**Scenario**

Our school has a year-long stock investment project, which every student who enrolled in economic courses must attend, having $1000 USD to invest. Justin, a friend of mine, is aiming to earn a higher amount of profit so that he can earn his pocket money for purchasing games from this project. However, it seems hard for him to decide to sell or buy stocks just with a single graph on websites such as yahoo finance. So, I offered him a plausible possibility by developing an application that would allow Justin to track his stock records and offer him suggestions of when to sell or buy based on statistical algorithms.

Justin agreed with this idea, which we spend another 30 minutes on WeChat messages and phone calls to make up with a blueprint. Both as students who want to minimize the time spent on projects, we agreed that information must be simple but direct. I suggested that the account should be secured well enough so that only the owner can change the purchase list on the application. Justin suggested adding visual representation such as bar graphs to show individual’s performance on the project.

We then discussed the format of this application, which we agreed to let the application be a simulation rather than a well-restricting time product. This is due to the consideration of students might not track their stocks on time while time lag exists when students communicate to school to purchase the stock.

**Rationale**

Java programming language was chosen to develop the frontend of the application. Not only because it was my most familiar programming language, but also because of the Oriented Programming Feature it has that fits my topic. There are inbuilt GUI features within java and has better performance speed compared to Python while also handle garbage collection that programming languages such as C++ doesn’t have.

Backend, however, uses python is used in this application. Even though statical algorithms are involved, languages like R are not used in this application due to the complexity of data not that huge and complex that need an actual statical analysis. In addition, one feature of this application is scrapping from databases to get the stock data that we need, which Python has inbuilt request library that communicate with those database APIs using HTTP request.

CSV files are also get used for data storing. Since many of the features are based on stock data and its communication, CSV file has the advantage of lightweight, easy to read, and better compatibility compared to building a new database.

**Success Criteria**

1. User can successfully create and log-in/log-out to an account with notification of “login/logout successfully”, or notification of error there is incorrect input of password or account when login. The account should have $1000 default cash flow available.
2. Menu page is shown with data performance of stock value’s change compared to last day, with sections such as the account’s list of purchased stocks, search bar, cash flow, overall stock value, view performance, and download performance.
3. Stock can be searched by stock code or by its name and be added with the input of number of stock and notifies the user to confirm the information for purchase or shows error message of account not having enough cash flow.
4. Stock being purchased can be find in the account’s stock list with names and the amount being purchased by clicking on purchased stock button. Each stock should have purchase, sell, or history function.
5. User can sell the stock that is below or equal to the amount of available number of stock and receive the cashflow with that value according to the login day to the account. User can add the number of stocks they want to purchase and deduct the cashflow if its available, otherwise display error message of having not enough cashflow.
6. The stock prices should be changed according to the login date of the system which update once daily, and user can track history record of a specific stock with an input of a specific day.
7. The user should be notified if one stock’s value has been decaying or growing consecutively over a certain amount of time (3 days) when adding/purchasing/selling the stock.
8. When view performance is clicked, analysis such as money spent on stock, stock value, Return of Investment (ROI), and available cashflow should be displayed. User can also click in detail (except the cashflow) of individual stock’s performance of the 3 categories.
9. A bar/line graph is displayed to show daily/monthly performance of stock ROI of the account
10. An excel spreadsheet can be executed to show the stock price history when pressed download performance with data of individual day’s ROI and stock value with their changes