|  |  |
| --- | --- |
| Intermediate Java Programming | University of Southern Indiana  OOP PROJECT DESCRIPTION  JAVA STOCK FORECASTER | SYED HASAN ABBAS  ZAIN-UL-ABIDEEN  MUHAMMAD ABDUL RAFEY FAROOQI |

National University of Sciences & Technology (NUST)

School of Electrical Engineering and Computer Science (SEECS)

Department of Computing

**Semester Project: Spr 2022**

**CS212:Object Oriented Programming (3+1)**

**BESE12-B**

**Due Date: 10th July 2022**

**Group Assignment**

**Note: You will upload your submission on LMS.**

**Course Learning Outcomes (CLOs)**

1. Understand the difference between procedural and Object Oriented Programming paradigms.
2. Demonstrate the ability to create and use OOP constructs to map real world scenarios.
3. Develop programs using object-oriented techniques.
4. Use the latest IDEs to enable quick development, testing, documentation, and packaging of programs.

The semester project for this course is an open-ended assessment which allows you to apply the skills acquired in this course on a real-world complex engineering problem. This is a group assignment where students are expected to work in groups to identify a real-world problem, and apply concepts related to OOP to propose a solution. Your solution **must be submitted as a formal report** in which you are expected to demonstrate the work done to achieve the solution for the identified problem. At the end of the semester, your group is required to present your work to the class as per schedule.

The report must contain

* **A 1-2 page proposal of your selected project, explaining the overall aims and achievable targets. You should concisely explain the problem you are aiming to solve using the skills acquired in this course.**

## INTRODUCTION

We are developing a Desktop based application that will retrieve real-time stock market data shows the user interactive visualization of graphs. This application will also forecast the price of stocks for the next two hundred days with an approximate tolerance of 85 percent. The application will replicate a reality-based buy/sell situation where the user will buy or sell according to the predicted values shown to him.

## REAL WORLD PROBLEM

People used to buy/sell stocks or invest in the International after thinking a lot. It would sometimes churn up their whole day. So, to facilitate newbies and the old ones in the industry, we thought of bringing a solution to this problem. It made us ponder upon the technologies that were available to us. So, by sorting the things out, we agreed upon making a user-friendly interactive system that would help the customer that he is making the right decision by investing in a particular stock.

## AIMS

* The system will use appropriate data visualization mechanisms such as graphs and charts to aid user-system interaction. i.e., intuitive data representation.
* The system will be a decision-making support system that will assist users to make decisions on the intended scenario, for instance, users may specify the profit or coin they intend to buy.
* We will apply suitable techniques and algorithms to make our system derive decisions based on the previous evidence given to it.
* We will also combine multiple machine-learning models and algorithms to evidence sophistication.
* The system will be interactive and able to handle real-time information, i.e., any alteration to the data from the interface will reflect in atmost a day. For instance, the change in price or currency should reflect in the predicted value.
* The system will allow the users to plan and forecast investment using a range of scenarios, i.e., the system permits alteration to answer ‘What-If’ questions (for instance, what will be the profit if **TSLA** is sold for $1000 or $3000. Users should be able to change the quantity of the coin etc).

## TECHNOLOGIES

* Java
* JavaFX, SceneBuilder for Desktop Application
* Long Short Term Memory (LSTM) for Time Series Forecasting
* Matplotlib for Graphs
* Pandas and Numpy for Data Analytics
* Tiingo API for retrieval of real-time data
* GoogleNews API
* **Appropriate sections in the report in which you need to explain various modules of the project and possible solutions considered. You also need to justify why you have applied a particular concept when you had other options available. Please highlight the choices made clearly.**

**The report is attached in the zip file.**

* **An elaboration of your project idea by mapping various OOP concepts to various modules used in your project. You are required to mention briefly how a particular concept was applied in your project. and what were the alternative options.**

|  |  |
| --- | --- |
| Topic | Application in your project |
| Flow Control | If else statements are being used to control the flow of control in java. In java flow control can be changed depending upon the conditions. |
| Composite Data Types | Bought Stocks (String Array) that stores the number of stock that customer has bought and can be used in displaying the asset’s information of the customer. |
| Setters & Getters, Default and No-Argument Constructors | Default and no argument constructors are being used in the different files such as Trade.java |
| Method (Function) Overloading | Buy Stock function in which user can either enter only amount of money or user can enter interest rate in addition to amount of money. |
| Static Class Members, this Reference | Static class members are basically used in check fields and this reference is basically used in default constructors. |
| Arrays and strings / functions | Bought Stocks (String Array) that stores the number of stock that customer has bought and can be used in displaying the asset’s information of the customer. |
| Inheritance In Java | Trading Stocks (super class) :: Buy stock (subclass) : Sell stock (subclass). The super class contains data member such as requiredInterestRate which is inherited by subclasses and is used in method Calculate Tolerance. |
| Super classes and sub classes | Trade is declared superclass (abstract) and buy and sell is its subclasses. |
| Method Overriding, Constructors in subclasses | Trade method has been overrided to either buy or sell stocks. |
| Polymorphism | Polymorphism has been used in the Trade class as well as buy and sell subclasses. |
| Abstract Classes & Methods | Fetch Stock Data abstract class whose main function would be to fetch and process the particular Stock value in inherited classes. So the user does not need to know where the data is being fetched from so it is perfect implementation of abstract class. Similarly for abstract methods you can have a show trend method which will be in each stock subclass. |
| Final Methods and Classes | calculateProfit in stock value (Final method). It will be used to return the calculated profit value for a particular stock. |
| Interfaces and Abstract Classes | Trading Stocks (Abstract super class):: Buy stock (subclass) : Sell stock (subclass) |
| Exception Handling | Data validation and exception handling while buying/selling stocks (user prompt). |
| Graphical User Interfaces | JAVA FX and SCENE BUILDER |
| File Manipulation | For reading CSV and TXT files. |

P a g e **2** of **3**

SUBMISSION GUIDELINES

You need to present your findings and understanding in a report format. You can use code snippets and output snippets to demonstrate your understanding section-wise/ step-wise. Appropriate explanation should be provided along with supporting figures and results. You should use table of contents and include captions and figure numbers for all figures included. You need to provide your complete code at the end of the report.

**Marking Rubric (Total Marks 100)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Report** | **60 Marks:** All  specifications are met, and demonstrated in the report with appropriate explanation. This section should be well presented to earn full marks. | **45 Marks:** Most specifications are met, and demonstrated. However, there is room for improvement in presentation and completeness. | **30 Marks:**  Work is not satisfactory, however a reasonable attempt is made. | **Zero Marks:** Work does not qualify as a reasonable attempt. |
| **Presentations/**  **Demos** | **30 Marks:** All  specifications are met, and demonstrated | **25 Marks:** Most specifications are met, and demonstrated. However, there is room for improvement | **15 Marks:**  Work is not satisfactory, however a reasonable attempt is made. | **Zero Marks:** Work does not qualify as a reasonable attempt. |
| **Class**  **Participation** | **10 Marks:** Exceptional | **7 Marks:** Room for improvement | **4 Marks:** Satisfactory | **Zero Marks:**  Unsatisfactory |

P a g e **3** of **3**