

ECE-568: Software Engineering of Web Applications

FINAL PROJECT REPORT

Web-based Stock Forecaster

Instructor: Prof. Ivan Marsic

Group 802 Team Members

Shruthi Kiran
Manasi Jagannatha
Tripti Singh
Soumya Sampath

Project website: <http://software-ece.rutgers.edu/~group802/>

Submitted on: 05/07/2008

This document summarizes the architecture diagrams, use cases, class diagram, interaction diagram, hardware requirements, algorithms and other implementation details of Project for the Spring2008 – Web-based Stock Forecaster.

Table of Contents

1. Gantt Charts.....	3
2. Summary of Changes.....	9
3. Glossary of terms.....	10
4. Functional Requirements Specifications	11
4.1 Requirements	11
4.2 Goals and Features:.....	11
4.3 Actors and Goals.....	12
5. SOA and Web Services:	13
6. System Architecture	15
6.1 Architecture Diagram.....	15
6.2 LAMP	15
6.3 R – Prediction Model	16
7. Hardware to Software Interface Diagram.....	17
8. Algorithm for Stock Forecasting	19
9. Four Step Strategy for stock forecasting	20
10. Use case Diagrams:.....	22
10.1 Use Case 1: Get current stock quotes to database.....	22
10.2 Use Case 2: Set Timer Interval	23
10.3 Use Case 3: Display stock quotes at the client.	24
10.4 Use Case 4: Generate predicted stock values and store it in Stock Database.....	25
10.5 Use Case 5: Generate Recommendation.....	26
11. HTTP Class Diagram.....	27
12. Mathematical Models.....	28
12.1 Regression.....	28
12.2 Generating the Regression Model.....	31
12.3 Algorithm for regression model.....	33
13. System Screen shots.....	34
14. Format of tables in the data base for getstock().....	38
15. Web Service Implementation.....	39
15.1 List of files in Web Services.....	39
15.2 Soap Messages	41
16. Conclusion:	44
17. Future work:.....	45
18. References.....	46
19. Appendix.....	47
19.1 List of companies used along with their ticker symbols.....	47

1. Gantt Charts

Web Based Stock Forecaster

1/25/08-5/9/08

GanttProject Report

Project : Web Based Stock Forecaster

Start : 1/25/08

End : 5/9/08

Organization : Rutgers- The State University of New Jersey

Web Link : <http://software-ece.rutgers.edu/~group802/>

Description :

Date : May 7, 2008 9:26:15 PM

Tasks List

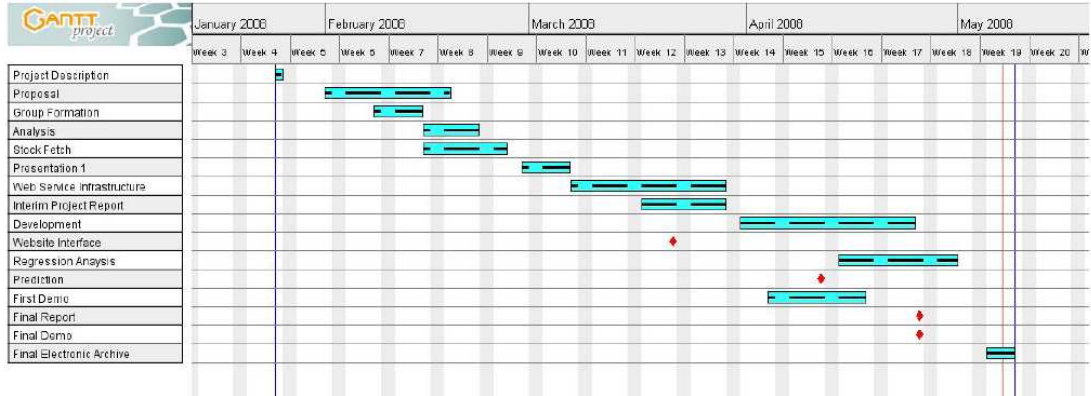
Name	Start	End	Milesto	%	Resources	Notes
Project Description	1/25/08	1/26/08	false	100		
Proposal	2/1/08	2/19/08	false	100		
Group Formation	2/8/08	2/15/08	false	100	Shruti GaviGowda Soumya Sampath Tripti Singh Richa Mishra Manasi Jagannatha	
Analysis	2/15/08	2/23/08	false	100	Shruti GaviGowda Soumya Sampath Tripti Singh Richa Mishra Manasi Jagannatha	
Stock Fetch	2/15/08	2/27/08	false	100	Shruti GaviGowda Soumya Sampath Tripti Singh Richa Mishra Manasi Jagannatha	
Presentation 1	2/29/08	3/7/08	false	100	Shruti GaviGowda Soumya Sampath Tripti Singh Richa Mishra Manasi Jagannatha	
Web Service Infrastructure	3/7/08	3/29/08	false	100	Shruti GaviGowda Soumya Sampath Tripti Singh Richa Mishra Manasi Jagannatha	
Interim Project Report	3/17/08	3/29/08	false	100	Shruti GaviGowda Soumya Sampath Tripti Singh Richa Mishra Manasi Jagannatha	
Development	3/31/08	4/25/08	false	100	Shruti GaviGowda Soumya Sampath Tripti Singh Richa Mishra Manasi Jagannatha	
Website Interface	3/21/08	5/3/08	true	100	Shruti GaviGowda Soumya Sampath Tripti Singh Richa Mishra Manasi Jagannatha	
Regression Anaysis	4/14/08	5/1/08	false	100	Shruti GaviGowda Soumya Sampath	

						Tripti Singh Richa Mishra Manasi Jagannatha
Prediction	4/11/08	5/3/08	true	100		Shruti GaviGowda Soumya Sampath Tripti Singh Manasi Jagannatha Richa Mishra
First Demo	4/4/08	4/18/08	false	100		Shruti GaviGowda Soumya Sampath Tripti Singh Manasi Jagannatha
Final Report	4/25/08	5/3/08	true	100		Shruti GaviGowda Soumya Sampath Tripti Singh Manasi Jagannatha
Final Demo	4/25/08	5/9/08	true	100		Shruti GaviGowda Soumya Sampath Tripti Singh Manasi Jagannatha
Final Electronic Archive	5/5/08	5/9/08	false	100		Shruti GaviGowda Soumya Sampath Tripti Singh Manasi Jagannatha

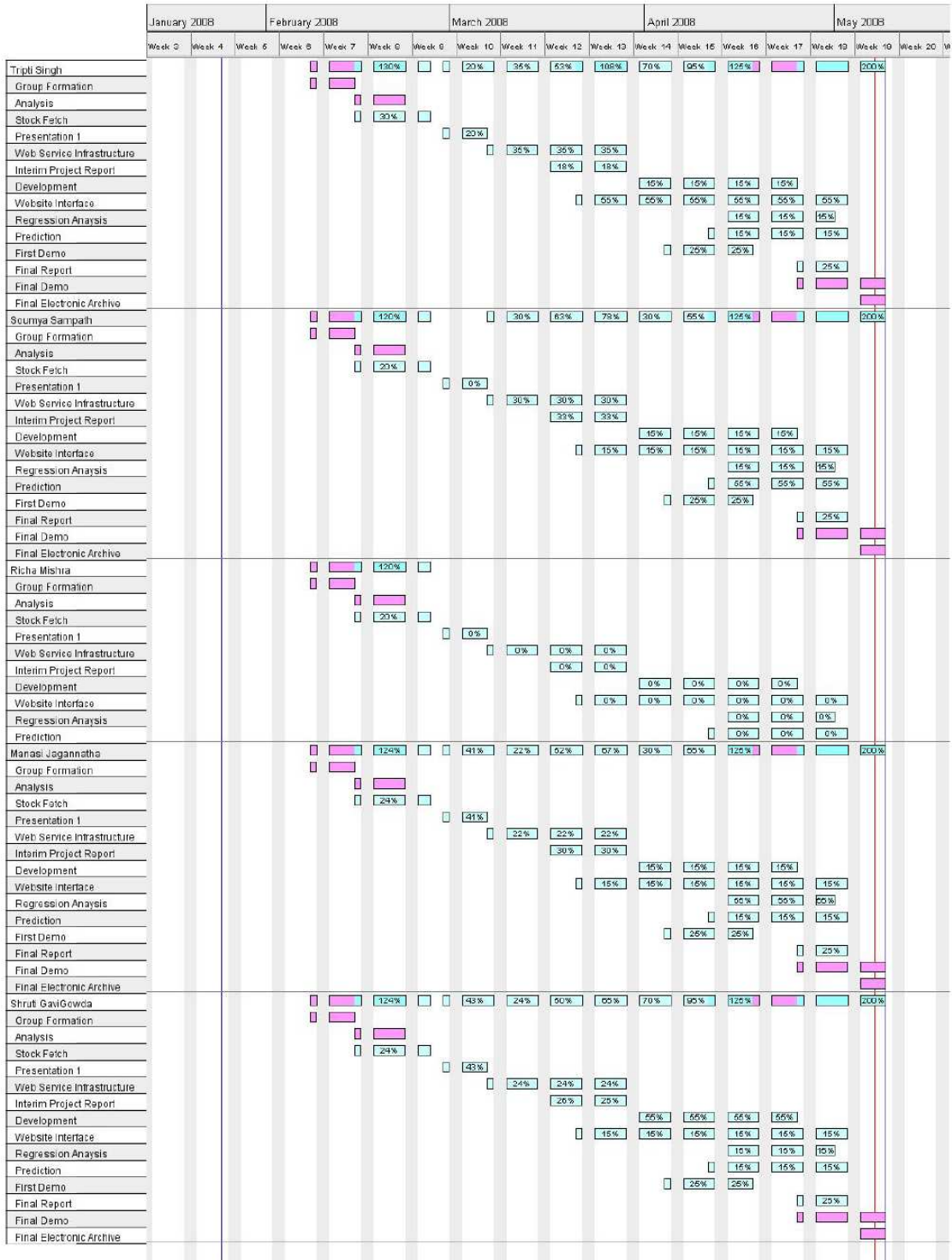
Resources List

Name	Default role	Mail	Phone
Name:Tripti Singh	undefined		
Name:Soumya Sampath	undefined		
Name:Richa Mishra	undefined		
Name:Manasi Jagannatha	undefined		
Name:Shruti GaviGowda	undefined		

Gantt Chart



Resources Chart



2. Summary of Changes

The key revisions since the last report are:

1. Inclusion of three new use cases, UC3, UC4 and UC5. UC3 gives the stock quotes displayed on the client's local machine. UC4 generates the predicated values and stored in the database. UC5 gives the recommendations for the selected stock.
2. Hardware and Software Interface Diagram.
3. Flow chart to give the complete description of the project.
4. Class diagram.
5. Number of ticker symbols increased to 60 companies.
6. Database tables and the variables.

3. Glossary of terms

Predictor: Provide a decision on whether the investor has to SELL, BUY or HOLD.

Confidence intervals - Used to indicate the reliability of an estimate.

Server Administrator – This is the server side user who starts the server and is in charge of changing update intervals, database information, and any other administrative issues.

Stock Database – This is the database that contains the current information about stocks being tracked by the web application. It also stores the predicated values generated by regression model.

Stock List – This is the list of stocks that the system keeps track, for which the system can give recommendation.

System Menu – This is the text based menu that the Server Administrator is shown once the server is running.

Timer – This is an abstract concept referring to an entity that notifies listeners on timeout.

Timer Interval – This is the amount of time that occurs between recurrent use cases.

Technical Indicators – Used to record the changes in the stock price.

Yahoo! Finance Library – This is a library whose capabilities include parsing Yahoo! Finance.

4. Functional Requirements Specifications

The goal of this project is to implement a Web service for stock-prediction. Each Web service (WS) will track different stocks and when queried, issue a forecast about the price movement for a given stock. The client module acts as a “facilitator” which gathers information about the stock prices.

4.1 Requirements

1. The system should download historic quotes for the chosen companies dynamically and store it in a database. The historic quotes are retrieved from [yahoo finance](http://www.finance.yahoo.com/) (<http://www.finance.yahoo.com/>).
2. The system authenticates the users who have registered and allows new users to register.
3. The system predicts if the user should SELL, BUY or HOLD for the chosen stock.
4. A performance evaluation is given to validate the prediction model.

4.2 Goals and Features:

1. Our target customers are individuals who are trading moderately (up to several times per week), but not very frequently (several times per day). We have aimed at short term prediction.
2. Given a list of about 60 companies, we record their quoted prices and volumes at the maximum available sampling density, also record some broad market indices. The stock prices are retrieved from a third party source such as [yahoo finance](#), that monitor stock exchange and maintain up to date stock prices. The data is gathered and stored locally.
3. The gathered data is used for developing the prediction model, which is a simple regression-curve fitting model. We are using both technical indicators and market proxies along with stock prices and trading volumes. The model considers both the individual company’s data as well as the broad market data.
4. Potential services that we offer to provide by the forecaster service include:
 - Given a stock A, we suggest an action, such as “buy,” “sell,” “hold,” or “sit-out;” we have assumed that the forecaster provides recommendation for one stock at a time.
 - Recommend a stock to buy, from all stocks that are being tracked, or from all in a given industry/sector

- Provide confidence level of how confident is the forecaster about the prediction; the range is 95 – 99 %.

5. Authentication and registration:

We have included this feature to ensure a user friendly and convenient experience for our users. The detailed user description is stored in the database for returning users. The user is asked to choose a login id and password for authentication.

User Functions

Web Client(Investor)
Registration
Stock browsing and searching
Viewing market prices and history
Recommendation on chosen stock

Backend operations

Retrieval of stock prices (Periodic)
On demand analysis Of data for prediction
Server to client connectivity

Figure 1: Logical Grouping of required functions for Stock Prediction
Authors: Manasi Jagannatha and Tripti Singh

4.3 Actors and Goals

Format - [Actor: Goal]

Investor: Risk-free investment in the world of stock investing

Investor Table: Provide relevant information about investor and relevant information about his or her portfolio.

Predictor: Provide a decision on whether the user has to SELL, BUY or HOLD.

Stock Database: Provide information about the historic current stock values.

Predicted value Table: Provide information from the regression model.

Timer: Notify listeners when the time interval runs out.

Yahoo! Finance: Provide current (or 10 minutes delayed) stock prices updates.

5. SOA and Web Services:

A Web Service is any piece of software that makes itself available over the Internet and uses a standardized XML messaging system. ([http:// www.W3schools.com/](http://www.W3schools.com/))

SOA is a concept that enables resources to be provided and consumed as services, allowing for dynamic information sharing. [8]. Web services are currently the preferred technology to implement an SOA. The main functional components of web services are the service provider, service consumer and service registry as illustrated in the figure. A service provider can publish the services it is willing to share with others in a service registry that announces their availability. A service consumer may browse the service registry to retrieve the relevant announcements that describe where and how the services may be invoked.

Web services are based on a set of XML-based standards. Three of the basic specifications are: Simple Object Access Protocol (SOAP), Web Services Description Language (WSDL) and Universal Description Discovery and Integration (UDDI).

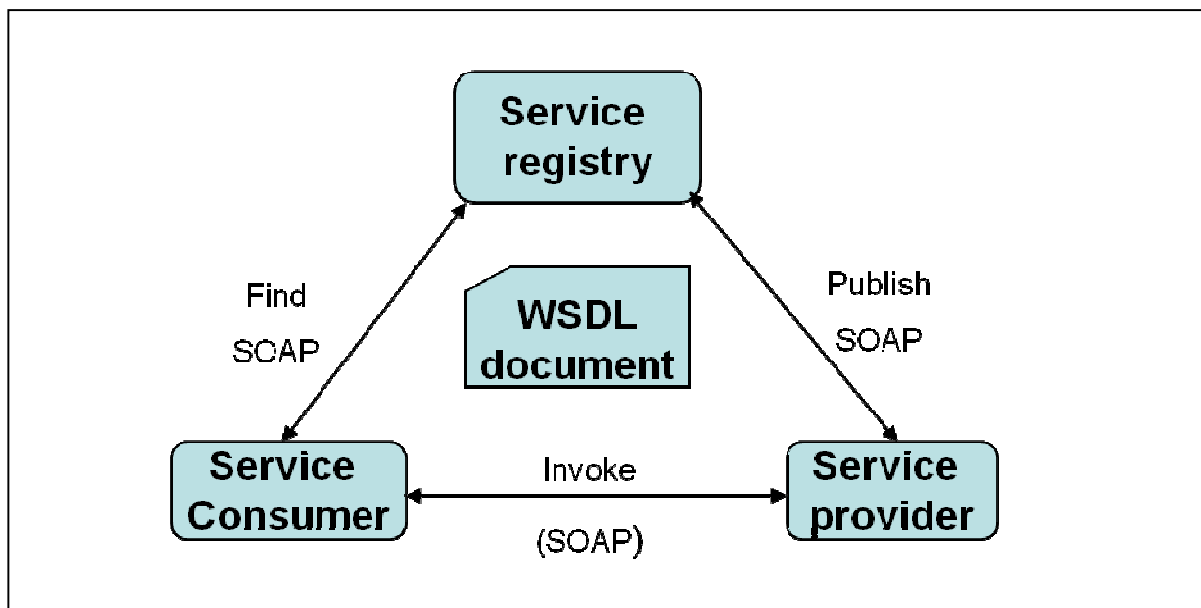


Figure 2: Representation of Web service and its components.
Author: Soumya Sampath

SOAP is the XML messaging protocol used for transport of information between the web services components. An important property of SOAP messages is that they can be delivered over a number of application, transport, and network protocols. Today HTTP is normally used for transporting the SOAP messages between hosts. Other possible protocols are Simple Mail Transfer Protocol (SMTP), File Transfer Protocol (FTP), Java Messaging Service (JMS), and so on.

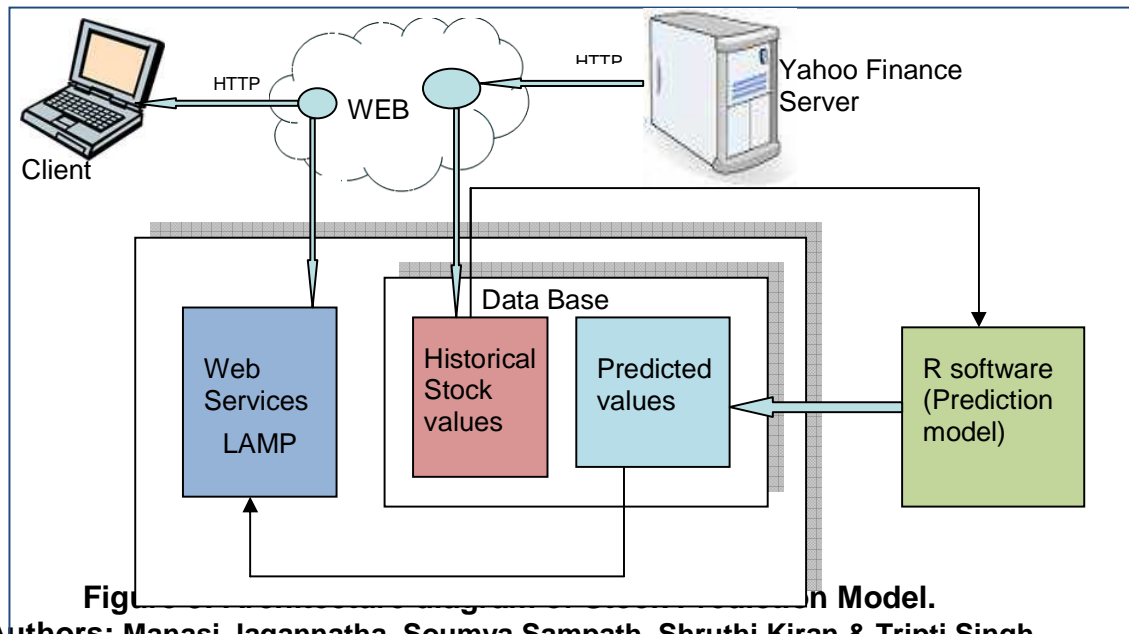
The service registry is often realized using UDDI, which provides mechanisms for publishing and discovery of services. More specifically, UDDI provides access to the

WSDL documents describing the protocol bindings and message formats required to interact with the web services listed in its directory. UDDI can be used for both design time and run time discovery of services. In an SOA, consumers and providers need not know each others locations; they only need to know where the service registry is. Likewise, the service registry need not know in advance where the consumers and providers are located. Because minimal preplanning is required there is:

- Easier sharing of information
- Faster fielding of new technologies
- Dynamic reconfiguration of functionality
- Adhoc organization of entities.

6. System Architecture

6.1 Architecture Diagram



6.2 LAMP

The acronym **LAMP** refers to a solution stack of software, usually free and open source software, used to run dynamic Web sites or servers. The original expansion is as follows:

- Linux, referring to the operating system;
- Apache, the Web server;
- MySQL the database management system (or database server);
- Perl, Python, and PHP, the programming languages.

The combination of these technologies is used primarily to define a web server infrastructure, define a programming paradigm of developing software, and establish a software distribution package.

Apache supports a variety of features, many implemented as compiled modules which extend the core functionality. These can range from server-side programming language support to authentication schemes. The language interfaces support mod_perl, mod_python, Tcl, and PHP. Popular authentication modules include mod_access, mod_auth, and mod_digest.

Compression methods on Apache include the external extension module, mod_gzip, implemented to help with reduction of the size (weight) of web pages served

over HTTP. Apache logs can be analyzed through a web browser using free scripts such as AWStats/W3Perl or Visitors.

Apache features configurable error messages, DBMS-based authentication databases, and content negotiation. It is also supported by several graphical user interfaces (GUIs) which permit easier, more intuitive configuration of the server.

Programmers developing web applications often use a locally installed version of Apache in order to preview and test code as it is being developed.(www.wikipedia.com)

6.3 R – Prediction Model

<http://www.r-project.org/>

R is a language and environment for statistical computing and graphics. R provides a wide variety of statistical (linear and non linear modeling, classical statistical tests, time-series analysis, classification, clustering ...) and graphical techniques, and is highly extensible. R provides an Open Source route to participation in that activity.

One of R's strengths is the ease with which well-designed publication-quality plots can be produced, including mathematical symbols and formulae where needed. Great care has been taken over the defaults for the minor design choices in graphics, but the user retains full control.

R is available as Free Software under the terms of the Free Software Foundation's GNU General Public License in source code form. It compiles and runs on a wide variety of UNIX platforms and similar systems (including FreeBSD and Linux), Windows and MacOS.

R is an integrated suite of software facilities for data manipulation, calculation and graphical display. It includes

- an effective data handling and storage facility,
- a suite of operators for calculations on arrays, in particular matrices,
- a large, coherent, integrated collection of intermediate tools for data analysis,
- graphical facilities for data analysis and display either on-screen or on hardcopy, and
- a well-developed, simple and effective programming language which includes conditionals, loops, user-defined recursive functions and input and output facilities.

7. Hardware to Software Interface Diagram

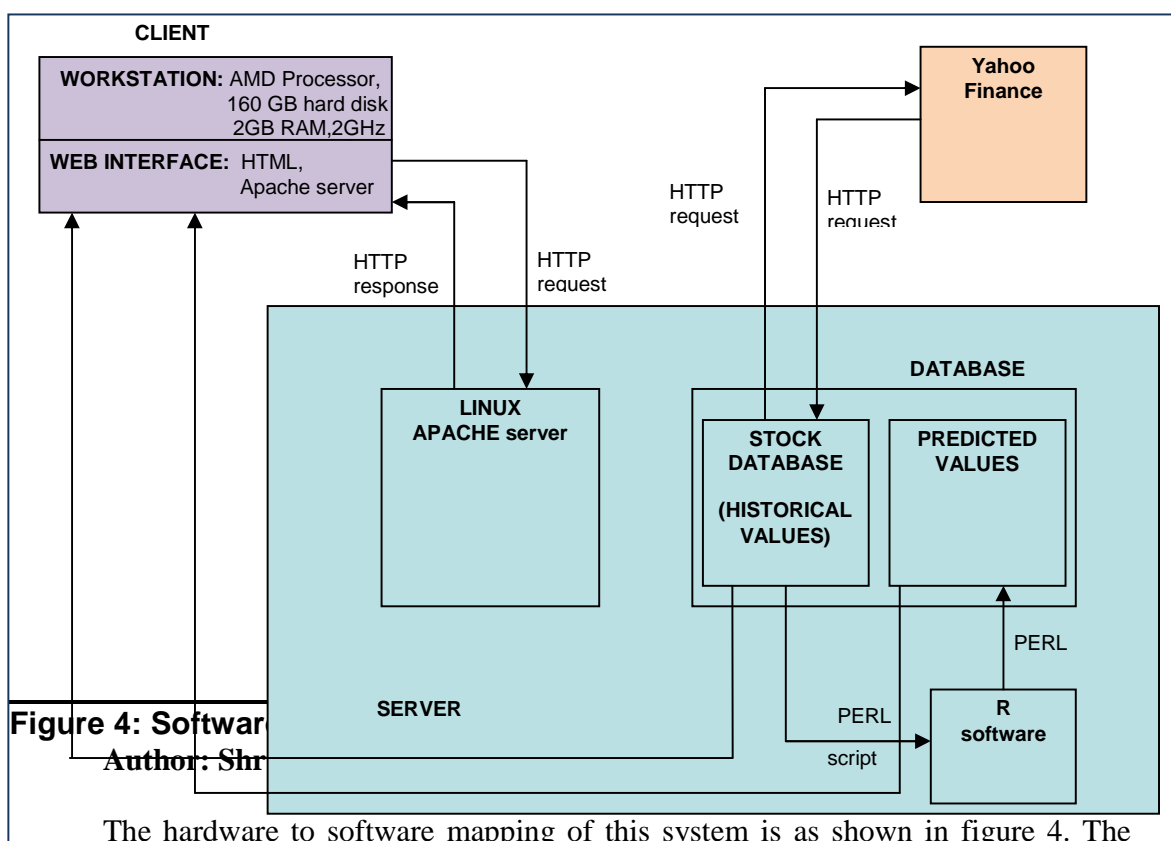


Figure 4: Software to Hardware Mapping
 Author: Shr

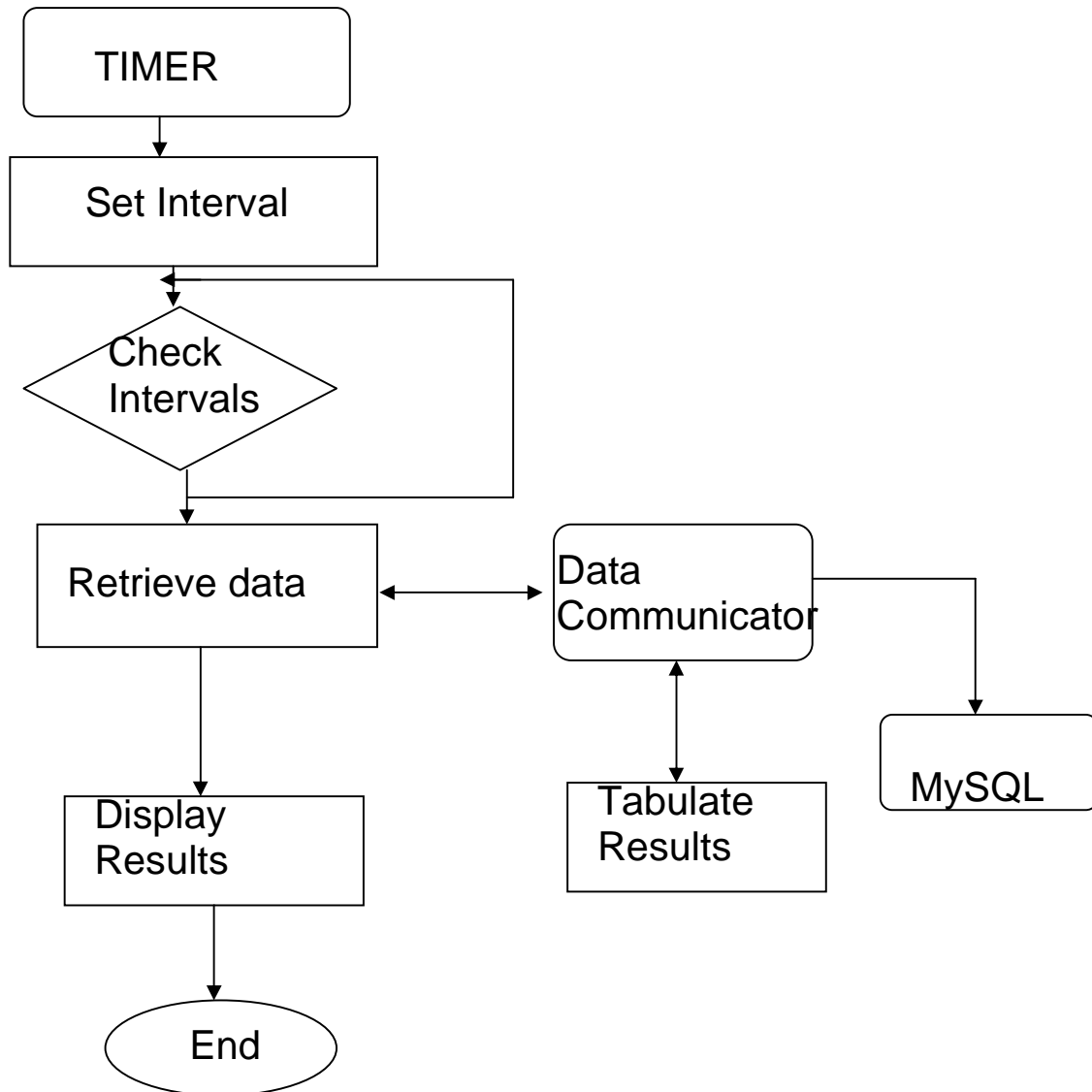
The hardware to software mapping of this system is as shown in figure 4. The hardware and software components work hand in hand in achieving the deliverable. The client as well as the server houses an Apache Server. A PHP script is run on this apache server thereby establishing a connection between the server and the client. The user interface, a web page written as a HTML code in which the PHP script is embedded is where the user makes a query.

When a query is made the entry is passed as a variable in the PHP script. Once the message reaches the server, the server performs a real time invocation of Yahoo finance (<http://finance.yahoo.com/>). The historical data of the company is downloaded on to the MySQL database. The server houses the MySQL database and the R application. The R application makes use of these historical values and fits it into its prediction model thereby giving the required prediction. This predicted output is then stored in the database which in turn is retrieved by the client using another PHP script.

As an analogy, let us consider a restaurant. Here, the PHP can be compared to the steward, the Apache server being the master chef and the MySQL database being the stock room. When a guest (user) arrives, he places an order (makes a query) for a dish (company) he chooses. This order is carried to the chef (Apache server) by the steward (PHP). The guest places the order and waits for the dish to arrive. It is of no concern to him how the dish is prepared or what goes into it. The user in our case is completely oblivious to how the query is passed, or how the data is retrieved and processed. He only

waits for the end results. The chef now has the details of the dish and he uses the ingredients (data) from his stock room (database) in order to prepare the dish. Once the dish is ready (predicted value), the steward (PHP) comes into picture again. He delivers the dish back to the guest. This highly interconnected system requires optimum synchronization to have the system up-and-running.

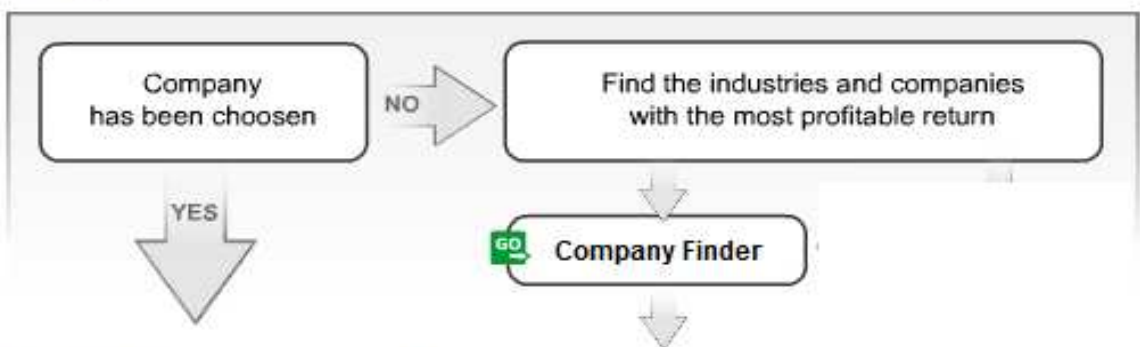
8. Algorithm for Stock Forecasting



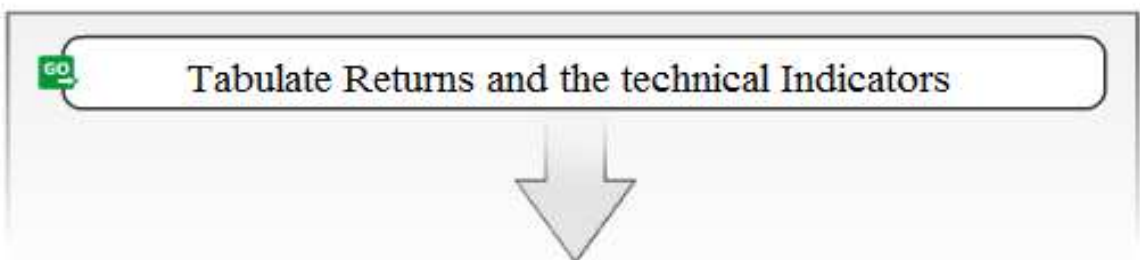
9. Four Step Strategy for stock forecasting

Four Steps Strategy

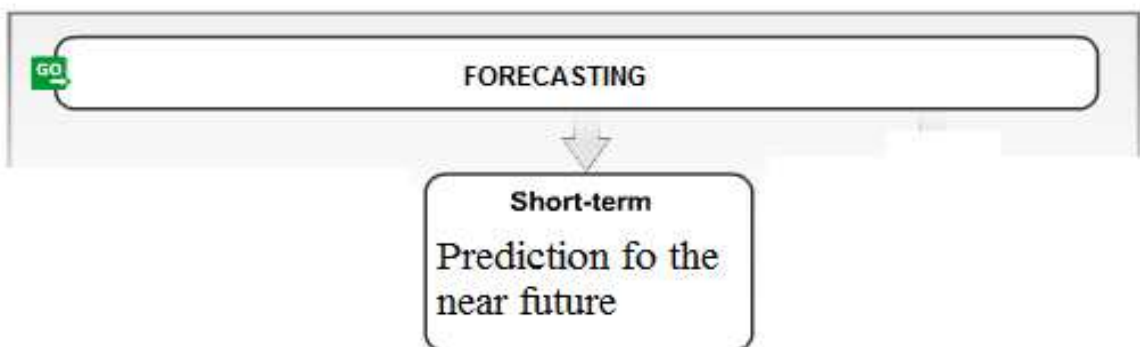
Step1: Create your portfolio



Step2: Regression Model



Step3: Predict company stock price trends, turning points, movement direction and BUY-HOLD-SELL signals





Step 4: Testing

Test the historical record of company prediction's accuracy by comparing forecasted quotes with actual trading stock quotes data base, years up to now.



TESTING and COMPARISON

Figure 6: Four steps strategy for developing stock forecaster.

<http://www.stock-forecasting.com/Content/Demo/CompanyFinder.aspx#t1>

Author: Manasi Jagannatha and Soumya Sampath

10. Use case Diagrams:

10.1 Use Case 1: Get current stock quotes to database.

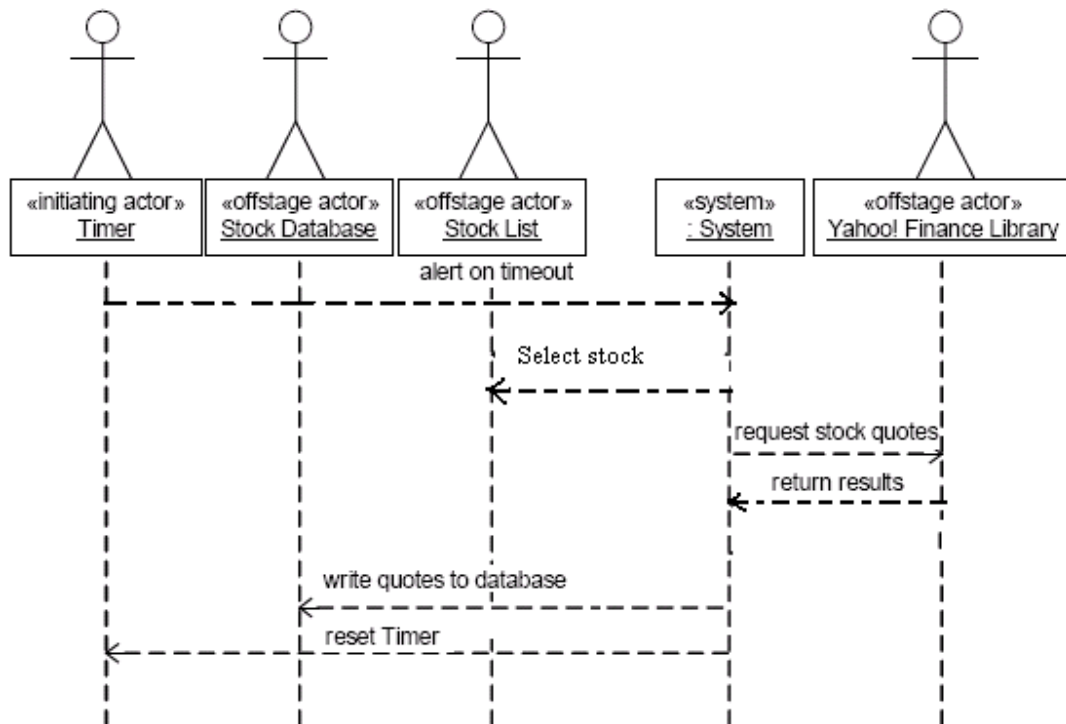


Figure 7: Use case diagram for getting current stock quotes.

- **Initiating Actor:** The initiating actor is the Timer, which is set in UC2 to initiate UC1 on regular basis.
- **Actor's Goal:** To gather stock quote information from online and store the results in Stock Database on the local machine.
- **Participating actors:** Stock List, Stock Database, Yahoo! Finance Library
- **Preconditions:** Timer Interval has been set, and the Stock List has been defined. The Stock Database exists and implements the expected schema.
- **Trigger:** The Timer times out.
- **Post conditions:** The Stock Database is updated. The Timer is reset. The System Menu is shown.
- **Main Success Scenario:**
 - 1. Timer initiates Stock Database update.
 - ← 2. System reads Stock List.
 - ← 3. System calls Yahoo! Finance Library.
 - ← 4. System copies the information returned by the Yahoo! Finance Library into the Stock Database.
 - ← 5. System resets Timer.

← 6. System returns to System Menu.

Authors: Shruthi Kiran and Tripti Singh

10.2 Use Case 2: Set Timer Interval

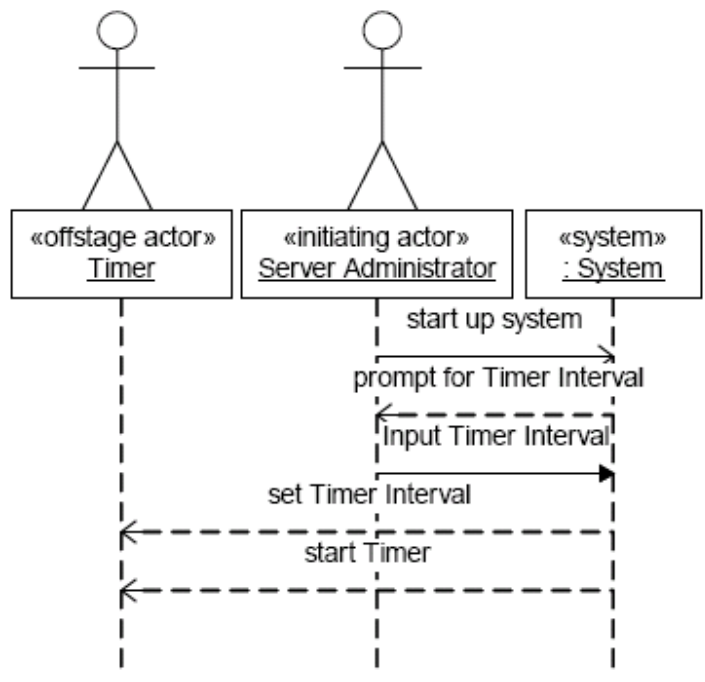


Figure 8: Use case diagram for setting timer interval

- **Initiating Actor:** Server Administrator
- **Actor's Goal:** Get the system to regularly update the Stock Database
- **Participating Actors:** Timer
- **Preconditions:** None.
- **Trigger:** The Server Administrator is started.
- **Post conditions:** The Timer Interval has been set and the Timer has started counting down to the next Database update. The System Menu is shown.

Authors: Shruthi Kiran and Tripti Singh

10.3 Use Case 3: Display stock quotes at the client.

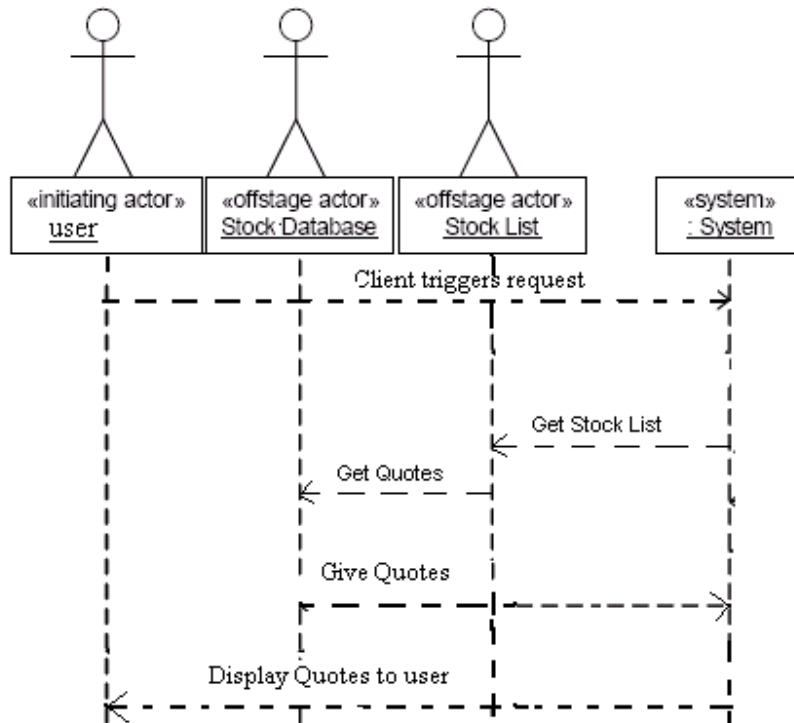


Figure 9: Use case diagram for displaying stock quotes at the client.

- **Initiating Actor:** The initiating actor is the User, the Web Client.
- **Actor's Goal:** To gather stock quote information from stock database and display the results at the client end on the local machine.
- **Participating actors:** Client, Stock List, Stock Database.
- **Preconditions:** Stock List has been defined. The Stock Database exists and implements the expected schema.
- **Trigger:** The Client requests stock quotes.
- **Post conditions:** The Stock quote is displayed on the client's browser window. The System Menu is shown.
- **Main Success Scenario:**
 - 1. Client initiates Stock quotes request.
 - ← 2. System reads Stock List.
 - ← 3. System calls Stock Database.
 - ← 4. The System gets the Stock quotes.
 - ← 5. The Stock quotes are displayed for the client.
 - ← 6. System returns to System Menu

Authors: Shruthi Kiran and Tripti Singh

10.4 Use Case 4: Generate predicted stock values and store it in Stock Database.

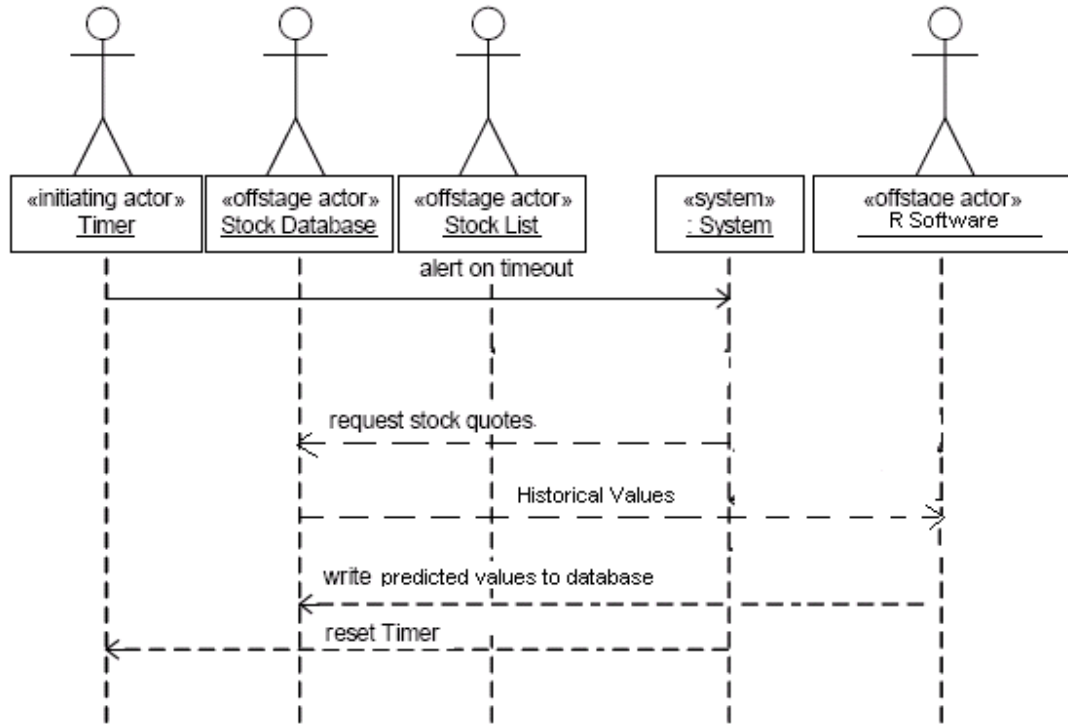
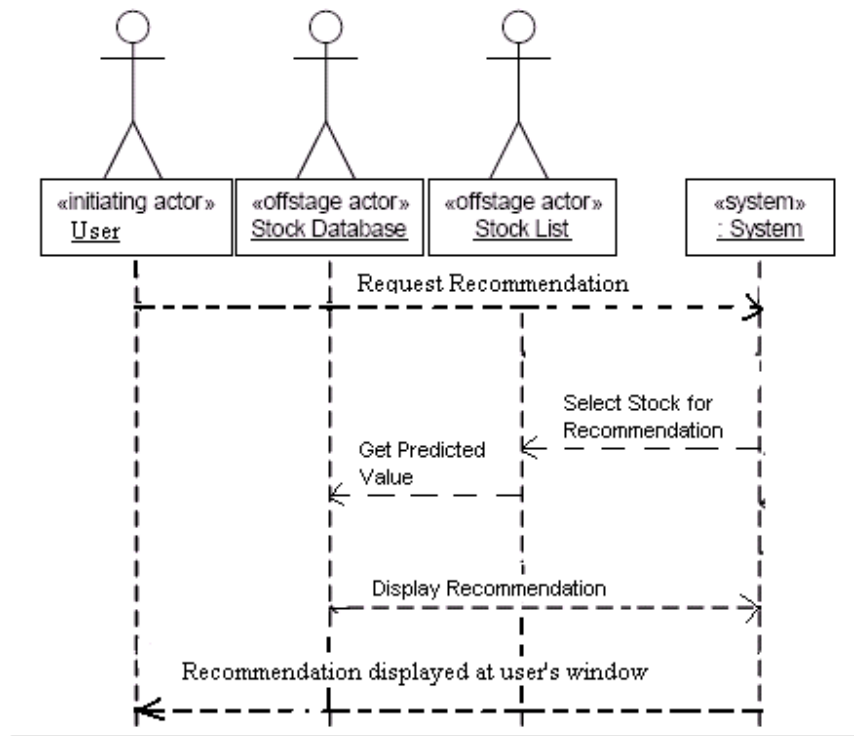


Figure 10: Use case diagram generating stock values and storing in database

- **Initiating Actor:** Timer, which is set for one day.
- **Actor's Goal:** To generate predicted stock quotes and store it in the Stock Database.
- **Participating actors:** Stock List, Stock Database, R .
- **Preconditions:** The Stock List has been defined. The Stock Database exists, implements the expected schema and R exists.
- **Trigger:** At the end of the day, the server administrator is started.
- **Post conditions:** The predicted values are stored in the Predicted values table. The Timer is reset.
- **Main Success Scenario:**
 - 1. The timer triggers the system administrator at the end of the day.
 - ← 2. System calls Stock Database.
 - ← 3. Stock quotes are given to R
 - ← 4. The R generates the Predicted values
 - ← 5. Predicted values are written to the Stock Database
 - ← 6. System resets Timer.

Authors: Shruthi Kiran, Manasi Jagannatha and Soumya Sampath.

10.5 Use Case 5: Generate Recommendation.



- **Initiating Actor:** The initiating actor is the Client, requesting for recommendation.
- **Actor's Goal:** To get recommendation for the selected stock and display on local machine.
- **Participating actors:** User, Stock List, Stock Database and Predicted Values table.
- **Preconditions:** Stock List has been defined. The Stock Database exists and implements the expected schema. Predicted values have been generated by R and stored in Predicted values table.
- **Post conditions:** Recommendations are generated for the requested stock. The recommendations are displayed at client's local machine. The System Menu is shown.
- **Main Success Scenario:**
 - 1. User initiates recommendation request by selecting the stock.
 - ← 2. System reads Stock List.
 - ← 3. System calls for Recommendation
 - ← 4. System copies the information returned by R, stored in the Prediction values table.
 - ← 5. Recommendation is displayed at the client's local machine.
 - ← 6. System returns to System Menu.

Authors: Manasi Jagannatha, Tripti Singh and Soumya Sampath.

Reference: http://www.omg.org/gettingstarted/what_is_uml.htm

11. HTTP Class Diagram

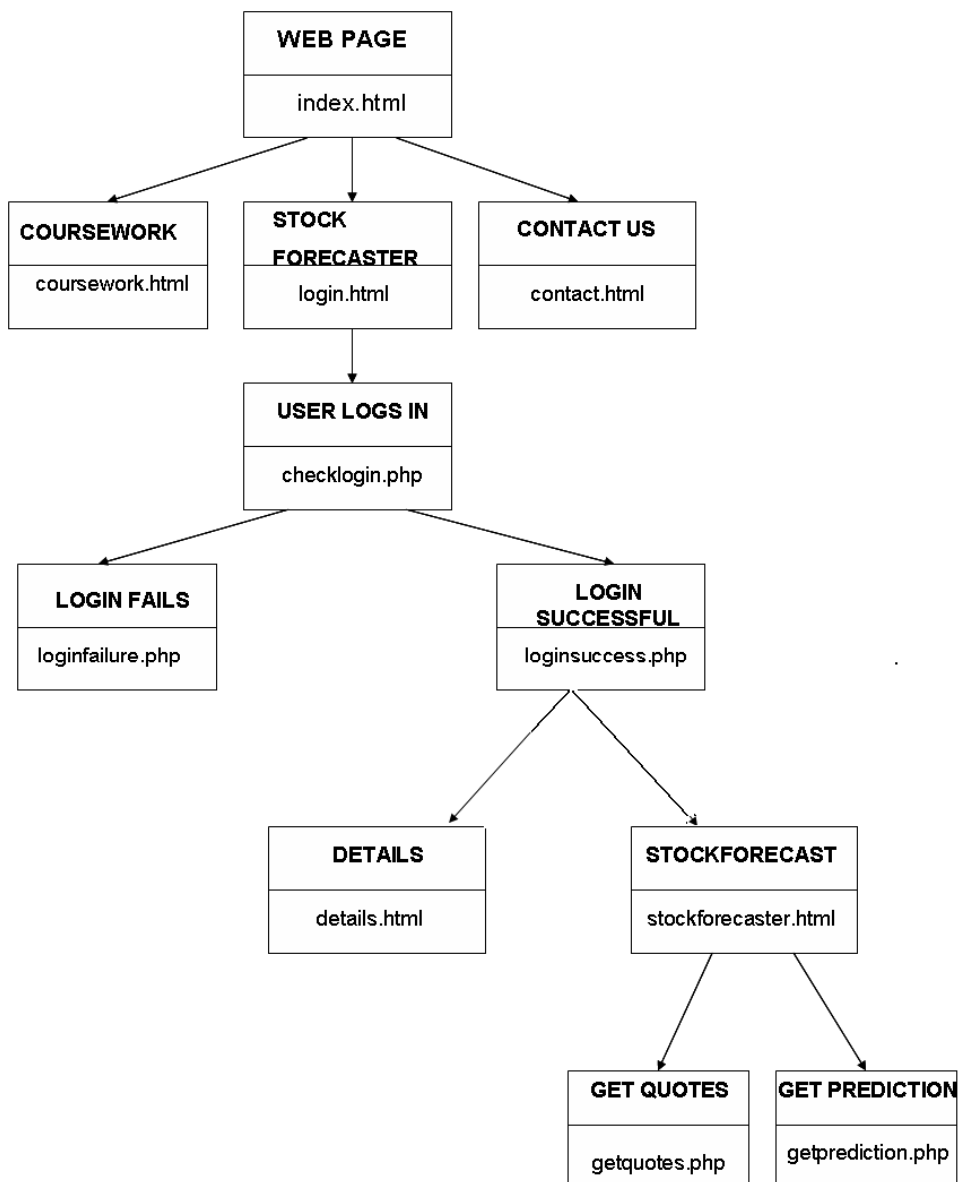


Figure 12: HTTP Class diagram

Authors: Soumya Sampath, Shruthi Kiran and Tripti Singh

12. Mathematical Models

I used to think that if there was reincarnation, I wanted to come back as the President or the Pope. But now I want to be the bond market: you can intimidate anyone.

- Political consultant James Carville.

Financial market efficiency is an important topic in the world of Finance. While most financiers believe the markets are neither 100% efficient, nor 100% inefficient, many disagree where on the efficiency line the world's markets fall.

However, market efficiency - championed in the efficient market hypothesis (EMH), formulated by Eugene Fama in 1970 - suggests that, at any given time, prices fully reflect all available information on a particular stock and/or market. Thus, according to the EMH, no investor has an advantage in predicting a return on a stock price since no one has access to information not already available to everyone else. (<http://www.investopedia.com/articles/02/101502.asp>)

If the above hypothesis were unarguably true then the world of stocks would no more be a mystery that it is today .In the age of information technology (IT), however, markets all over the world are gaining greater efficiency. IT allows for a more effective, faster means to disseminate information, and electronic trading allows for prices to adjust more quickly to news entering the market. However, while the pace at which we receive information and make transactions quickens, IT also restricts the time it takes to verify the information used to make a trade. Thus, IT may inadvertently result in less efficiency if the quality of the information we use no longer allows us to make profit-generating decisions.

12.1 Regression

Regression analysis is a technique used for the modeling and analysis of numerical data consisting of values of a dependent variable (response variable) and of one or more independent variables (explanatory variables). The dependent variable in the *regression equation* is modeled as a function of the independent variables, corresponding parameters ("constants"), and an *error term*. The error term is treated as a *random variable*. It represents unexplained variation in the dependent variable. The parameters are estimated so as to give a "best fit" of the data. Most commonly the best fit is evaluated by using the least squares method, but other criteria have also been used. (http://en.wikipedia.org/wiki/Regression_analysis)

Regression models could be used to predict the value of one variable from another or more variables whose values are pre-determined .The first stage of the process is to identify the variable that we want to predict(the dependent variable) and then carry out multiple regression analysis focusing on the variables that we want to use as predictors(explanatory /independent variables).The multiple regression analysis would

then identify the (causal)relationship between the dependent variable and the independent variables. This is then finally presented as a mathematical model.

Stepwise regression includes models in which the choice of the predictors is carried out by a sequential automatic procedure

The main approaches are:

a) Forward selection, which involves starting with no variables in the model, trying out the variables one by one and including them if they are 'statistically significant'.

b) Backward elimination, which involves starting with all candidate variables and testing them one by one for statistical significance, deleting any that are not significant.

A sequence of statistical significance is often used to control the inclusion or exclusion of the predictor variables. Although the p –values associated with these tests might be difficult to readily interpret, since each is conditional on the previous tests of inclusion and exclusion, the methodology readily lends itself to yielding a parsimonious representation of the data.

Stepwise forward regression

One of the goals of this project is to develop a predictive model that yields estimates of the Closing Price of an equity asset class within the imminent future. We give below the regression model used.

$$P = \beta_0 + \beta_1 * AD + \beta_2 * MFI + \beta_3 * Qstick + \beta_4 * S\&P\ 500 + Error$$

Where ,

P = Predicted closing price

AD, MFI, Qstick are standard technical indicators as defined below

Accumulation Distribution (AD): Accumulation/Distribution Technical Indicator is determined by the changes in price and volume. The volume acts as a weighting coefficient at the change of price — the higher the coefficient (the volume) is, the greater the contribution of the price change (for this period of time) will be in the value of the indicator.(http://stockcharts.com/school/doku.php?id=chart_school:technical_indicators:accumulation_distribution_line)

Market Facilitation Index (MFI): Market Facilitation Index Technical Indicator (BW MFI) is the indicator which shows the change of price for one tick. Absolute values of the indicator do not mean anything as they are, only indicator changes have sense. (http://www.metaquotes.net/techanalysis/indicators/market_facilitation_index)

Qstick: A technical indicator developed to numerically identify trends in candlestick charting. It is calculated by taking an 'n' period moving average of the difference between the open and closing prices. A Qstick value greater than zero means that the majority of

the last 'n' days have been up, indicating that buying pressure has been increasing. (<http://www.paritech.com/paritech-site/education/technical/indicators/trend/qstick.asp>)

The forward stepwise regression model was run for each of the 60 firms to yield 60 unique regression models. For each of these, indicators are added in sequentially until a parsimonious model is selected. The forward stepwise model retained only MFI and S&P 500 as significant predictors. As such, only MFI and S&P500 were the predictors used in all the models. The new model is as given below:

$$P = \text{beta0} + \text{beta1} * \text{MFI} + \text{beta2} * \text{S\&P}$$

Results:

As described above, the forward stepwise regression model was run for each of the 60 firms and a snapshot of the results is as shown below:

(NOTE: Please visit <http://software-ece.rutgers.edu/~group802>)

Under the assumption that the recent past would provide a legitimate glimpse into the near future, most recent day by day trading data could be utilized to obtain a prediction. As such, for each of the 60 firms, Closing Price data, for the 66 most recent trading days was downloaded from Yahoo Finance. Subsequently, the data was fitted to the regression model.

We provide the following statistics as given by the regression model for each firm:

- Regression coefficients(beta0,beta1,beta2,beta3)
- *t*-statistic corresponding to each regression coefficient as well as the *p*-values for each *t*-test. It may be noted that, the *t*-test is a test of statistical significance that indicates the efficacy of the independent variables in predicting the Closing Price. To this end, the *p*-value provides a statistical measure of significance of the respective predictors in the model. A small *p*-value is indicative of a relatively more statistically significant variable.
- Predicted Closing Price for day 'd+1', given data until day 'd'. This is the predicted Closing Price of the corresponding stock, one day into the future.
- 95% confidence interval for the aforementioned predicted Closing Price. It may be noted that the 95% confidence interval could be loosely interpreted such that the predicted Closing Price would fall out of the interval on only 5 trading days out of every 100 such trading days.
- To corroborate results from the prediction model, for each firm the data was divided into two sets- a training sample(59 trading days) and a test sample(7 trading days). The training sample was fit to the regression model thereby providing estimates of the regression parameters (betas). Subsequently, for each firm the regression model was used to predict the Closing Prices, for the remaining 7 days. The predicted prices could then be compared to the actual

Closing Prices, in the test sample, to obtain an idea of the accuracy of the regression model.

As the results in the Figure 13, indicate, the stepwise forward regression model has performed commendably in predicting the Closing Price. Although, a variety of different, potentially more complex statistical models could be used for the same, the model described in this report yields an optimal blend of simplicity as well as performance.

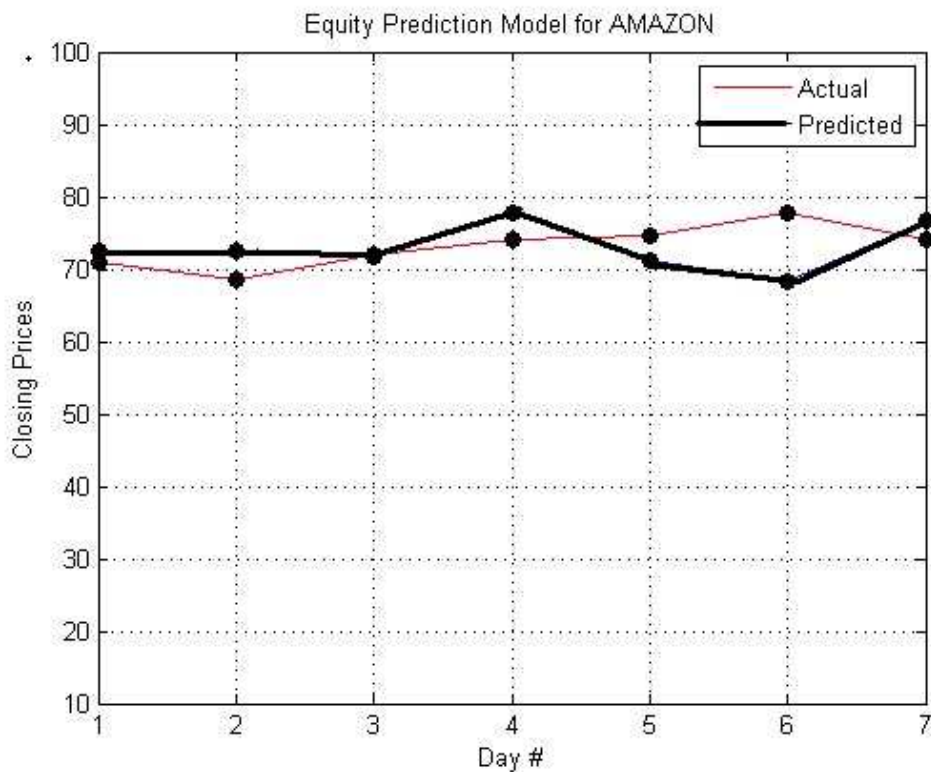


Figure 13: Equity Prediction model

12.2 Generating the Regression Model

MFI

The Market Facilitation Index (MFI) is one indicator that synthesizes both price and volume analysis. The MFI is the ratio of the current bar's range (high-low) to the bar's volume. The MFI is designed to gauge the efficiency of price movement. The efficiency is measured by comparing the current bar's MFI value to the previous bar's MFI value. If the MFI increased, then the market is facilitating trade and is more efficient, implying

that the market is trending. If the MFI decreased, then the market is becoming less efficient, which may indicate a trading range is developing that may be a trend reversal.

To calculate Market Facilitation Index you need to subtract the lowest bar price from the highest bar price and divide it by the volume.

$$\text{MFI} = (\text{HIGH}-\text{LOW})/\text{VOLUME}$$

(<http://www.forexrealm.com/technical-analysis/technical-indicators/market-facilitation-index.html>)

S&P 500

Widely regarded as the best single gauge of the U.S. equities market, this world-renowned index includes 500 leading companies in leading industries of the U.S. economy. S&P 500 is a core component of the U.S. indices that could be used as building blocks for portfolio construction. It is also the U.S. component of S&P Global 1200.

With more than US\$ 1.53 trillion in indexed assets, the S&P U.S. indices have earned a reputation for being not only leading market indicators, but also investable portfolios designed for cost efficient replication or the creation of index-linked products. The history of the S&P 500 dates back to 1923, with an expansion to include 500 companies in 1957.

All of the stocks in the index are those of large publicly held companies and trade on the two largest US stock markets, the New York Stock Exchange and Nasdaq. After the Dow Jones Industrial Average, the S&P 500 is the most widely watched index of large-cap US stocks. It is considered to be a bellwether for the US economy and is a component of the Index of Leading Indicators. It is often quoted using the symbol SPX or INX, and may be prefixed with a caret (^) or with a dollar sign (\$).

In stock and mutual fund performance charts, the S&P 500 index is often used as a baseline for comparison. The chart will show the S&P 500 index, with the performance of the target stock or fund overlaid.

In statistics, a **confidence interval (CI)** is an interval estimate of a population parameter. Instead of estimating the parameter by a single value, an interval likely to include the parameter is given. Thus, confidence intervals are used to indicate the reliability of an estimate. How likely the interval is to contain the parameter is determined by the **confidence level** or confidence coefficient. It provides critical values corresponding to 2.5% area to the left and 2.5% area to the right of the point estimate, that is, the predicted closing price estimate. Increasing the desired confidence level will widen the confidence interval.

http://www2.standardandpoors.com/portal/site/sp/en/us/page.topic/indices_500/2,3,2,2,0,0,0,0,0,0,0,0,0,0,0.html

12.3 Algorithm for regression model

- 1) The indicators MFI and S&P 500 are computed using the raw data retrieved from the stock quotes.
- 2) The values are tabulated for a number of readings.
- 3) The Returns are computed for the corresponding values using the equation

$$(P_{t+1} - P_t) / P_t \quad (1)$$

Where, P_t - The principal amount invested at time t and P_{t+1} - The amount at the end of t+1

- 4) We obtain the regression equation

$$R = \text{beta0} + \text{beta1} * \text{MFI} + \text{beta2} * \text{S\&P} \quad (2)$$

Where, beta0, beta1...beta6 are the regression coefficients which are to be computed.

13. System Screen shots

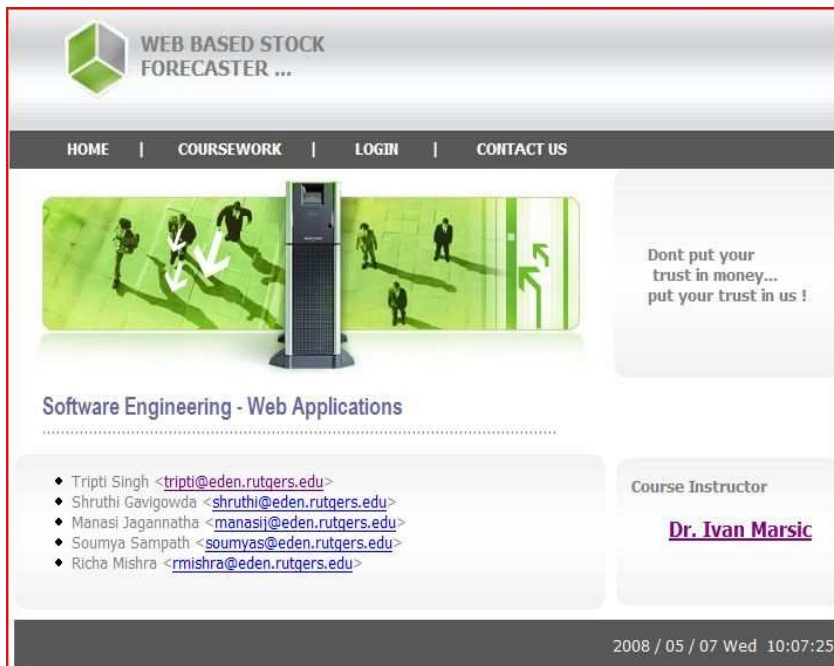


Figure 14: Screen shot of Web based Stock forecaster homepage.
Authors: Shruthi Kiran and Tripti Singh

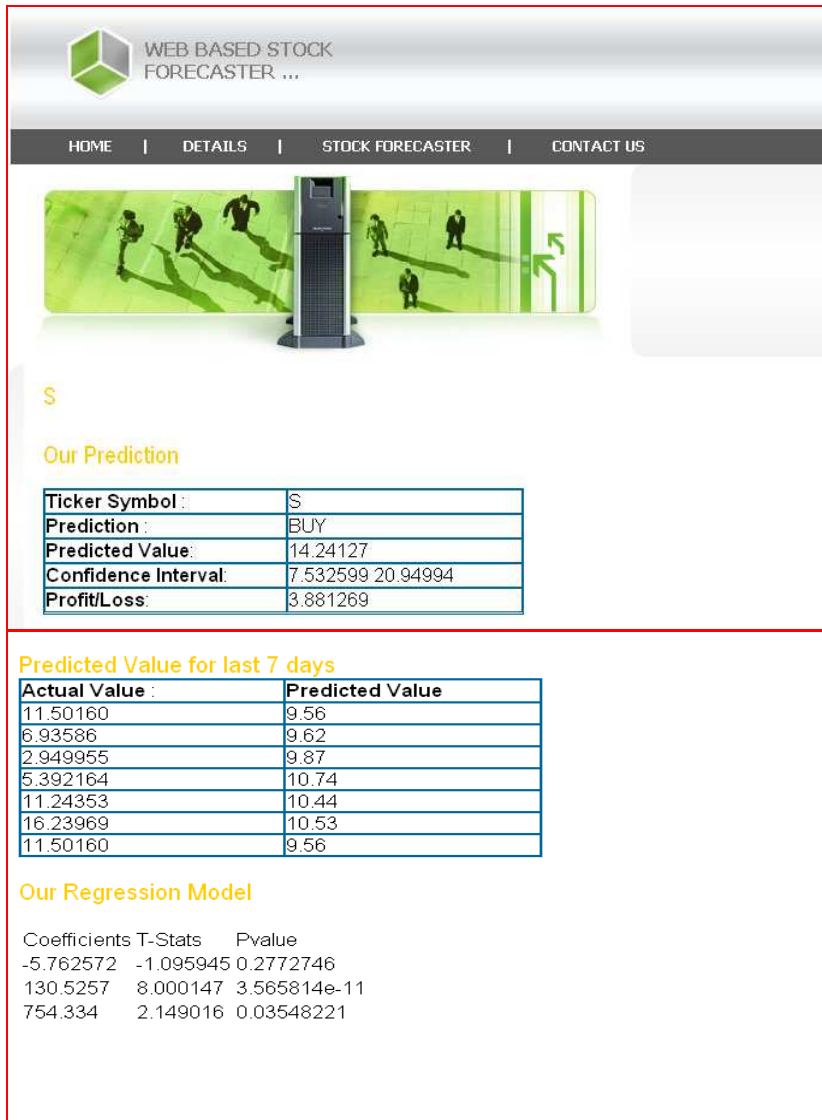


Figure 15: Screen shot for the recommendation given to the client
Author: Shruthi Kiran and Tripti Singh

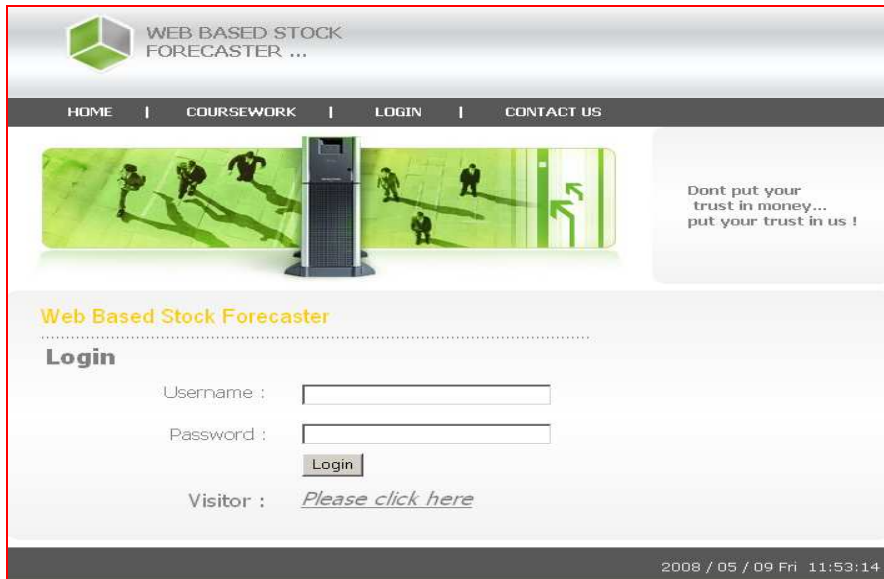


Figure 16: Screen Shot for user login
Authors: Shruthi Kiran and Tripti Singh



Figure 17: Screen shot for getting the stock quotes
Author: Tripti Singh

WEB BASED STOCK FORECASTER ...

HOME | DETAILS | STOCK FORECASTER | CONTACT US

Our Stock of the Day : DB

Our Prediction

Ticker Symbol :	DB
Prediction :	BUY
Predicted Value:	124.5523
Confidence Interval:	122.4015 126.7032
Profit/Loss:	9.992328

Dont put your trust in money...
put your trust in us !

Figure 18: Screen shot of Stock of the day
Author: Shruthi Kiran and Tripti Singh

14. Format of tables in the data base for getstock()

Tables in our database

Tables_in_group802
data
members
prediction
stocks

Table used for collecting data from yahoo finance

Field	Type	Null	Key	Default	Extra
ID	int(11)	YES		NULL	
stock	varchar(64)	YES		NULL	
value	varchar(16)	YES		NULL	
changepts	varchar(16)	YES		NULL	
open	varchar(16)	YES		NULL	
intra_top	varchar(16)	YES		NULL	
intra_down	varchar(16)	YES		NULL	
date	varchar(10)	YES		NULL	
time	varchar(6)	YES		NULL	
unixtime	int(12)	YES		NULL	
realloctime	int(12)	YES		NULL	
md5	varchar(32)	YES		NULL	

Table for checking login

Field	Type	Null	Key	Default	Extra
ID	int(11)		PRI	NULL	auto_increment
USERNAME	varchar(65)				
PASSWORD	varchar(65)				

3 rows in set (0.00 sec)

Table for storing final data from regression

Field	Type	Null	Key	Default	Extra
ticker	varchar(4)	YES		NULL	
prediction	varchar(4)	YES		NULL	
value	decimal(7,3)	YES		NULL	
confidence	decimal(5,2)	YES		NULL	
profit	decimal(7,2)	YES		NULL	

Author: Tripti Singh, Shruthi Kiran, Manasi Jagannatha and Soumya Sampath

The change points in the second table is the change in stock price, which is the day's opening price minus day's closing price. The profit in the last table is calculated per share, which is the difference between predicted price and actual price. If it is positive, then profit otherwise loss.

15. Web Service Implementation

We have used php and Apache server for the webservice interface. SOAP messages are exchanged between the client and server. The list of SOAP messages are given in section 15.2. The detailed description is given with the architecture diagram in Figure.3. The corresponding description is also available in the section 6.

15.1 List of files in Web Services

The lists of files are given below.

1. Public_html

This folder contains all the html pages of our website along with the images and the java script.

- a. **index.html** : This is the index page of our website.
- b. **coursework.html** : This page is kind of e-archive of our project and contains links to all the codes we need for the successful compilation of the project.
 - i. This page gives links to :
 - i) PHP code to get data from yahoo finance.
 - ii) Perl code to evaluate indicators for R software.
 - iii) PHP code to put data back into the database.
 - iv) R code to get our Regression Model and Predicted values.
 - v) Project Report.
 - vi) Gantt Project Report.
- c. **login.html**: This page authenticates the user and let it enter our stock forecaster only if he is a valid user.
- d. **checklogin.php**: This page is used to check the valid user name and password.
- e. **login_success.php**: If the login is successful, it directs you to the stockforecaster page.
- f. **stockforecaster.html**: On this page the user can make his/her choice forgetQuotes, getRecommendation or Stockoftheday.
- g. **page2.php**:If user selects Get Quotes, the website gets directed to this page. This page displays the current stock prices.
- h. **page3.php** : It displays our Recommendation of the stock selected by the user and with the predicted values for the last 7 days and regression coefficients.
- i. **page4.php**: This page displays the Stock of the Day.

2. regression_code.txt

This is the R code which we run on R software to get out Regression coefficients and Predicted Prices. It gives the results in the form of the text file, stats.txt.

3. parse.pl

This perl code gives calculated values of the indicators for all the 60 companies to be used by the R software. The output is stored in the form of the txt file, out.txt to be used by R.

4. pred.php

This php code is used to put the values calculated by the R to our database. That is it takes the stats.txt file and puts it into 3 different tables in the database.

5. stockfetch.php

This php code is used for getting the real time stock values from yahoo finance and puts it into our database.

6. shell.txt

It contains the lines of code to be typed at the command line to get our stockfetch.php code to retrieve data after every 15 min.

7. ticker symbols.txt

It contains the list of 60 companies we used for Prediction.

8. databasetables.txt

It gives the list of tables in our database.

15.2 Soap Messages

XML schema

```
<?xml version="1.0" encoding="UTF-8"?><!-- Published by JAX-WS RI at http://jax-ws.dev.java.net. RI's
version is JAX-WS RI 2.1.2_01-hudson-189-. --><!-- Generated by JAX-WS RI at http://jax-ws.dev.java.net.
RI's version is JAX-WS RI 2.1.2_01-hudson-189-. --><definitions
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://stock/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://schemas.xmlsoap.org/wsdl/"
targetNamespace="http://stock/" name="StockWebServiceService">

<types>
<xsd:schema>
<xsd:import namespace="http://stock/"
schemalocation="http://localhost:8082/StockForecaster/StockWebServiceService?xsd=1"></xsd:import>
</xsd:schema>
</types>
```

Soap messages for recommendation, getPriceForecaster and StockoftheDay

```
<message name="getPriceForecaster">
<part name="parameters" element="tns:getPriceForecaster"></part>
</message>
<message name="getPriceForecasterResponse">
<part name="parameters" element="tns:getPriceForecasterResponse"></part>
</message>
<message name="getRecommendation">
<part name="parameters" element="tns:getRecommendation"></part>
</message>
<message name="getRecommendationResponse">
<part name="parameters" element="tns:getRecommendationResponse"></part>
</message>
<message name="StockoftheDay">
<part name="parameters" element="tns:StockoftheDay"></part>
</message>
<message name="StockoftheDayResponse">
<part name="parameters" element="tns:StockoftheDayResponse"></part>
</message>
<message name="GetQuotes">
<part name="parameters" element="tns:GetQuotes"></part>
</message>
<message name="GetQuotesResponse">
<part name="parameters" element="tns:GetQuotesResponse"></part>
</message>
```

```

<portType name="StockWebService">
  <operation name="getPriceForecaster">
    <input message="tns:getPriceForecaster"></input>
    <output message="tns:getPriceForecasterResponse"></output>
  </operation>
  <operation name="getRecommendation">
    <input message="tns:getRecommendation"></input>
    <output message="tns:getRecommendationResponse"></output>
  </operation>
  <operation name="StockoftheDay">
    <input message="tns:StockoftheDay"></input>
    <output message="tns:StockoftheDayResponse"></output>
  </operation>
  <operation name="GetQuotes">
    <input message="tns:GetQuotes"></input>
    <output message="tns:GetQuotesResponse"></output>
  </operation>
</portType>

```

SOAP Binding

```

<binding name="StockWebServicePortBinding" type="tns:StockWebService">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"></soap:binding>
  <operation name="getPriceForecaster">
    <soap:operation soapAction=""></soap:operation>
    <input>
      <soap:body use="literal"></soap:body>
    </input>
    <output>
      <soap:body use="literal"></soap:body>
    </output>
  </operation>

  <operation name="getRecommendation">
    <soap:operation soapAction=""></soap:operation>
    <input>
      <soap:body use="literal"></soap:body>
    </input>
    <output>
      <soap:body use="literal"></soap:body>
    </output>
  </operation>

```

```
<operation name="StockoftheDay">
  <soap:operation soapAction=""></soap:operation>
  <input>
    <soap:body use="literal"></soap:body>
  </input>
  <output>
    <soap:body use="literal"></soap:body>
  </output>
</operation>

<operation name="GetQuotes">
  <soap:operation soapAction=""></soap:operation>
  <input>
    <soap:body use="literal"></soap:body>
  </input>
  <output>
    <soap:body use="literal"></soap:body>
  </output>
</operation>
</binding>

<service name="StockWebServiceService">
  <port name="StockWebServicePort" binding="tns:StockWebServicePortBinding">
    <soap:address location="http://localhost:8082/StockForecaster/StockWebServiceService"></soap:address>
  </port>
</service>
</definitions>
```

16. Conclusion:

In developing Use Case 4 and 5, the first technical challenge we encountered was that the selection of technical indicators and the Regression model. The Regression model involves accumulation and processing of large amount of data which require 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 some difficulties trying to integrate 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 UML and strengthened our knowledge about web services and its interface. It has also taught us to work on a project as a team and to coordinate with team members.

17. Future work:

Some of the work we could have done, had we continued this project

1. Write a more precise code by including all Perl libraries.
2. Use more technical indicators like AD, QStick to improve the confidence level.
3. Use the CGI-BIN directory access to have better user interface
4. Provide ratings to the stocks based on the predicted values generated using our model.
5. Provide the investor the opportunity to view the current best stocks in the market. The best stocks would mean the ones with high gains as predicted by our prediction model.

18. References

[1] Technical Indicators

www.investopedia.com

[2] Regression Analysis

http://en.wikipedia.org/wiki/Regression_analysis

[3] Performance Evaluation

http://www2.standardandpoors.com/portal/site/sp/en/us/page.topic/indices_500/2,3,2,2,0,0,0,0,0,0,0,0,0,0,0,0.html

[4]UML Standards

http://www.omg.org/gettingstarted/what_is_uml.htm

[5] Regression Software

<http://www.r-project.org/>

[6] Lecture Notes by Prof. Ivan Marsic

http://www.caip.rutgers.edu/~marsic/books/SE/book-SE_marsic.pdf

[7] Stock Forecasting

<http://www.stock-forecasting.com/Content/Demo/CompanyFinder.aspx#t1>

[8] SOA and Web Services

<http://ieeexplore.ieee.org/Xplore/login.jsp?url=/iel5/35/4342808/04342822.pdf?arnumber=4342822> (Lund, K.; Eggen, A.; Hadzic, D.; Hafsoe, T.; Johnsen, F.T. Communications Magazine, IEEE)

19. Appendix

19.1 List of companies used along with their ticker symbols

"MSFT"	MICROSOFT CP
"MO"	Altria Group Inc.
"AAPL"	APPLE INC
"AIG"	American International Group, Inc.
"BSC"	BEAR STEARNS COS THE
"AMZN"	AMAZON.COM INC
"AMD"	ADV MICRO DEVICES
"T"	AT & T Inc.
"ADBE"	Adobe Systems Inc.
"ANF"	ABERCROMBIE & FITCH
"GS"	GOLDMAN SACHS GRP
"BARC.L"	BARCLAYS
"BKS"	Barnes & Noble, Inc.
"HPQ"	Hewlett-Packard Co.
"MER"	Merrill Lynch & Co., Inc.
"LMT"	Lockheed Martin Corporation
"HON"	Honeywell International Inc.
"TWX"	TIME WARNER INC
"GOOG"	GOOGLE
"GT"	GOODYEAR TIRE RUBBER
"NYT"	The New York Times Company
"TXN"	TEXAS INSTRUMENTS

"NATI"	National Instruments Corp.
"QCOM"	QUALCOMM Inc.
"AVR.F"	AVAYA
"LEH"	Lehman Brothers Holdings Inc.
"VZ"	VERIZON COMMUN
"VOD"	VODAFONE GRP PLC ADS
"S"	Sprint Nextel Corp.
"IBM"	INTL BUSINESS MACH
"YHOO"	Yahoo! Inc.
"DIS"	Walt Disney Co.
"WMT"	Wal-Mart Stores Inc.
"WAG"	Walgreen Co.
"SI"	Siemens AG
"BLK"	BlackRock, Inc.
"XLNX"	Xilinx Inc.
"ADI"	Analog Devices Inc.
"BAC"	Bank of America Corporation
"INTC"	INTEL CP
"WB"	Wachovia Corp.
"LHL.F"	LENOVO GROUP
"TOA.F"	TOSHIBA MACHINE CO
"DELL"	DELL INC
"TTM"	Tata Motors Ltd.
"INFY"	INFOSYS TECHN ADS

"TCS.NS"	TATA CONSULTANCY SERV LT
"CTSH"	Cognizant Technology Solutions Corp.
"ACN"	ACCENTURE LTD.
"CSCO"	CISCO SYS INC
"DFG"	DELPHI FIN GRP INC A
"JPM"	JPMorgan Chase & Co
"SNE"	Sony Corp.
"PHG"	Koninklijke Philips Electronics NV
"JNJ"	Johnson & Johnson
"SLB"	Schlumberger Limited
"C"	Citigroup Inc.
"DB"	Deutsche Bank AG
"^DJI"	DOW JONES INDUSTRIAL AVERAGE IN