# ECS414U/A Miniproject form

Queen Mary University of London

2021/22

|  |  |
| --- | --- |
| Name | Michał Staszewski |
| Student ID | 210338883 |
| Submitted file name | Stock Simulation – Michal210338883 |
| Level of this program (1, 2, 3, Extra) | Extra |

|  |
| --- |
| Brief description of the program. Write the chosen theme and a high-level overview of the features (two or three sentences should suffice). |
| I chose the theme of stock simulation app that is based on a GUI from a library Java Swing. Each window in the program is created as a class and all except the main are Subclasses of an abstract class new window. Each simpler window Is referenced as the superclass and overwrites the abstract methods that set them up. I also implemented a system that changes the value of the stocks (In the StockValueChanger) depending on what was the last time user opened the app counted in days in the DateChecker class. All the information about the user, the previous date and the stock prices are saved externally on the .txt files using the IO\_System class. |
|  |

|  |  |
| --- | --- |
| List all your source code files, and briefly describe their roles. Add as many rows as necessary. Mark the main file used for compilation in bold. | |
| File name | Description |
| Main.java | Responsible for the whole program running, all the information is stored on it, and it’s passed to most objects for accessibility. Also is the main window with user data and stock data with option to buy or sell them. Switches between itself and Portfolio using menu bar. |
| NewUser.java | Window where the user can type in the name they want to have in the program. Handles reusing of the same name as past users. |
| ExistingUser.java | Window where user types in the name they used when they last used the program. |
| StartingWindow.java | First window that appears, gives the choice of going as a new user or load in the last save. |
| Portfolio.java | Window that shows the user their stocks Portfolio, it switches between main and itself using menu bar. |
| NewWindow.java | Abstract class and base for creating the 4 windows above. Has multiple methods used by all subclasses and abstract methods for setting appearance of the window. |
| ActionListenerMenu.java | Allows switching between Main and Portfolio. |
| ActionListenerStarting.java | Detects the button detected and shows either the New or Existing User window. Given to StartingWindow. |
| BuySellActionListener.java | Allows the buying and selling of stocks by clicking button on the main window. Handles stuff like not sufficient funds or no stocks owned. |
| TextFieldMouseListener.java | Deletes the text in the textfield on mouse click. |
| SubmitExistingNameActionListener.java | Allows the program to check if the file of that name exists on button click and show error is file doesn’t exist. |
| SubmitNewNameActionListener.java | Allows to submit new name for the new user and checks for stuff like too long name and name being taken. |
| SubmitNameActionListener.java | Base for the 2 ActionListeners above. Abstract class with basic information like Main window and the field assigned on constructor. |
| DateChecker.java | Does 2 things. First subtracts 2 dates from each other and gets the number of days that passed. Also gets the current date and passes it to the IO System to save. |
| IO\_System.java | Saves the data into the external files when program is closed and retrieves them as well when the program starts. |
| Stock.java | Stocks data, base class for the program theme. Has quantity, value, name and symbol. |
| User.java | Stores information about the user and what stocks they own, gives access to those attributes. |
| StockValueChanger.java | Object that applies a function of value change to the stock, given by the amount of days that passed since program was last used (min 1 day). |
| WindowCloser.java | Once either Main window or Portfolio is closed it saves the data about the user and their stocks as well as the stocks with their new value. |

|  |
| --- |
| Class diagram, in the format specified in the instructions. |
| **class** Main{}  **class** Stock{}  **class** User{}  **class** DateChecker{}  **class** IO\_System{}  **class** NewUser **extends** NewWindow{}  **class** ExistingUser **extends** NewWindow{}  **class** StartingWindow **extends** NewWindow{}  **class** Portfolio **extends** NewWindow{}  **class** SubmitExistingNameActionListener **extends** SubmitNameActionListener[]  **class** SubmitNewNameActionListener **extends** SubmitNameActionListener[]  **class** StockValueChanger{}  **class** BuySellActionListener{}  **class** TextfieldMouseListener{}  **class** ActionListenerStarting{}  **class** ActionListenerMenu{} |

|  |
| --- |
| Usage instructions. Describe briefly what features are available to the user and how to use them. If File I/O is used, list and describe the files involved. |
| First user has to either create a new user with 5000 currency of balance and their name, or enter their name from when they last used the program. The I/O system will check if the files exist and load the data from them. Then user is left by themselves, they are given their balance on the Main window and can buy or sell stocks. They can also view their stocks in the Portfolio tab. Once the user has finished, they can close the program and their data will be saved. When they come back to the program and open their previous account the values of the stocks will be updated like on a normal stock market. Then again they can sell/buy stocks.  (For testing purposes the date can be manually changed in the Date.txt file) |

|  |
| --- |
| Other comments. |
| 2 .txt files are required to run the program:  Stocks.txt - It contains all the information about the stocks in the program  Date.txt – it contains the date of the last usage of the program  Other .txt files are user data or a stock backup (if something got erased) |