Members: Adam Toms-Sheridan, Unais Qureshi, Maciej Madejsza, Rajababu Kushwaha

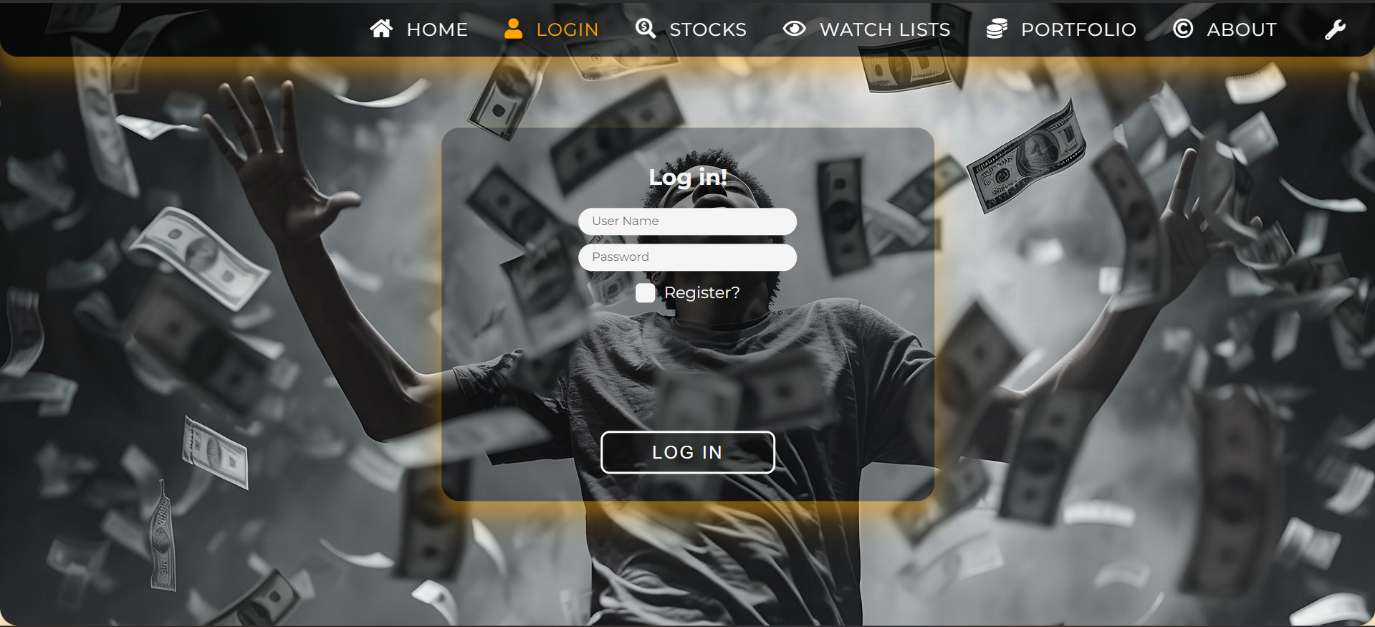
The design of our system followed the n-tiered style within the compound components. We utilised the n-tiered style to follow the business model as there were multiple layers that fell into different areas of the business model. This would allow us to separate into clear boundaries what components would fall into what part of the business concept.

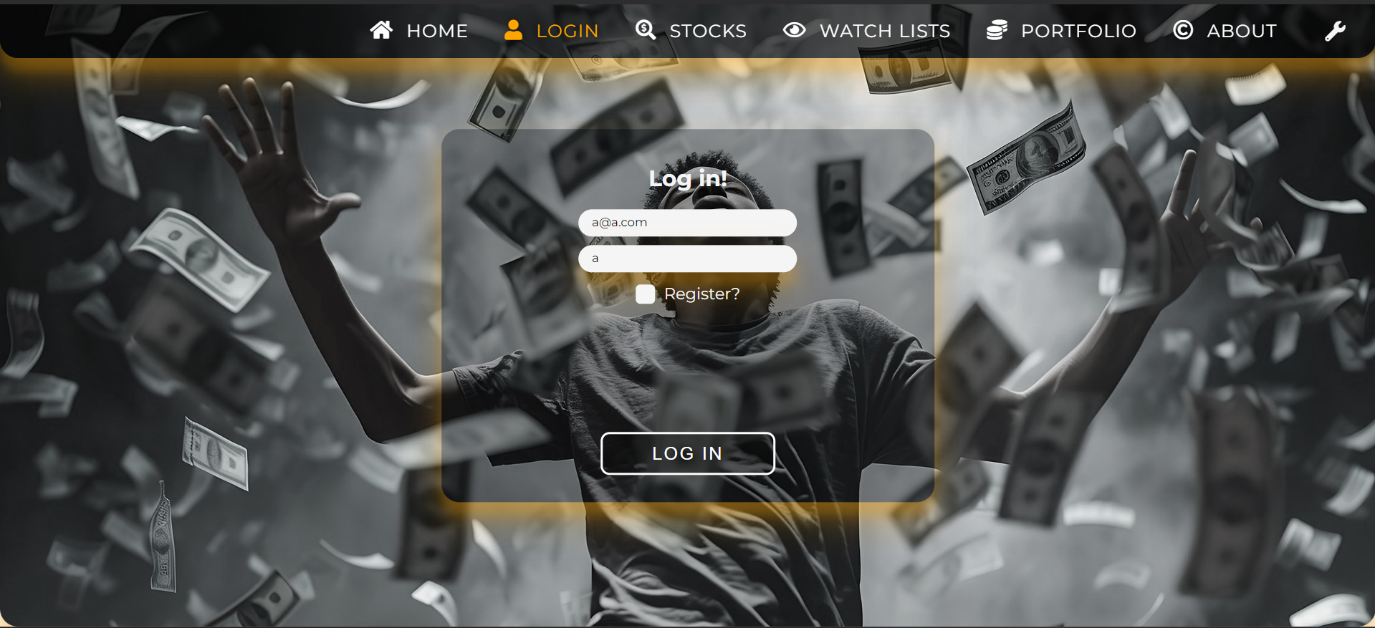
The compound components used were Stocks, Database, Graph, and User. There are outlying subcomponents that did not belong in the compound components, Notification System and Stock Site. The StockInformation subcomponent in the Stocks component gets data from the StockSite, in our case yahoo finance, and then temporarily stores the data. This information is then transferred into the Database component using the IGetStocks interface. The information is then permanently stored within the Database using the Update database class. Once the database has been updated the information is then retrieved using the retrieve database class. There is a split in the interface to notification system and the Graph component. When the data is sent to the notification system, it is checked to see if the information has reached a certain price to then notify the user if they should buy the stock based on a set level. The Graph component gets the data from the database and filters it by using the Graph Filters compound. This filtered information is sent to the Plot Graph component, this then plots the graph for the users and sends the data to the GUI. The GUI then shows the user the frontend code and is the interactable part of the code. The GUI then connects the Login, User Preferences, Portfolio, Watchlist, and Stock Options components. The user is then able to visualise the data and access what stocks they desire.

The architecture used in the system uses the main three components and then centralises them under the core management system. This core management system is the logic behind all the components. The core management system takes all of the data from the manager classes through interfaces and then sends the information to the core management system to then process the data.

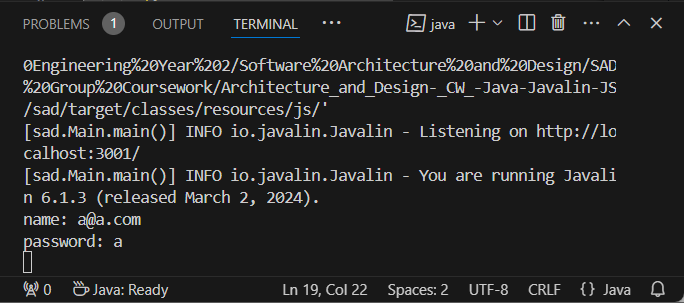
The team meetings occurred three times a week since the beginning of the semester. They would be every Sunday (7.00pm), Wednesday (10.30 pm), and Friday (9.00 pm). No one missed a meeting during this. The first week of meetings were to get to know each other and see what everyone’s strengths were and what everyone was happy doing. The second week focused on architecture and how group members were getting on with their work, the third week was finalising the GitHub setup and the code of conduct as well as the architecture. Sprint 2 commenced on week 4 and we hadn’t been given much information how to start the work, so we started planning everything that week, week 5 started focusing on the business concept model and anything extra we could start working on. Week 6 carried on with the development of the business concept models and initial system architecture and focused on cleaning up the Java code following SOLID Principles, week 7 and 8 was spend finalising designs and development of software to provide a very basic terminal interface. Week 9 was spend trying to develop the advanced compound component styles alongside week 10 to ensure that we were happy with the style the architecture has gone in.

Test Case 1: Logging in with email “a@a.com” and password “a”.

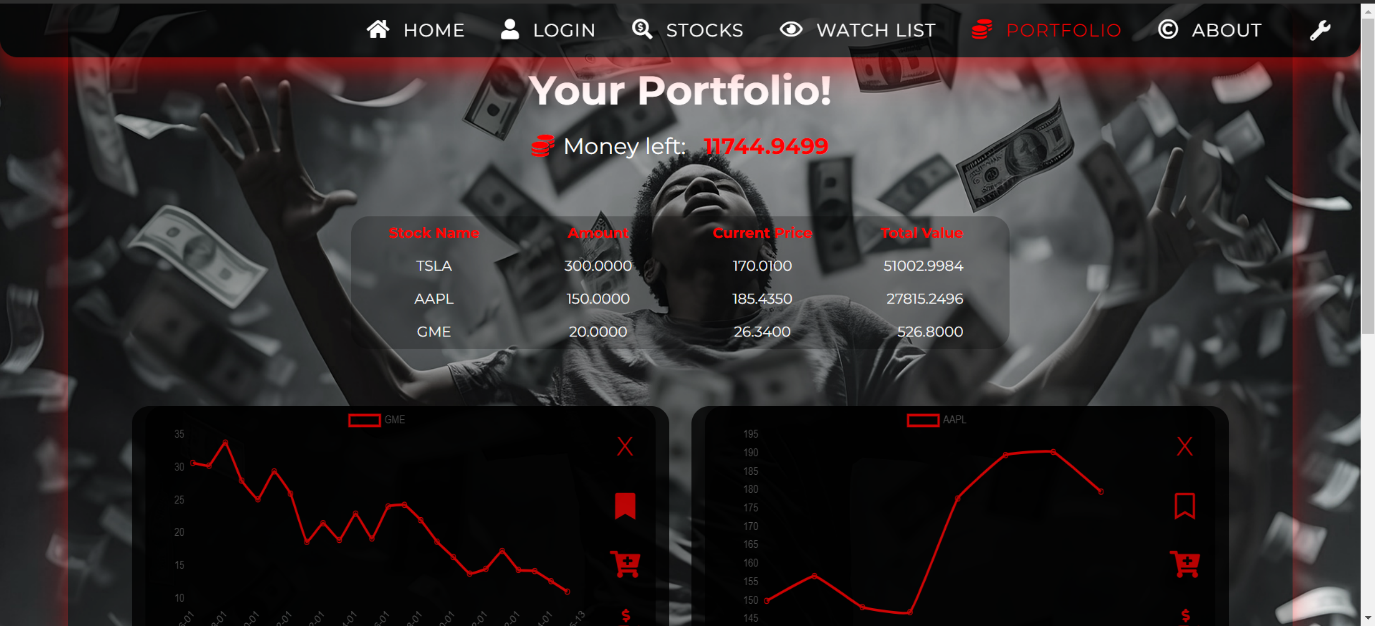


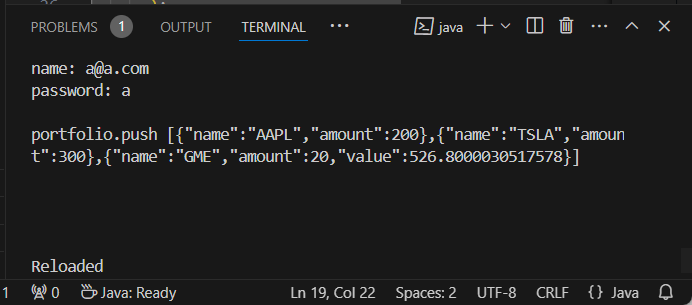




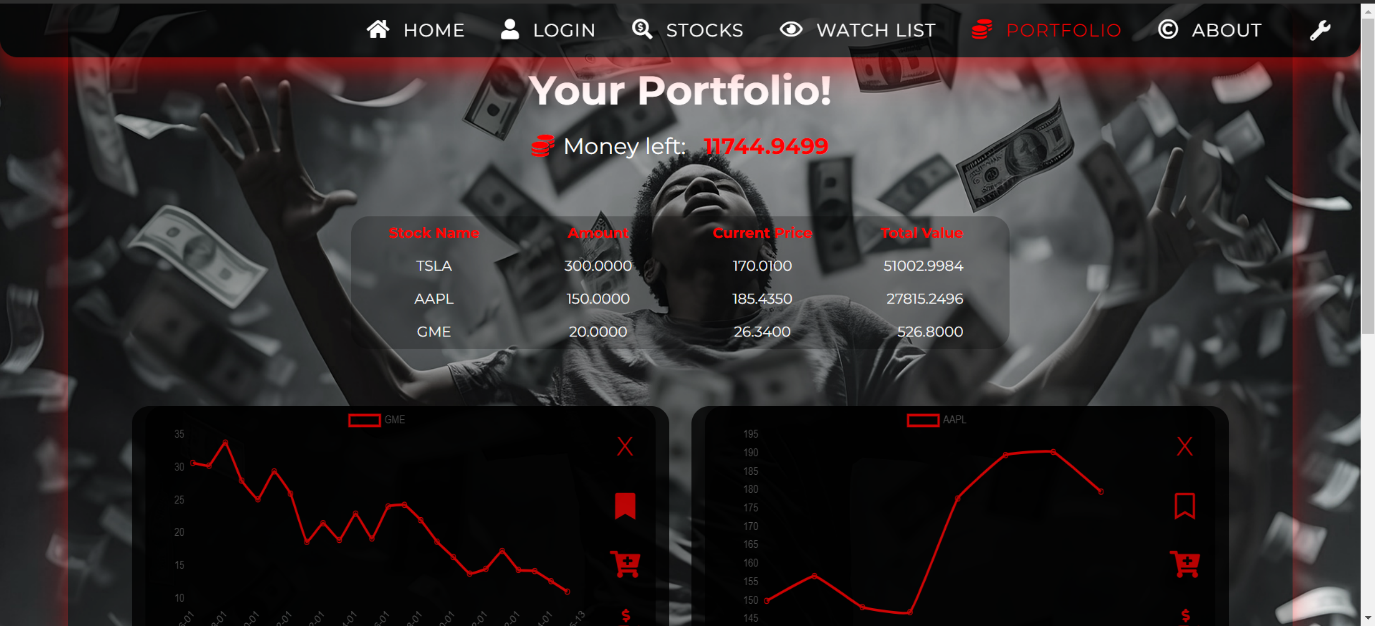


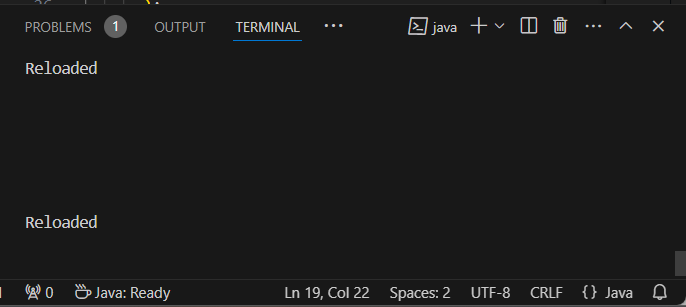
Test Case 2: Buying 20 GameStop (GME) stocks.





Test Case 3: Selling 50 Apple (AAPL) stocks.

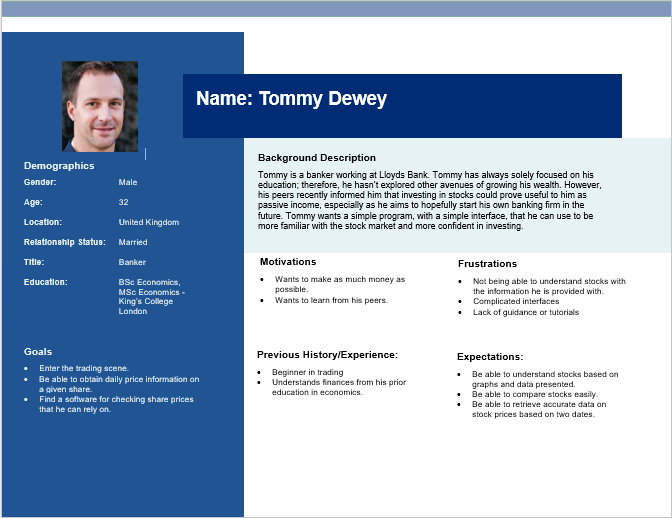


****

A screenshot of a computer

Description automatically generatedA person in a military uniform

Description automatically generatedA blue and white card with a person's face

Description automatically generated****