Software Design Document

for

ProfitsRUS Stock Market System
Version 2.0

Prepared by GROUP: D

Submitted to Dr. Juerge Rilling

Tapsvini Shingala	**40000729**
Kaichen Zhang	**40000160**
Deepakkumar Subramaniam	**27894856**
Manjinder Kaur	**40038081**
Hanbing Zuo	**40000345**
Hassan Khalid	**27885490**

April 15, 2017

Version History

Sr.	Date	Name	Change Reason	Document Version
1	February 06, 2017	Group D	Beginning of Document/Project	1.0
2	March 24, 2017	Group D	Merging Yahoo Finance API with Watch list Features	2.0

Note: Highlighted area indicate the major changes in version 2.0

Table of Contents

1.0 Introduction:

The main aim to make this project is that to develop an application for software company which provides enormous services to the customers with some basic and advanced modifications. We used some methods/functions and techniques to develop this application. This application is work for both online as well as offline i.e. Desktop application. ProfitRUS created the many new products which ultimately helps to expand the business and the future of features of the application. Team members will implement the stock market analysis software as well as we will make this project in programming language Java. Our documentation will be based on the Professor's Sample Document which is provide by the professor to each group team.

1.1 Purpose:

The purpose of this documents is to provide the design of the application which fulfill the course requirements. In this we describe the following design features Architectural Styles and Patter, Module interface design, internal module design interface, each group member effort and time and at the end the whole flow the application using a diagram. that make this software application for customers who have a keen interest in stock market. This Software Requirement Specification Will Include many functions which will be perform by customers such as to read the CSV files, to select the date range, selection of Mas to displayed, to display charts for stock, options to sell/buy stock as well as cost estimation.

Software design document version 2.0 has describes other extended features like selecting stock, creating panel, managing watch list and API verification.

1.2 Scope:

ProfitRUS is a new technical software application for stock market which provides many guidelines to the customers which facilitate them to take decision during buy/sell. It is intended to provide detailed design specifications of the application that will be used as a basis for the implementation phase. It contains graph that will show the actual user outcome and some implementation coding interface. Which show that how functions/method, objects and their attributes are related among each other.

The system also provides the user flexibility to choose the CSV.file and analyze the details of the stock with the extended features like selecting stock, creating panel, managing watch list and API verification. By using watch list the user select the stock in which he might be interested.

1.3 References:

[1] J. Rilling, "Case study #2", January 2017, Concordia University, Department of Computer Science provided by the professor.

[2] J. Rilling, "Specification Sample Document", February 2017, Concordia University, Department of Computer Science.

[3] https://en.wikipedia.org/wiki/List_of_architectural_styles

[4] https://msdn.microsoft.com/en-us/library/dd409432.aspx

[5] http://www.csci.csusb.edu/dick/cs375/fullydressed.html

[6] https://www.techwalla.com/articles/how-to-draw-a-sequence-diagram-in-visio

[7] N. Nagappan et al [N. Nagappan, L. Williams and M. Vouk]. (2003). Software reliability estimation plug-in for Eclipse, North Carolina State University, leigh, oi:10.1145/965660.965667

[8] IEEE standard reading material - Systems and software engineering—Measurement process. IEEE standard adoption of ISO/IEC 15939:2007

1.4 Overview:

This software application provides many services to the end-users which contains various features that are given below:

- Application start and connect with the yahoo finance API.
- Select Stock
- Starting Year
- Starting Month
- Starting Date
- Ending Year
- Ending Month
- Ending Date
- Select moving average (MA) last days
- Select the moving average(MA) for particular period
- Choose CSV and show option.
- Display the option to buy/sell.

Display chart

- User select the company stock from the dropdown list;
- Select the start year, start month, start date
- Select the end year, end month, end date
- Select the moving average and browse the file location to read

To display Moving Averages

- System connect with the yahoo API and received the data including the all the history of stock.
- User select the moving average range from 20-100 day.
- User add a stock to watch list and also update it.

To Buy/Sell

- User select the stock based on the MA.

2.0 Architectural Design:

2.1 Architectural Design Purpose:

The purpose of architectural design is to build fundamental level view of the whole system, describes the components, their relationships to each other, so the programmer can construct the application and match the requirements of stockholders.
The architecture which we decided for the whole system is Client-Server Architecture design. Since the most important functionality of our application is to provide the client with real-time stock monitoring. And in such relationship two of the most critical entities are stock market software user which is considered as client in the Client-Server Architecture, and the server of Yahoo finance which plays the role of server in the Client-Server Architecture. When the client initiates a request and the server will reply the appropriate responses to the client, if it's a proper request for Yahoo finance data, then the server will send back the requested data through Yahoo Finance API, and the client will store the data in his or her local host.

The architecture for the client side implementation is Model-View-Controller (MVC) architecture. And we'll discuss it in detail in the next section.

2.2 Program Structure

Model-View-Controller (MVC) Architecture

In the client side implementation, we used a Model-View-Controller architecture. The component of model will play the role of the background engine, it takes charge of processing and analyzing data and messages, and it uses Yahoo Finance API to communicate with server side to connect the client and server. The component of view will invoke external libraries to generate the visualization charts for users, gap the bridges between background and frontend. The component of controller can handle the requests from client users, pass those request to the model so the results will be shown in the component of view. Here we introduce the detailed implementation of MVC model. In our implementation, the Stock.java class is the model component, the Chart.java class is the view component, and the FX.java class is the controller component.
We use Array List<Series<String, Number>> as the data structure of stock price series, Array List<String> list as the data structure of watch lists, and the model component will take Chart chart and Stock stock as parameters in its own logic. When we invoke the Yahoo Finance API ,we invoke the function void download(String stock, int YS, int MS, int DS, int YE, int ME, int DE) , and use String stringUrl ="http://chart.finance.yahoo.com/table.csv?s=" + getSymbol(stock) + "&a=" + (MS - 1) + "&b=" + DS + "&c=" + YS + "&d=" + (ME - 1) + "&e=" + DE + "&f=" + YE + "&g=d&ignore=.csv"; to download the csv files to the local host.

2.3 Client-Server Architecture - Alternative

The program is clearly separated into two components, client side and server side. For the server side, since it's already built to be used through API we don't need to consider the server component implementation. So our main focus will be client side programing implementation.

2.4 System Topology:

The application connects with Yahoo API, to get data for user needs in order run the application. User view the interested stock and add to watch list. User changes the MA according to the date and year. The connection from user to application including yahoo API third party.

2.5 Design Change:

In the first case study we used jfree library to draw the graph. Reading from existing csv file, and give buy/sell options by compare the close data with MA line. Also the crosscut

In the second one we used javaFX to create the panels and draw the graph, also we used Yahoo Finance API to download real-time stock data. Also the data is downloaded to local host, so it can be read as CSV file. (download method coding as mention above).
We produce watch list in which stock can be added or deleted. And the txt file is created and stored in the project. The buy/sell options are given by 20/100 MA crosscuts, and shown in the watch list.

2.6 Architectural Views 4+1 views:

The above mentioned architectural documents not have sufficient information to address all the stakeholders needs who involved in the project. So it's better to choose the 4+1 architecture view it concerns about each and every person involved in the project. The concerning stakeholders for the system are,

1. User
2. System Engineer
3. System Integrator
4. Management
5. User Scenarios

2.6.1 User or Logical View:

It is connected the functionality of system that provide to the user. Here the User interface part is the logical part.

2.6.2 System Engineer or physical view

It depicts the system from a system engineer point of view. It shows the topology of the system is connected with the yahoo finance and the user.

2.6.3 System integrator or process view

It deals with the dynamic aspect of the system.

2.6.4 Management or development view

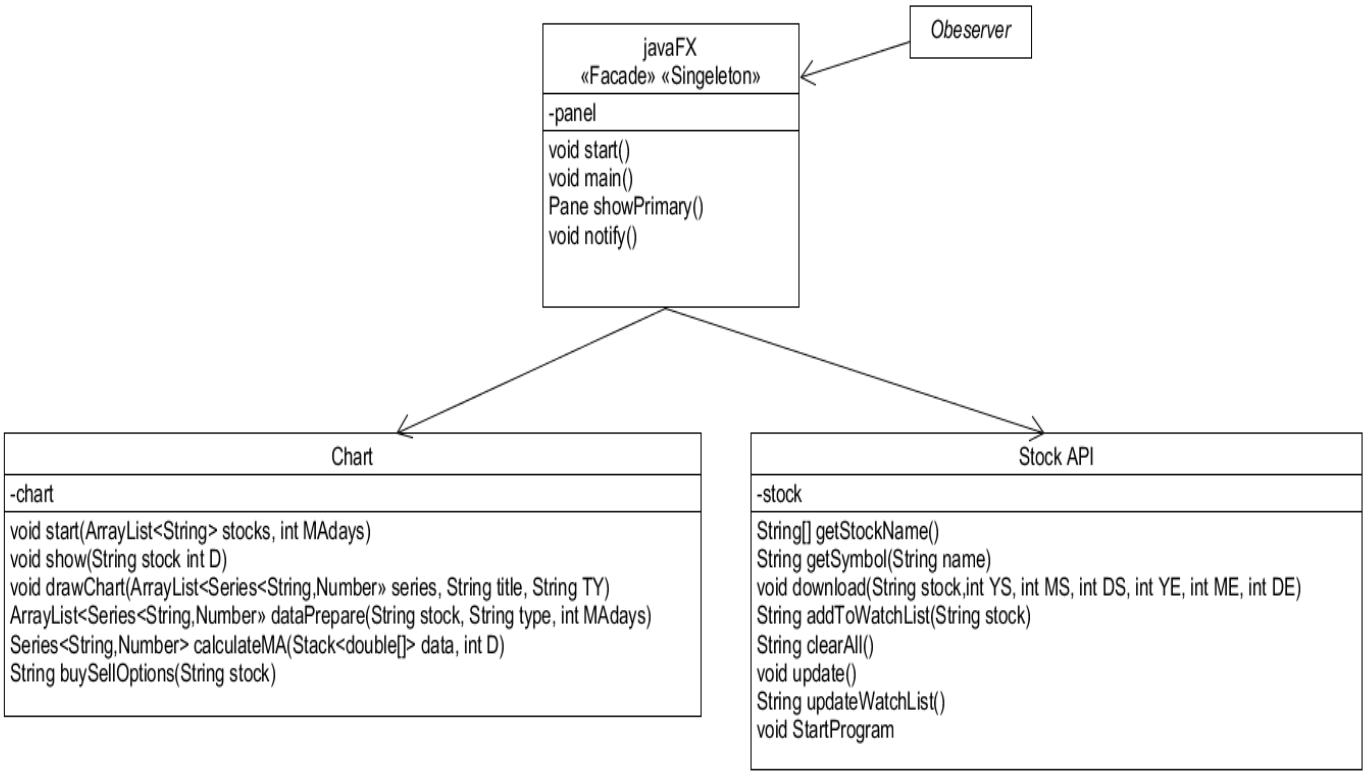
Programmer perspective and concern with software management. It mentions each and every classes in the system including the components and their interfaces and their connections.

2.6.5 User Scenarios or Scenarios

It is similar to the user view or logical view. It contains the requirements of the system. The overall of working of the system is validated by the User Scenarios.

3.0 Class Diagram:

We are using three classes in our application. JavaFX the main class, Javachart which displays the charts and graph on the screen according to selected the data values by the user and third one is stock API which connect with the Yahoo API for stock data. A good design is with max. cohesion and min. coupling.



4.0 Detail Design

4.1 Section Overview

In our application we use classes attributes methods and their relationship between each other to run the program in a sequence and control manner.

4.2 Design Patterns

Patterns are used to follow the advancement and expertise of design patterns which are define by the expert's programmer and developer in order to main the high quality standers and maximum achievements.

Singleton
Used to write and read the files and data.

Observer
It is used to observe any change in the notification panel. Either a new stock is selected or a new MA is selected.

Threads
Improved performance and concurrency by using multithreading and multi context together coding of remote procedure calls and conversations. Calling inside the threads of API and web call allow to continually threading without any interruption.

Controller
In our program the JavaStock acts as the controller, it certainly controls the flow of action between the user and the model which is the Yahoo API. It contains the following methods
WatchList():
Watch list manages the stocks that the user may be interested. Once the user has shown some interest in the stock it will be added to the watch list.
Stock():
Stock() method validates the stock details with the start and end date it picks stock as user's interest.

5.0 Internal Module Design with UI Description:

5.1 Interface

JavaFx

Class Name	Fx.java
Inherits	No
Class Import	Java.io.IOException; Import javafx.application.Application import javafx.collections.FXCollections;

```
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.ComboBox;
import javafx.scene.control.Label;
import javafx.scene.control.TextArea;
import javafx.scene.control.TextField;
import javafx.scene.layout.BorderPane;
import javafx.scene.layout.Pane;
import javafx.scene.layout.VBox;
import javafx.stage.Stage
```

Description

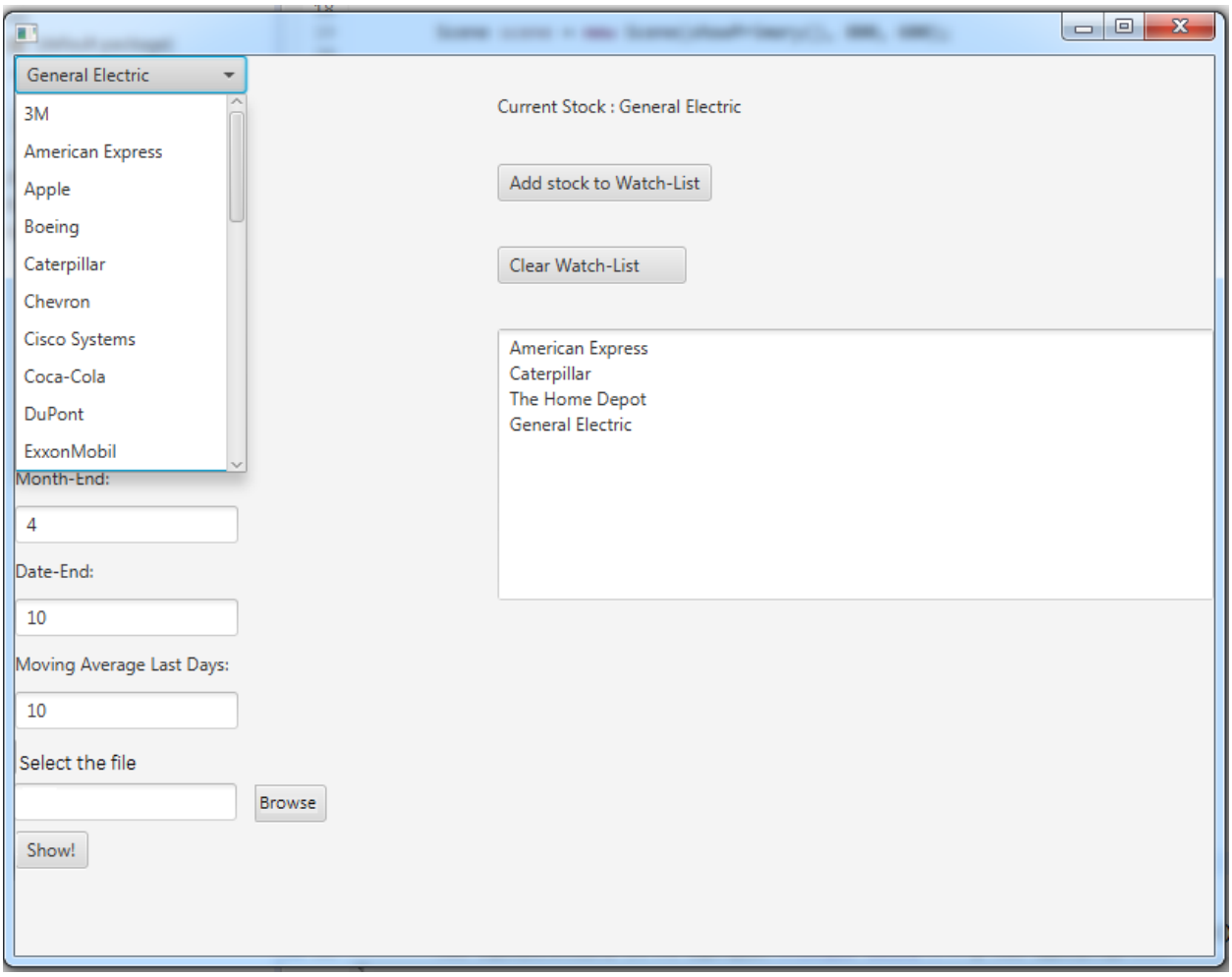
It is the main class that run the main function for the application. It contains the user interface (layouts) to interact with the application. Data is bounded in some text fields (Data loaded when application starts).

Attributes

Visibility	Data Type	Name	Description
Private	input	Scanner	It is used to take the inputs
Private	int	Boolean	Either the input is correct/incurrent
Private	int	User input	Its default value is 0.
Private	valid	User input	It validate the user input
Private	variable	Demo	Object of JavaFX class
Private	variable	Final	Object used in interface class
Private	variable	Vbox	Get it children values and draw in the graph as dot as a single
Private	variable	VboxWatch	Attribute of vbox

Methods

Visibility	Method Name	Description
Public	Main Mathod()	It is the main method to run the client interface
Public	interfaceControl	It provide an interface to user and give the option to select the following options
Public	Valid()	It take the user input in a loop and break when the input is false
Public	Pane showPrimary()	It load data when the application is load for the first time and allow user to select the values from the given options
Public	stock.download()	It connects with the yahoo finance to download the data for live.
Public	ADD.setOnAction()	Its allow to bind the list.
Public	CLE.setOnAction	After bind, it close the list
Public	Userdatetimeswitch()	It take the start and end date, time period from the user and make a graph on x axis and y axis.
Public	Userrangerswitch()	It take the range time period and date and time from the user.
Public	Exit()	It terminate the application.



5.2 Engine:

Java Stock

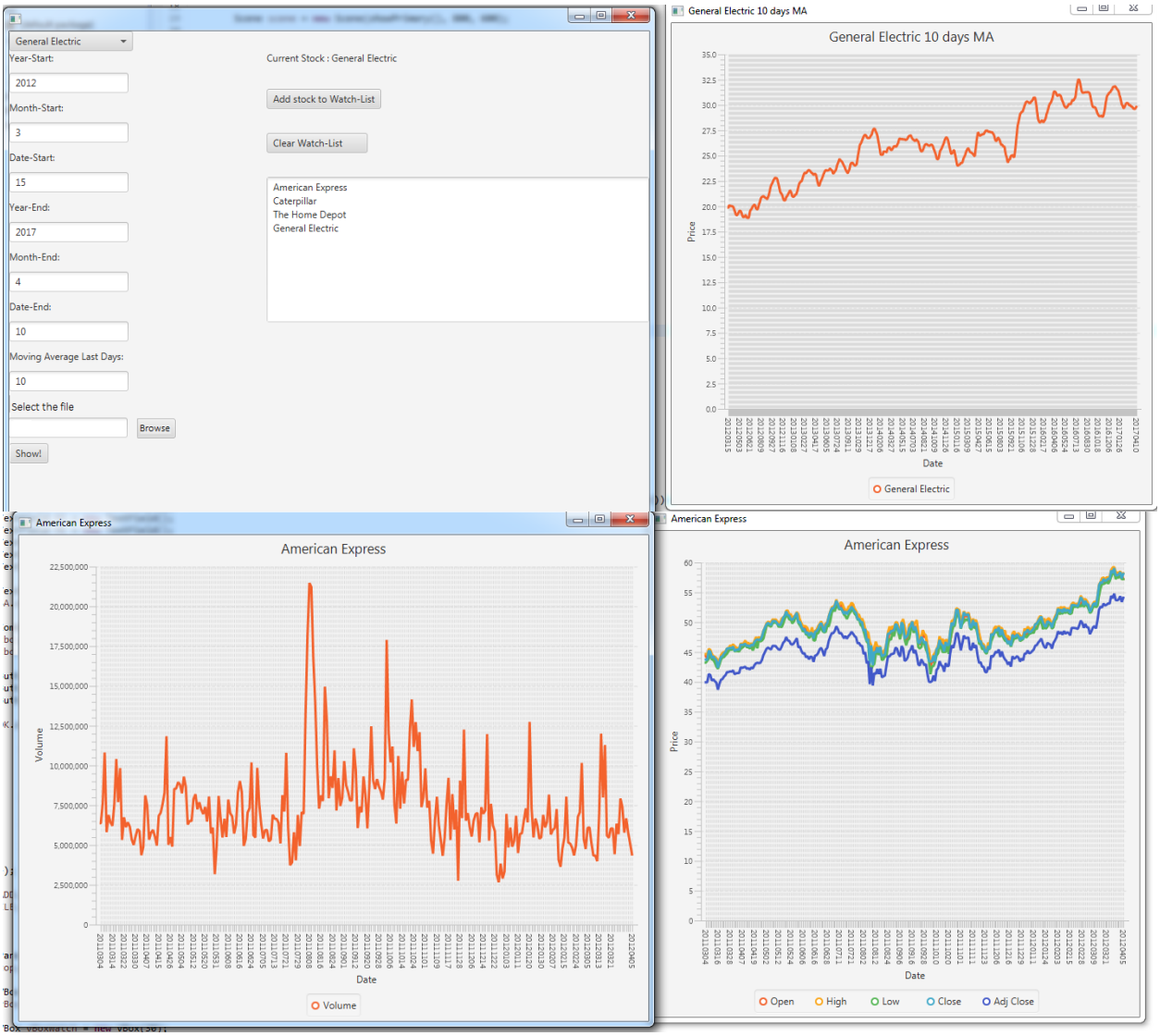
Class Name	Stock.java
Inherits	JavaFx
Class Import	Java.awt.Color import java.io.File; import java.io.FileNotFoundException; import java.io.FileWriter; import java.io.IOException; import java.io.PrintWriter; import java.net.MalformedURLException; import java.net.URL; import java.util.ArrayList; import java.util.Date; import java.util.Scanner;
Description	It handles the yahoo live data for both MA and Price chart. It is connected for live data
Attributes	

Visibility	Data Type	Name	Description

Private	Final long	Serial version	It is used provide the version
Private	dataset	time Series	It provide the time collection
Private	dataset	chart	Its provide the chart
Private	variable	printwriter	It read the scanner and csv file
Private	int	Chart height	It define the chart height
Private	variable	Malformned	Throws exception when URL invalid
Private	int	Chart width	It define the chart width
Private	Datetime	Date	It is used to set/get date
Private	Datetime	day	It is used to set/get day
private	double	head	It provide the head or starting point
Private	Double	Data	It contain the class data

Methods

Visibility	Method Name	Description
Public	Engine()	It initialized the attributes and their properties
Public	Array List()	It collects all the data from different variables and store using this method
Public	Stockname()	Its shows the name of all the companies
Public	Getstockname()	It is used to get/set the stock
Public	Getsymbol()	It is used to read all the companies name in loop
Public	Download()	It download the company live data
Public	Addtowatch()	It is used to add an item to watch list
Public	Clearall()	It is used to clear the things
Public	Filewritrer()	It is used to read the file
Public	Createdataset()	It creates the dataset regarding time series where the define the following: Open price, high price, low price, close price, volume, adjacent close price, moving average
Public	dataprocessing()	It take the data in a loop using string line argument and split the data into I and j indexes
Public	timePointer()	It pints the time period using month day and years. It take data for X,Y,Z,O,A,Q,P, and average and bind all the data in closedata().
Public	Calculatemovingavg()	It calculate the data by using average data and closing data
Public	Selecttime()	It start the time period on X,Y,Z,0,P,Q, and moving average data by using day month and year
Public	Buysignal()	It shows the buysignal by using the computation average.rangeMA==mathfloor&&closedata.get(i-1)
Public	Displaycontent()	Its show the display as outcome
Public	Showall()	It shows the data on all the axis
Public	Movingaverage()	Add both the dataset to show an average on x-axis and y axis
Public	Filenotfound()	If the dataset not able read any kind of file or available file
Public	Chart()	It show the chart with components time and value / true and false.



5.3 Graph:

Java Chart

Class Name	Chart
Inherits	JavaFx
Class Import	Java.awt.Basicstrocke import java.io.File; import java.io.FileNotFoundException; import java.util.ArrayList; import java.util.Scanner; import java.util.Stack; import javafx.scene.Scene; import javafx.scene.chart.CategoryAxis; import javafx.scene.chart.LineChart; import javafx.scene.chart.NumberAxis; import javafx.scene.chart.XYChart.Data;

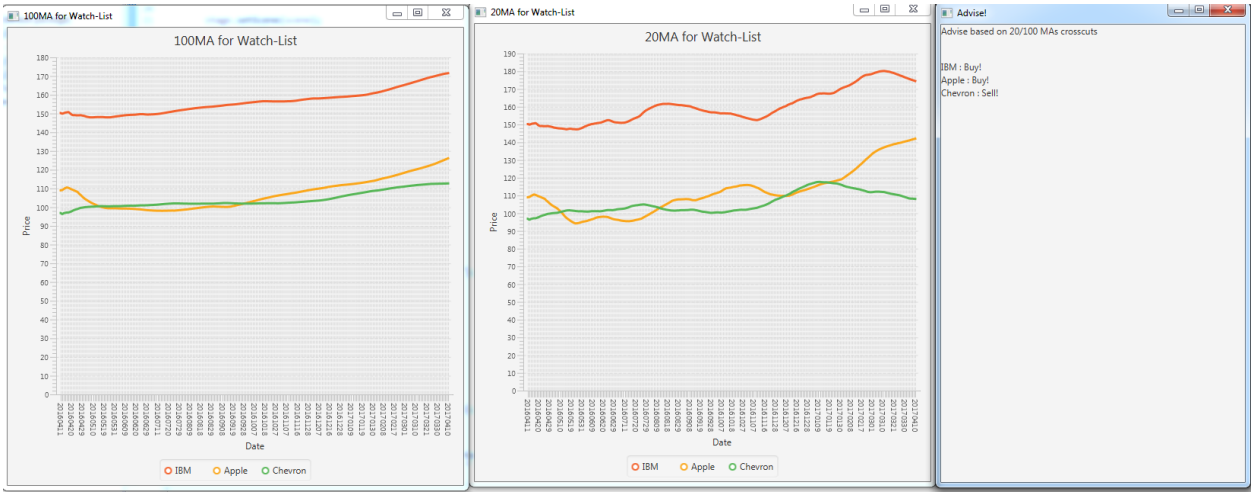
	import javafx.scene.chart.XYChart.Series; import javafx.scene.control.Label; import javafx.scene.layout.VBox; import javafx.stage.Stage;
Description	It display the MA and prices charts for the selected company, date and time period

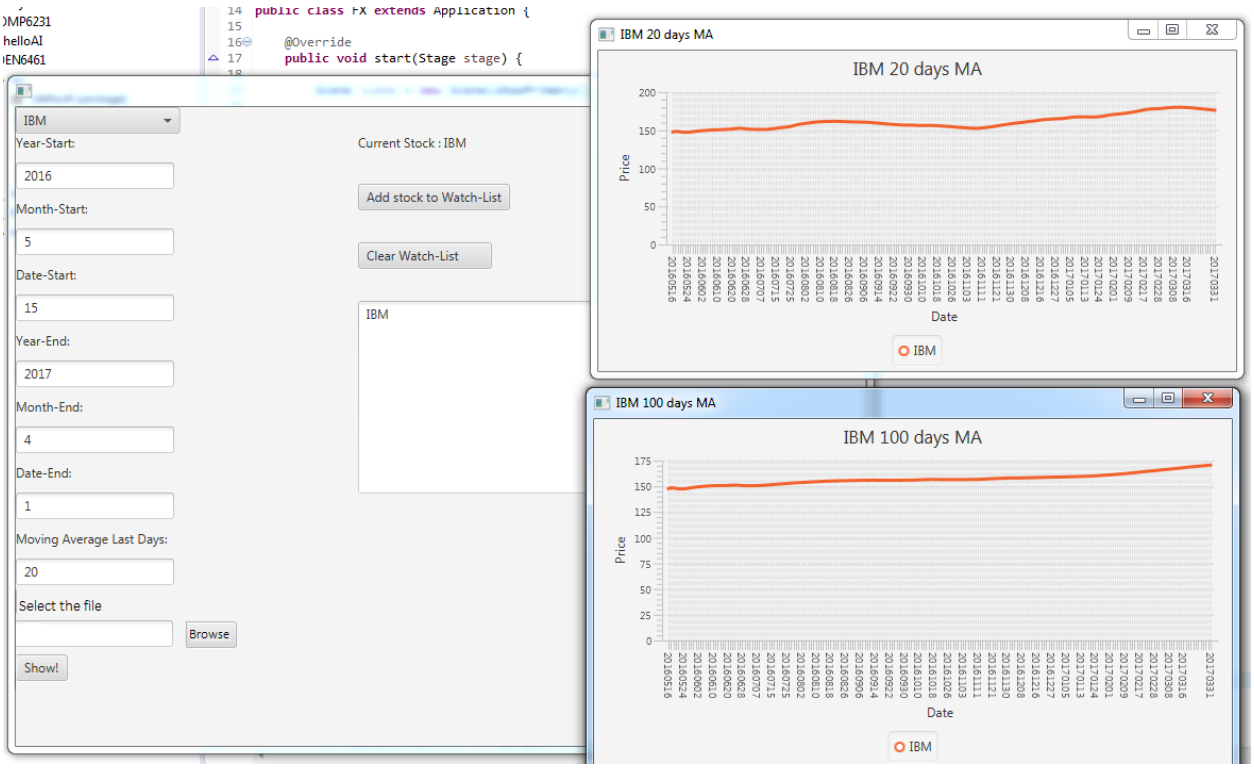
Attributes

Visibility	Data Type	Name	Description
Private	Int	width	It is used to set the width
Private	Int	height	It is used to set the height
Private	Int	padding	It is used to set the padding
Private	Int	labelpadding	It is used to set the label padding
Private	variable	BOS	It is beginning of string
Private	variable	Vbox	Get the children values for chart
Private	variable	stage	Show the current stage of stock
Private	Variable	Category axis	used to label the date
Private	Variable	Number axis	Manage the chart on x-y axis
Private	Color	Line color	It is used to set the line color
private	Color	Point color	It is used to set the point color
Private	Color	Grid color	It is used to set the grid color
Private	Int	Point width	It is used to set the point width
Private	Int	Number division	It is used to divide the numbers
private	int	Score	It is used to store the score

Methods

Visibility	Method Name	Description
Public	Paintcomponents()	It draw the graph its points padding and labeling it
Public	Start()	It is used to prepare the date that shown in chart
Public	Drawchart()	Lable the chart, box and add children to chart.
Public	Show()	It is used to chart on x-y axis
Public	Linechart()	Its handle the moving line on x-y axis
Public	Arraylist()	Contain a series of number, conditions and loop
Public	Series	Return the values to the chart
Public	Decidebos()	Read the csv file and stock and return data
Public	whitebackgroung()	It draw a blank background graph





6.0 Major Challenges and Redo Project Again.

While implementing the application we facing some challenges regarding API. And the major challenges are

- How to implement the API, and read real time data.
- How to change from Jfree library to javaFX library.

6.1 Redo Project Again

If we are going to redo the project, we will refactor the project so that our external behavior will be same only the internal things will get changed with better design. Refactoring removes the bad smells in the design and improves the readability of the code, provides flexibility in the design and removes the unwanted methods in the code. So, the design will not decay as soon, it provides continuous improvement.

7.0 Team Member and log sheets

7.1 Tapsvini Shingala – 40000729 Group D

Date	Task	Duration
March. 24	Initial Design meeting	1 Hour
March. 25	Group meeting with new use case(Create Panels)	1.5 Hour
March. 27	Research on the internet. Watch tutorials	1.5 Hour
March. 29	Start the design	1 Hour
March. 31	Implementation the code and architecture	3.1 Hour
April. 02	Draw the diagrams	1 Hour
April. 05	Discuss the work with other group members	2 Hour
April. 07	Improve errors and problems	1 Hour
April. 12	Improvement in the code and design	1 Hour
April. 14	Final version 2.0	2 Hour
	Total:	**16.1 Hours**

7.2 Hassan Khalid – 27885490 Group D

Date	Task	Duration
March. 24	Initial Design meeting	1 Hour
March. 26	Group meeting with new use case(API Verification)	1.5 Hour
March. 28	Research on the internet. Watch tutorials	1.5 Hour
March. 30	Start the design	1 Hour
March. 31	Implementation the code and architecture	3.1 Hour
April. 03	Draw the diagrams	1 Hour
April. 05	Discuss the work with other group members	2 Hour
April. 08	Improve errors and problems	1 Hour
April. 11	Improvement in the code and design	1 Hour
April. 15	Final version 2.0	2 Hour
	Total:	**16.1 Hours**

7.3 Deepakkumar Subramaniam – 27894856 Group D

Date	Task	Duration
March. 24	Initial Design meeting	1 Hour
March. 27	Group meeting with new use case(Select Stock)	1.5 Hour

March. 29	Research on the internet. Watch tutorials	1.5 Hour
March. 30	Start the design	1 Hour
March. 31	Implementation the code and architecture	3.1 Hour
April. 04	Draw the diagrams	1 Hour
April. 05	Discuss the work with other group members	2 Hour
April. 09	Improve errors and problems	1 Hour
April. 13	Improvement in the code and design	1 Hour
April. 15	Final version 2.0	2 Hour
	Total:	**16.1 Hours**

7.4 Manjinder Kaur – 40038081 Group D

Date	**Task**	**Duration**
March. 24	Initial Design meeting	1 Hour
March. 26	Group meeting with new use case(Create MA Panel)	1.5 Hour
March. 28	Research on the internet. Watch tutorials	1.5 Hour
March. 30	Start the design	1 Hour
March. 31	Implementation the code and architecture	3.1 Hour
April. 03	Draw the diagrams	1 Hour
April. 07	Discuss the work with other group members	2 Hour
April. 10	Improve errors and problems	1 Hour
April. 14	Improvement in the code and design	1 Hour
April. 15	Final version 2.0	2 Hour
	Total:	**16.1 Hours**

7.5 Kaichen Zhang – 40000160 Group D

Date	**Task**	**Duration**
March. 24	Initial Design meeting	1 Hour
March. 25	Group meeting with new use case(Manage Watch list)	1.5 Hour
March. 29	Research on the internet. Watch tutorials	1.5 Hour
March. 30	Start the design	1 Hour
March. 31	Implementation the code and architecture	3.1 Hour
April. 04	Draw the diagrams	1 Hour
April. 08	Discuss the work with other group members	2 Hour
April. 10	Improve errors and problems	1 Hour
April. 13	Improvement in the code and design	1 Hour
April. 15	Final version 2.0	2 Hour
	Total:	**16.1 Hours**

7.6 Hanbing Zuo – 40000345 Group D

Date	**Task**	**Duration**
March. 24	Initial Design meeting	1 Hour
March. 28	Group meeting with new use case(Add Stock)	1.5 Hour
March. 29	Research on the internet. Watch tutorials	1.5 Hour

March. 30	Start the design	1 Hour
March. 31	Implementation the code and architecture	3.1 Hour
April. 06	Draw the diagrams	1 Hour
April. 08	Discuss the work with other group members	2 Hour
April. 11	Improve errors and problems	1 Hour
April. 14	Improvement in the code and design	1 Hour
April. 15	Final version 2.0	2 Hour
	Total:	**16.1 Hours**

8.0 Conclusion:

All the group member participates in the whole project equally. We design the application in such a way it meets the requirements and able to adopt the changes in future. It will meet the requirements either it's a live API data or given CSV file or web base. Architectural and design pattern make it easy to handle the application and to understand the flow of methods, objects and their relationships. We adopt Agile for quality and fast development. Design the code and classes in such a way that max. methods can be reuse and provide code flexibility and use less resource.

9.0 Group Member Contributions

ID	Name	Contribution	Percentage
40000729	**Tapsvini Shingala**	**Prepare UML diagrams, Handle the use case, participate in coding and documentation, meeting and review the final document.**	**100%**
40000160	**Kaichen Zhang**	**Prepare UML diagrams, Handle the use case, participate in coding and documentation, meeting and review the final document.**	**100%**
27894856	**Deepakkumar Subramaniam**	**Prepare UML diagrams, Handle the use case, participate in coding and documentation, meeting and review the final document.**	**100%**
40038081	**Manjinder Kaur**	**Prepare UML diagrams, Handle the use case, participate in coding and documentation, meeting and review the final document.**	**100%**
27885490	**Hassan Khalid**	**Prepare UML diagrams, Handle the use case, participate in coding and documentation, meeting and review the final document.**	**100%**
40000345	**Hanbing Zuo**	**Prepare UML diagrams, Handle the use case, participate in coding and**	

		documentation, meeting and review the final document.	