Yilin Wang

CS 15-112

15 April 2020

Project Proposal

**PROJECT DESCRIPTION**

The name of the project is “Trading Simulation”. It is a trading app that allows the user to access information of financial assets (e.g. price, sentiment analysis about the stock, and corporate finance information, etc.) and to construct a user-specific portfolio to simulate the trading process.

**COMPETITIVE ANALYISIS**

Similar applications that I’ve seen online are rather simpler than this app. Most of them consist of two functions, one that allows the user to access the stock price and the other allows the reader to buy and sell stocks.

This app will be similar to those apps, but it will be more complex as it allows the user to customize their analysis (for example, the user may want to add a moving average line to the plot, or the user might access the trading volume of the stock). Also, the app will include more graphical presentations (e.g. the sentiment analysis and portfolio part will include pie charts, and there will be more graphical features to add).

**STRUCTURAL PLAN**

After the app started, the user will be directed to a start page where the user is asked to register or login. Different popup information will occur depending on user input (e.g. wrong password, no user found). When the user info is correctly entered, the user will be directed to the main page, where the user will choose between 5 parts (Price info, sentiment analysis, corporate finance, modify portfolios, and view portfolio). There might be more features to add after MVP (e.g. I might include an option trading page or bit-coin specific live price chart).

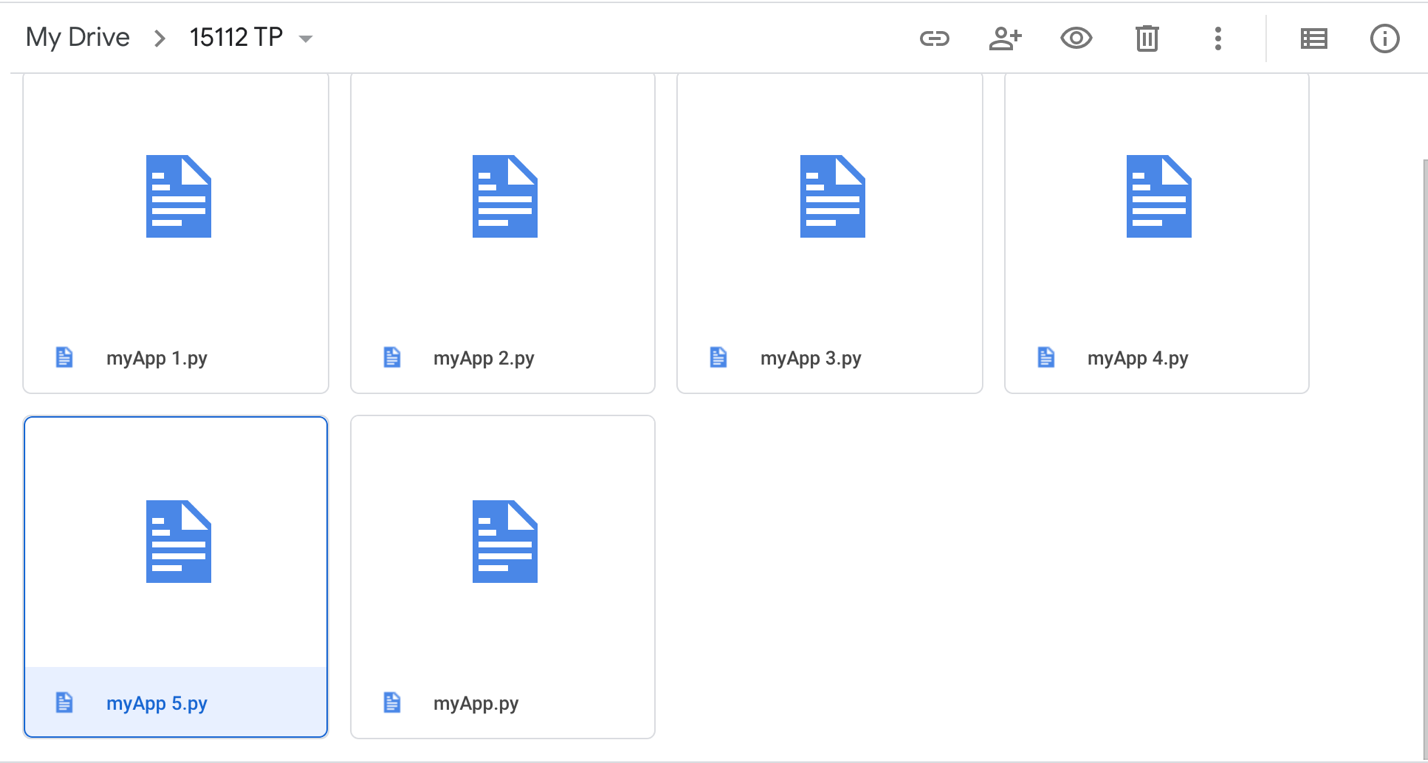
In the price info page, the user will be able to see and customize the price chart. The sentiment analysis page will show the recent 100 news about a company of the user’s choice and give statistical visualization of the composition of news. The corporate finance page will allow the user to see the financial statement of a firm at specific year/seasons of user’s choice. The user may modify and view the portfolio at the last two pages.

**ALGORITHIMIC PLAN**

I think that the complexity is evenly spread out for this project. But I would say that the login page (mostly about how to transmit information among frames) and the sentiment analysis part. I used the tk.TopLevel object to take user input and show it in the frame. I used a recursive solution in the login page to make it more natural. For the sentiment analysis part, it is hard as most financial websites are very protective of their data and therefore made it very hard to webscrape the data. To solve that issue, I found a relatively simpler website and scrapped news headlines according to a special pattern I found in the originally scrapped website data. The sentiment analysis tool that I used, thought, is not very accurate. Therefore, I might implement my own ML algorithm after MVP, and that will be the most complex part if that happens.

**VERSION CONTROL.**

I back-up my file using Google Drive (but more frequently, WeChat files). I save the major version on Google Drive and save minor changes in WeChat files.



**MODULE LIST**

I have been using the tkinter as the main module. For webscrapping, I used the beautifulsoup and requests modules to extract the data from the internet. I used yfinace to get the stock price information. I used matplotlib on some graphical presentation (though I might replace it with tkinter canvas). And I also used pandas to handle data and vaderSentiment to do sentiment analysis (might be replaced with self-implemented ML algorithm).

Some trivial modules that I used includes os (to write files), PIL (to present image), and webbrowser (to open websites). They are very easy to use and is very often one-time-use only.

**COMPLETE MODULE LIST**

* Matplotlib
* Tkinter
* PIL
* Urllib
* Yfinance
* Requests
* Pandas
* bs4
* lxml
* vaderSentiment
* webbrowser
* io
* os