ECE 16:332:568

Software Engineering of Web Applications

GROUP # 5

WEB-BASED
STOCK FORECASTERS

(Project Website: http://software-ece.rutgers.edu/~group805/)

Submitted by:

AMARINDER CHEEMA
ATEET VORA
CHETAN JAIN
PUNEET KATARIA
RONAK SHAH
SIDDHARTH WAGH

Submission date:

31 March, 2008

- **Work Distribution among group members:**

Name	Amarinder	Ateet	Chetan	Puneet	Ronak	Siddharth
Introduction						
Concept of Project	15%	15%	15%	20%	15%	20%
System Design						
Block Diagram	50%	10%	10%	10%	10%	10%
Use Case	70%			10%	10%	10%
Data Mining		80%		10%	10%	
Web Service		80%				20%
Website Design	10%		80%	5%	5%	
Implementation and Results						
Indicators	10%	10%	50%	10%	10%	10%
Trend Analysis	10%			10%	70%	10%
Matlab Implementation	5%	5%	10%	5%	5%	70%
Graphical Data Representation	5%	5%	5%	70%	10%	5%
Machine Learning and Neural Networks	15%	15%	15%	15%	20%	20%
Presentation 1						
Slide Preparation	20%	15%	20%	15%	15%	15%
Project Report 1						
Documentation	15%	15%	15%	20%	20%	15%
Plan of Work						
Gantt Chart Diagram	5%	5%	5%	40%	40%	5%
References						
References	15%	15%	15%	15%	15%	15%

Fig. 1: Breakdown of individual contributions using Gantt chart

Table of Contents:

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. The stock market is not an efficient market. Herding behavior is common among investors, all investors do not get all information at the same time and the time it takes to evaluate information before they act differs between investors. Many investors do not show rational behavior. Greed and fear are strong feelings and may result in panic sales and stock market bubbles. Hence, to regulate the stock market to obtain maximum profit or achieve a certain objective in general without falling prey to inconsistencies, predicting stock behavior is a pressing requirement.

Prediction methodologies fall into two broad categories. They are fundamental analysis and technical analysis. Fundamental analysis of a business involves analyzing its income statement, financial statements and health, its management and competitive advantages, and its competitors and markets. It is more subjective compared to technical analysis. Fundamental analysis maintains that markets may misprice a security in the short run but that the "correct" price will eventually be reached. Profits can be made by trading the mispriced security and then waiting for the market to recognize its "mistake" and reprise the security. On the other hand, Technical Analysis is an approach that uses information of past stock behavior in order to forecast future price movements. Within the technical analysis community there exist several schools with different techniques, but they all have in common that they use price and volume history. A basic thought is that it takes time before the market reacts upon new information and that pattern often occur in price behavior which makes forecasting possible.

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. It is part of a company's nature to rest in good or bad periods. In good periods, the probability that good news will be followed by more good news is larger than the normal. Because the market often reacts on every single event this may result in a chain effect - a positive trend for the stock.
3. 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.
4. 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.

Hence, we limit our focus to technical analysis which has time and again proved its supremacy over other methods.

There are many tools available to investors using technical analysis but none of them removes entirely the element of chance from investment decisions. Large trading organizations can employ sophisticated computer systems and armies of analysts. We, as students, attempt to employ a simple set of formidable techniques to achieve the same result for the benefit of small-time investors who cannot afford to hire experts or buy costly softwares to make their investment decisions.

1.1 Project Goals and Requirements:

Formulation of Project Goals

1. **Selection of target customers:** Our customers are small-time daily or weekly investors who trade on an individual basis and who do not have the time or resources to avail of commercial forecasting services or hire agents.

2. **Selection of information to be tracked:** We aim to collect and use at least 10 indicators or patterns for performing technical analysis and provide predictions. We also plan to dole out information like RSS feeds, current stock prices and more to facilitate better understanding of the decision made by an investor while buying or selling stocks.

3. **Data collection:** Our major data source is the Google Finance service which provides us with daily stock prices for an entire year. Further, we plan to gather other indicators like S&P 500, accumulation/distribution line, average directional index, commodity channel index for comparison and re-enforcement of our prediction.

4. **Charting and pattern recognition:** Based on the information in the database, we plan to display the stock prices to the end-user using charts. Further, we need to recognize specific patterns like head and shoulders, cup and handle etc. which hint at the trends of the stock.

5. **Machine learning:** The logic to recognize such patterns and make wise decisions based on statistical inference should be coded in the machine in the form of efficient algorithms. Various options like mathematical regression models, neural networks, etc. are available that can frame such algorithms.

6. **Implement Web services:** Finally, we construct various web services to use prediction models that track different stocks as queried by user and issue forecasts about price movement of a given stock. The client module should gather information from multiple web services and combine their answers into a single recommendation.

7. **Design web interface:** A user-friendly web interface needs to be created and hosted to aid users to maintain their profiles and get timely recommendations and tips about dealing with their stock options.

System requirements:

1. The system should collect data from web sources like Yahoo Finance, Google finance at specific time intervals (say every 15 minutes) and update its database from a set of about 50-100 companies.
   http://finance.google.com
   http://finance.yahoo.com

2. The system should maintain a host of web services working on different prediction models and assemble their predictions together to give a fair recommendation to the customer whether to buy/sell/hold stock. Various algorithms like Caterpillar SSA, mathematical regression, curve fitting etc. need to be evaluated and their performance must be compared.

3. The system should track various stock movements simultaneously that might not be owned currently by customers and send alerts if it notices a favorable tilt in the graph that hints at immediate purchase.

4. Various customer profiles should be maintained by the system and confidential customer data should be protected at all costs from prying softwares.

5. The system should regularly send emails or short messages to the customer informing him about the current trends and future scenarios. It could feed the user current news and conduct surveys to judge its performance.

6. If time permits, the system may provide in-detail risk analysis of user portfolios to generate Sharpe Ratio to evaluate their current and future risk standing in the market and take appropriate corrective measures.

- **Statistical comparison of current softwares in market:**

Softwares	Attributes					
	Charting	Scanner	Built-in indicators	Data feed	Online/ Download	Alerts
Optimal Trader	Yes	Yes	15	EOD, Delayed	D	No
Ninja Trader	Yes	Yes	100	EOD, RT, Delayed	D	Yes
Stocker	No	Yes	13	Delayed	D	No
TA- Lib	No	No	125	No	D	No
Meta Stock	Yes	Yes	200	EOD, RT, Delayed	D	Yes
"Timing the Market"	**Yes**	**Yes**	**20**	**EOD, Delayed**	**OL**	**Yes**

Fig. 2 Comparison of softwares

Terminology:

Charting: Drawing elaborate charts based on stock information stored in database.
Scanner: Scanning entire database to detect trends and make predictions.
Built-in indicators: Various indicators used for studying trends like Dow, S&P 500 etc.
Data feed: How the data is extracted from web sources
EOD: End of day
RT: Real-time
Delayed: After a specific time interval
Online/Download: Whether the software is available online or is to be downloaded.
Alerts: Email, web, SMS services.

2. System Description

2.1 Block Diagram

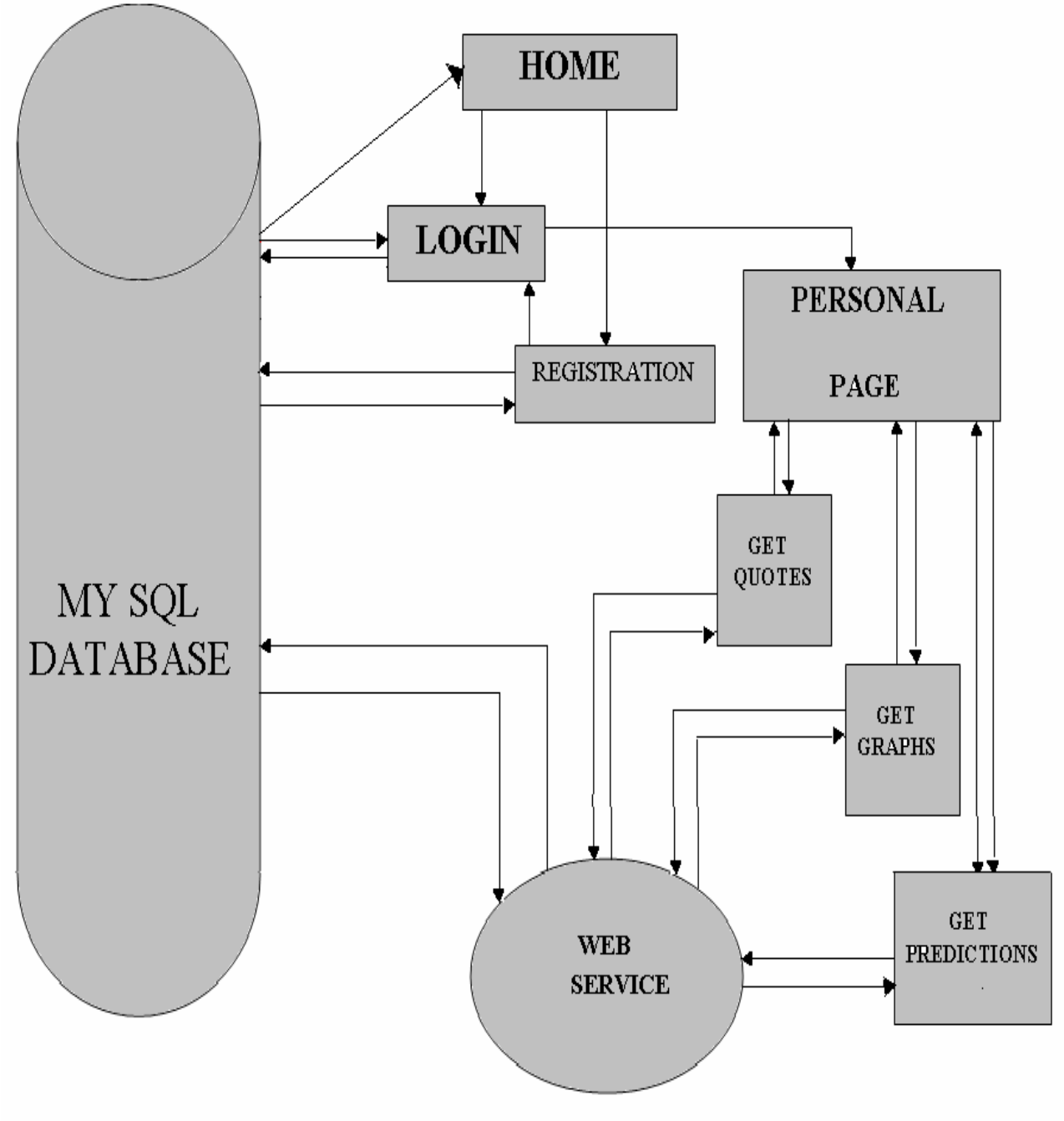

Fig. 3 System Block Diagram

2.2 Functional Specification Requirements

ACTOR	ACTOR GOALS	USE CASE
ADMINISTRATOR	To Login into the Admin account to view the application.	Use Case# 3
ADMINISTRATOR	Maintenance and Management of web service and the server	Use Case# 1
ADMINISTRATOR	To send/receive emails/opinions to/from the users	Use Case# 8
USER	To create a new account and login into his account	Use Case# 2
USER	To login into his account via the login page and see his account information	Use Case# 3
USER	To manage the account and check its details	Use Case# 4
USER	To get Quotes, Track Stocks, get Prediction	Use Case# 5, Use Case# 6,
GMAIL	To provide e-mail facility to the administrator and the user	Use Case# 8
DATABASE	To provide/store all necessary information regarding the stocks and the players	Use Case# 1, Use Case# 2, Use Case# 3, Use Case# 4, Use Case# 5, Use Case# 6, Use Case# 7, Use Case# 8

Fig. 4 Specification Diagram

2.3 Use cases

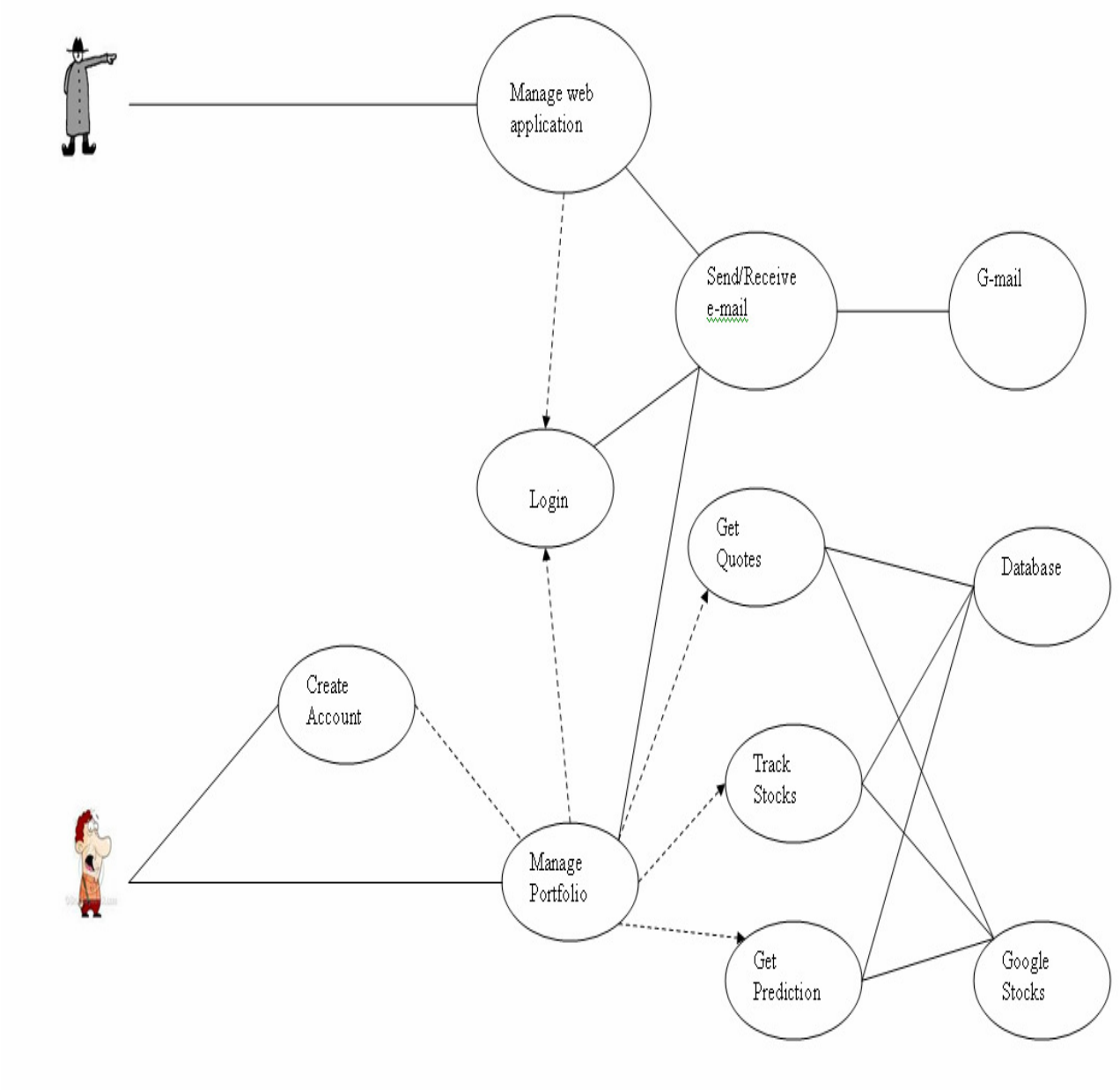

Fig. 5 Use Case Diagram

2.4 Use Case Description

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	Create Account	User	The user will be able to create a new account if he is a new user
3	Login	User/ Administrator	By providing the correct combination of user name and password any user or the administrator is allowed to login to his respective accounts
4	Account/Portfolio management	User	This will allow a user to manage his account, check details of his account and is allowed to perform several other functions
5	Get Quotes/Graphs	User	The User will be allowed to get quotes/graphs for selected stocks
6	Track Stocks	User	The user will be allowed to track selected stocks for a specified amount of time before actually buying that stock
7	Get Stock Prediction	User	The user will be allowed to get the prediction of selected stocks
8	Send/Receive e-mails	User/ Administrator	Both the user and the administrator will be allowed to send and receive e-mails whenever required.

Fig. 6 Use Case Description

USE CASE# 1 DESCRIPTION:

ACTOR'S GOALS: Maintenance and management of web application and server
PARTICIPATING ACTORS: None
PRE CONDITIONS: The administrator should be properly authenticated by the system after he provides the correct user name and password
POST CONDITIONS: Check the details of the application and make changes if required
SUCCESS SCENARIO:
1) Opens the Login Page
2) Fills the correct Username
3) Fills the correct Password
4) Checks the details and make modifications if required

USE CASE # 2 DESCRIPTION:

USE CASE# 2	Create Account
INITIATING ACTOR: User	
ACTOR'S GOALS: To create a new account	
PARTICIPATING ACTORS: Database	
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 SCENARIO:	
1) The user opens main page	
2) Selects new user option	
3) Fills in the registration form	
4) Select the create account option	

USE CASE # 3 DESCRIPTION:

USE CASE# 3 Login
INITIATING ACTOR: User/ Administrator
ACTOR'S GOALS: To login into the account
PARTICIPATING ACTORS: User / Administrator
PRE-CONDITIONS: The user/ administrator has not logged into his account
POST-CONDITIONS: The user/ administrator has login into his account by providing the correct username and password
SUCCESS SCENARIO:
1) The user/ administrator opens login page
2) Fills in the correct user name
3) Fills in the correct password
4) Select the Login option

USE CASE # 4 DESCRIPTION:

USE CASE# 4 Account/Portfolio Management
INITIATING ACTOR: User
ACTOR'S GOALS: To manage the account and check the details
PARTICIPATING ACTORS: Database
PRE-CONDITIONS: The user has not logged into his account
POST-CONDITIONS: The user after logging in can check the details of his portfolio
SUCCESS SCENARIO:
1) The user opens Login page
2) Fills in the correct user name and password
3) Check the details of the portfolio and make decisions accordingly

USE CASE # 5 DESCRIPTION:

USE CASE# 5 Get Quotes/Graphs
INITIATING ACTOR: User
ACTOR'S GOALS: To get the quotes and graphs of selected stocks
PARTICIPATING ACTORS: None
PRE-CONDITIONS: The user does not have any qoutes of a particular stock
POST-CONDITIONS: The user has the quotes of that selected stock
SUCCESS SCENARIO:
1) The user opens the Login page
2) Fills in the correct username and password
3) Select the Company for which he wants the quotes/ graphs
4) Select the get quoted option
5) Check the quotes and the graphs obtained

USE CASE # 7 DESCRIPTION:

USE CASE# 7 Get Stock Prediction
INITIATING ACTOR: User
ACTOR'S GOALS: To get the prediction of selected stocks
PARTICIPATING ACTORS: Database
PRE-CONDITIONS: The user does not have any prediction of a particular stock
POST-CONDITIONS: The user has the prediction of that selected stock and is provided with the decision to Buy, Sell or Hold the stock
SUCCESS SCENARIO:
1) The user opens the Login page
2) Fills in the correct username and password
3) Select the Company for which he wants the prediction
4) Select the get prediction option
5) Check the predictions obtained and make decisions accordingly

2.5 Data Mining

This section gives a brief description of the backend stock data retrieval for the Software Engineering project.

As per the specifications we have been able to develop a PHP script supported by a MySQL database to be able to automate data collection in our database. The script is set to execute every 15 minutes and update the stock market information existing in the database as well as inserting new data.

Database schema

The figures below give a snapshot view from MySQL of the tables currently existing in our database. We have also shown a snapshot view of the current data existing in our database.

We have currently added two tables namely

1. OrganizationTable

Columns:
OrganizationID (Primary Key)
OrganizationName (Name of the Organization)

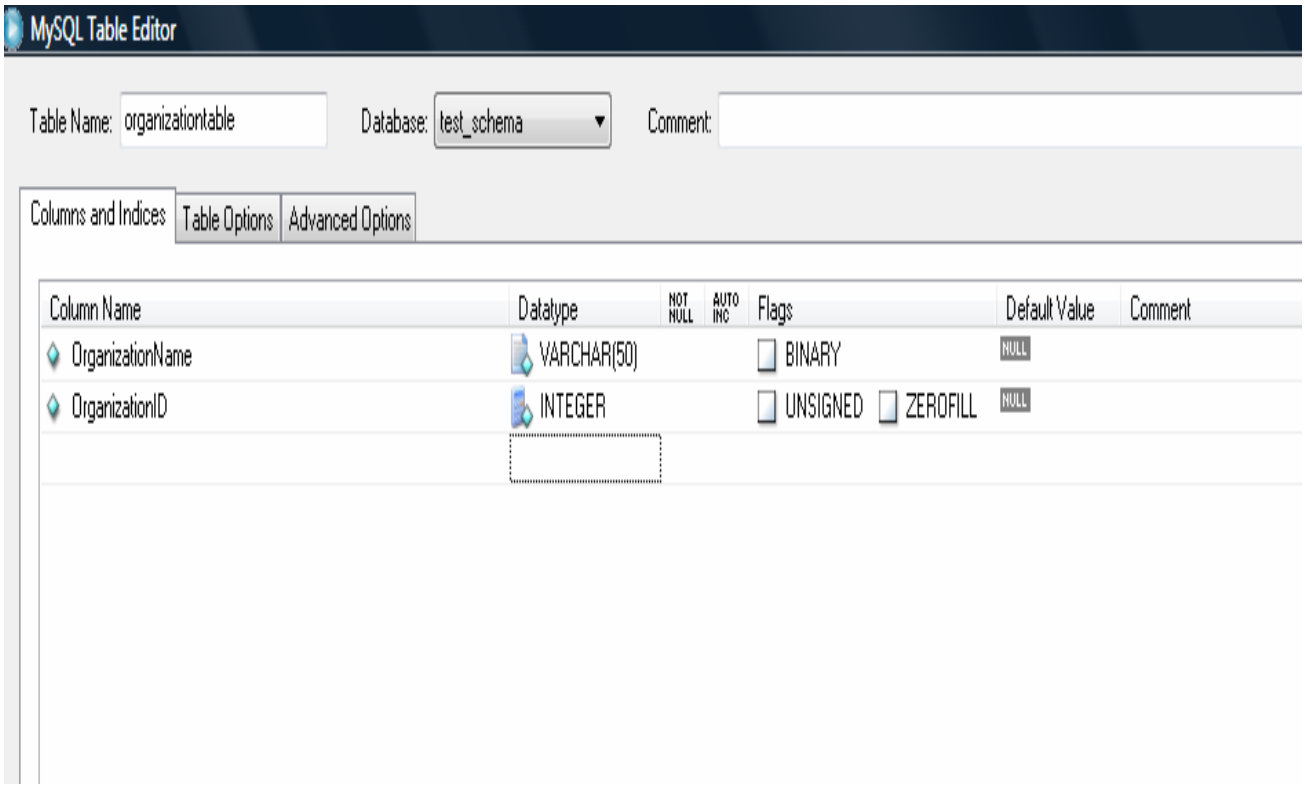

Fig. 7 Snapshot of Organization Table from MySQL database

2. OrganizationHistory

Columns:
OrganizationID (Foreign Key)
Date (displays the date)
Open (displays the opening price for the particular stock on a given day)
High (displays the highest price for the particular stock on a given day)
Low (displays the lowest price for the particular stock on a given day)
Close (displays the closing price for the particular stock on a given day)
Volume (displays the volume of shares traded for the particular stock on a given day)

These columns store information about the stock prices of the various companies which are listed in the OrganizationTable.

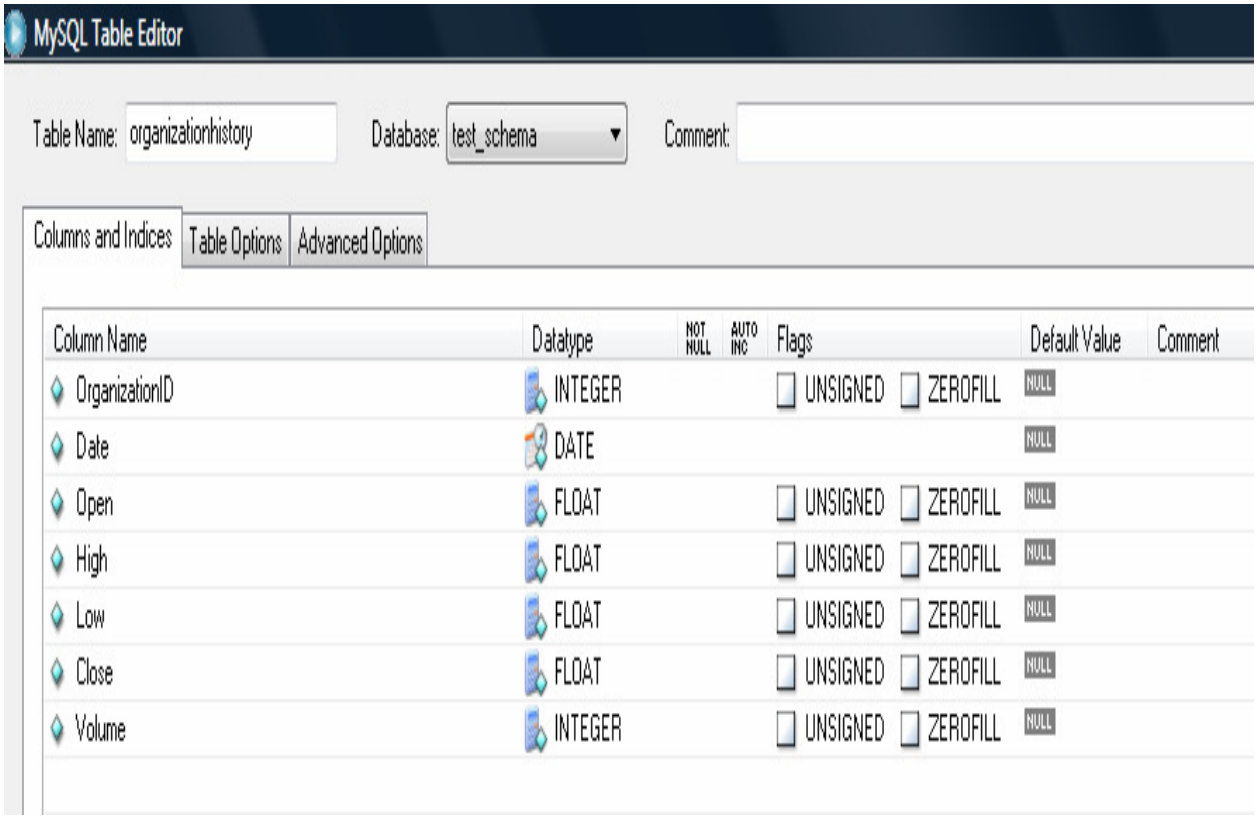

Fig. 8 Snapshot of OrganizationHistory Table from MySQL database

The script uses the stock data available on the Google Finance (http://finance.google.com/finance). We can easily store stock information of more than 2 organizations. And there would be no need to change the script submitted. What we have to do is just add the new organization name and organization ID to the OrganizationTable.

2.6 Web Services

Web Services remain the most integral and important feature which we will be implementing in our project. In our case we shall be using java based web services. The web services will be developed using Apache Axis.

Apache Axis is an open source product developed by Apache. It is a new SOAP engine which makes web service creation much easier. Axis works as a front end for our services. This means that Axis intercepts the SOAP request, desterilizes the XML to a Java Object and passes it to the service. This service is a pure Java Service. Axis exposes the simple java method as service. It makes the deployment of services very easy.

Currently we have been able to create the web service, which we have planned to use. It is ready for deployment. Though we only have the interface ready without any implementation inside. The interface looks somewhat like this:

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package axis.netbeans.ws.stockpredict;


import java.util.Date;

/**
 *
 * @author Ateet
 */
public interface StockPredictInterface {

   public int sPredict1(Long sPrice, Date sCurrentDate);

   public long sPredict2(Long sPrice, Date sCurrentDate, Long sVolume);

}
```

We have decided to develop a web service client server architecture, where the database access will be managed by the web service. Hence the client application will call a web service for all database related function like predicting stocks, real time data for plotting graphs and charts and also for acquiring real time stock quotes.

2.7 Overview of Our Website and its features:

We have used technology like PHP, HTML and CSS for the development of our website. On the homepage the user will find 20 min delayed quotes scrolling from right to left. Also he can see the index values and latest business news. The login will be secure. There will be a tab called the "Intro" where the user will find all information about the usage of the service and our approach towards the development of the project. There is also a Google search bar on our homepage so if user wants to search anything he can do it without moving from our web service. There is also a calculator if he requires it. The user will also get access to the home page of the team members and guide. User can search for the quotes through homepage which will display the price graph along with volumes, day-range, open and close price, days high and low. User can access his personal portfolio after signing in where he can get all features as on homepage and apart from that he can also get the buying or selling suggestion depending on our technical analysis. We have taken care that our website will open on all browser and platform as intended. More features will be included as time permit. Below is the snap shot of our website.

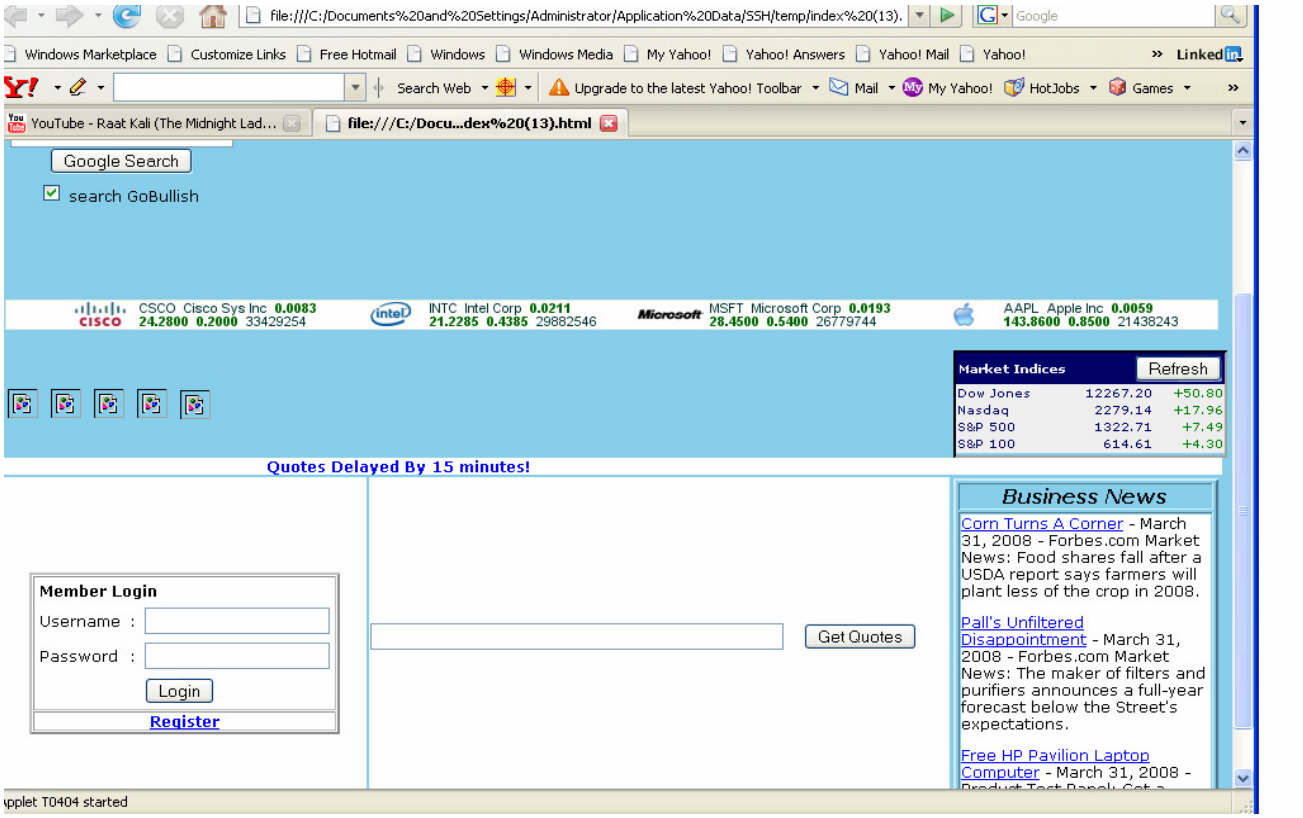

Fig. 9 Snapshot of the website

The URL of our website is ***http://software-ece.rutgers.edu/~group805***

3. Implementation and Results:

3.1 Indicators

Definition: Technical analysis indicators are the mathematical formulae that day traders on their charts to decide when to make their trades.

In our project we are targeting to implement two to three most reliable and often used indicators. Following is the list of the indicators.

Moving Average Convergence/Divergence (MACD)

MACD is one of the simplest and most reliable indicators available. MACD uses moving averages, which are lagging indicators, to include some trend-following characteristics. These lagging indicators are turned into a momentum oscillator by subtracting the longer moving average from the shorter moving average. The resulting plot forms a line that oscillates above and below zero, without any upper or lower limits.

MACD Formula: The most popular formula for the "standard" MACD is the difference between a security's 26-day and 12-day Exponential Moving Averages (EMAs). This is the formula that is used in many popular technical analysis programs. Of the two moving averages that make up MACD, the 12-day EMA is the faster and the 26-day EMA is the slower. Closing prices are used to form the moving averages. Usually, a 9-day EMA of MACD is plotted along side to act as a trigger line. A bullish crossover occurs when MACD moves above its 9-day EMA, and a bearish crossover occurs when MACD moves below its 9-day EMA.

Explanation: The Merrill Lynch (MER) chart below shows the 12-day EMA (thin blue line) with the 26-day EMA (thin red line) overlaid the price plot. MACD appears in the box below as the thick black line and its 9-day EMA is the thin blue line. The histogram represents the difference between MACD and its 9-day EMA. The histogram is positive when MACD is above its 9-day EMA and negative when MACD is below its 9-day EMA. MACD measures the difference between two Exponential Moving Averages (EMAs). A positive MACD indicates that the 12-day EMA is trading above the 26-day EMA. A negative MACD indicates that the 12-day EMA is trading below the 26-day EMA. If MACD is positive and rising, then the gap between the 12-day EMA and the 26-day EMA is widening. This indicates that the rate-of-change of the faster moving average is higher than the rate-of-change for the slower moving average. Positive momentum is increasing, indicating a bullish period for the price plot. If MACD is negative and declining further, then the negative gap between the faster moving average (blue) and the slower moving average (red) is expanding. Downward momentum is accelerating, indicating a bearish period of trading. MACD centerline crossovers occur when the faster moving average crosses the slower moving average.

Fig. 10 Graph showing Moving Average Convergence/Divergence Indicator

Even though moving averages are lagging indicators, notice that MACD moves faster than the moving averages. In this example, MACD provided a few good trading signals as well:
In March and April, MACD turned down ahead of both moving averages, and formed a negative divergence ahead of the price peak.
In May and June, MACD began to strengthen and make higher Lows while both moving averages continued to make lower Lows.
Finally, MACD formed a positive divergence in October while both moving averages recorded new Lows
MACD Benefits:

One of the primary benefits of MACD is that it incorporates aspects of both momentum and trend in one indicator. As a trend-following indicator, it will not be wrong for very long. The use of moving averages ensures that the indicator will eventually follow the movements of the underlying security. By using Exponential Moving Averages (EMAs), as opposed to Simple Moving Averages (SMAs), some of the lag has been taken out.

MACD can be applied to daily, weekly or monthly charts. MACD represents the convergence and divergence of two moving averages. The standard setting for MACD is the difference between the 12 and 26-period EMA. However, any combination of moving averages can be used. The set of moving averages used in MACD can be tailored for each individual security. For weekly charts, a faster set of moving averages may be appropriate. For volatile stocks, slower moving averages may be needed to help smooth the data. Given that level of flexibility, each individual should adjust the MACD to suit his or her own trading style, objectives and risk tolerance.

Accumulation/Distribution Line:

The basic premise behind volume indicators, including the Accumulation/Distribution Line, is that volume precedes price. Volume reflects the amount of shares traded in a particular stock, and is a direct reflection of the money flowing into and out of a stock. Many times before a stock advances, there will be period of increased volume just prior to the move. Most volume or money flow indicators are designed to identify early increases in positive or negative volume flow to gain an edge before the price moves.

Accumulation/Distribution Line Formula:

(((Close-low)-(High-Low))/(High-Low))*Volume

There are basically five combinations:
If the stock closes on the high, the top of the range, then the value would be plus one.
If the stock closes above the midpoint of the high-low range, but below the high, then the value would be between zero and one.
If the stock closes exactly halfway between the high and the low, then the value would be zero.
If the stock closes below the midpoint of the high-low range, but above the low, then the value would be negative.
If the stock closes on the low, the absolute bottom of the range, then the value would be minus one.

Explanation:

A bullish signal is given when the Accumulation/Distribution Line forms a positive divergence. Be wary of weak positive divergences that fail to make higher reaction highs or those that are relatively young. The main issue is to identify the general trend of the Accumulation/Distribution Line. A two-week positive divergence may be a bit suspect. However, a multi-month positive divergence deserves serious attention.
Bullish Signal Example

Fig. 11 Accumulation/Distribution Line Representation (Bullish Market)

On the chart for Alcoa, Inc. (AA), the Accumulation/Distribution Line formed a huge positive divergence that was over 4 months in the making. Even though the stock fell from above 35 to below 30, the Accumulation/Distribution Line continued on a relentless march north. If one did not know better, it would seem that the two plots did not belong together. However, the stock finally caught up with the Accumulation/Distribution Line when it broke resistance in November.

Bearish Signal Example

The Delta Air Lines (DAL) chart shows a negative divergence that developed within the confines of a clear downtrend. The stock had clearly broken down, and the Accumulation/Distribution Line was declining in line with the stock. A deteriorating Accumulation/Distribution Line confirmed weakness in the stock. During the June-July rally, the stock recorded a new reaction high, but the Accumulation/Distribution Line failed, thus setting up the negative divergence.

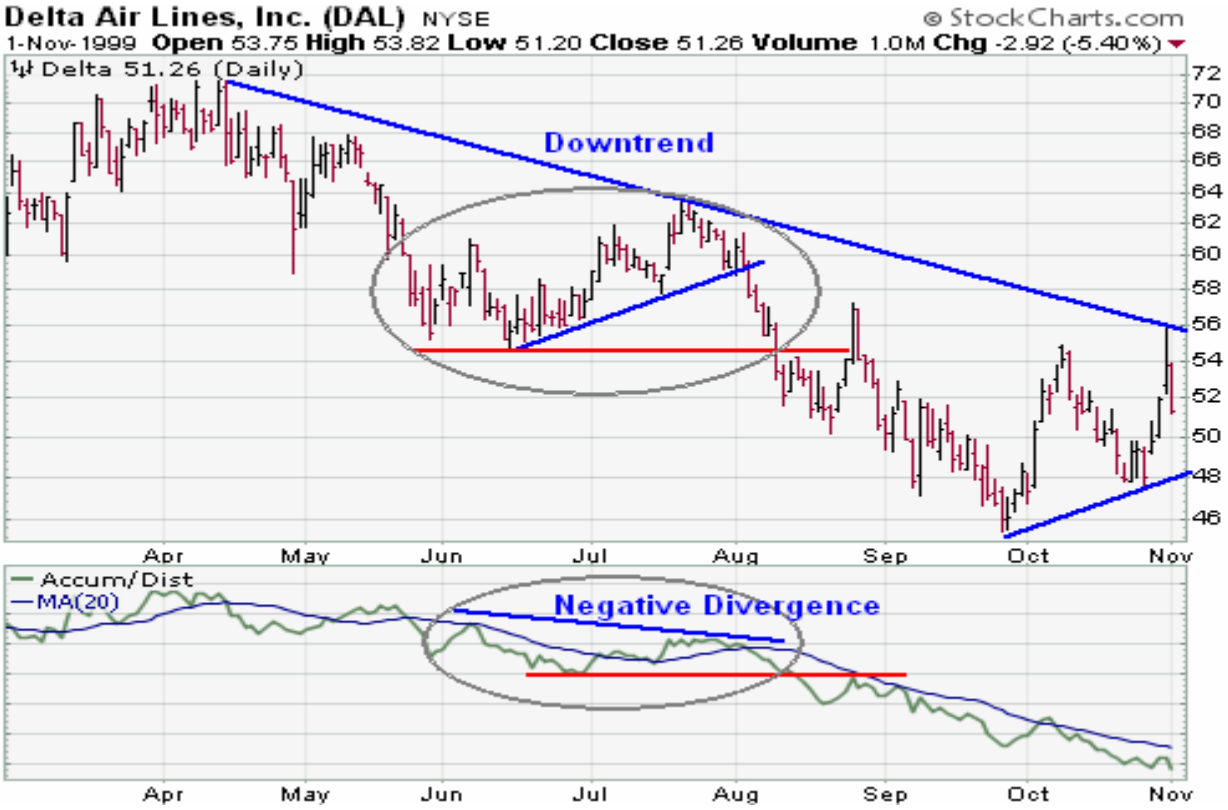

Fig. 12 Accumulation/Distribution Line Representation (Bearish Market)

How we will Implement this Indicators:

So far we were able to plot the graph using the prehistoric data. Now we are working on how to trigger the buying or selling light once the particular condition meet. We are learning Matlab Financial tool to achieve this.

3.2 Graphical Representation of Data Using JPGraphs

JPGraph is an Object Oriented graph library, which can help us to draw graphs. The real advantage in using JPGraph is its simplicity. Creating a graph at runtime may sound complex. But JPGraph class has been designed in such a way that they hide the complex part from the user. We provide a set of values from database to be represented on a graph and we can display the same to the user with few lines of coding.

Here are a few **features** of the software:

- Flexible scales, supports text-linear, text-log, log-linear and log-log and integer scales.
- Supports PNG, GIF and JPG graphic formats. Note that the available formats are dependent on the specific PHP installation where the GD library is used.
- Supports caching of generated graphs to lessen burden of a HTTP server.
- Intelligent autoscaling which gravitates towards esthetical values, i.e. multiples of 2s
- Supports background images with different formatting options
- Supports color and brightness adjustments of images directly in PHP.
- User specified grace for autoscaling
- Supports, line-plots, filled line-plots, accumulated line-plots, bar plots, accumulated bar plots, grouped bar plots, error plots, line error plots, scatter plots, gantt-charts, radar plots, 2D and 3D pie charts.

Requirements:

1. JP graph uses the GD library to create and display images. We can download the library file ‘jpgraph.php’ from ‘http://www.aditus.nu/jpgraph/jpdownload.php’.

2. We have to make sure that the PHP installation is at least 4.04 and that we have compiled support for GD library.

3. We must set up the directory paths in jpgraph.php where the cache directory should be where the TTF directory is. Apache/PHP must have write permission in the cache directory.

4. PHP must have write privileges to the cache directory if we plan on using the cache feature.

Implementation of JP graphs:

As a general rule each PHP script which generates an image must be specified in a separate file which is then called in an <IMG> tag reference. For example, the following HTML excerpt includes the image generated by the PHP script in "fig1.php".

<img src="fig1.php" border=0 align =center width=300 height= 200>

The library will automatically generate the necessary headers to be sent back to the browser to correctly recognize the data stream as an image of either PNG/GIF/JPEG format. The browser can then correctly decode the image. To include several images together with text on a page we

need to have a parent page with several <IMG> which each refers to an image script. When it comes to the structure of our imaging script they will generally have the structure.

```
// ... Include necessary headers
$graph  = new Graph($width, $height, ...);
// ... code to construct the graph details
$graph->Stroke();
```

JpGraph is completely Object oriented so all calls will be action on specific instances of classes. One of the fundamental classes is the Graph() class which represents the entire graph. After the creation of the Graph() object, we add all your lines of code to construct the details of the graph. As the final call, we will send the generated image back to the browser with a call to the Stroke() method.

In addition to this standard usage pattern, we can also send the graph directly to a file, get the GD image handler for the image and also make use of the built-in cache system. The cache system, which lessens the burden of the PHP server, works by avoiding to run all the code that follows the initial Graph() call by checking if the image has already been created and in that case directly send back the previously created (and filed) image to the browser. When using the cache system we must specify a filename which is used to store the image in the cache system and possibly also a timeout value to indicate how long the image in the cache directory should be valid.

By default JpGraph automatically chooses the image format to use in the order PNG, JPEG and GIF. The exact format depends on what is available on your system. There are two ways one can influence the way the graphic format is chosen.

i. Change the default graphic format by changing the DEFINE:
   DEFINE ("DEFAULT_GFORMAT" ,"auto");
ii. Set the graphic format in the script by calling the method SetImgFormat().
   //force to use JPEG image
   $graph ->img->SetImgFormat("jpeg")

If we want to save the image directly to a file instead of streaming it back to the browser then we just have to specify an absolute filename in the final call to Graph::Stroke(), i.e.

$graph ->Stroke("/usr/home/peter/images/result2002.png");

The enabling-disabling of the cache system is controlled by two defines (in jpgraph.php)

```
DEFINE ("USE_CACHE",true);
DEFINE("READ_CACHE", true);
```

The first of these, USE_CACHE, is the master-switch which must be set to true to enable the caching system. The second switch READ_CACHE very seldom needs to be changed. This second switch basically tells whether or not JpGraph should ever look in the cache. Setting this to false and the master-switch to true would then always generate a new updated image file in the cache and this new image would be sent back to the browser. The main use for this (admittedly)

strange setting is if one likes to have the side effect of the script that a fresh image is always stored in the cache directory. Once the cache is enabled, we must also make sure that a valid cache directory is setup. The cache directory is specified with:

DEFINE ("CACHE_DIR","/tmp/jpgraph_cache/");

We can of course change the default directory to whatever directory one fancies. But, the cache directory must be writable for the user running Apache/PHP. To use caching in the script, we must supply a suitable file name which will be used to store the image in the cache.

We can also supply a timeout value indicating how many minutes the cached image should be considered valid. These parameters are supplied in the initial Graph() method call which should be among the first in the script. Instead of manually specifying a file name to be used you could often use the special name "auto". If the filename is specified as "auto" the cashed image will then be named the same as the image script but with the correct extension depending on what image format have been chosen.

These are a few instances we can use as we begin to build graphs using JPGraphs. We are currently in the process of learning more about the software to present visual displays to the customer about stock prices and trends.

3.3 Machine Learning and Pattern Recognition

We plan to apply a combination of two technologies: ‘Wavelet Decomposition’ and ‘Neural Networks’ to create a web service which recommends the customer to Buy or Sell stocks. Here is a brief description of the two which explains why we are keen on using them to predict stock trends.

Wavelet theory:

1. Wavelets can localize data in time-scale space. At high scales (shorter time intervals), the wavelet has a small time support and is thus, better able to focus on short lived, strong transients like discontinuities, ruptures and singularities. At low scales (longer time intervals), the wavelet's time support is large, making it suited for identifying long periodic features.
2. Wavelets have a intuitive way of characterizing the physical properties of the data. At low scales, the wavelet characterizes the data's coarse structure; its long-run trend and pattern. By gradually increasing the scale, the wavelet begins to reveal more and more of the data's details, zooming in on its behavior at a point in time.
3. Wavelet analysis is the analysis of change. A wavelet coefficient measures the amount of information that is gained by increasing the frequency at which the data is sampled, or what needs to be added to the data in order for it to look like it had been measured more frequently. For instance, if a stock price does not change during the course of a week, the wavelet coefficients from the daily scale are all zero during that week. Wavelet coefficients that are non-zero at high scales typically characterize the noise inherent in the data. Only those wavelets at very fine scales will try to follow the noise, whereas

those wavelets at coarser scales are unable to pick up the high frequency nature of the noise.

Neural Networks:

They are artificial intelligence adaptive software systems that have been inspired by how biological neural networks work. Their use comes in because they can learn to detect complex patterns in data. In mathematical terms, they are universal non-linear function approximators meaning that given the right data and configured correctly; they can capture and model any input-output relationships. This not only removes the need for human interpretation of charts or the series of rules for generating entry/exit signals but also provides a bridge to fundamental analysis as that type of data can be used as input.

In addition, as ANNs are essentially non-linear statistical models, their accuracy and prediction capabilities can be both mathematically and empirically tested. In various studies neural networks used for generating trading signals have significantly outperformed buy-hold strategies as well as traditional linear technical analysis methods. While the advanced mathematical nature of such adaptive systems have kept neural networks for financial analysis mostly within academic research circles, in recent years more user friendly neural network software has made the technology more accessible to traders.

Requirements:

1. MATLAB 7.0 version
2. Image processing toolbox
3. Wavelet toolbox
4. Neural networks toolbox

We are currently working on connecting MATLAB module with SQL database and acquiring data to process it. Concurrently, we are learning more about the mentioned toolboxes to create an algorithm to suit our goals.

4. **Plan of Work**

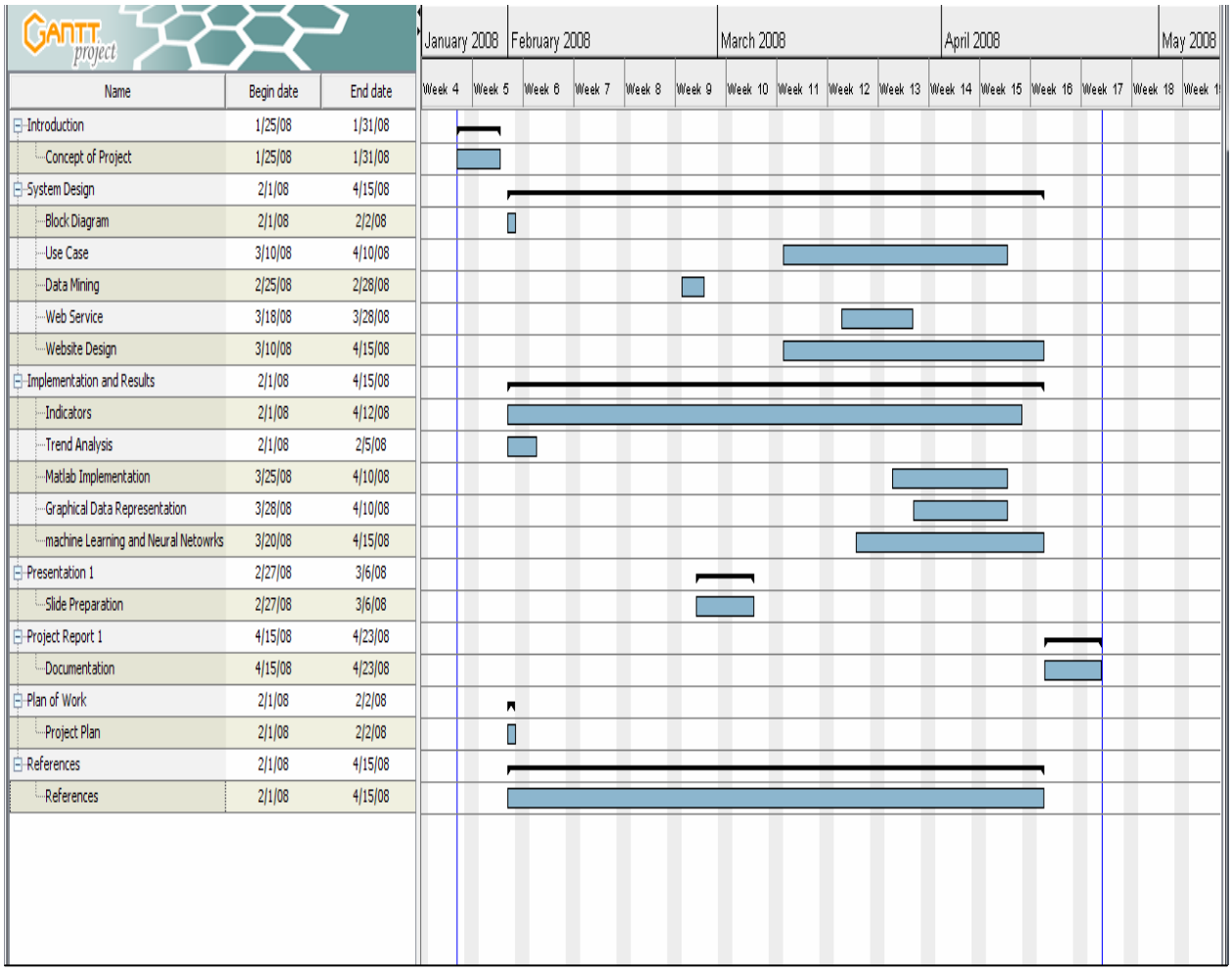

Fig.13 Gantt Chart representing Plan of Work

5. References

http://tomcat.apache.org/

http://ws.apache.org/axis/

http://www.netbeans.org/kb/60/websvc/rest-tomcat.html

Software Engineering (Chapter 8 – Web Services) by Prof. Ivan Marsic5.

www.investopedia.com

http://www.mathworks.com/

Pattern Recognition and Machine Learning – Christopher M Bishop

http://www.php.net/docs.php