SOFTWARE ENGINEERING OF WEB APPLICATION

StockOutluk

Submitted By
Group 8
Anjanibhargavi Ragothaman
Antons Treikalis
Anurag Ekkati
Priya Padmanaban
Sneha Bojja

To
Instructor: Dr. Shiyu Zhou

Date: 2nd May 2014

ACKNOWLEDGEMENT

It is a moment of great pleasure and satisfaction for us to express our sense of profound gratitude to all people who have contributed in making our project work successfully.

We convey our sincere thanks to Prof. Dr. Shiyu Zhou for giving us an opportunity to take up the project work and also helping us in all phases of the project work and providing our team sound encouragement. We are also thankful to him for designing the project in such a way that it helped us learn the concepts of software engineering and we are extremely grateful for providing us all the facilities and help which lead to the successful completion of our project.

TABLE OF CONTENT:

Break Down of Responsibility:

Name	Responsibility	% Contribution
Anjanibhargavi Ragothaman	Database creation & population SVM Longterm Prediction Testing Report	20
Antons Treikalis	Database creation & population Bayesian Prediction integration MART Longterm Prediction Web interface support Testing Report	20
Anurag Ekatti	Web Interface support Testing Report	20
Priya Padmanaban	Database creation & population Web Interface Development Testing	20
Sneha Bojja	Web Interface support Testing Report	20

Abstract

Stock prediction has been an attractive project for a long time for different perspectives. For the past few years, along with the traditionally available indicators, research has been undertaken to use machine learning techniques to predict the stock price, stock indices, the movement of the stock market and the like. It helps the trader to learn accurately, the forthcoming changes which would yield him profits or avoid loss. In this project, we are developing a web application to predict stock prices and their movement using Bayesian short term stock prediction for day traders and GBRT/MART and SVM for long term stock prediction and predict its trend.

1. Introduction

Stock market prediction is the act of trying to determine the future value of a company stock or other financial instrument traded on a financial exchange. The successful prediction of a stock's future price could yield significant profit.

Will Rogers once said, "The way to make money is to buy stock at a low price, then when the price goes up, sell it. If the price doesn't go up, don't buy it." Obviously, the rule of "Buy low and sell high" is the simple key to successful investment. However, it's never been easy to fulfil this rule as it might seem to be. According to some statistics, more than 50% of the US households directly or indirectly own stocks through mutual funds, retirement accounts or other managed assets. The hardcore question is how to make right decisions on stock markets and do we really understand what we are buying.

Even though predicting stock market with 100% precision is just a mission impossible, people still manage to develop some analysis techniques to help us get much more comprehensive understanding of the market and make better decisions in our investments. There are two broad categories of market predicting methodologies – Fundamental Analysis and Technical Analysis.

Fundamental Analysis of a business involves analyzing its financial statements and health, its management and competitive advantages, and its competitors and markets. It includes economic analysis, industry analysis and company analysis. On the basis of these three analyses the intrinsic value of the shares are determined. This is considered as the true value of the share. If the intrinsic value is higher than the market price it is recommended to buy the share. If it is equal to market price, hold the share; and if it is less than the market price, sell the shares. Fundamental Analysis is a much more subjective way compared with Technical Analysis.

Technical Analysis employs models and trading rules based on price and volume transformations. It stands in contrast to the fundamental analysis approach to security and stock analysis. Technical analysis analyzes price, volume and other market information, whereas fundamental analysis looks at the facts of the company, market, currency or commodity. Technical Analysis utilizes objective mathematical indicators to reflect all relevant information of the market. It holds that prices already reflect all such trends before investors are aware of them, and uncovering those trends is what technical indicators are designed to do. Technical analysis is widely used among traders and financial professionals and is very often used by active day traders.

There are several factors that explain why technical analysis works:

1. Most speculators on the market act upon fundamental analysis, so that kind of facts influence stock prices strongly. But all operators do not get this information at the same time. When there are positive news of a company, those acting immediately can buy shares for a lower price than those getting the news later.

2. Large investors such as mutual funds and banks are often not placing their whole block orders at the same time when they are buying larger quantities of securities because this would risk triggering an unnecessary high price advance. Instead, the orders are spread over a period that can last several weeks. The resulting increased purchase pressure may result in a steady advancing trend under the period the purchases continue.

3. It is more psychological stressing to go against the trend than to follow it. People are herding animals and like to do as others are doing. This is why a rising stock price is a signal in itself that the price will advance even more. Of course one has to be careful with stocks that have been rocketing, because they will often recoil.

This report describes the overall structure, behaviours and interactions of various modules of our system. It also contains the details of the algorithm, prediction modules, databases and web services used in successfully implementing our project.

1.1 **Project Overview:**

Our project is aimed at developing a web application which provides access to users, a reliable prediction of stock values of companies they are interested in.

In this project, we are using Technical analysis, and are especially designed for active daily or weekly short-term investors, since they usually do not have the time or resources to avail of commercial forecasting services or hire agents.

As the technical analysis is based on the analysis if historical market data, we are using Yahoo Finance API from which we are collecting both the historical and real time data. The programming language Python is used to query the Yahoo Finance API and the data is stored into PostgreSQL database.

Our project mainly focuses on three aspects:

Real time data feed:
We have used real time data that is collected from Yahoo Finance API. So the prediction is based on real time data feeding and long term prediction is based on historical data, thus making it a realistic prediction advisory.

Prediction Strategies:
We are providing prediction for both long term and short term. For short term we are using Bayesian predictor algorithm and for long term we are using SVM and MART.

Easy Access to Web Interface:
We developed a sophisticated web application, offering various functionalities to the end users like getting valuable information, timely recommendations and tips on how to deal with their stocks.

The user interface of our web site is user-friendly and simple. We have implemented complex and smart algorithms to perform price predictions. These algorithms run as a back-end task and compute the prediction values for the various stocks.

2. Software Environment:

This project collects information of 25 stocks mentioned in Table 1. Yahoo Finance API is used as the source of stock data. The end to end project development is based on Python. The database system used is PostgreSQL and the web development framework used is Django.

2.1 Python:

Python is a widely used general purpose, high-level and interpreted programming language. Python supports multiple programming paradigms including object-oriented, imperative and functional programming, and styles. It features a dynamic type system and automatic memory management and has a large and comprehensive standard library.

Python efficiently supports many Internet protocols and offers powerful frameworks for web development. Its standard library and the Python Package Index support modules for creating graphical user interfaces, web frameworks, connecting to relational databases, arithmetic with arbitrary precision decimals, manipulating regular expressions, testing frameworks, scientific and numeric computing and many other useful tools.

This project uses the potent tools like **Django**, python based tool for web development, **PostgreSQL** for connecting to database and **numpy**, **scipy** and **scikit** for scientific and numeric computing required for programming the prediction algorithms incorporated in the project.

2.2 PostgreSQL Backend:

PostgreSQL is a powerful, open source object-relational database system and has a strong reputation for reliability, data integrity, and correctness. PostgreSQL is an enterprise class database which boasts sophisticated features such as Multi-Version Concurrency Control (MVCC), point in time recovery, tablespaces, asynchronous replication, nested transactions (savepoints), online/hot backups, a sophisticated query planner/optimizer, and write ahead logging for fault tolerance.

Multi version concurrency control (MCC or MVCC) is used to make PostgreSQL extremely responsive in high volume environment which is one of the basic requirements that a database suitable for this project must have. It has an efficient concurrency control method that fits our requirement to provide concurrent access to the database in the later stages of this project.

PostgreSQL has significant savings on costs as there is no associated licensing cost. It has been designed and created to have much lower maintenance and tuning requirements than leading databases, retaining all necessary features, stability, and performance.

2.3 Scientific Computing Tools:

Python supports lot of libraries for scientific and numeric computing. It provides simple and efficient tools for data mining and data analysis. This project uses **scikit** machine learning library for prediction strategies. It is based on modules **numpy**, **scipy** and **matplotlib** which help in computing and visualization of the data.

2.4 Django Web Framework:

Django is a high-level Python web framework that aids in rapid development and clean, pragmatic design. It is object oriented, supports automatic admin interface, provides elegant URL design, has efficient cache system for better performance and provides designer friendly template system to separate design, content and the python code.

2.5 BitBucket Version Control:

Bitbucket is a web-based hosting service for projects that use the Git revision control systems. Bitbucket offers free accounts with an unlimited number of private repositories and also offers commercial plans. It is similar to GitHub. It is ad-free and provides a clean interface for archiving files, providing version control and web hosting. It has many other useful features like code review, bug tracking and release binaries that help manage multi-developer projects in an effortless way.

We have used Bitbucket, source code repository for this project to enable consistency of file versions among the team members and for enabling multi-developer code development environment.

2.6 Open Stack VM:

For persistent stock data collection we are using an OpenStack Havana cloud which is available on Future Grid. Future Grid is a scientific grid, providing Grid, Cloud and HPC resources to its users. Currently under its umbrella are nine HPC clusters from five geographically distributed sites. OpenStack is cloud computing software for public and private clouds. OpenStack can be viewed as infrastructure as a service (IaaS) platform. Originally OpenStack was jointly launched by Rackspace Hosting and NASA. For this project we are using Ubuntu 12.04.4 LTS virtual machine running on OpenStack Havana. OpenStack Havana on Future Grid was chosen due to its ease of use and reliability.

2.7 System Setup:

This section explains the required steps to get the complete software environment setup and running. The whole project has been developed on Ubuntu platform, hence the installation process will guide through the steps required on Ubuntu platform.

To work with postgresql, having Python Language, Postgresql database and psycopg2 language binding installed on the system are pre-requisites.

Installation of PostgreSql

$ *sudo apt-get install postgresql*

Once installed, we check if the PostgreSQL server is running. If not, we need to start the server.
After postgresql is installed, the psycopg2 module is installed.

$ *sudo apt-get install python-psycopg2*

Installation of Scikit

The dependencies to install Scikit are Python (>= 2.6), NumPy (>= 1.3), SciPy (>= 0.7), setuptools, Python development headers and a working C++ compiler.

```
$ sudo apt-get install build-essential python-dev python-numpy python-setuptools python-scipy libatlas-dev
$ sudo apt-get install python-matplotlib
$ sudo apt-get install python-sklearn
```

Installation of Django

This section runs through the steps required in installing the official Django release. It is as simple as running two commands.

Install pip

```
$ sudo apt-get install python-pip
$ sudo pip install Django
```

Dependencies Installation related to Django

```
$ pip install django-registration
$ pip install django-localflavor
$ pip install django-bootstrap3
```

3. System Description:

3.1 System Block Diagram:

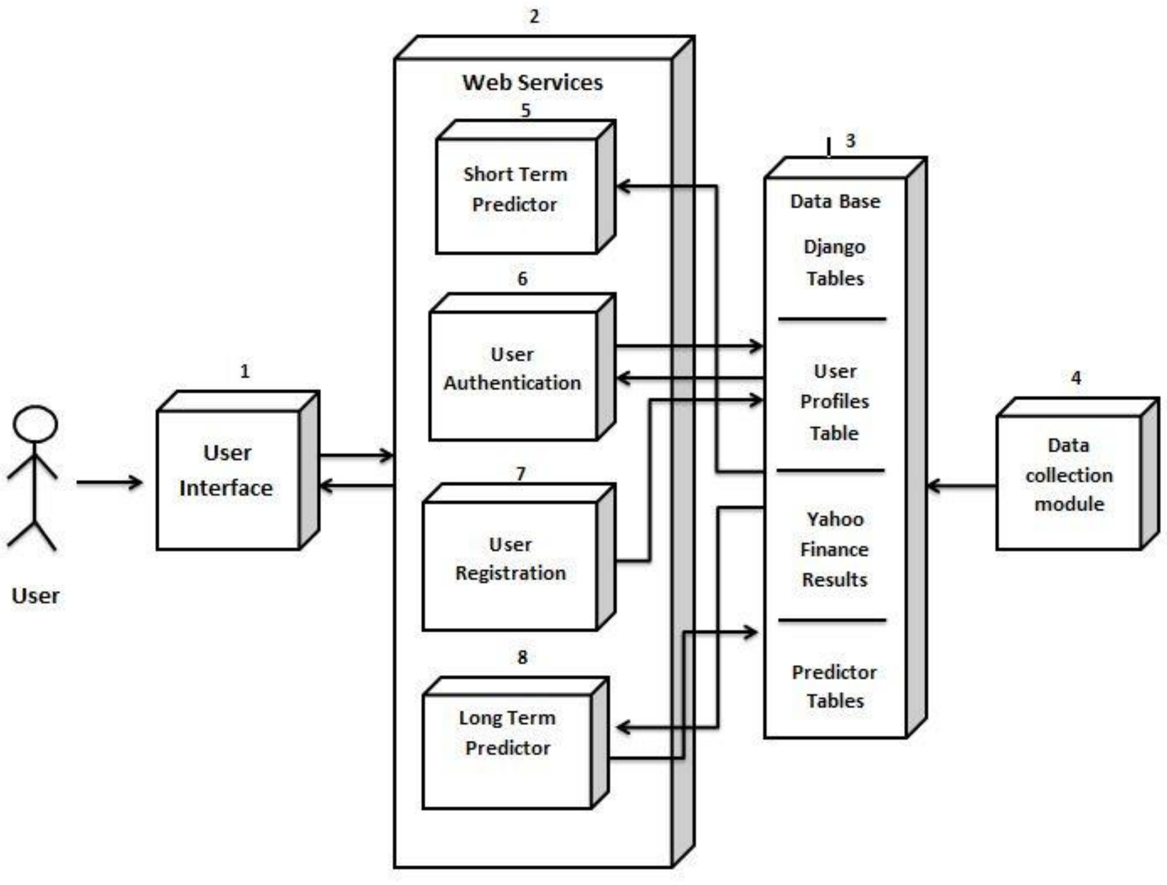


Figure 1: System Block Model of StockOutluk.

It illustrates 1.User Interface, 2.Web Service, 3.Database, 4.Yahoo Finance API, 5.Short Term Predictor, 6.User Authentication, 7.User Registration, 8.Long Term Predictor

It is clearly seen from the diagram that how all the modules interacts with each other and are integrated seamlessly. Our project is designed entirely in python and we are using Django web framework which is a python based web framework and so no delays are occurred during predictions.

We are also providing the user authentication and registration modules which helps the user to maintain a personal preference for his/her stock price analysis and predictions.

Unbiased prediction from other web service can be carried out pretty easily after making use of our website, for both the short term investors and long term investors.

3.2 System Description and General working:

The system under consideration has three levels: presentation level, service level and storage level. User gives the name of stock as input and gets the required output through the interface in the presentation level.

Registration and authentication is required to get access to our services. The service level performs prediction on the data collected from Yahoo Finance API using Bayesian Predictor, SVM and GBRT when the user requests. The storage level which is at the bottom most level, comprises of the PostgreSQL database. The user is provided with the “home” page when he opens the application. The user is asked to provide a correct combination of username and password. This pair of username and password is sent to the database for verification. If the combination is correct, the user is redirected to the “profile”. If the user is new to the system then he is allowed to register as a new user. The user selects the “registration” button and is then allowed to fill in the registration form, when this is done, the corresponding information is sent and stored into the database and then a email is sent to the user for activating the account. The user now got the username and password to login into the system.

Once the user enters the profile page, the user information is displayed and the user can view his 3 favourite stocks and can select any one of the 25 different stocks we are providing.

For the each favourite stock he gets 4 options

- Stock information – This page displays the real time data for the stock selected and gives the graphical representation in the form of the charts.
- Short term prediction – This page gives the stock term prediction value for both ask price and bid price using the Bayesian Predictor.
- Long term prediction – This page gives the long term prediction by suggesting the user whether the requested stock is ready to SELL, BUY or HOLDS.
- Compare Stocks – this page displays the comparison of the short term predictions for the users favourites.

If a user wants to see the graph of a particular stock, the information is directly sent to the database, corresponding data is fetched, required actions are performed on it and the required graph is sent back to the top layer that is the interface. If the user wants to get the prediction for a particular stock, he selects that stock and asks for the prediction. In this case the Web Service is called which takes the data from the data base, performs the required actions and send the required data to the user.

The database is kept up to date with the latest stock data. This is because the Python script collects the data every 1 minute. After every 1 minute, new data is updated into the database. Yahoo Finance API is being used for fetching both the historical and real time data for our system.

3.3 Use Case:

3.3.1 Use Case Diagram:

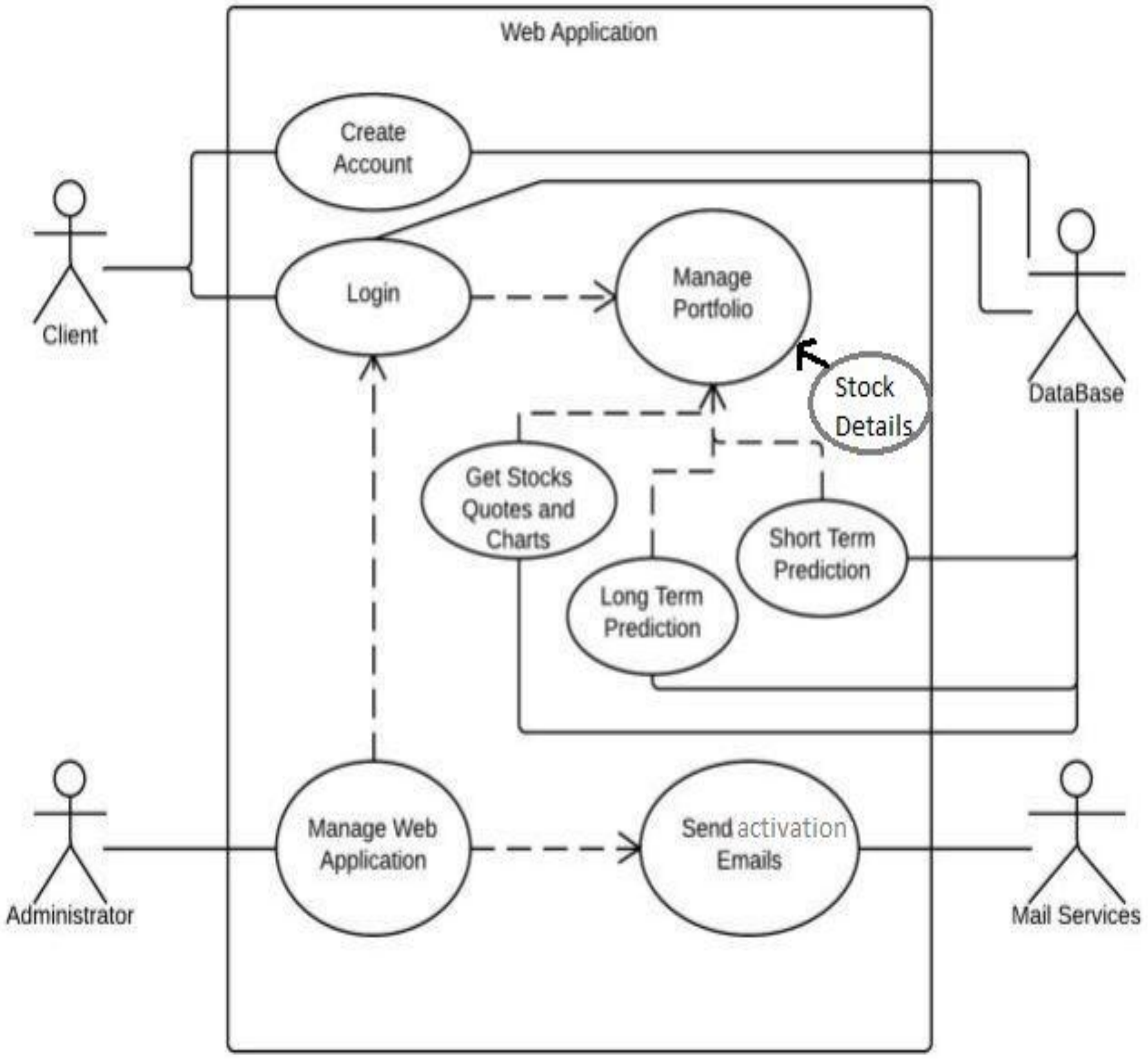


Figure 2: Use Case Diagram

3.3.2 Use Case Casual Description:

Use Case	Use Case Name	Active Participants	Description
1	Web- Service management	Administrator	The Administrator will be maintaining and managing the web-service and the server
2	Registration	User	The user will be able to create a new account by clicking on the activation link sent to his mail id provided during registration
3	Login	User	By providing the correct user name and password combination any user is allowed to login in his respective account
4	Account/Portfolio management	User	This will allow the user to manage his account, check details of his account and will allow to perform several other activities in his profile page
5	Short term Prediction	User	This will allow the user to get the short term prediction of selected stock
6	Long term Prediction	User	This will allow the user to get the long term prediction of selected stock
7	Get quotes/ graphs	User	The user will be allowed to get quotes and graphs for the selected stocks.
8	Stock Details	User	The user will be provided with the stock price details and will be given the option to view short and long term prediction.

3.3.3 Functional Specification Requirement:

Actor	Actor Goals	Use Case
Administrator	Management and maintenance of the web service and the server	Use case # 1
User	To create a new account and log in into his account	Use case # 2
User	To login into his account via the home page and access his account information	Use case #3
User	To manage his account and details	Use case #4
User	To get long and short term predictions	Use case #5 Use case #6
User	To get the real time data and 5 days historical data	Use case #8
User	To be able view the quote values and charts for the selected stock symbols	Use case #7
Mail Service(Gmail)	To send the activation email to complete the user registration	Use case # 2
Database	To store all the necessary information for user authentication, stock values and the corresponding stock names	Use case #1 Use case #2 Use case #3 Use case # 4 Use case #5 Use case # 6 Use case #7 Use case #8

USE CASE #1 DESCRIPTION

USE CASE #	Web service Management
INITIATING ACTOR:	Administrator
ACTOR'S GOALS:	Maintenance and management of web application and server
PARTICIPATING ACTORS:	None
PRE CONDITIONS:	The administrator should be correctly authenticated by the system after he/she provides the correct credentials.
POST CONDITIONS:	Check the application and make changes if required
SUCCESS SCENARIOS:	
1) Enters into the admin login page	
2) Provides correct user name and password	
3) Checks the details and make modifications if required	

USE CASE #2 DESCRIPTION

USE CASE #2	Registration
INITIATING ACTOR:	User
ACTOR'S GOALS:	To create a new account and activate the account
PARTICIPATING ACTORS:	Database/Mail service
PRE CONDITIONS:	The user does not have an account
POST CONDITIONS:	The user has an account and can use the application as and when required
SUCCESS SCENARIOS:	
1) The user opens home page	
2) Selects new user option and fills in the registration form by providing his details	
3) Select create account option	
4) Complete the registration by clicking on the activation link sent to his email id provided	

USE CASE #3 DESCRIPTION

USE CASE #3	Login
INITIATING ACTOR:	User
ACTOR'S GOALS:	To login into the account
PARTICIPATING ACTORS:	User
PRE CONDITIONS:	Not Logged in
POST CONDITIONS:	Logged in by giving correct username and password
SUCCESS SCENARIOS:	
1) The user enters the home page	
2) Fills in the correct user name and password	
3) Login by selecting the login option	

USE CASE #4 DESCRIPTION

USE CASE #4	Account /Portfolio Management
INITIATING ACTOR:	User
ACTOR’S GOALS:	To manage his account and details
PARTICIPATING ACTORS:	Database
PRE CONDITIONS:	Login status: not logged in
POST CONDITIONS:	Login status: Logged in by providing correct credentials a and use services provided
SUCCESS SCENARIOS:	
1) The user opens the home page and gets authenticated	
2) Check his interested stock details and manage his stock accordingly	
3) Manage his account details	

USE CASE #5 DESCRIPTION

USE CASE #5	Short Term Prediction
INITIATING ACTOR:	User
ACTOR’S GOALS:	To get short term predicted values for ask and bid price
PARTICIPATING ACTORS:	Database
PRE CONDITIONS:	Logged in
POST CONDITIONS:	Get the predicted ask and bid price values for the stock of interest
SUCCESS SCENARIOS:	
1) The user opens the home page and gets authenticated	
2) Select the stock for which user needs short term prediction	

USE CASE #6 DESCRIPTION

USE CASE #6	Long Term Prediction
INITIATING ACTOR:	User
ACTOR’S GOALS:	To get short term predicted values for ask and bid price
PARTICIPATING ACTORS:	Database
PRE CONDITIONS:	Logged in
POST CONDITIONS:	Get the suggestion about the stock status
SUCCESS SCENARIOS:	
1) The user opens the home page and gets authenticated	
2) Select the stock for which user needs long term prediction	

USE CASE #7 DESCRIPTION

USE CASE #6	Get quotes/ graphs
INITIATING ACTOR:	User
ACTOR'S GOALS:	To be able view the quote values and charts for the selected stock symbols
PARTICIPATING ACTORS:	Database
PRE CONDITIONS:	Logged in
POST CONDITIONS:	Get the quotes and Charts
SUCCESS SCENARIOS:	
1) The user opens the home page and gets authenticated	
2) Select the stock for which user needs the quotes and graphs	
3) Select the get quoted option	
4) Check the quotes and graphs obtained	

USE CASE #8 DESCRIPTION

USE CASE #8	Stock Details
INITIATING ACTOR:	User
ACTOR'S GOALS:	To get the real time data and 5 days historical data
PARTICIPATING ACTORS:	Database
PRE CONDITIONS:	Logged in
POST CONDITIONS:	Knows the current stock values
SUCCESS SCENARIOS:	
1) The user opens the home page and gets authenticated	
2) Select the stock for which user needs stock details	

3.3.4 Interactive Diagrams for Key Use Cases:

Use Case #2 (Registration)

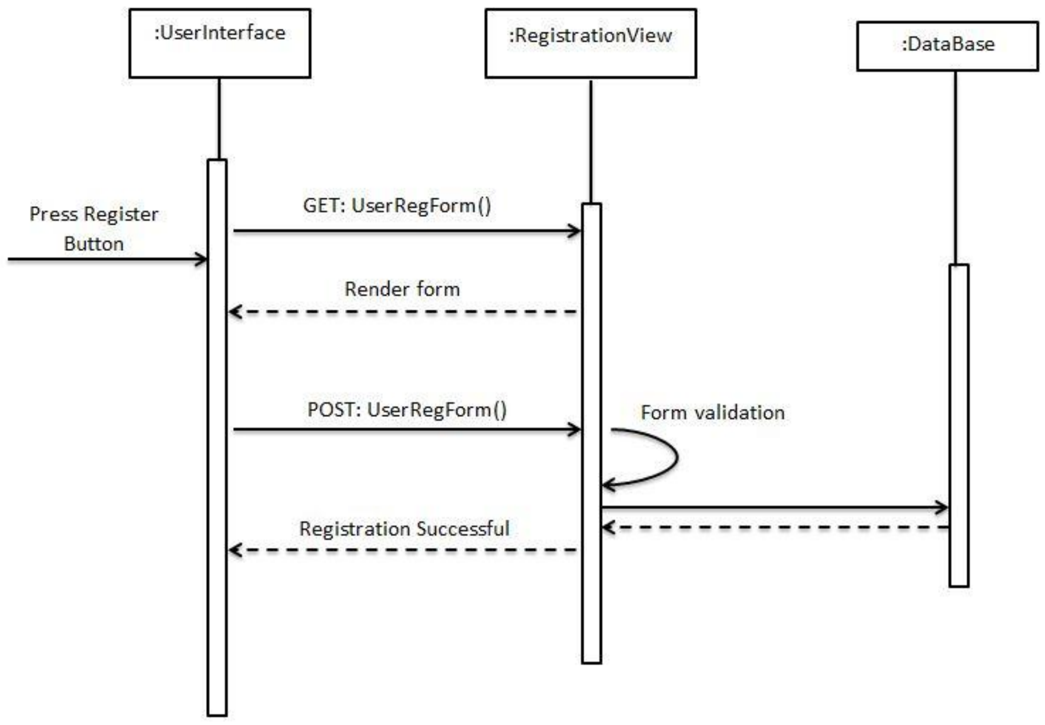


Figure 3: Interactive diagram for Registration

When a new user tries to register, a GET request is generated and a registration form is rendered. The new user need to fill in complete details in the registration form as instructed, else it creates an error in creating account. Once the user provides all the valid details in the registration form, form validation occurs and credentials will be populated in the data base and an activation email is sent to the email id that was provided during the registration. The user needs to click on the activation link that was sent to activate the account. This will display registration successful message to the user as acknowledgement for successful creation of account

Use Case# 3 (LOGIN)

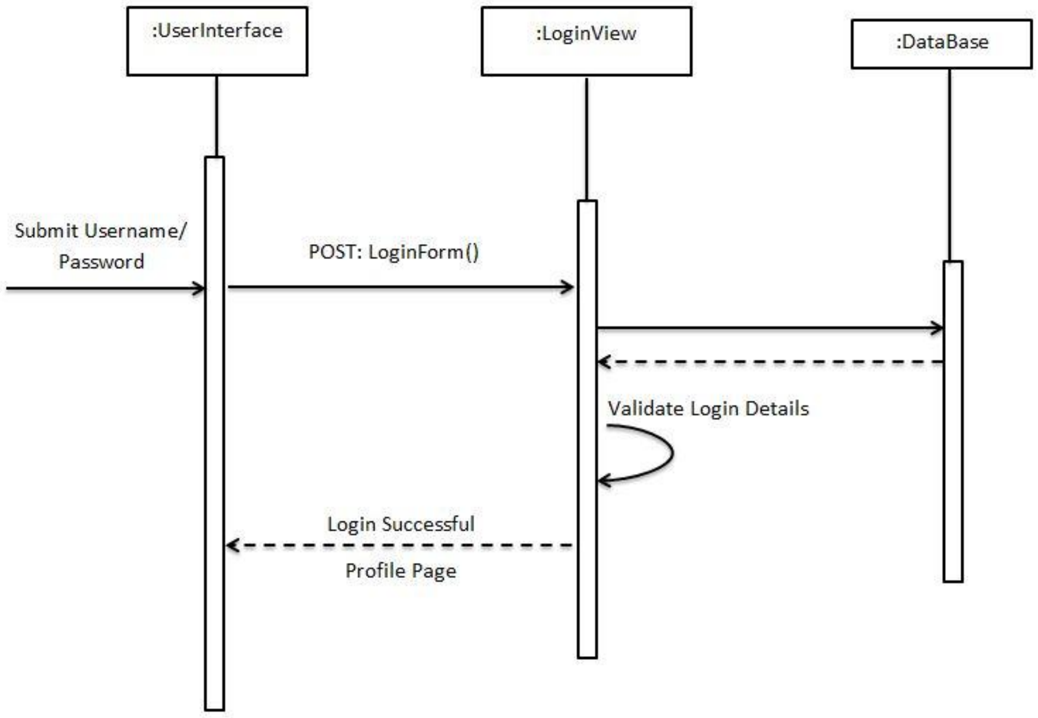


Figure 4: Interactive Diagram for Login

The user is required to provide the correct username and password for the successful login. When he submits the login button, the user name and password are validated in the login view using the information stored in the database. After the successful validation the user is directly to the profile page where he can manage and get the predictions of the various stocks. If the validation is not successful an error message is sent to the user for entering his username and password again for authentication purpose.

Use Case#5 (Short Term Prediction)

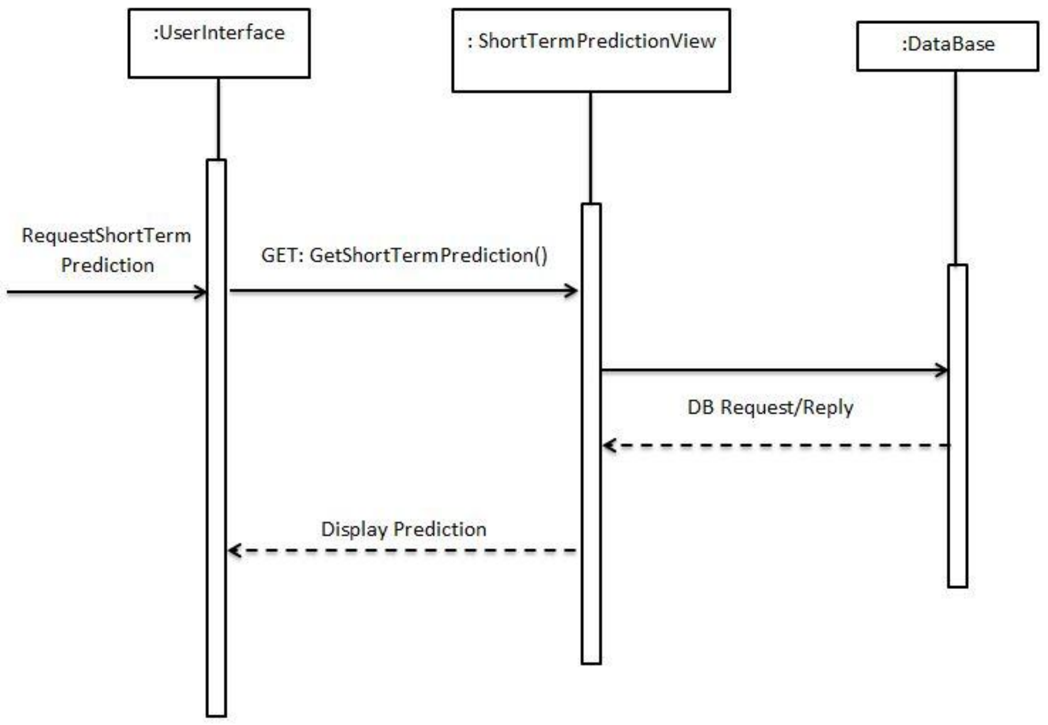

:UserInterface
: ShortTermPredictionView
:DataBase
RequestShortTerm Prediction
GET: GetShortTermPrediction()
DB Request/Reply
Display Prediction

Figure 5: Interactive diagram for Short Term Prediction

When the user requests for the short term prediction of a particular stock, he his redirected to the short term prediction page and a request for the calculation of short term prediction is sent to the short term prediction view where the Bayesian estimator calculates the prediction values for the ask price and bid price for that particular stock using the information from the real time data of that stock which is being stored into the database every 1 minute. To reflect the changes in the predictions the short term prediction page refreshes every one minute.

The short term prediction for a stock is calculated for every one minute up to ten minutes and displayed to the user. This provides the user with intricate information of the fluctuations in both the ask and bid prices that enables the user in taking the wiser decision.

We have a special feature provided for the convenience of the user. This calculates the short term predictions for his favourite stocks and displays them to the user enabling the user to compare predictions for his favourite stocks.

Use Case#5 (Long Term Prediction)

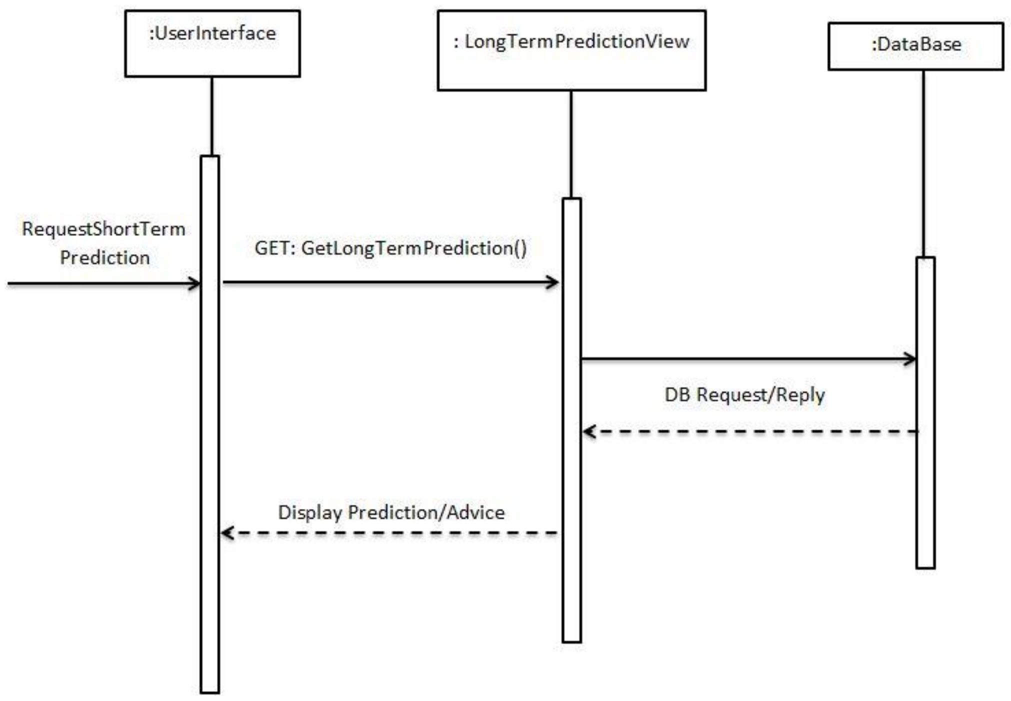

:UserInterface
: LongTermPredictionView
:DataBase
RequestShortTerm Prediction
GET: GetLongTermPrediction()
DB Request/Reply
Display Prediction/Advice

Figure 5: Interactive diagram for Long Term Prediction

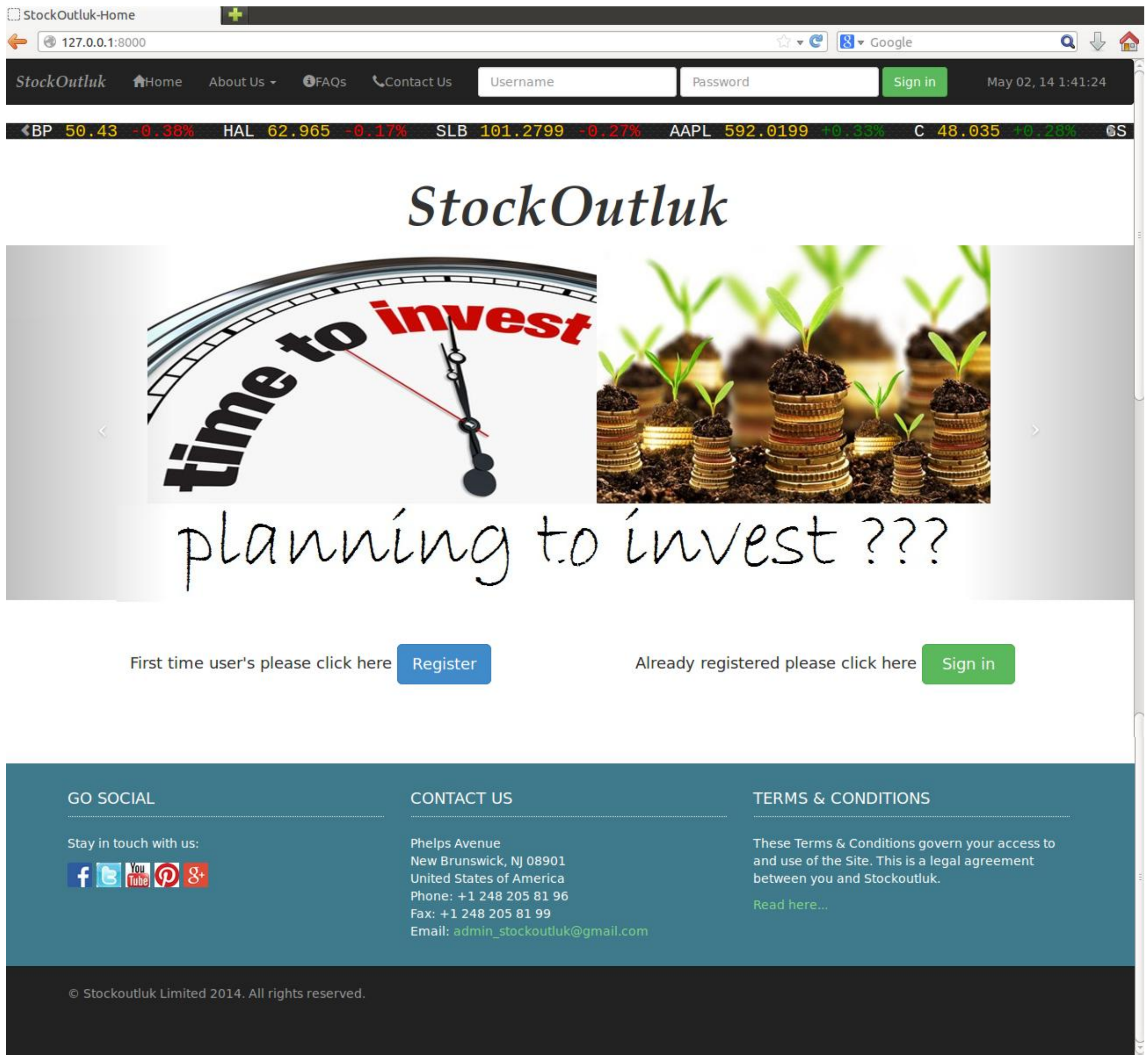
When the user requests for the long term prediction of a particular stock, he his redirected to the long term prediction page and a request for the calculation of long term prediction is sent to the long term prediction view where the GBRT calculates the prediction for Day's high and Day's low for the next 5 days. This predicted value is now fed to SVM as the test data set. SVM validates if the predicted value follows the trend of the historical high and low stock price (which is the training data) and provides a confidence level (Strong/Weak prediction). The predicted stock price is compared with the computed 50 days MACD and suggests the trend of the stock (rising/falling). These values are stored in the database in the table named temp_prediction_data for that day.

4. User Interface:

We tried to make our user interface easy to understand for the user and also made it simple by separating the backend complex engines. Our website requires users to create profiles to avail the facilities we provide. This also helps in providing users better user experience by enabling them to specify user preferences which provides faster browsing options to the user.

Home Page:

In the home page of our web application, a new user can register while the existing user can directly login. On successful registration an activation email will be sent to the user. This allows only genuine users to register in our website. Our navigation bar provides easy access to various pages like About Us, F.A.Qs, Contact Us, etc. which provide important information about the website.



First time user's please click here Register

Already registered please click here Sign in

Stay in touch with us:

Phelps Avenue
New Brunswick, NJ 08901
United States of America
Phone: +1 248 205 81 96
Fax: +1 248 205 81 99
Email: admin_stockoutluk@gmail.com

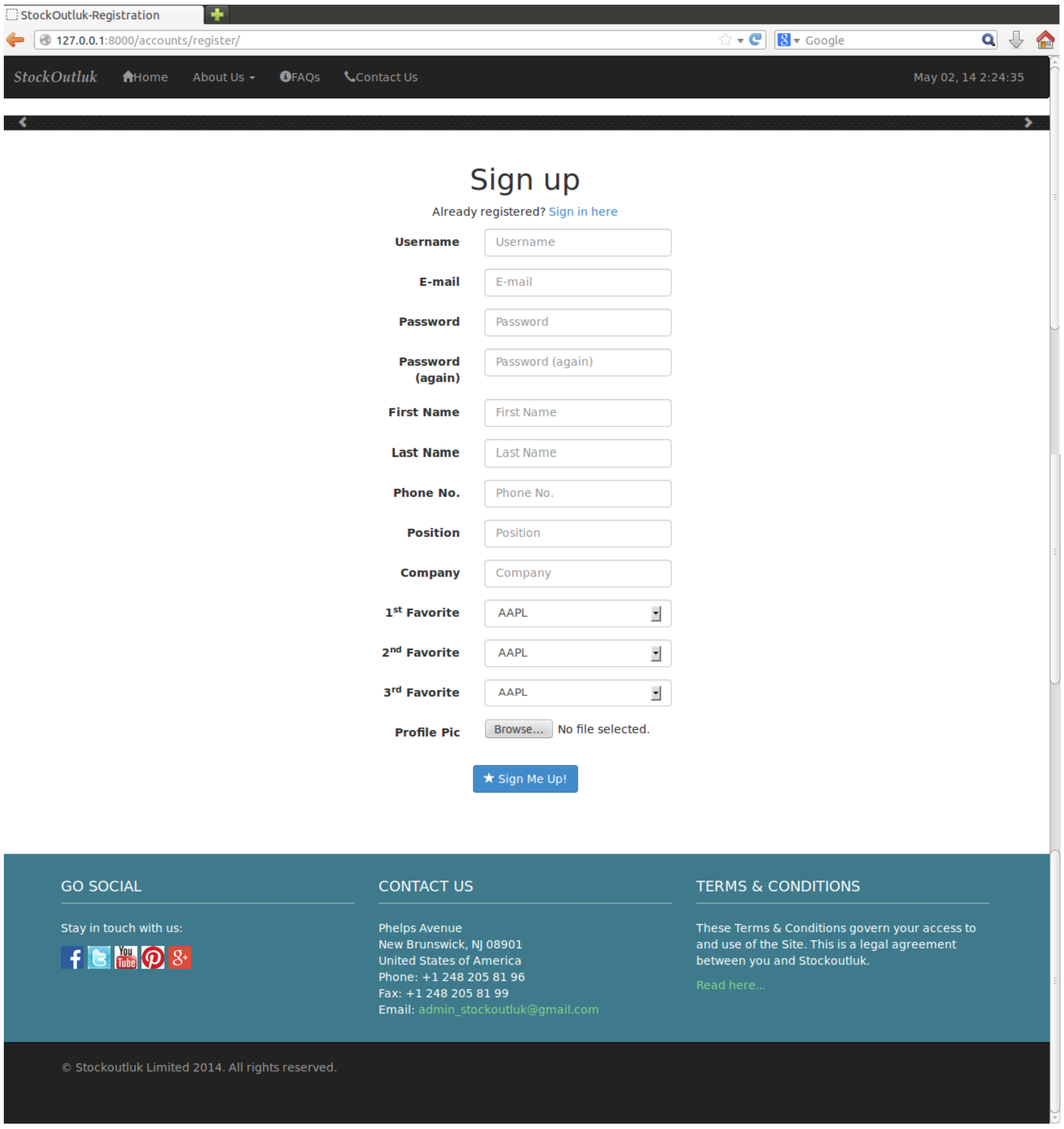
These Terms & Conditions govern your access to and use of the Site. This is a legal agreement between you and Stockoutluk.

© Stockoutluk Limited 2014. All rights reserved.

Figure 6: Screenshot of the home page

User Registration Page:

When a new user wants to register with StockOutluk, he/she needs to provide the requested user information. Our website also provides a feature to upload his/her profile picture. The user will also get the facility specifying three favourite stocks during registration. This would provide the user easy navigation to information related to these stocks.



Sign up

Already registered? Sign in here

Username	Username
E-mail	E-mail
Password	Password
Password (again)	Password (again)
First Name	First Name
Last Name	Last Name
Phone No.	Phone No.
Position	Position
Company	Company
1st Favorite	AAPL
2nd Favorite	AAPL
3rd Favorite	AAPL
Profile Pic	Browse... No file selected.

★ Sign Me Up!

GO SOCIAL

Stay in touch with us:

CONTACT US

Phelps Avenue
New Brunswick, NJ 08901
United States of America
Phone: +1 248 205 81 96
Fax: +1 248 205 81 99
Email: admin_stockoutluk@gmail.com

TERMS & CONDITIONS

These Terms & Conditions govern your access to and use of the Site. This is a legal agreement between you and Stockoutluk.

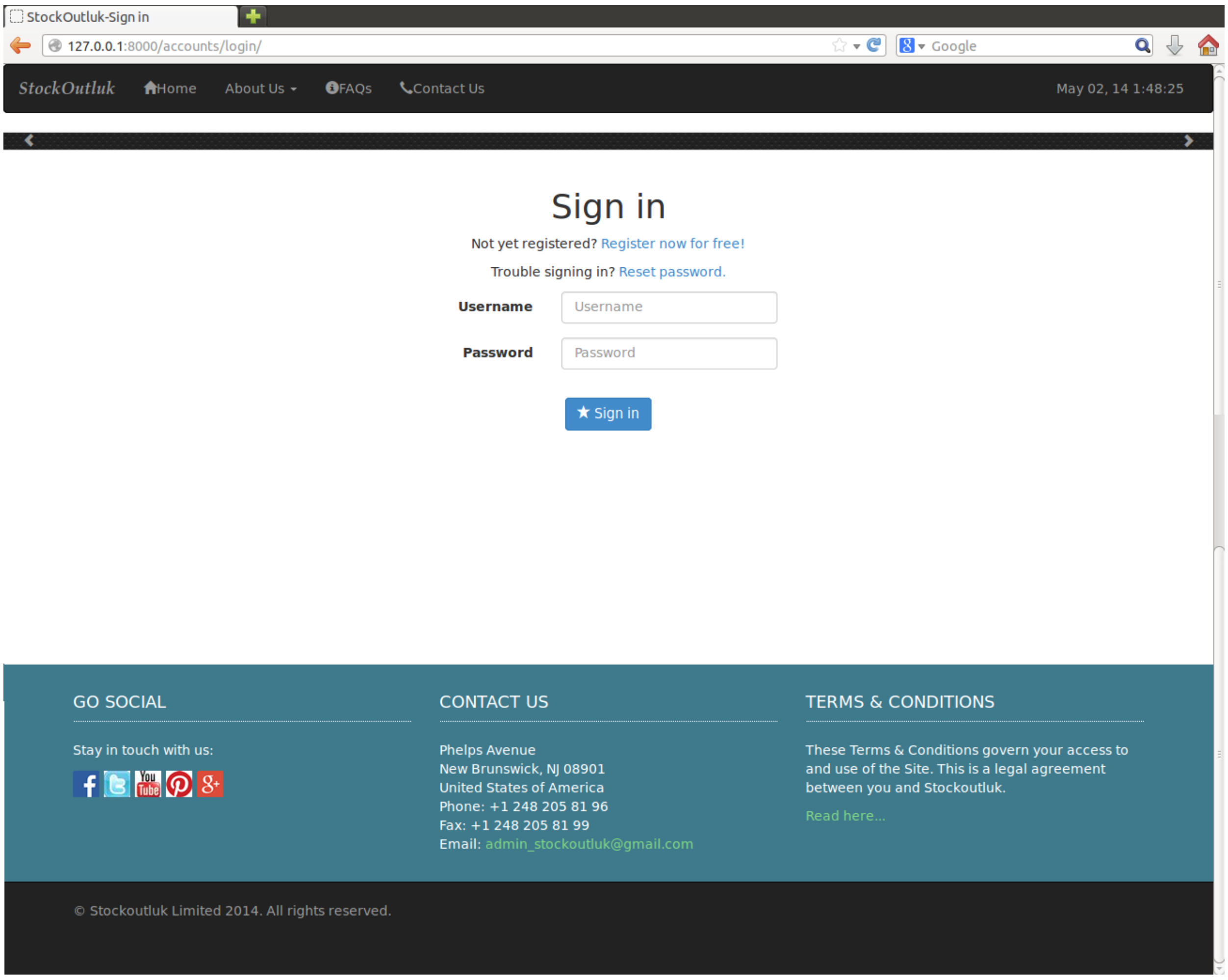
Read here...

© Stockoutluk Limited 2014. All rights reserved.

Figure 7: Screenshot of the registration page

Login Page:

When the user wants to login with Stockoutluk, he needs to provide the correct username and password to access his profile page. He has given an option to reset the password in case he forgets it.



Sign in

Not yet registered? Register now for free!

Trouble signing in? Reset password.

Username: Username

Password: Password

Sign in

GO SOCIAL

Stay in touch with us:

CONTACT US

Phelps Avenue
New Brunswick, NJ 08901
United States of America
Phone: +1 248 205 81 96
Fax: +1 248 205 81 99
Email: admin_stockoutluk@gmail.com

TERMS & CONDITIONS

These Terms & Conditions govern your access to and use of the Site. This is a legal agreement between you and Stockoutluk.

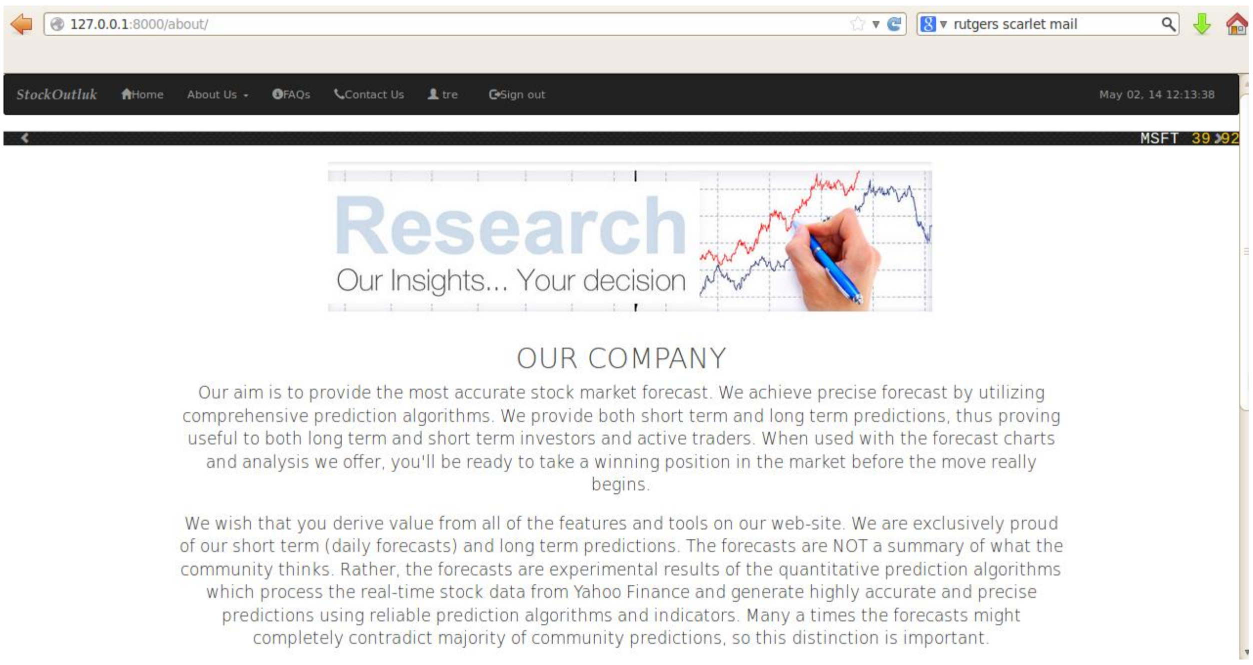
Read here...

© Stockoutluk Limited 2014. All rights reserved.

Figure 8: Screenshot of Login page

About Us Page:

This page displays the brief details about the StockOutluk website.



OUR COMPANY

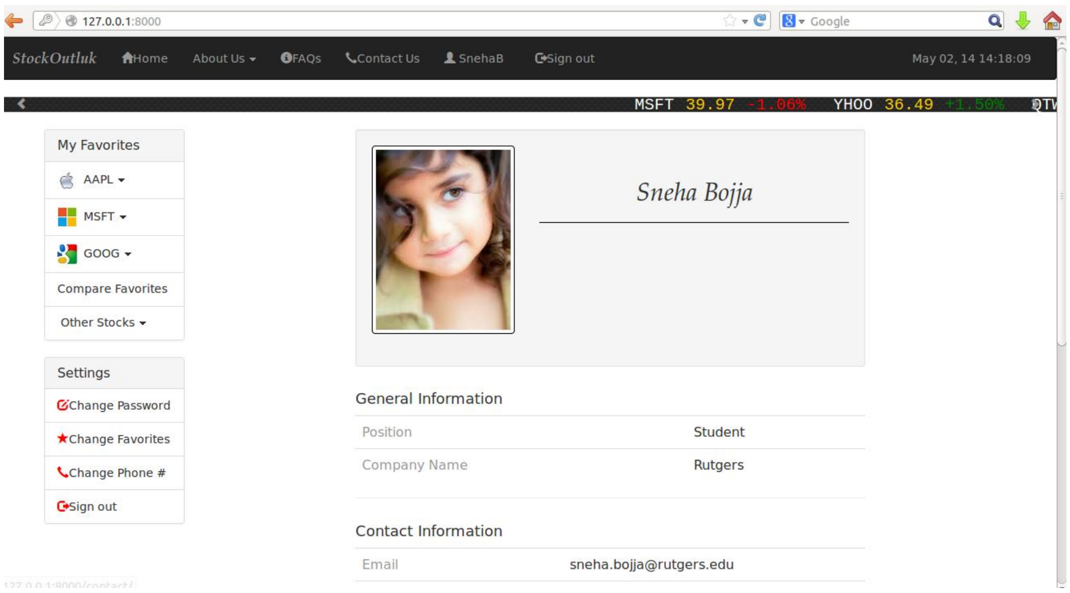
Our aim is to provide the most accurate stock market forecast. We achieve precise forecast by utilizing comprehensive prediction algorithms. We provide both short term and long term predictions, thus proving useful to both long term and short term investors and active traders. When used with the forecast charts and analysis we offer, you'll be ready to take a winning position in the market before the move really begins.

We wish that you derive value from all of the features and tools on our web-site. We are exclusively proud of our short term (daily forecasts) and long term predictions. The forecasts are NOT a summary of what the community thinks. Rather, the forecasts are experimental results of the quantitative prediction algorithms which process the real-time stock data from Yahoo Finance and generate highly accurate and precise predictions using reliable prediction algorithms and indicators. Many a times the forecasts might completely contradict majority of community predictions, so this distinction is important.

Figure 8: Screenshot of About Us page

Profile Page:

The users profile page displays the user's information and three stocks of his interest. When the user selects one of this stock symbols he is given options like stock details, short term prediction and long term prediction as drop down box. By selecting one of the above options he is redirected to its corresponding page. Additionally the user can view the predictions of other stocks by selecting them from the drop down box present in his profile page.



General Information

Position	Student
Company Name	Rutgers

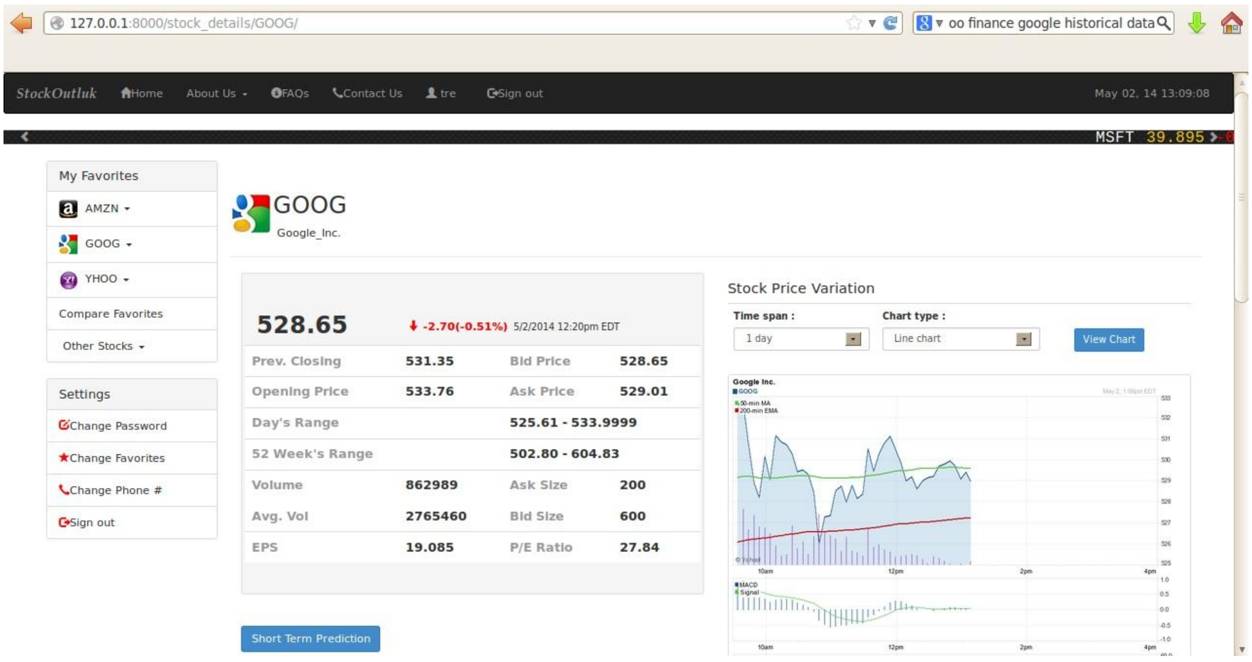
Contact Information

Email	sneha.bojja@rutgers.edu

Figure 9: Screenshot of User Profile page

Stock Info Page:

The Stock details page displays the real time data and is refreshed every one minute to reflect the changes in the stock in real time. We also provide graphical representation of stock price fluctuations using charts for the real time data using the Yahoo API charts. We have given the option for choosing the prediction stratergies.

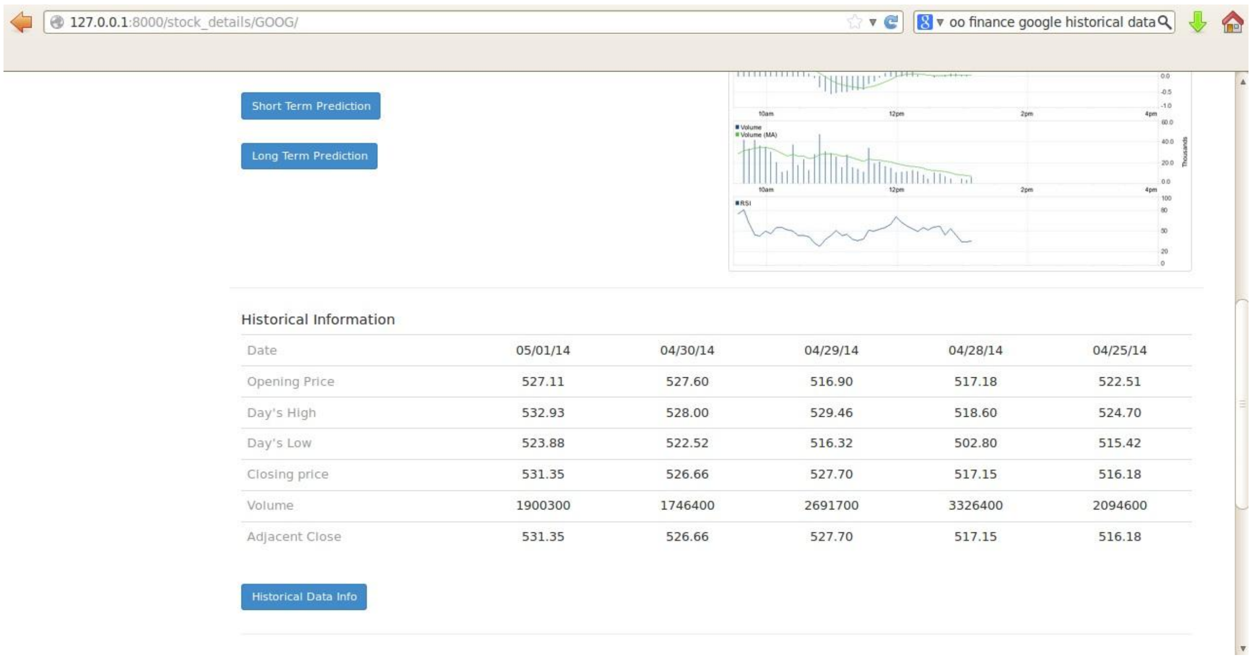


GOOG
Google_Inc.

528.65 -2.70(-0.51%) 5/2/2014 12:20pm EDT

Prev. Closing	531.35	Bid Price	528.65
Opening Price	533.76	Ask Price	529.01
Day's Range		525.61 - 533.9999	
52 Week's Range		502.80 - 604.83	
Volume	862989	Ask Size	200
Avg. Vol	2765460	Bid Size	600
EPS	19.085	P/E Ratio	27.84

Figure 10.1: Screenshot of Stock Info page


127.0.0.1:8000/stock_details/GOOG/
oo finance google historical data
Short Term Prediction
Long Term Prediction
Historical Data Info

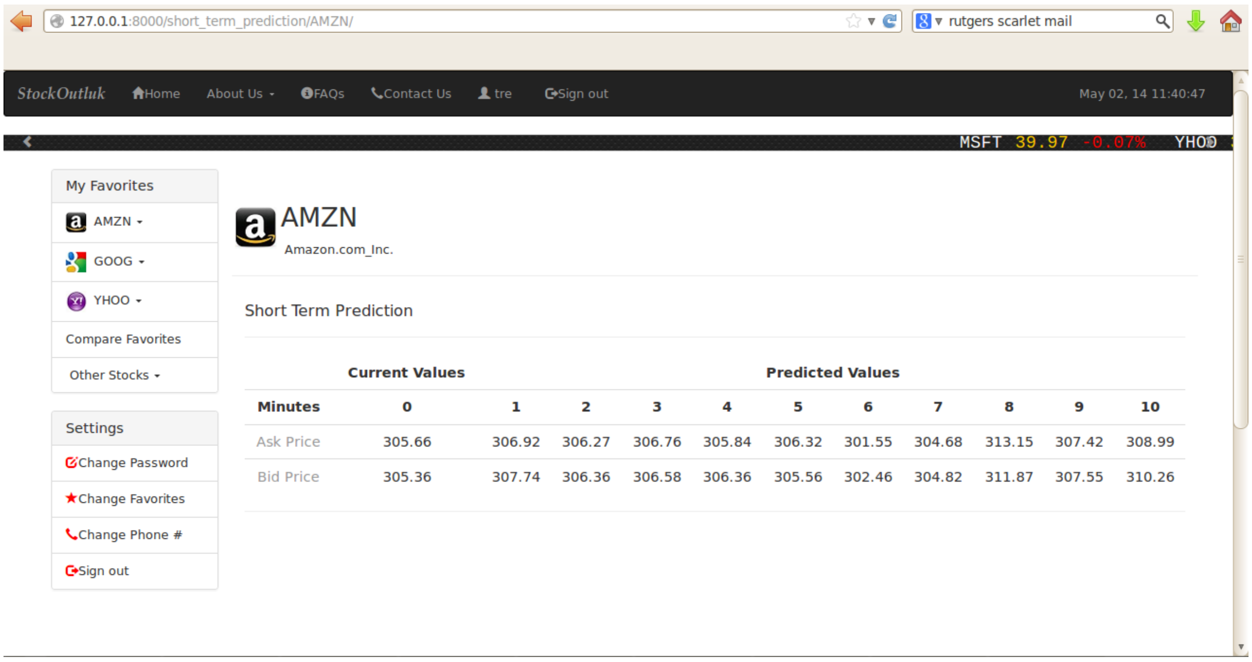
Historical Information

Date	05/01/14	04/30/14	04/29/14	04/28/14	04/25/14
Opening Price	527.11	527.60	516.90	517.18	522.51
Day's High	532.93	528.00	529.46	518.60	524.70
Day's Low	523.88	522.52	516.32	502.80	515.42
Closing price	531.35	526.66	527.70	517.15	516.18
Volume	1900300	1746400	2691700	3326400	2094600
Adjacent Close	531.35	526.66	527.70	517.15	516.18

Figure 10.2: Screenshot of Stock Info page continuation

Short Term Prediction Page:

When the short term prediction is selected user will be redirected to its page. This page consists of a table which is populated with stock data corresponding to the stock value at that particular instant. The table also contains stock predictions for next 10 min at 1 min intervals as shown below.



127.0.0.1:8000/short_term_prediction/AMZN/

StockOutluk | Home | About Us | FAQs | Contact Us | tre | Sign out | May 02, 14 11:40:47

MSFT 39.97 -0.07% YHOO

My Favorites: AMZN, GOOG, YHOO, Compare Favorites, Other Stocks

Settings: Change Password, Change Favorites, Change Phone #, Sign out

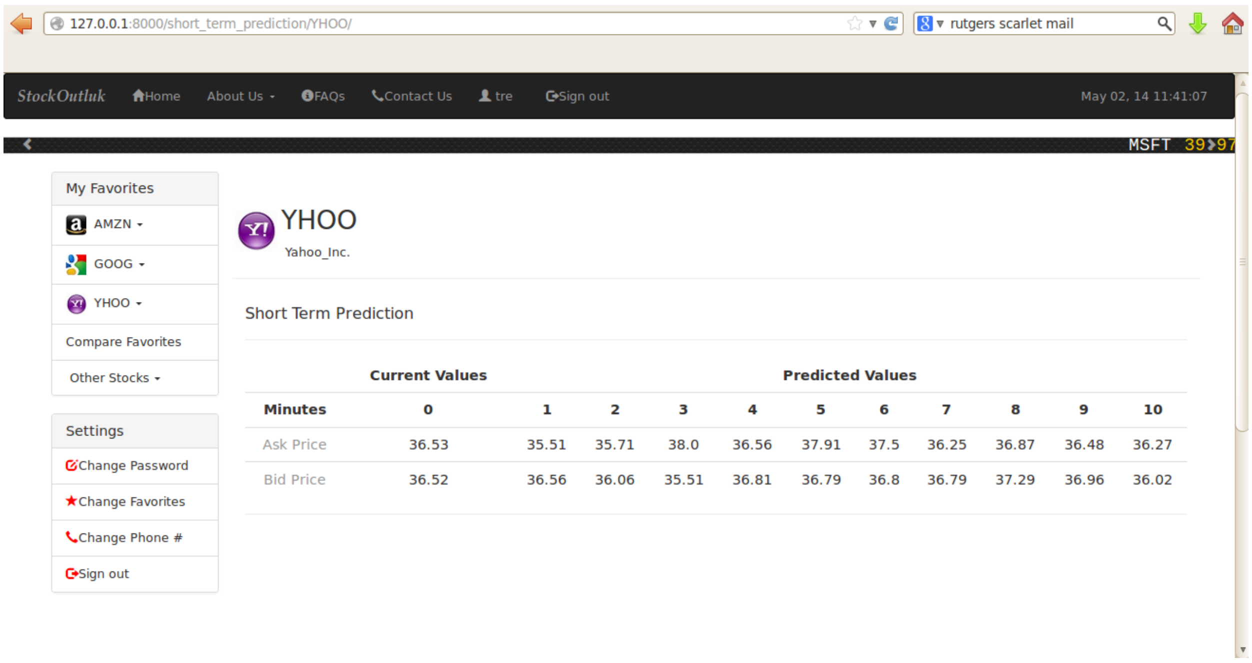
AMZN

Amazon.com_Inc.

Short Term Prediction

	Current Values					Predicted Values					
Minutes	0	1	2	3	4	5	6	7	8	9	10
Ask Price	305.66	306.92	306.27	306.76	305.84	306.32	301.55	304.68	313.15	307.42	308.99
Bid Price	305.36	307.74	306.36	306.58	306.36	305.56	302.46	304.82	311.87	307.55	310.26

Figure 11.1: Screenshot of Stock Term Prediction page



127.0.0.1:8000/short_term_prediction/YHOO/

StockOutluk | Home | About Us | FAQs | Contact Us | tre | Sign out | May 02, 14 11:41:07

MSFT 39.97

My Favorites: AMZN, GOOG, YHOO, Compare Favorites, Other Stocks

Settings: Change Password, Change Favorites, Change Phone #, Sign out

YHOO

Yahoo_Inc.

Short Term Prediction

	Current Values					Predicted Values					
Minutes	0	1	2	3	4	5	6	7	8	9	10
Ask Price	36.53	35.51	35.71	38.0	36.56	37.91	37.5	36.25	36.87	36.48	36.27
Bid Price	36.52	36.56	36.06	35.51	36.81	36.79	36.8	36.79	37.29	36.96	36.02

Figure 11.2: Screenshot of Stock Term Prediction page continuation

Long Term Prediction Page:

When the long term prediction is selected user will be redirected to its page. This page consists of a table which is populated with predicted data of the selected stock for past 5 days and future 5 days. The page also contains a chart giving the comparison between the predicted and actual stock price.

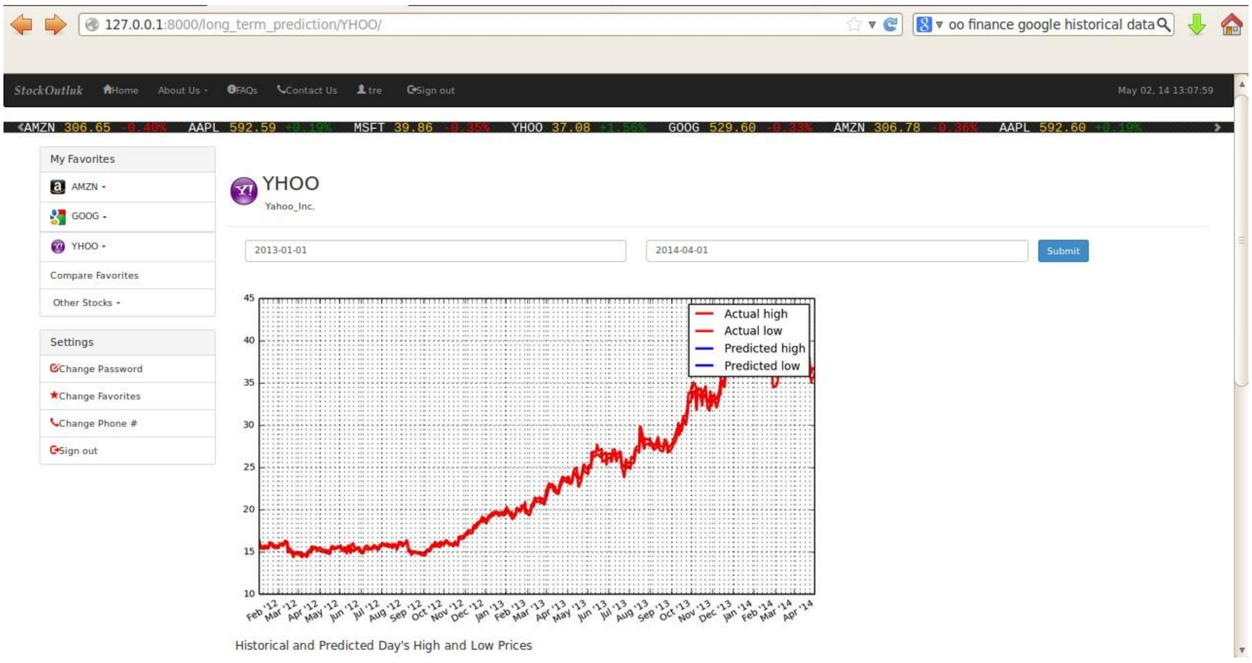
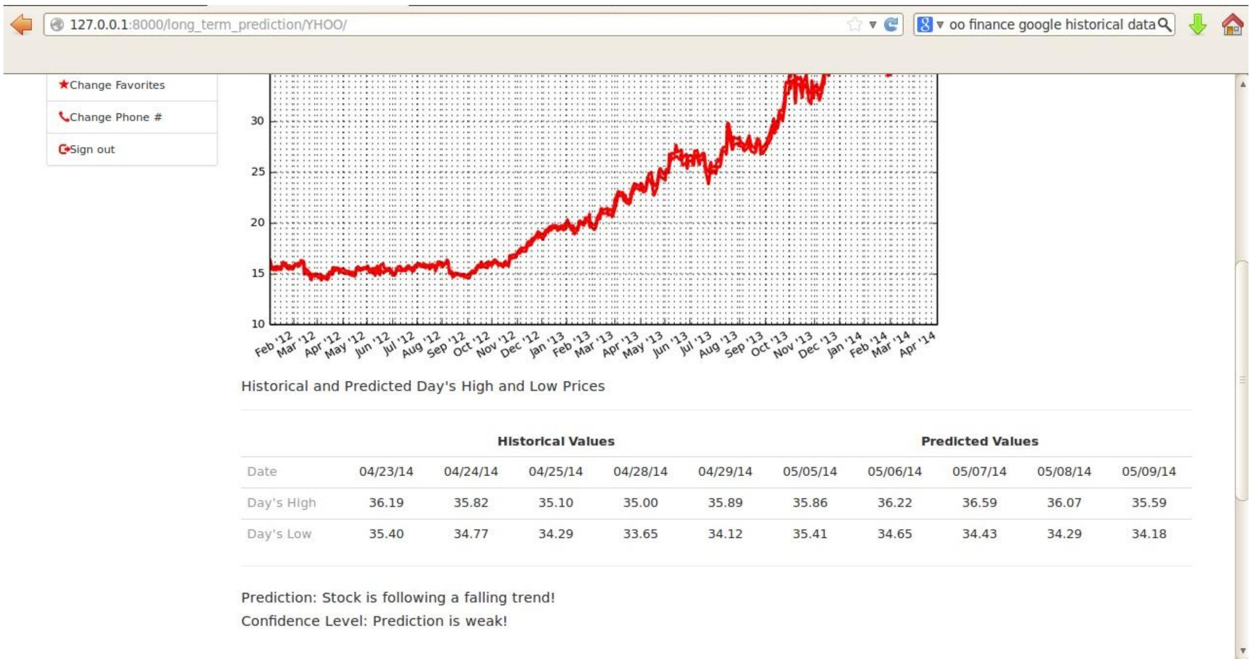


Figure 12.1: Screenshot of Yahoo Long Term Prediction page


127.0.0.1:8000/long_term_prediction/YHOO/
oo finance google historical data
Change Favorites
Change Phone #
Sign out
Historical and Predicted Day's High and Low Prices

	Historical Values					Predicted Values				
Date	04/23/14	04/24/14	04/25/14	04/28/14	04/29/14	05/05/14	05/06/14	05/07/14	05/08/14	05/09/14
Day's High	36.19	35.82	35.10	35.00	35.89	35.86	36.22	36.59	36.07	35.59
Day's Low	35.40	34.77	34.29	33.65	34.12	35.41	34.65	34.43	34.29	34.18

Prediction: Stock is following a falling trend!
Confidence Level: Prediction is weak!

Figure 12.2: Screenshot of Yahoo Long Term Prediction page continuation

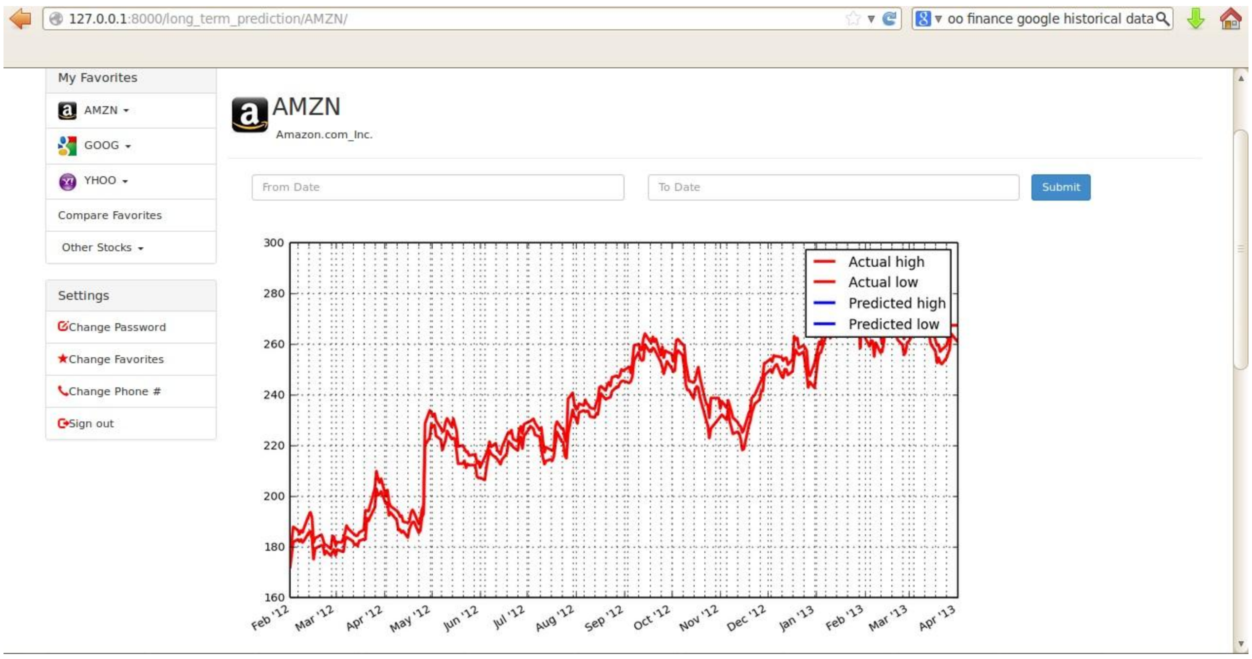
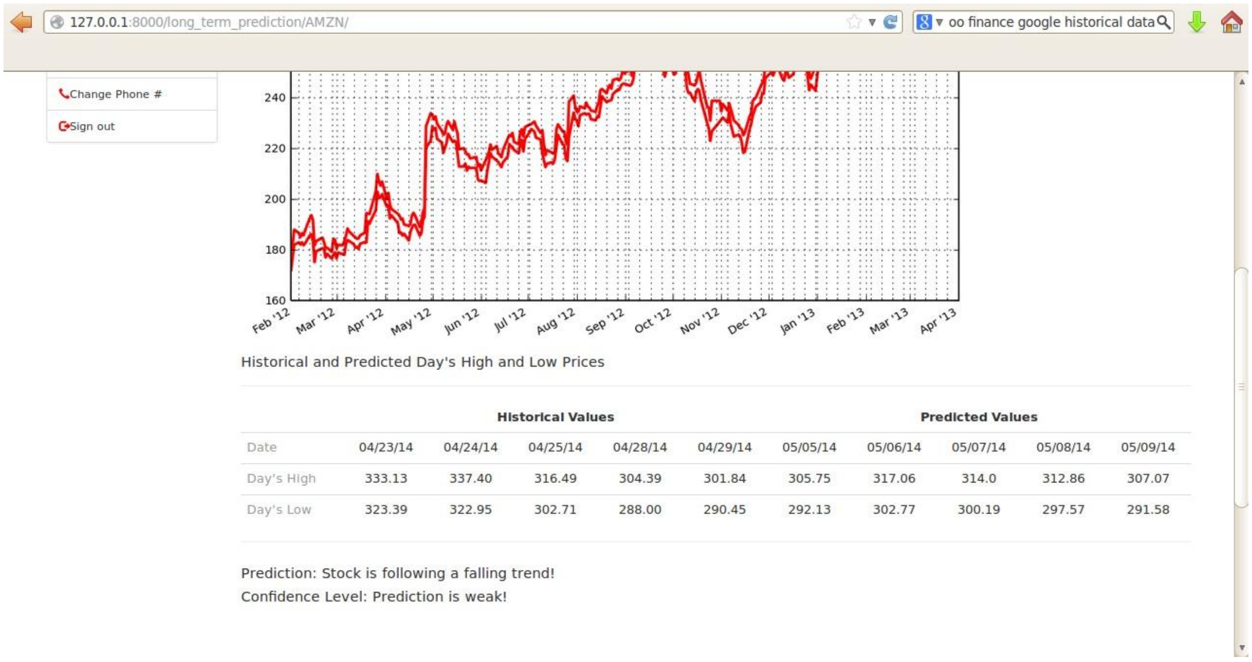


Figure 13.1: Screenshot of Amazon Long Term Prediction page



127.0.0.1:8000/long_term_prediction/AMZN/
oo finance google historical data
Change Phone #
Sign out

Historical and Predicted Day's High and Low Prices

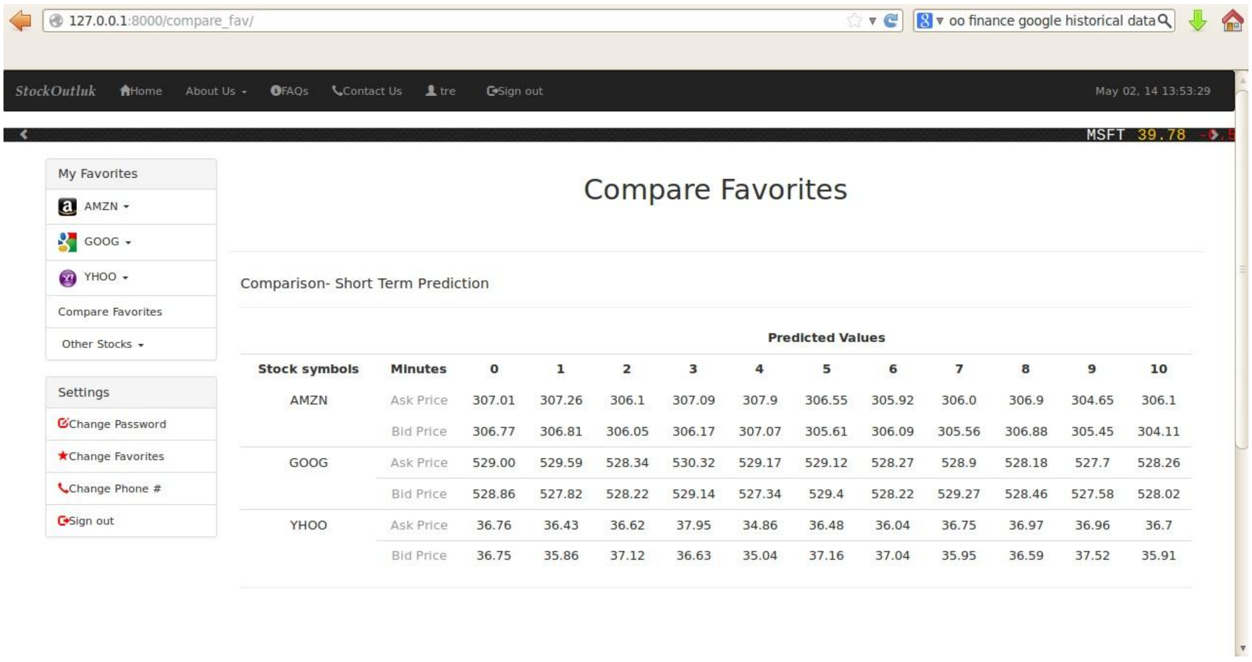
	Historical Values					Predicted Values				
Date	04/23/14	04/24/14	04/25/14	04/28/14	04/29/14	05/05/14	05/06/14	05/07/14	05/08/14	05/09/14
Day's High	333.13	337.40	316.49	304.39	301.84	305.75	317.06	314.0	312.86	307.07
Day's Low	323.39	322.95	302.71	288.00	290.45	292.13	302.77	300.19	297.57	291.58

Prediction: Stock is following a falling trend!
Confidence Level: Prediction is weak!

Figure 13.2: Screenshot of Amazon Long Term Prediction page continuation

Compare Favourites Page:

In this page the user can compare his favourite stocks behaviours



Compare Favorites

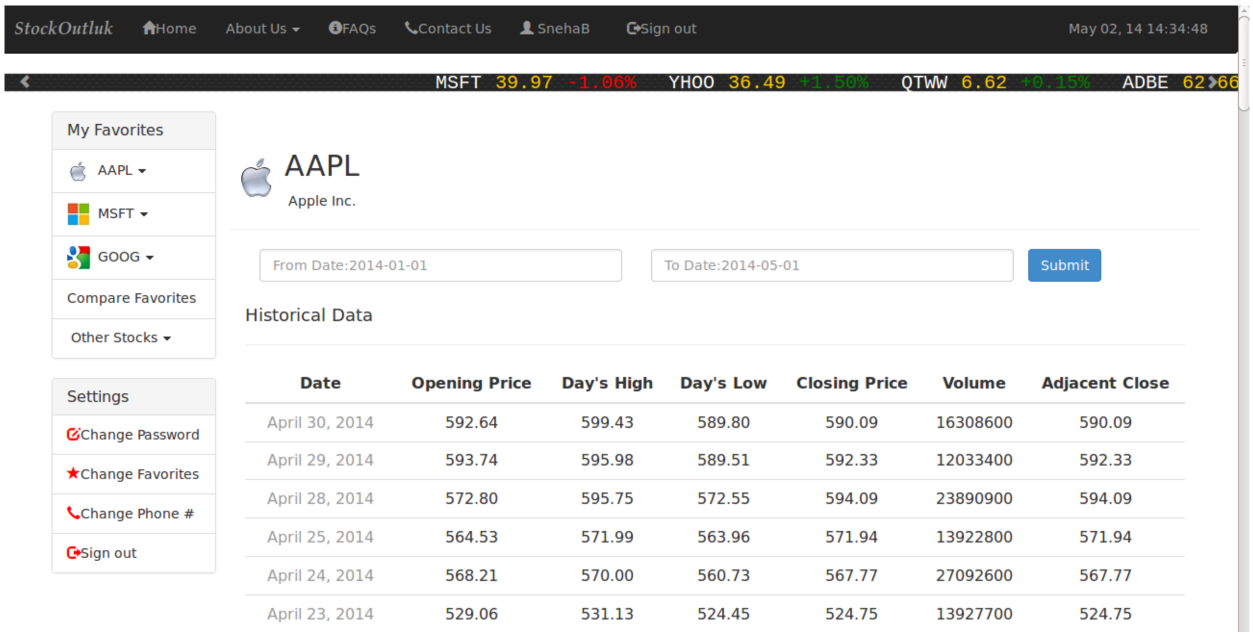
Comparison- Short Term Prediction

		Predicted Values										
Stock symbols	Minutes	0	1	2	3	4	5	6	7	8	9	10
AMZN	Ask Price	307.01	307.26	306.1	307.09	307.9	306.55	305.92	306.0	306.9	304.65	306.1
	Bid Price	306.77	306.81	306.05	306.17	307.07	305.61	306.09	305.56	306.88	305.45	304.11
GOOG	Ask Price	529.00	529.59	528.34	530.32	529.17	529.12	528.27	528.9	528.18	527.7	528.26
	Bid Price	528.86	527.82	528.22	529.14	527.34	529.4	528.22	529.27	528.46	527.58	528.02
YHOO	Ask Price	36.76	36.43	36.62	37.95	34.86	36.48	36.04	36.75	36.97	36.96	36.7
	Bid Price	36.75	35.86	37.12	36.63	35.04	37.16	37.04	35.95	36.59	37.52	35.91

Figure 14: Screenshot of users Compare Favourites page

Historical Data Info:

This page displays the range of historical information for the duration requested by the user.



Historical Data

Date	Opening Price	Day's High	Day's Low	Closing Price	Volume	Adjacent Close
April 30, 2014	592.64	599.43	589.80	590.09	16308600	590.09
April 29, 2014	593.74	595.98	589.51	592.33	12033400	592.33
April 28, 2014	572.80	595.75	572.55	594.09	23890900	594.09
April 25, 2014	564.53	571.99	563.96	571.94	13922800	571.94
April 24, 2014	568.21	570.00	560.73	567.77	27092600	567.77
April 23, 2014	529.06	531.13	524.45	524.75	13927700	524.75

Figure 15: Screenshot of the Historical Info page

5. Prediction Strategies:

Stock prices are in general considered to be very dynamic and they are susceptible to quick changes due to lot of known and unknown factors influencing the financial domain. In relation to it, lot of interesting work has been happening in the area of applying machine learning algorithms to predict stock prices, analyze price patterns and index changes. This helps to identify the fluctuations in the market before it actually happens thus helping the trader to gain profit at the right time or avoid a loss before incurring it.

In this project, we are using short term and long term prediction strategies to help the trader make decision for day trading purposes and for daily and weekly monitoring and trading. We are using Bayesian Prediction methodology for short term prediction. For long term prediction we chose two prediction strategies namely, MART (a decision tree based boosting algorithm) and SVM (Unsupervised classification model Support Vector Machine)

5.1 Short Term Prediction:

5.1.1 Bayesian Estimator:

A naive Bayes classifier assumes that the value of a particular feature is independent of other features, given the class variable. Bayesian estimator is based upon the simple Bayes' theorem. Given a set of observed data, it is straight forward to get the probability of choosing the model. It is called a prior probability. But, given a model, what is the probability of choosing the data set becomes tricky. This is called the posterior probability. Bayes' theorem helps to identify the posterior probability with the help of a likelihood function. In very simple terms, posterior α likelihood prior

Bayesian Estimator using Gaussian Distribution:

The input to the Bayesian estimator are the training data set stock prices(x) at time points (t) and the future time point (**T**) for which the stock price (X) has to be predicted. The mean and variance 2 are determined from the given data set. The future stock price is given by the probability function

$$p(X \mid T, t,x) = N(X \mid m(x), s2(x))$$

Where the mean and variance are given by

$$m(x) = (t)T\ S\ n=1N\ (tn)\ xn$$
$$s2(x) = -1 + (t)T\ S\ (t)$$

Here the Smatrix is given by

$$S-1 = I + n=1N\ (tn)\ (t)T$$

Where I is the unit matrix and the vector (t) is defined by the elements n(t) = tn for n = 0,1... M

where M is the order of the polynomial. The values are chosen to be 5 10-3 and 11.1 respectively.

The input data set per stock used for intra-day prediction using the Bayesian estimator is the historical intra- day ask and bid values collected every minute during the market hours over a period of a month. The accuracy of the prediction is calculated using the absolute and relative error.

5.2 Long Term Prediction:

5.2.1 GBRT:

For the long term prediction strategy, GBRT algorithm is used to predict the stock price and unsupervised classification type SVM algorithm is used to predict the direction (rising or falling trend) of the predicted stock price.

GBRT is a novel multi-feature prediction based on a decision tree boosting algorithm. It uses machine learning techniques for regression problems. GB builds an additive model in a forward stage-wise fashion; it allows for the optimization of arbitrary differentiable loss functions. In each stage a regression tree is fit on the negative gradient of the given loss function. One of the advantages of using GBRT is, it supports using multiple features to train the model to predict accurate values.

The training dataset over ***X*** axis for GBRT algorithm constitutes of two years of historical prices, namely, open price, high, low, close price, volume and adjusted close price. In addition to that S&P 500 stock market index is used in this data set. It is important to point out that for predictions of high and low prices are used different data sets over ***X*** axis - training set for prediction of high price includes low price but does not include high price and conversely training set for low price includes high price. Training data set over ***Y*** axis is stock's high and low pricing for high and low price prediction respectively? It is worth mentioning that for one day prediction there is one row offset between data sets for ***X*** and ***Y*** axis - we use today's ***X*** axis data set to predict tomorrow's stock price.

We make two types of predictions using GBRT - one day prediction and five day prediction. The difference between these is in offset between data sets for ***X*** and ***Y*** axis - for one day prediction offset is one and for five day prediction offset is five. This difference is due to the fact that we can't just build prediction for the day after tomorrow based on the prediction for tomorrow. Our training set included not only the date for next day and predicted value for the previous day, but also a set of other features, which we can't easily obtain. Of course we can predict each of these features separately for the next day in order to avoid increasing the offset, but it was proven that this approach result in a lower accuracy.

Library Used:

We have implemented GBRT algorithm using ***scikit-learn*** package - an open source machine learning library for the Python Programming Language. ***scikit-learn*** is designed to be used in conjunction with ***NumPy***, ***SciPy*** and ***Matplotlib***. In our application we have used function provided by ***scikit-learn*** package as a black box. To perform prediction using GBRT with ***scikit-learn*** first we fit a Gradient Tree Boosting model to the data using ***GradientBoostingRegressor()*** function. This function accepts the following parameters:

- ***n_estimators*** : int (default=100; used=500) - The number of boosting stages to perform. GB is fairly robust to over-fitting so a large number usually results in better performance.
- ***max_depth*** : integer, optional (default=3; used=6) - The maximum depth of the individual regression estimators. The maximum depth limits the number of nodes in the tree. The best value depends on the interaction of the input variables.
- ***learning_rate*** : float, optional (default=0.1; used=0.1) - learning rate shrinks the contribution of each tree by learning_rate. There is a certain trade-off between learning_rate and n_estimators.
- ***loss***: {‘ls’, ‘lad’, ‘huber’, ‘quantile’}, optional (default=’ls’; used=’huber’) - loss function to be optimized. ‘ls’ refers to least squares regression. ‘lad’ (least absolute deviation) is a highly robust loss function solely based on order information of the input variables. ‘huber’ is a combination of the two. ‘quantile’ allows quantile regression.
- ***alpha***: float (default=0.9; used=0.95) - The alpha-quantile of the huber loss function and the quantile loss function. This parameter should be used only if loss='huber' or loss='quantile' is used.

There are also other optional parameters for ***GradientBoostingRegressor()*** function, but in our application we used only parameters specified above. After the model is built we fit our training data sets to this model by calling ***fit()*** function. ***GradientBoostingRegressor()*** function returns an object, which we can then use to make our prediction by calling ***predict()*** function on this object. ***predict()*** function accepts a list of parameters over ***X*** axis corresponding to price we make our prediction for (taking offset into account) and returns predicted value for high or low price.

5.2.2 Support Vector Machine (SVM):

Support vector machine (SVM) is a very specific type of learning algorithms characterized by the capacity control of the decision function, the use of the kernel functions and the sparsity of the solution. Established on the unique theory of the structural risk minimization principle to estimate a function by minimizing an upper bound of the generalization error, SVM is shown to be very resistant to the over-fitting problem, eventually achieving a high generalization performance. Another key property of SVM is that training SVM is equivalent to solving a linearly constrained quadratic programming problem so that the solution of SVM is always unique and globally optimal, unlike neural networks training which requires nonlinear optimization with the danger of getting stuck at local minima.

The SVM algorithm uses two years of historical stock high price per stock as the training dataset. The predicted stock price using GBRT is used as a test dataset. SVM helps determine whether the predicted stock price follows a rising or falling trend.

SVM is proven to be good for single feature classification, whereas GBRT has higher accuracy when multiple features are used for training dataset.

Library Used:

Trading guide which tells the direction of movement of the stock is more useful to the trader than certain amount of forecast with a forecast error. This project uses the scikit python machine learning library for using the SVM classifier. The OneClassSVM() classifier available in the scikit library is an unsupervised SVM outlier detector. The kernel function used is Radical Basis Function (rbf) kernel. It helps to predict whether the test dataset is a member of the training dataset or not.

6. Conclusion:

The technical challenge we faced was with regression models. It involved accumulation and processing of large amount of data which demanded utmost precision and accuracy. Though we found the task of handling and consolidating large statistical data to be challenging at the incipient stages of the project, we manage to implement the model successfully with a holistic understanding of the model. The incrementally progressing steps of software engineering development style have helped us to build these use cases from a concept entity to a realized module.

We also faced difficulties integrating all the codes written by each team member. We were able to overcome this cross communication between different languages by pure hard work, team work and perseverance.

The course Software Engineering has been of great help to us in terms of developing knowledge of languages like Python, XML and strengthened our knowledge about web framework like Django and its interface. It has also taught us to work on a project as a team and to coordinate with team members.

7. Future Work:

- The system can be integrated with price alert feature. With this feature whenever the stock price value falls in a particular range or reaches a particular value a notification e-mail or text message will be sent to the user.
- Additionally the system can continuously send emails or text messages at regular intervals requested by the user. This will be an appealing feature for the users who wish to be informed quite often.
- Articles written by eminent economist and published in famous magazines can be posted on our website blog.

APPENDIX

Data Collection Module:

Data selection and pre processing constitute a crucial step in any modelling effort. There are so many stocks that are traded at the stock market. As a stock prediction developer we need to choose wisely the range of stocks that we would advise our investors to invest in. There are different types of stocks that different investors would be interested in depending on what type of investor he/she is. However for our system we employed the following thought process:

Phase one of our project aimed at creating a database collection module for collecting and storing stock information of various companies over a considerable period of time. This information was used to make stock value predictions.

Collecting Stock Data from Yahoo Finance API

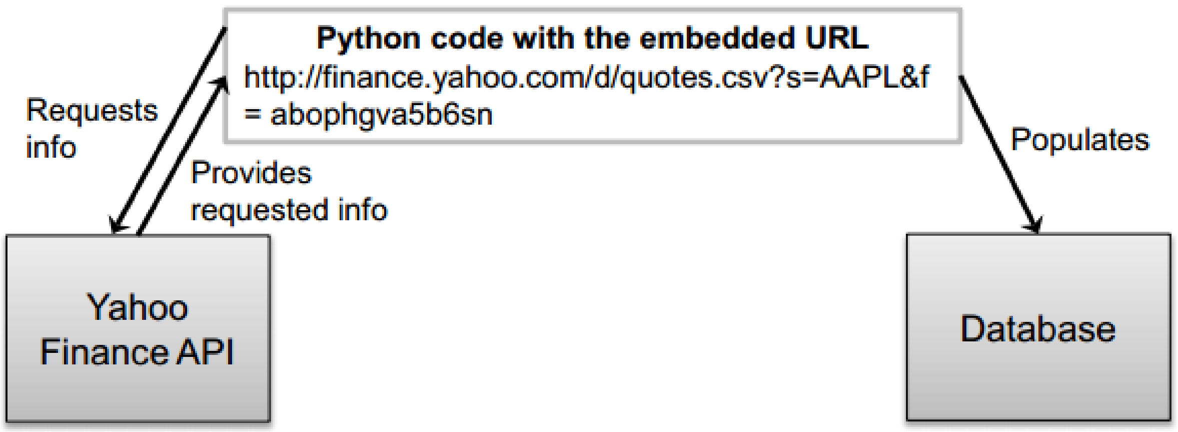
This project required a huge repository of stock information both for historical and current data for efficient prediction of stock values. Yahoo Finance API provides a great and simple way to download stock related data. It is a REST based API. This service returns stocks information in a comma delimited file format (.CSV). Python code is written to retrieve information using this API and subsequently stores data into the respective tables created with the characteristics of the retrieved data.

Assembling the URL in a particular format enables us to retrieve required data into a .CSV file. The URL construction method for retrieving current data is given below,

- The base URL for retrieving current data is, "http://finance.yahoo.com/d/quotes.csv".
- Add a "?s=<stock symbols separated by commas>", examples of stock symbols are APPL, GOOG, MSFT etc.
- Add a "&f=<attributes list>", the attribute list is used to specify the information required about the stock symbols mentioned above.
- For example, "http://finance.yahoo.com/d/quotes.csv?s=AAPL&f= abophgva5b6sn"

The URL construction method for retrieving historical data is given below,

- The base URL for retrieving historical data is, "http://ichart.yahoo.com/table.csv".
- Add a "?s=<stock symbols separated by commas>", examples of stock symbols are APPL, GOOG, MSFT etc.
- Add a "&a=<month>&b=<day>&c=<year>", the month field must be populated with month-1. It specifies the day from which the historical data is requested.
- For example, "http://ichart.yahoo.com/table.csv?s=GOOG&a=0&b=1&c=2000".
- The data is retrieved from the date mentioned in the URL till today.


Python code with the embedded URL
http://finance.yahoo.com/d/quotes.csv?s=AAPL&f = abophgva5b6sn
Requests info
Provides requested info
Populates
Yahoo Finance API
Database

The stock symbols we are using for this project are,

Symbols	Symbol Names
GLD	SPDR gold shares
YHOO	Yahoo, Inc.
INTU	Intuit Inc.
INTC	Intel Corporation
AMZN	Amazon.com Inc.
GOOG	Google Inc.
NTAP	NetApp, Inc.
CSCO	Cisco Systems, Inc.
ADBE	Adobe Systems Inc.
QTWW	Quantum Fuel Systems Technologies Worldwide Inc.
HPQ	Hewlett-Packard Company
T	AT&T Inc.
DVN	Devon Energy Corporation
LG	The Laclede Group Inc.
F	Ford Motor Co.
BAC	Bank of America Corporation
WMT	Wal-Mart Stores Inc.
GS	Goldman Sachs Group Inc.
AAPL	Apple Inc.
SLB	Schlumberger Limited

HAL	Halliburton Company
BP	British Petroleum plc
GE	General Electric Company
C	Citigroup_Inc.
MSFT	Microsoft Corporation

Table 1: Symbols for which stock data are collected

Many attributes are provided by the Yahoo Finance API. The attributes being used in this project and the description of the information they provide for each of the symbols mentioned above are as follows,

Current Data from http://finance.yahoo.com	**Historical Data from http://ichart.yahoo.com**
a: Ask	Open
b: Bid	Close
o: Open	Adjacent Close
p: Previous Close	High
h: Day’s High	Low
g: Day’s Low	Volume
s: Symbol	Date
v: Volume	
a5: Ask Size	
a6: Bid Size	
l1: last trade	
d1:trade date	
t1:trade_time	
c1:change	
p2:percen_change	
a2 :Average Daily Volume	
e : EPS	
r : P/E ratio	

Table 2: Attributes used from the Yahoo Finance API for current data

These attributes are described below,

- ASK - Ask or ask price, also called offer price, offer, asking price, or simply ask, is the price a seller states she or he will accept for a good. The seller may qualify the stated asking price as firm or negotiable. Firm means the seller is saying he or she won't change the price. Negotiable means the seller is inviting the potential buyer to attempt to convince the seller to lower the price.

- BID - A Bid or bid price is the highest price that a buyer (i.e., bidder) is willing to pay for a good. It is usually referred to simply as the "bid."

  In bid and ask, the terms ask price is used in contrast to the term bid price. The difference between the bid price and the ask price is called the spread.

- OPEN PRICE - The price at which a security first trades upon the opening of an exchange on a given trading day. A security's opening price is an important marker for that day's trading activity, especially for those interested in measuring short-term results, such as day traders.

- CLOSE PRICE - The final price at which a security is traded on a given trading day. The closing price represents the most up-to-date valuation of a security until trading commences again on the next trading day.

- PREVIOUS CLOSE - A security's closing price on the preceding day of trading. Previous close can refer to the prior day's value of a stock, bond, commodity, futures or option contract, market index, or any other security. By comparing a security's closing price from one day to the next, investors can see how the security's price has changed over time.

- ADJACENT CLOSE - A stock's closing price on any given day of trading that has been amended to include any distributions and corporate actions that occurred at any time prior to the next day's open. The adjusted closing price is often used when examining historical returns or performing a detailed analysis on historical returns.

- DAY’S LOW - Day's low is the lowest price at which a stock trades over the course of a trading day. Day's low is typically lower than the opening or closing price.

- DAY’S HIGH - Day's high is the highest price at which a stock traded during the course of the day. Day's high is typically higher than the closing or opening price. More often than not this is higher than the closing price.

- SYMBOL - A unique series of letters assigned to a security for trading purposes. They are also known as "ticker symbols."

- VOLUME - The number of shares or contracts traded in a security or an entire market during a given period of time. It is simply the amount of shares that trade hands from sellers to buyers as a measure of activity.

- ASK SIZE - Ask size is the amount of a security that a company is offering to sale.

- BID SIZE - Bid size is the opposite of ask size.

Bid size and ask size are thought to have a relationship which imply that if bid sizes are higher than ask sizes, then there may be a high demand for the stock.

- LAST TRADE - The final day that a futures contract may trade or be closed out before delivery of the underlying asset or cash settlement must occur. By the end of the last trading day, the contract holder must be prepared to accept delivery of the commodity, or settle in cash if the position is not closed.

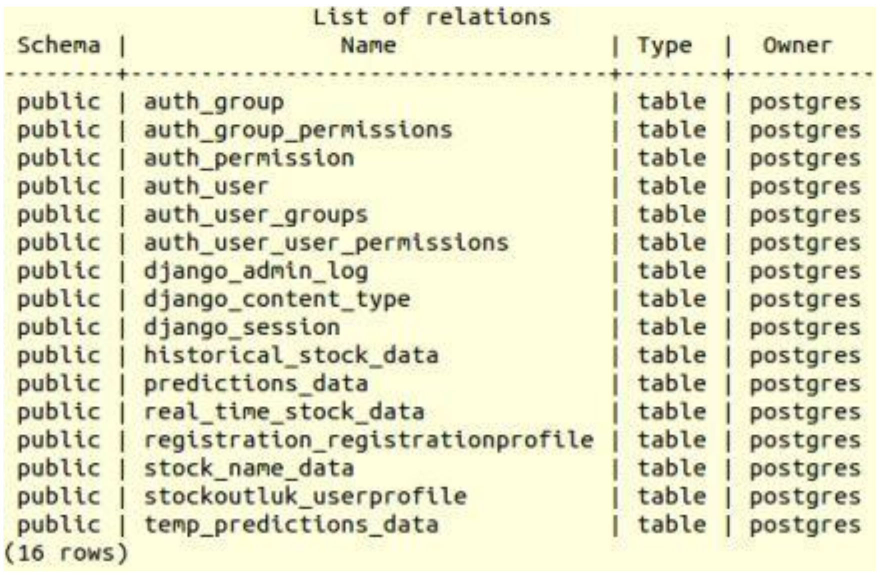
- TRADE DATE - The month, day and year that an order is executed in the market. The trade date is when an order to purchase, sell or otherwise acquire a security is performed. The trade date can apply to the purchase, sale or transfer of bonds, equities, foreign exchange instruments, commodities, futures, etc.

- CHANGE - The difference between the current price and the previous day's settlement price.

The current data is collected only during market hours from Monday to Friday. The current data for a particular stock is collected every minute and is stored on to real_time_stock_data table. The collection of historical data is a one-time process. Historical data of two year’s time till before the start of current data collection is required to be collected. Historical data from 02/01/2010 has been collected and stored on to historical_stock_data table.

Database Tables Used:

We have hosted different tables for the purpose of this project; they are described as follows,



```
                    List of relations
 Schema |               Name                | Type  |  Owner
--------+-----------------------------------+-------+----------
 public | auth_group                        | table | postgres
 public | auth_group_permissions            | table | postgres
 public | auth_permission                   | table | postgres
 public | auth_user                         | table | postgres
 public | auth_user_groups                  | table | postgres
 public | auth_user_user_permissions        | table | postgres
 public | django_admin_log                  | table | postgres
 public | django_content_type               | table | postgres
 public | django_session                    | table | postgres
 public | historical_stock_data             | table | postgres
 public | predictions_data                  | table | postgres
 public | real_time_stock_data              | table | postgres
 public | registration_registrationprofile  | table | postgres
 public | stock_name_data                   | table | postgres
 public | stockoutluk_userprofile           | table | postgres
 public | temp_predictions_data             | table | postgres
(16 rows)
```

Table 1: Various Tables Used in the project

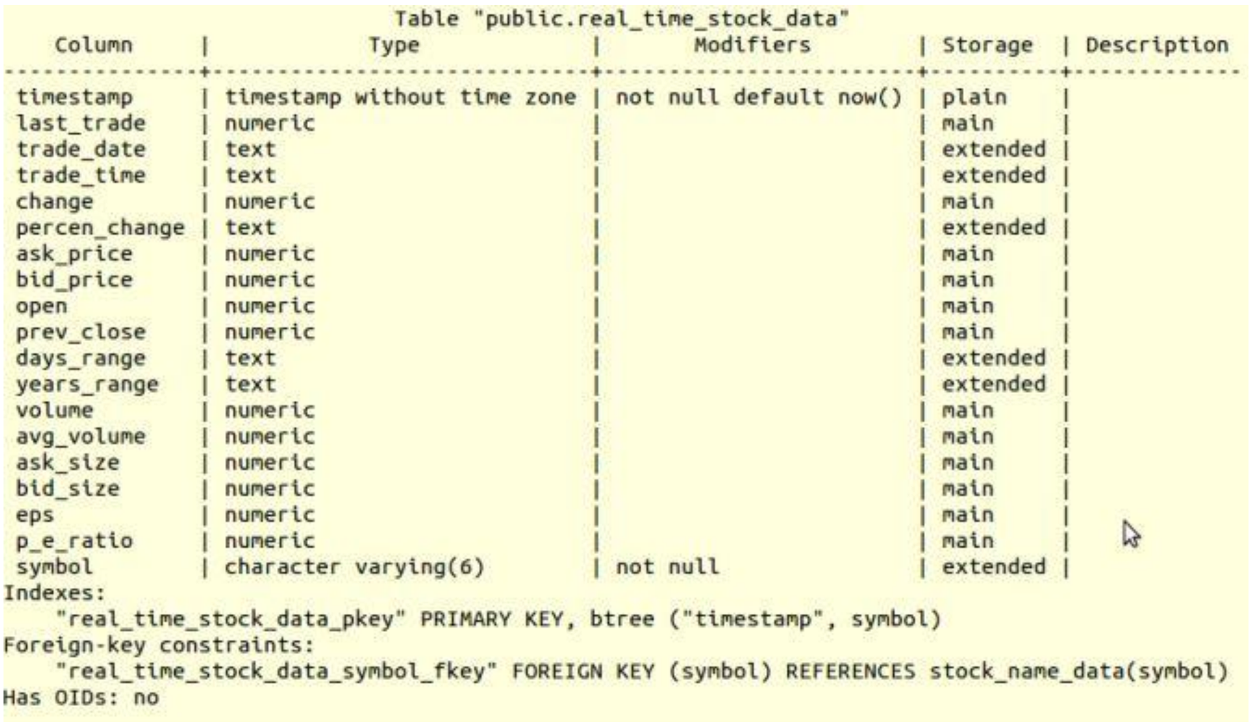
From the above figure the first few tables namely

- auth_group
- auth_group_permissions
- auth_permission
- auth_user
- auth_user_groups
- auth_user_user_permissions
- django_admin_log
- django_content_type
- django_session

These tables were generated automatically as we used django-registration API for the user authentication.

Other tables that we created are used for storing the data collected from Yahoo Finance API and for storing the long term prediction results

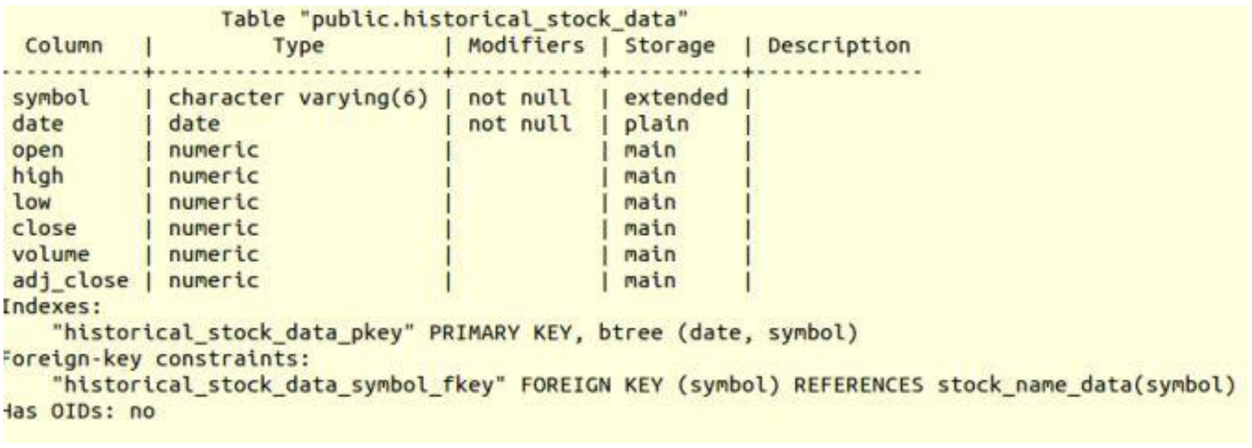
- real_time_stock_data – This table maintains a repository of current stock data which is retrieved from http://finance.yahoo.com. This table has a timestamp field which is populated using the system timestamp at the time of loading the table with the corresponding vale which was just retrieved from the Yahoo Finance API.



```
                              Table "public.real_time_stock_data"
    Column     |            Type             |       Modifiers        | Storage  | Description
---------------+-----------------------------+------------------------+----------+-------------
 timestamp     | timestamp without time zone | not null default now() | plain    |
 last_trade    | numeric                     |                        | main     |
 trade_date    | text                        |                        | extended |
 trade_time    | text                        |                        | extended |
 change        | numeric                     |                        | main     |
 percen_change | text                        |                        | extended |
 ask_price     | numeric                     |                        | main     |
 bid_price     | numeric                     |                        | main     |
 open          | numeric                     |                        | main     |
 prev_close    | numeric                     |                        | main     |
 days_range    | text                        |                        | extended |
 years_range   | text                        |                        | extended |
 volume        | numeric                     |                        | main     |
 avg_volume    | numeric                     |                        | main     |
 ask_size      | numeric                     |                        | main     |
 bid_size      | numeric                     |                        | main     |
 eps           | numeric                     |                        | main     |
 p_e_ratio     | numeric                     |                        | main     |
 symbol        | character varying(6)        | not null               | extended |
Indexes:
    "real_time_stock_data_pkey" PRIMARY KEY, btree ("timestamp", symbol)
Foreign-key constraints:
    "real_time_stock_data_symbol_fkey" FOREIGN KEY (symbol) REFERENCES stock_name_data(symbol)
Has OIDs: no
```

Table 2: Table for real time data

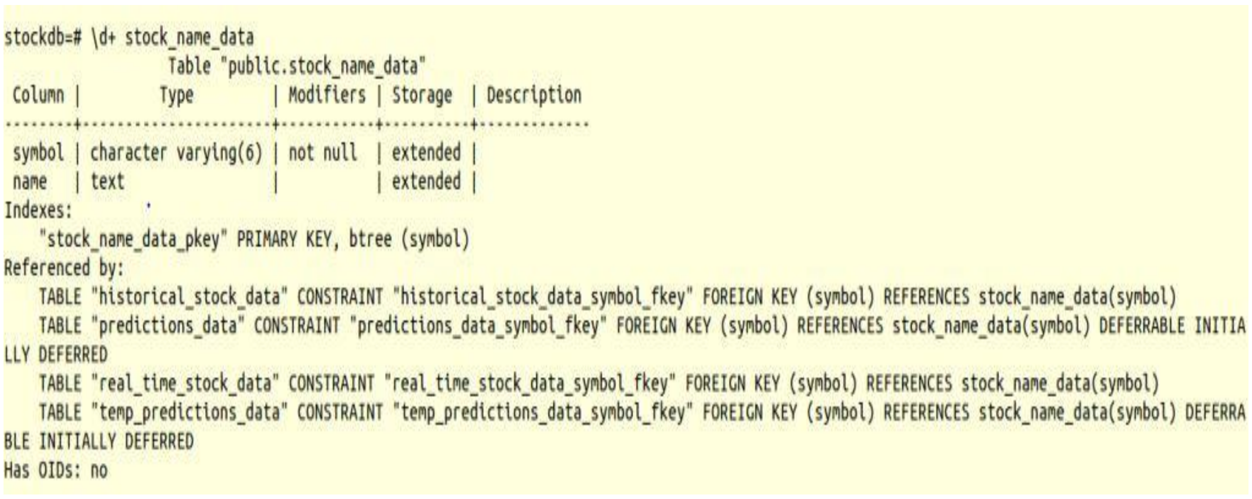
- historical_stock_data – This table maintains a repository of historical stock data from the past two years which has been retrieved from http://ichart.yahoo.com .



```
                   Table "public.historical_stock_data"
  Column    |         Type          | Modifiers | Storage  | Description
-----------+----------------------+-----------+----------+-------------
 symbol    | character varying(6) | not null  | extended |
 date      | date                 | not null  | plain    |
 open      | numeric              |           | main     |
 high      | numeric              |           | main     |
 low       | numeric              |           | main     |
 close     | numeric              |           | main     |
 volume    | numeric              |           | main     |
 adj_close | numeric              |           | main     |
Indexes:
    "historical_stock_data_pkey" PRIMARY KEY, btree (date, symbol)
Foreign-key constraints:
    "historical_stock_data_symbol_fkey" FOREIGN KEY (symbol) REFERENCES stock_name_data(symbol)
Has OIDs: no
```

Table 3: Table for historical data

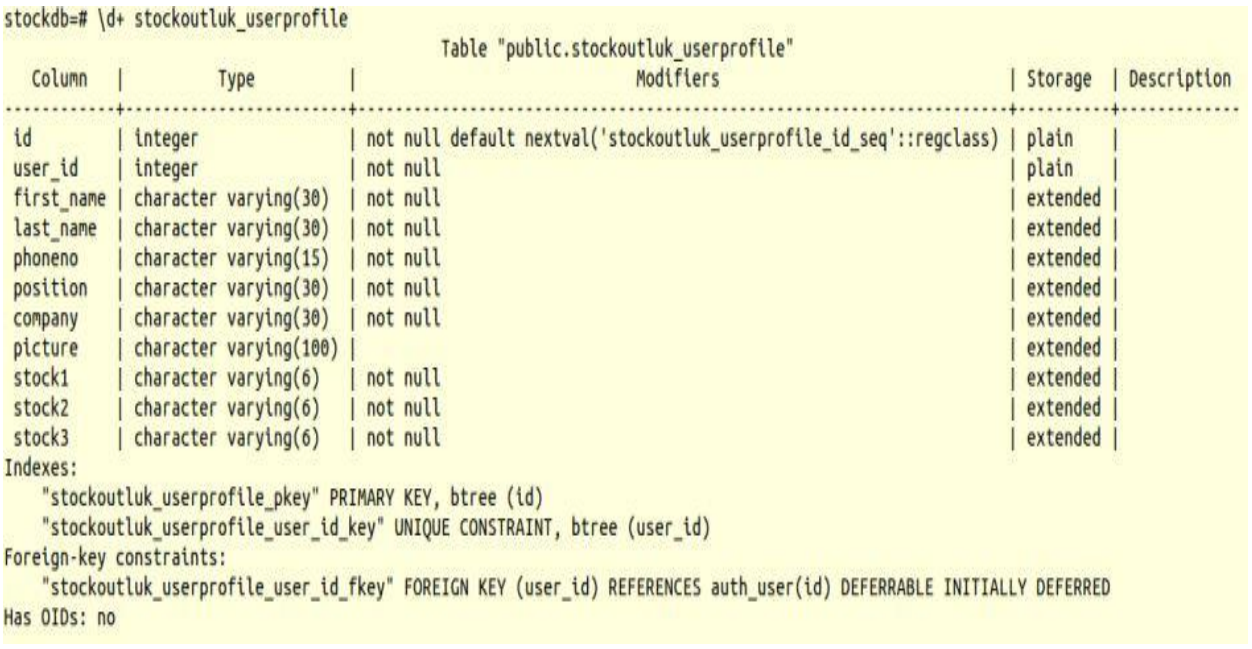
- stock_name_data – This table maintains the list of all the symbols used in our project and their corresponding names. We are using this table to maintain the names of the various symbols as it is an efficient way to maintain the names in the database rather than maintaining them in the application. It also avoids duplication of data which would otherwise exist if this information is stored in the above mentioned tables.



```
stockdb=# \d+ stock_name_data
                  Table "public.stock_name_data"
 Column |         Type         | Modifiers | Storage  | Description
--------+----------------------+-----------+----------+-------------
 symbol | character varying(6) | not null  | extended |
 name   | text                 |           | extended |
Indexes:
    "stock_name_data_pkey" PRIMARY KEY, btree (symbol)
Referenced by:
    TABLE "historical_stock_data" CONSTRAINT "historical_stock_data_symbol_fkey" FOREIGN KEY (symbol) REFERENCES stock_name_data(symbol)
    TABLE "predictions_data" CONSTRAINT "predictions_data_symbol_fkey" FOREIGN KEY (symbol) REFERENCES stock_name_data(symbol) DEFERRABLE INITIA
LLY DEFERRED
    TABLE "real_time_stock_data" CONSTRAINT "real_time_stock_data_symbol_fkey" FOREIGN KEY (symbol) REFERENCES stock_name_data(symbol)
    TABLE "temp_predictions_data" CONSTRAINT "temp_predictions_data_symbol_fkey" FOREIGN KEY (symbol) REFERENCES stock_name_data(symbol) DEFERRA
BLE INITIALLY DEFERRED
Has OIDs: no
```

Table 4: Table for stock names

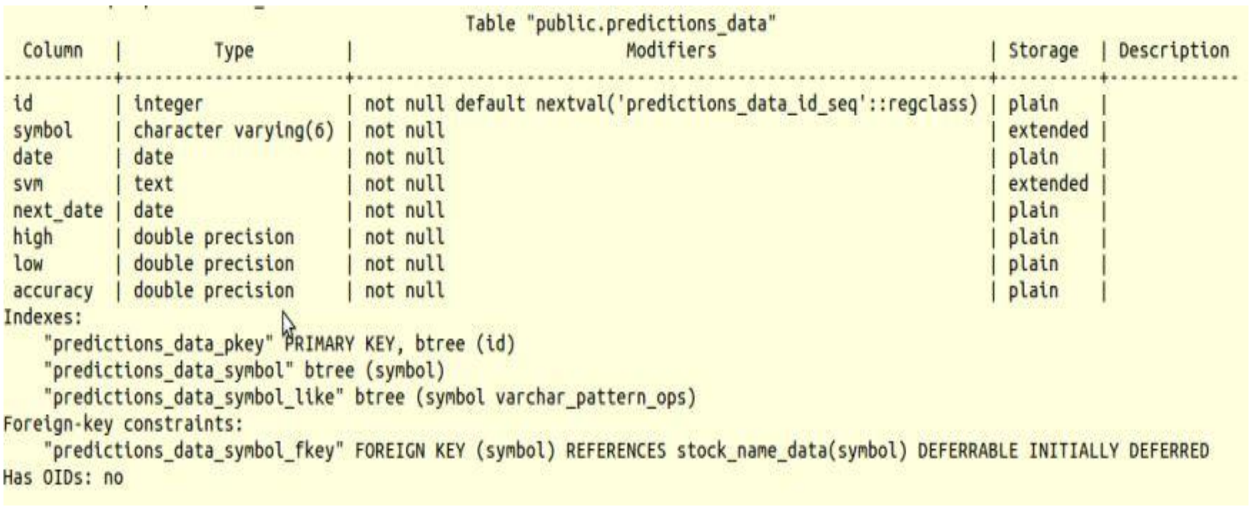
- stockoutluk_userprofiles – This table maintains the additional details of the user that are provided during the registration which are different from the information provided for authentication



```
stockdb=# \d+ stockoutluk_userprofile
                                        Table "public.stockoutluk_userprofile"
  Column   |         Type          |                              Modifiers                               | Storage  | Description
-----------+-----------------------+----------------------------------------------------------------------+----------+------------
 id        | integer               | not null default nextval('stockoutluk_userprofile_id_seq'::regclass) | plain    |
 user_id   | integer               | not null                                                             | plain    |
 first_name | character varying(30) | not null                                                            | extended |
 last_name | character varying(30) | not null                                                             | extended |
 phoneno   | character varying(15) | not null                                                             | extended |
 position  | character varying(30) | not null                                                             | extended |
 company   | character varying(30) | not null                                                             | extended |
 picture   | character varying(100) |                                                                     | extended |
 stock1    | character varying(6)  | not null                                                             | extended |
 stock2    | character varying(6)  | not null                                                             | extended |
 stock3    | character varying(6)  | not null                                                             | extended |
Indexes:
    "stockoutluk_userprofile_pkey" PRIMARY KEY, btree (id)
    "stockoutluk_userprofile_user_id_key" UNIQUE CONSTRAINT, btree (user_id)
Foreign-key constraints:
    "stockoutluk_userprofile_user_id_fkey" FOREIGN KEY (user_id) REFERENCES auth_user(id) DEFERRABLE INITIALLY DEFERRED
Has OIDs: no
```

Table 5: Table for storing user profiles

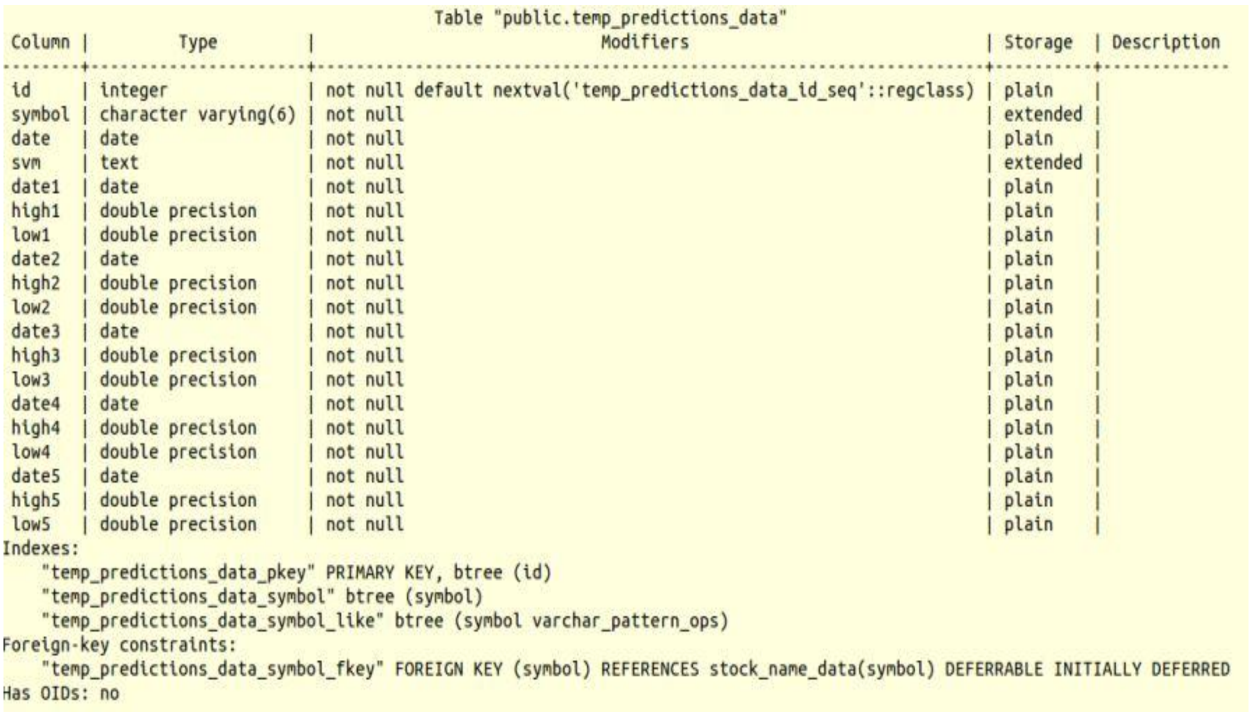
- prediction_data – This table is polulated with values only after we are able to obtain real data, so for example if we have made prediction for tomorrow and populated temp_prediction_data today, tomorrow our application will retrieve actual data from Yahoo API and will calculate accuracy of our prediction and only then we store an entry into prediction_data.



```
                                        Table "public.predictions_data"
  Column   |         Type         |                           Modifiers                            | Storage  | Description
-----------+----------------------+-----------------------------------------------------------------+----------+------------
 id        | integer              | not null default nextval('predictions_data_id_seq'::regclass) | plain    |
 symbol    | character varying(6) | not null                                                        | extended |
 date      | date                 | not null                                                        | plain    |
 svm       | text                 | not null                                                        | extended |
 next_date | date                 | not null                                                        | plain    |
 high      | double precision     | not null                                                        | plain    |
 low       | double precision     | not null                                                        | plain    |
 accuracy  | double precision     | not null                                                        | plain    |
Indexes:
    "predictions_data_pkey" PRIMARY KEY, btree (id)
    "predictions_data_symbol" btree (symbol)
    "predictions_data_symbol_like" btree (symbol varchar_pattern_ops)
Foreign-key constraints:
    "predictions_data_symbol_fkey" FOREIGN KEY (symbol) REFERENCES stock_name_data(symbol) DEFERRABLE INITIALLY DEFERRED
Has OIDs: no
```

Table 6: Table for storing prediction values

- temp_prediction_data – This table is used to store long term prediction data made during invocation of the long term prediction function. Basically for each stock prediction is made once a day, so that all users get the same prediction for the same stock. The first user who clicks on long term prediction button for a given stock invokes long term prediction function. as you can see in that table are stored dates and made prediction for the next five days.



```
                                              Table "public.temp_predictions_data"
 Column |          Type          |                              Modifiers                              | Storage  | Description
--------+------------------------+---------------------------------------------------------------------+----------+------------
 id     | integer                | not null default nextval('temp_predictions_data_id_seq'::regclass) | plain    |
 symbol | character varying(6)   | not null                                                            | extended |
 date   | date                   | not null                                                            | plain    |
 svm    | text                   | not null                                                            | extended |
 date1  | date                   | not null                                                            | plain    |
 high1  | double precision       | not null                                                            | plain    |
 low1   | double precision       | not null                                                            | plain    |
 date2  | date                   | not null                                                            | plain    |
 high2  | double precision       | not null                                                            | plain    |
 low2   | double precision       | not null                                                            | plain    |
 date3  | date                   | not null                                                            | plain    |
 high3  | double precision       | not null                                                            | plain    |
 low3   | double precision       | not null                                                            | plain    |
 date4  | date                   | not null                                                            | plain    |
 high4  | double precision       | not null                                                            | plain    |
 low4   | double precision       | not null                                                            | plain    |
 date5  | date                   | not null                                                            | plain    |
 high5  | double precision       | not null                                                            | plain    |
 low5   | double precision       | not null                                                            | plain    |
Indexes:
    "temp_predictions_data_pkey" PRIMARY KEY, btree (id)
    "temp_predictions_data_symbol" btree (symbol)
    "temp_predictions_data_symbol_like" btree (symbol varchar_pattern_ops)
Foreign-key constraints:
    "temp_predictions_data_symbol_fkey" FOREIGN KEY (symbol) REFERENCES stock_name_data(symbol) DEFERRABLE INITIALLY DEFERRED
Has OIDs: no
```

Table 7: Table for storing temporary values used in prediction

8. References:

- 'Python' - www.**python**.org
- 'PostgreSQL' - www.**postgresql**.org
- 'Psycopg' - http://initd.org/psycopg
- 'BitBucket' – https://bitbucket.org
- 'Django' – www.djangoproject.com
- 'Bootstrap3' – http://getbootstrap.com/components
- 'Django – Registration API' – http://django-registration.readthedocs.org/en/latest
- 'Local Flavor' – https://docs.djangoproject.com/en/1.4/ref/contrib/localflavor
- 'Use Case' – http://en.wikipedia.org/wiki/Use_Case_Diagram
- 'Math Plot Lib' – http://matplotlib.org
- 'Bayesian Predictor' – http://www.bayesian-inference.com/prediction
- 'SVM' – http://en.wikipedia.org/wiki/Support_vector_machine
- 'Scikit' – http://scikit-learn.org/stable/modules/svm.html
- 'Scikit Tool for GBRT' – http://scikitlearn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingRegressor.html
- 'GBRT' – http://en.wikipedia.org/wiki/Gradient_boosting
- 'Definition of the various fields' - http://www.jarloo.com/yahoo_finance
- 'Symbols-Information' - http://finance.yahoo.com/q/hp?s=GOOG+Historical+Prices
- http://finance.yahoo.com/q/hp
- http://finance.yahoo.com
- 'Yahoo Finance API' - https://code.google.com/p/yahoo-finance managed/wiki/YahooFinanceAPIs
- 'Stock market prediction' - http://en.wikipedia.org/wiki/Stock_market_prediction
- 'A Tutorial on Support Vector Machines for Pattern Recognition' - http://research.microsoft.com/~cburges/papers/SVMTutorial.pdf.
- http://www.sqlalchemy.org
- http://www.numpy.org
- http://www.w3schools.com/html/default.asp
- http://www.w3schools.com/xml/default.asp
- http://www.w3schools.com/javascript/default.asp
- http://www.w3schools.com/sql/default.asp
- Shunrong Shen, Haomiao Jiang., Tongda Zhang., "Stock Market Forecasting Using Machine Learning Algorithms," Computers & Operations Research.