq index to see if this stock is relatively performed better than others. This features is realized by REQ 14.

REQ 15 allows users to be able to search any stock for current to historical data. REQ 16 deals with allowing users to register for an account with the system and login.

REQ 17 and REQ18 provide users much more convenience to keep touch with stock price.

3.2 Enumerated Non-functional Requirement



INDENTIFIER	PW	REQUIREMENT
REQ19	5	The system shall protect stock and user's information security.
REQ20	5	The system shall retrieve one day history stock every market open day.
REQ21	5	The system shall retrieve current stock data every 30 seconds.
REQ22	4	The system shall backup user's information once per week.
REQ23	4	The system shall backup stock information once per week.
REQ24	3	The system shall allow user register and login to obtain customize information.
REQ25	3	The system shall response to user's each requirement no more than 3 seconds.
REQ26	2	The system shall be compatible to different platform.
REQ27	2	The system shall keep the user interface as concise as possible.

Table2: Enumerated non-functional requirement for a web based stock forecasting system

Since the system will be storing user accounts, REQ 19 needs to be specified for security purposes. The system insures that all user’s and stocks’ information will be backup at least once per week in case the database crack down at times. That’s REQ 22 and REQ 23 should realize.

For REQ 24., the system insures users can complete every time of register or login in a successful way.

Furthermore, the customer intends for the service to be as easily accessible as possible and thus REQ 26 finally insures that the system is able to be used for not only desktop browsers but also smartphones and tablets as well. This requirement is of the least priority because we will be working on the functionality for modern desktop web browsers first and then optimize for mobile app if we have enough time.

4. System Design

4.1 Overall system design

The below figure1 is the basic system design diagram

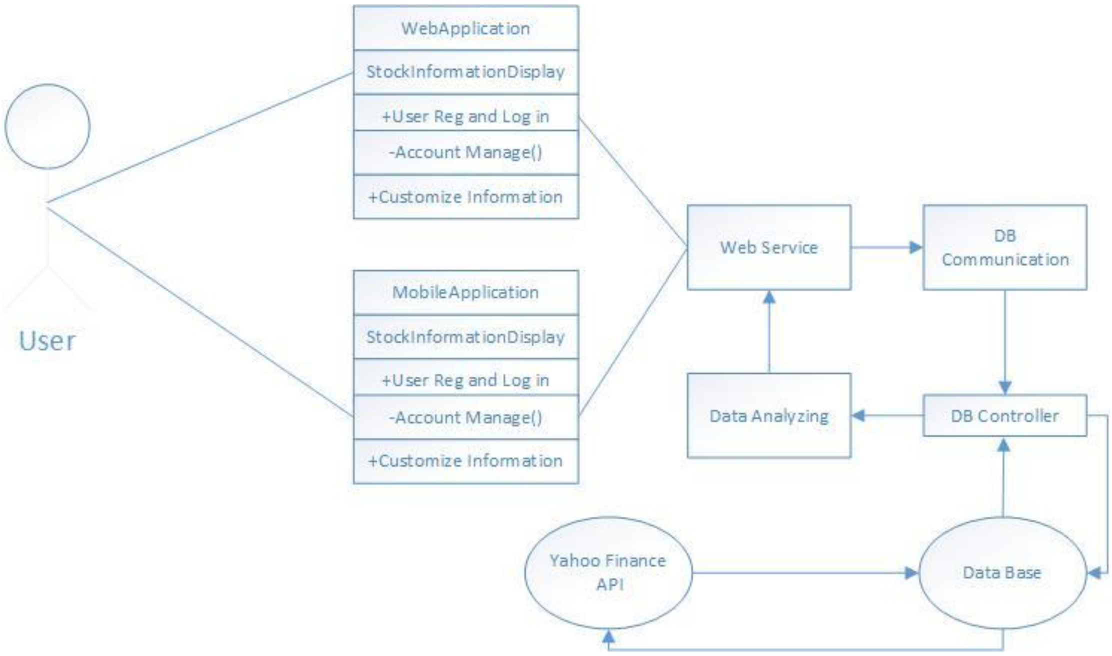


Figure 1. System level interaction diagram

The key point for the stock forecasting system design is to receive user's order, retrieving and processing data, and then send the processed data back to user. In this case, we need to design eight parts to complete the task. First, we decide to design two end points application, one for the web and one for the phone app. This is because stock market is changing at every moment, one minute may result in a huge turnover, therefore, to make the market information accessible for our customers in every place and every time is important, so design another mobile application is necessary.

Both endpoints may have the capability to display basic stock information, and let the user to register and log in to view some particular stock and information he or she may interested in. Besides, the application should have the ability to remember user's preference, so every time user log in, the application will display the most important information that the user want to see.

The web Service class is use to obtain the user’s command and send the command to background to process. Also, the web service need to send the processed data and information back to the endpoints to display.

The DB communication class act as two goals. First handle the request from web service and processed the command to data base understandable language. Second, it connect to the data base controller to ask for requested data.

The DB controller class use to decide if the request is legible, if yes, allow the database to process the request, if not, deny the request. This is use to protect the data base to become overloaded and secure the database.

The database is used to store all the information including the stock market data and user’s account information. Also, the database need to store the data comes from Yahoo Finance.

4.2 Endpoint interaction design

The end point should response to the user’s request. The ability is shown in figure 2.

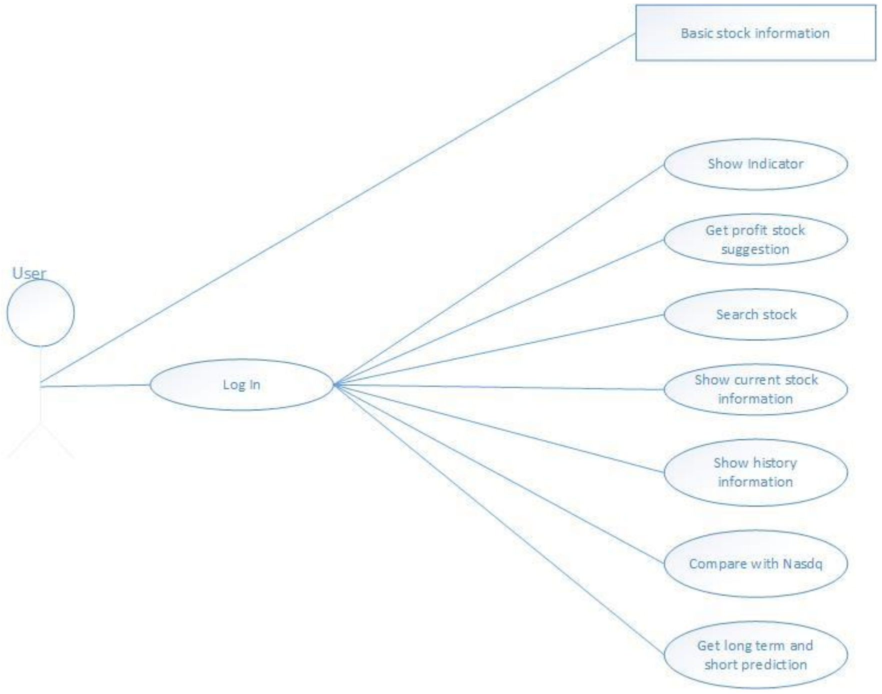


Figure 2 Client side use case

3. Background interaction design

The most important goal for background design is to have the ability to collect data and processing them. The detail can be shown in figure 3.

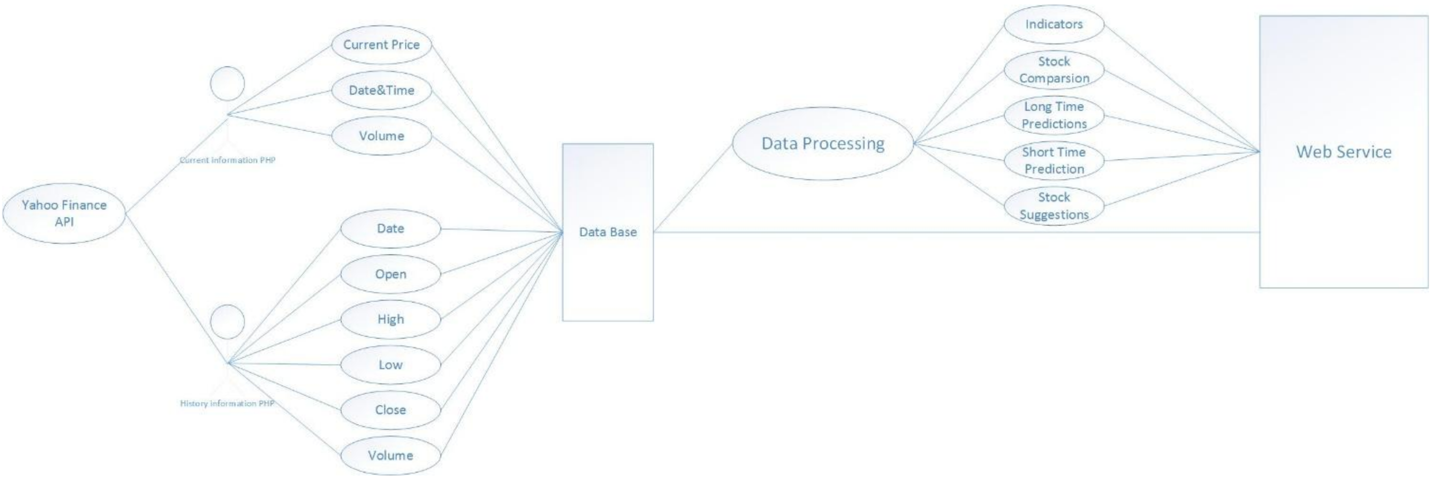


Figure 3 Back interaction design

First, there are two basic PHP scripts that runs background continuously and obtain basic stock data from Yahoo Finance periodically and data base receive and store all the data. Then the data processing class call for this basic data and do analyzing, then the web service receive both the basic data and processed data as user required.

4. Overall interaction.

As the background and endpoints part work together, the system can display request information to the user. The overall interaction can be shown in figure 4.

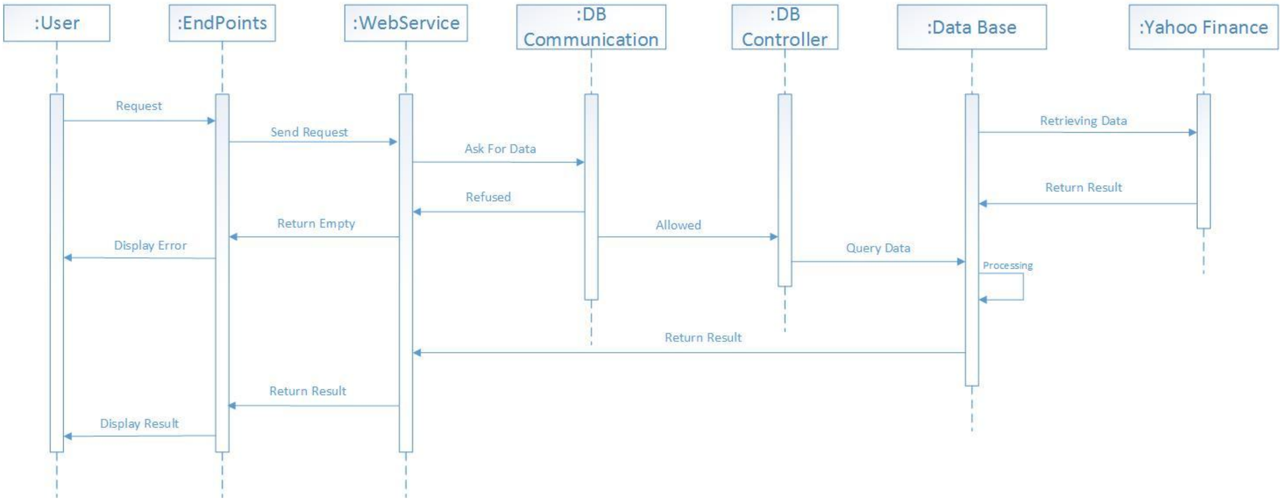


Figure 4 Interaction Diagram

The user ask for information from endpoints and trigger the system, if the ask is not legal, then refuse the request, if the ask is legal, then processing and return the result from database. Meanwhile, the data base retrieve necessary from Yahoo Finance continuously without interruption.

Stock information retrieve from Yahoo Finance

History information	Real time information
Stock	Stock
Date	Current Price
Open Price	Date
High Price	Time
Low Price	Volume
Close Price	
Volume	

Table 3 Stock Information

5. Functional Requirements Specification

5.1 Stakeholders

The stock forecast system concentrates on providing the prediction of stock price in the future time on stocks that users require.

Register Users

Register users are defined as the most principal user objects. They have access to pick any stock they interest and would like to observe from the stock list within the database. They have access to add indicators on stock interactive charts so as to observe and monitor the tendency line in detail of stocks. They can also calculate stock return profits by inputting stock parameters. Moreover, the system will provide ask or bid suggestion of preferred stocks and notify users about the recent price change of stocks.

Systems architects and developers

System architects and developers are defined as system supporters. They take responsibility on building the stock forecast system satisfying the users' requirements, testing and maintaining they system, and providing technical support to other stakeholders.

5.2 Actors and Goals

User (Initiating type)

Goals: to interact with the stock forecast system, to choose preferred stocks and obtain ask and bid suggestions.

Administrator (Initiating type)

Goals: to collect current and history stock price, access, manage and maintain the database as well as the website, providing stock service to the users.

Yahoo Finance server and database (Participating type)

Our server and database (Participating type)

5.3 Use Cases

5.3.1 Casual Description

The summary of use cases are illustrated as following.

UC-1: Login

Allowing a user to access and manage the account and using the system features provided specifically for registered users. [Derived from REQ16]

UC-2: Register

Allowing a visitor to become a registered user through completing the registration procedure. [Derived from REQ16]

UC-3: Manage Stock

Allowing the administrator to add or remove a stock from stock forecast database. (<include> login) [Derived from REQ13]

UC-4: Search Stock

Allowing the user to search from the prediction list one specific stock which has been analyzed and are available to users through the stock ticker or the company name. [Derived from REQ2, REQ15]

UC-5: Add Stock Indicator

Allowing the user to add some indicators on the stock price data to analysis the stock performance from distinct perspectives in details. (<include> login) [Derived from REQ6-REQ12]

UC-6: Suggest Ask or Bid

Allowing the user to execute the system predicting stock price and affording the asked and bid strategies. (<include> login) [Derived REQ3, REQ4, REQ5]

UC-7: Track Stock Price

Allowing the user to create a watch list or portfolio to track the performance of stocks. (<include> login) [Derived from REQ13]

UC-8: Manage Portfolio

Allowing the user to check the profit return of the portfolio by inputting stock parameters. (<include> login) [Derived from REQ13, REQ14, REQ17]

UC-9: Update Stock Price

Allowing the timer to send the request of the loading the current stock price from Yahoo Finance to the database. (<include> login) [Derived from REQ1]

UC-10: Notify Price Change

Allowing the user to get notification from the system such as email if the prediction has been changed. (<include> login) [Derived from REQ18]

5.3.2 Use Case Diagram

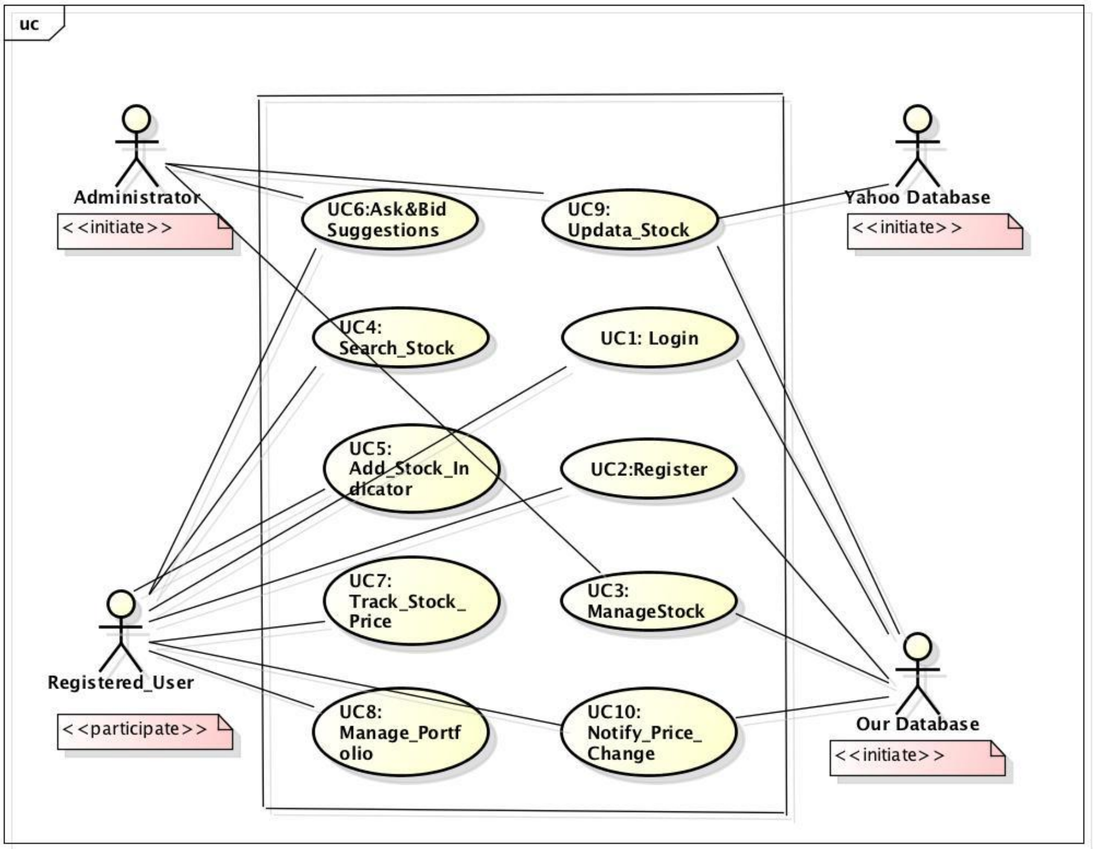


Figure 5 Use Case Diagram

5.3.3 Traceability Matrix

REQ	PW	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8	UC9	UC10
REQ1	5									X	
REQ2	5				X						
REQ3	5						X				
REQ4	5						X				
REQ5	5						X				
REQ6	4					X					
REQ7	4					X					
REQ8	4					X					
REQ9	4					X					
REQ10	4					X					
REQ11	4					X					
REQ12	4					X					
REQ13	3							X	X		
REQ14	3								X		
REQ15	3				X						
REQ16	2	X	X								
REQ17	2								X		
REQ18	2										X
MAX PX		2	2		5	4	5	3	3	5	2
TOTAL		2	2		8	28	15	3	8	5	2

Table 4 Traceability Matrix

5.3.4 Fully Dressed Description

Use case UC-3: ManageStock, allows the administrator to manage stocks in the system. The administrator can add stocks into the system or remove stocks from the system. After choosing stocks, the system would retrieve the data from the price provider and build the stock database. Usually each stock has the unique identifier, ticker symbol. The system would use the ticker to mark the stock for searcher to visit the stock database.

Use Case UC-3 ManageStock

Related Requirements: REQ13

Initiating Actor: Administrator

Actor's Goal: to add or remove stocks in the database that predictions are made on.

Participating Actors: Price Provider, Database

Preconditions: Administrator is logged in.

Success End Condition: All of the chosen stocks are added to the list of stocks within the database that predictions are made on. Historical prices for the given stocks are retrieved from the Price Provider and stored within the database.

Failed End Condition: The administrator is notified that price provider cannot find the particular stock in the web database.

Flow of Events for Main Success Scenario:

Include::Login(UC-1)

1. Administrator enters ticker symbols for the stocks he wants to add or remove.

2. Administrator clicks the "Add Stocks" button of "Remove Stocks" button.

(loop: for each ticker symbol entered, use Remove Stocks for example)

1. System verifies the stock currently already exist within the **Database**.

2. System sends the remove stock request to the **Database**.

3. Database deletes the stock data and returns the information

4. System notifies **Administrator** that the stock is successfully removed.

Flow of Events for Alternate Scenarios:

1a. System cannot find the stock in the database.

1. **System** notifies **Administrator** the stock that has failed to be added.
2. **System** lets the **Administrator** to remove another stock.

Use case UC-5: Add Indicator, allows the user to add some indicators on the stock price data to analysis the stock performance comprehensively.

Use Case UC-5 Add Indicator

Related Requirements: REQ6-REQ12

Initiating Actor: User

Actor's Goal: to add the indicator on the price data.

Participating Actors: Price Provider, Database

Preconditions: User is logged in.

Success End Condition: The user can view the indicator on the price data.

Failed End Condition: The user is notified the indicator requested is not available for the particular stock data.

Flow of Events for Main Success Scenario:

Include::Login(UC-1)

Include::Search(UC-4)

1. **User** enters the ticker symbols for the stocks to search.
2. **System** verifies that the given stock exists within the Database.
3. **System** offers an indicator list for user to choose (such as SMA, EMA, MFI, MACD, RSI).
4. **User** chooses the indictor he wants and sends to system.
5. **System** gets the historical data from the **Database**.
6. **System** calculates the indictor data and sends to Grapher.
7. **System** displays the graph and historical data to the **User**.

Flow of Events for Alternate Scenario:

3a. The historical data cannot support the particular indicator.

1. System notifies the user that the indicator cannot be calculated based on the stock data in the database.

6. Implementation and Results

To get start, we need to implement to methods: downloading current price and history price, and store them into the database.

6.1 Creating Database and Table

To create a database, it's very convenient to do it via phpmyadmin, and we create a database called the StocketForcasting. The next step is to create tables to store the data. Consider there are a lot of tables, we create the database through PHP scripts instead of create them through phpmyadmin. The Following PHP sentence will create a table called CurrentPrice in the database:

```
$sql = "CREATE TABLE IF NOT EXISTS CurrentPrice
      (
        ID INT(6)   NOT NULL AUTO_INCREMENT PRIMARY KEY,
        Stocks VARCHAR(15),
        Price DECIMAL(10,3),
        Volume int(10),
        VolumeChange int(10),
        Date VARCHAR(20) DEFAULT '0',
        Time Varchar(20) Default '0'
              )ENGINE=MyISAM";
```

As we can see from above, this table use to store data related to stocks current price information. The data include stocks name, real time price, the date and time, accumulative total volume per day and the volume change in a short period. We choose to use the engine MyISAM instead of InnoDB because InnoDB has some problem in display the whole information—though it's not effect the result.

The next table use to store information related to history price from Google.

```
$sql = "CREATE TABLE IF NOT EXISTS Google
        (
          ID INT(6)    NOT NULL AUTO_INCREMENT PRIMARY KEY,
          Stocks VARCHAR(15),
          Open    DECIMAL(10,3),
          High DECIMAL(10,3),
          Low DECIMAL(10,3),
          Close DECIMAL(10,3),
          Volume INT(10),
          Date VARCHAR(20) DEFAULT '0'
          )ENGINE=MyISAM";
```

This table store the stocks name, open price, high price, low price, close price and the volume and date. The other table to obtain history price have the same structure except different company names.

Below figures are the structure of the database, current price table and history table with real data obtain from Yahoo Finance.

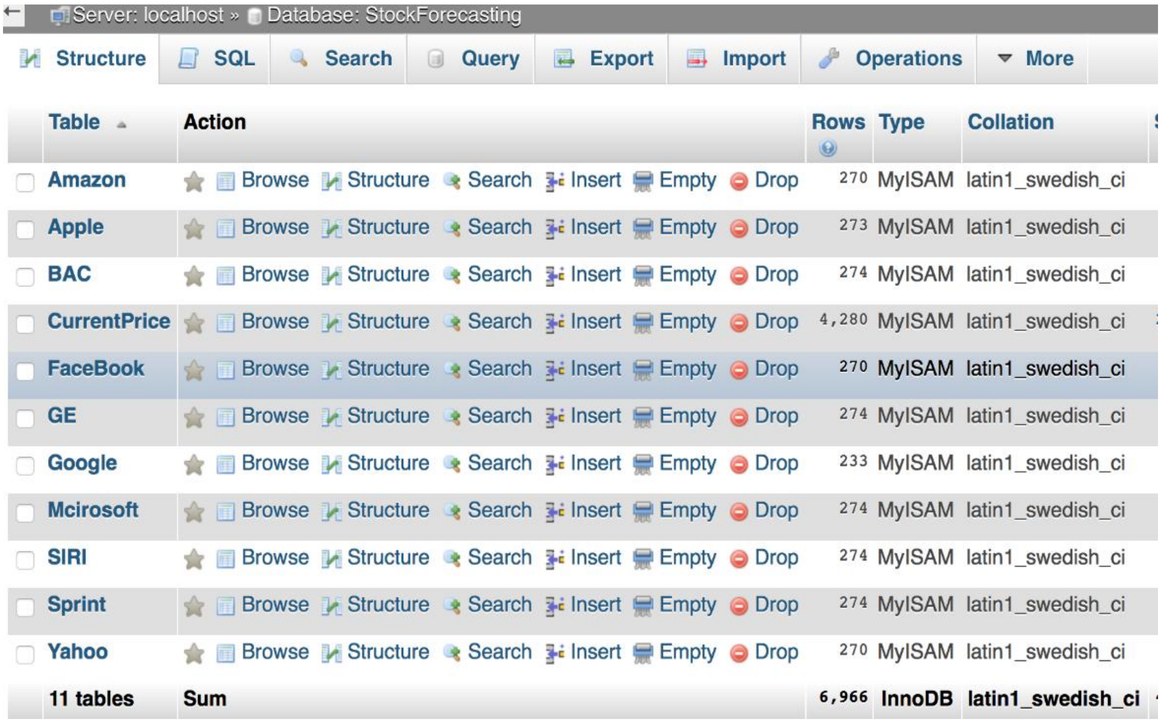
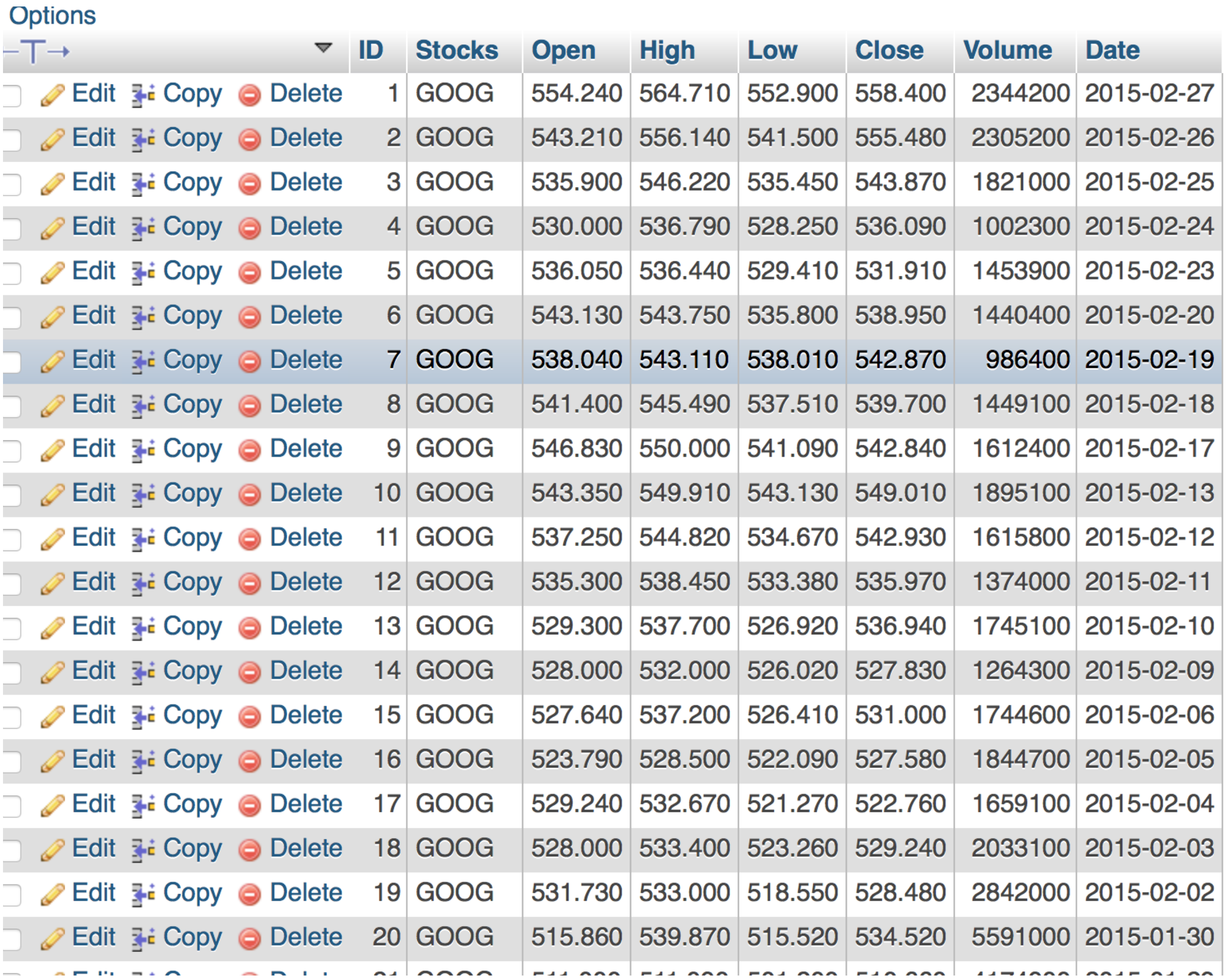


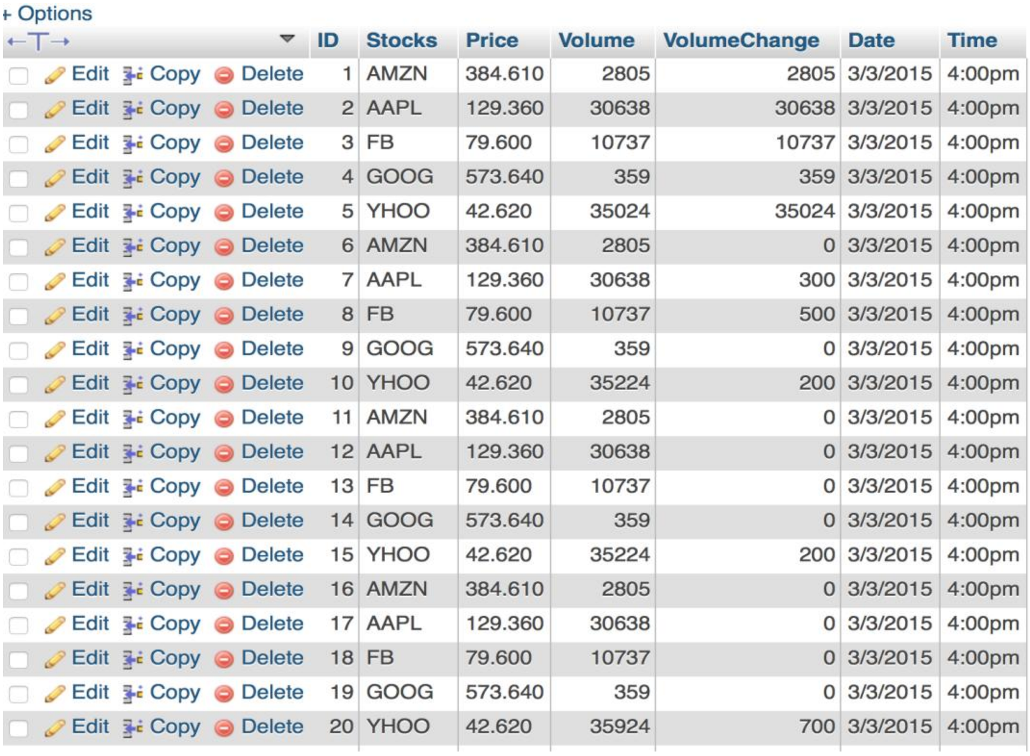
	Table	Action	Rows	Type	Collation
☐	Amazon	Browse Structure Search Insert Empty Drop	270	MyISAM	latin1_swedish_ci
☐	Apple	Browse Structure Search Insert Empty Drop	273	MyISAM	latin1_swedish_ci
☐	BAC	Browse Structure Search Insert Empty Drop	274	MyISAM	latin1_swedish_ci
☐	CurrentPrice	Browse Structure Search Insert Empty Drop	4,280	MyISAM	latin1_swedish_ci
☐	FaceBook	Browse Structure Search Insert Empty Drop	270	MyISAM	latin1_swedish_ci
☐	GE	Browse Structure Search Insert Empty Drop	274	MyISAM	latin1_swedish_ci
☐	Google	Browse Structure Search Insert Empty Drop	233	MyISAM	latin1_swedish_ci
☐	Mcirosoft	Browse Structure Search Insert Empty Drop	274	MyISAM	latin1_swedish_ci
☐	SIRI	Browse Structure Search Insert Empty Drop	274	MyISAM	latin1_swedish_ci
☐	Sprint	Browse Structure Search Insert Empty Drop	274	MyISAM	latin1_swedish_ci
☐	Yahoo	Browse Structure Search Insert Empty Drop	270	MyISAM	latin1_swedish_ci
	11 tables	Sum	6,966	InnoDB	latin1_swedish_ci

Figure 6 Database structure



Options				ID	Stocks	Open	High	Low	Close	Volume	Date
□	Edit	Copy	Delete	1	GOOG	554.240	564.710	552.900	558.400	2344200	2015-02-27
□	Edit	Copy	Delete	2	GOOG	543.210	556.140	541.500	555.480	2305200	2015-02-26
□	Edit	Copy	Delete	3	GOOG	535.900	546.220	535.450	543.870	1821000	2015-02-25
□	Edit	Copy	Delete	4	GOOG	530.000	536.790	528.250	536.090	1002300	2015-02-24
□	Edit	Copy	Delete	5	GOOG	536.050	536.440	529.410	531.910	1453900	2015-02-23
□	Edit	Copy	Delete	6	GOOG	543.130	543.750	535.800	538.950	1440400	2015-02-20
□	Edit	Copy	Delete	7	GOOG	538.040	543.110	538.010	542.870	986400	2015-02-19
□	Edit	Copy	Delete	8	GOOG	541.400	545.490	537.510	539.700	1449100	2015-02-18
□	Edit	Copy	Delete	9	GOOG	546.830	550.000	541.090	542.840	1612400	2015-02-17
□	Edit	Copy	Delete	10	GOOG	543.350	549.910	543.130	549.010	1895100	2015-02-13
□	Edit	Copy	Delete	11	GOOG	537.250	544.820	534.670	542.930	1615800	2015-02-12
□	Edit	Copy	Delete	12	GOOG	535.300	538.450	533.380	535.970	1374000	2015-02-11
□	Edit	Copy	Delete	13	GOOG	529.300	537.700	526.920	536.940	1745100	2015-02-10
□	Edit	Copy	Delete	14	GOOG	528.000	532.000	526.020	527.830	1264300	2015-02-09
□	Edit	Copy	Delete	15	GOOG	527.640	537.200	526.410	531.000	1744600	2015-02-06
□	Edit	Copy	Delete	16	GOOG	523.790	528.500	522.090	527.580	1844700	2015-02-05
□	Edit	Copy	Delete	17	GOOG	529.240	532.670	521.270	522.760	1659100	2015-02-04
□	Edit	Copy	Delete	18	GOOG	528.000	533.400	523.260	529.240	2033100	2015-02-03
□	Edit	Copy	Delete	19	GOOG	531.730	533.000	518.550	528.480	2842000	2015-02-02
□	Edit	Copy	Delete	20	GOOG	515.860	539.870	515.520	534.520	5591000	2015-01-30

Figure 7 History price structure

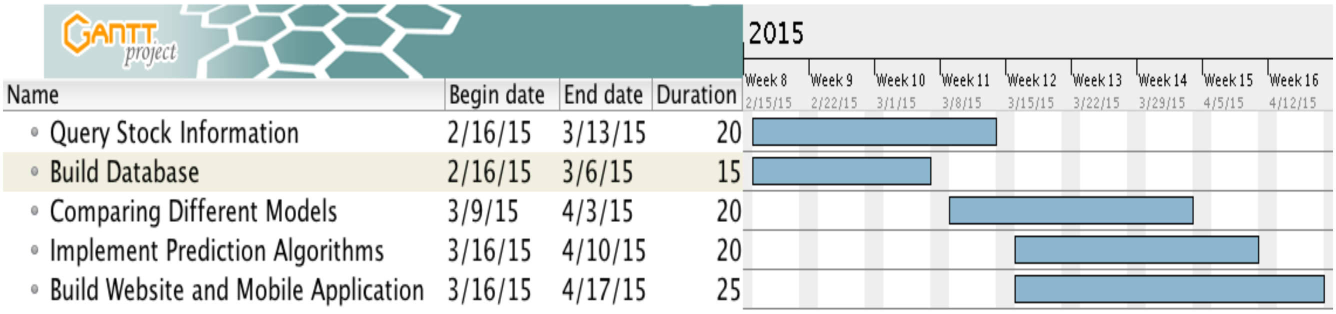


+ Options

←T→	ID	Stocks	Price	Volume	VolumeChange	Date	Time
□ Edit Copy Delete	1	AMZN	384.610	2805	2805	3/3/2015	4:00pm
□ Edit Copy Delete	2	AAPL	129.360	30638	30638	3/3/2015	4:00pm
□ Edit Copy Delete	3	FB	79.600	10737	10737	3/3/2015	4:00pm
□ Edit Copy Delete	4	GOOG	573.640	359	359	3/3/2015	4:00pm
□ Edit Copy Delete	5	YHOO	42.620	35024	35024	3/3/2015	4:00pm
□ Edit Copy Delete	6	AMZN	384.610	2805	0	3/3/2015	4:00pm
□ Edit Copy Delete	7	AAPL	129.360	30638	300	3/3/2015	4:00pm
□ Edit Copy Delete	8	FB	79.600	10737	500	3/3/2015	4:00pm
□ Edit Copy Delete	9	GOOG	573.640	359	0	3/3/2015	4:00pm
□ Edit Copy Delete	10	YHOO	42.620	35224	200	3/3/2015	4:00pm
□ Edit Copy Delete	11	AMZN	384.610	2805	0	3/3/2015	4:00pm
□ Edit Copy Delete	12	AAPL	129.360	30638	0	3/3/2015	4:00pm
□ Edit Copy Delete	13	FB	79.600	10737	0	3/3/2015	4:00pm
□ Edit Copy Delete	14	GOOG	573.640	359	0	3/3/2015	4:00pm
□ Edit Copy Delete	15	YHOO	42.620	35224	200	3/3/2015	4:00pm
□ Edit Copy Delete	16	AMZN	384.610	2805	0	3/3/2015	4:00pm
□ Edit Copy Delete	17	AAPL	129.360	30638	0	3/3/2015	4:00pm
□ Edit Copy Delete	18	FB	79.600	10737	0	3/3/2015	4:00pm
□ Edit Copy Delete	19	GOOG	573.640	359	0	3/3/2015	4:00pm
□ Edit Copy Delete	20	YHOO	42.620	35924	700	3/3/2015	4:00pm

Figure 8 Current Price Structure

7. Plan of Work



Name	Begin date	End date	Duration
Query Stock Information	2/16/15	3/13/15	20
Build Database	2/16/15	3/6/15	15
Comparing Different Models	3/9/15	4/3/15	20
Implement Prediction Algorithms	3/16/15	4/10/15	20
Build Website and Mobile Application	3/16/15	4/17/15	25

Reference

[1]. Wikipedia, “Broker” [Online]. Available: http://en.wikipedia.org/wiki/Broker

[2]. Wikipedia, “Stock market prediction”
http://en.wikipedia.org/wiki/Stock_market_prediction