Criterion A

Contents

The Scenario

My client is Mr. (Vorlow), a member of the Economics department at St. Catherine's British School. He wants to gradually familiarise the students with the rudiments of financial economics, an invaluable asset that will inexorably give them an advantage in their adult lives.

During the month of April 2019, in an economics lesson (Appendix, **Communication with client (Mr. Vorlow)**), he expressed his desire for the creation of a resource that would improve students' financial literacy with regard to how the stock market functions. Since financial economics is not part of the Economics curriculum, he could not spend an extensive amount of time on it - as it detracted from the preparation of the other parts of the syllabus. Also, the development of the resource would invariably require coding literacy and there are no free online resources of the type outlined above.

In addition, I conducted a survey, whose form is (Appendix, page 3) in my school year's economics cohort to find out whether such a resource would be of interest. The results were encouraging with more than 80% replying positive (Appendix, **Survey**).

After I initially approached him with my plan for a solution, he seemed interested and we booked an interview (Appendix **Interview with Mr. (Vorlow)**), to analyse the issue and discuss the details of my future software.

Rationale for proposed solution

As my client and I were talking about the problem, I thought that it would be a good opportunity for my Internal Assessment to create an IT solution, namely a functioning product that would be intuitive, user friendly and highly interactive.

After some deliberation, I have decided that a web application (executable file) based on a java program following an object oriented approach would be most suitable to help solve Mr. (Vorlow's) problem. This is because the programming language Java is one of the best when it comes to OOP[1]. There are alternative approaches such as using off the shelf programs, but I decided to use the bespoke software development approach (Appendix, **Summary and Assessment of alternative approache**s). This will enable me to implement various requirements through the use of classes and objects with relative ease.

More specifically, I am planning on using the open source NetBeans IDE as it has a powerful GUI[3] builder which is going to be an integral part of the development of the software.It will enable me to create GUI's quickly and favours the iterative prototyping approach which has the advantage of quick building times for the initial product. It also has a feature whereby I can insert libraries such as JFreeChart[1] for plotting. My future software will be based on such an interface so as to meet the requirement of intuitivity and interactivity of the product.

In addition, I plan on adding the JDBC[2] driver to my NetBeans project path in order to establish a connection with the database browser SQLite. This will facilitate the development of the code as I will need to store historical stock data of companies in a flexible manner by having access to a database stored locally with each record being comprised of the name, password, balance and information about the stocks they own - stored in an abstract data structure. Also, using a database, I will be able

[1] http://www.jfree.org/jfreechart/

to efficiently query data such as the volume of stocks of a particular company owned by a user or their balance. Not to mention, I will attempt to implement a stock market API[2] (AlphaVantage) to track and access stock prices in real time.

Lastly, I am going to incorporate mathematical and statistical models for my stock market simulator in order to implement basic monte carlo simulations.

Success Criteria

1. The product will simulate an interactive stock market using monte carlo / stochastic methods
   1.1. The product will successfully model an efficient market
   1.2. The product will model market volatility

2. The product will allow the students to trade virtual stocks using a fixed initial budget
   2.1. The product will enable the user to buy shares
   (for instance, Symbol: AAPL, Volume: 100)
   2.2. The product will enable the user to sell shares
   2.3. The product will enable the user to short shares

3. The product will model the changes in the portfolio

4. Multiple portfolios - each for a user - will be stored in a database once initialised and their values tracked in real time using a pertinent API for Java.

5. The Product will use historical data (Appendix, **Summary and Assessment of alternative approaches)** from companies stored in a database format to make predictions for their future price using monte carlo/stochastic methods.

[2] API: Application Programming Interface
[3] OOP. Object Oriented Programming
[4] GUI: Graphical User Interface
[5]JDBC: Java Database Connectivity

6. The product will display historical data of companies on a graph.

7. The product will give projections of future price movements of a company’s stock as line graphs using simple statistical procedures incorporating “drifts” and “shocks”

8. The product will produce and display a distribution (approximating a lognormal distribution) given the historical data

9. The product will use statistical methods and models. More specifically, random walks and Geometric Brownian Motion (Appendix, **Geometric Brownian Motion Illustration**).
10. The product will include social media posts, more specifically, tweets generated from a simple query using an API.

Word Count: 603