MaxTrade

Independent Study Proposal

Max Weinreb

Objective

The objective of this project is to create software designed for advanced stock analysis and trading automation, and then shipping it out to consumers. The name of the software will be MaxTrade. In order to accomplish this goal I will need to be timely, work hard, and focus on reaching every goal which I plan. The major features which I plan on implementing include but are not limited to:

- The ability to look at a price graph of a user selected stock over a specified time period.
- Overlaying technical indicators on the price graph. The technical indicators which I will implement are moving averages, RSI, and EMA.
- Add a volume subgraph below the main price graph.
- Implement a live news feed for each ticker, which will be compiled from multiple sources.
- On top of this GUI interface which the user will interact with, there will be a “stock scanner” running in the background locally which finds entry points for shorting or opening a long position with a stock. It will be based on user requirements which they choose.
- Finally, certain brokerages can be connected to the software (currently it will be just robinhood) to automatically execute trades. This will be the “trading bot” aspect of the software. Based on certain user set requirements, such as when a stock reaches a certain price, the bot will be able to execute trades automatically.

This project will be primarily coded in python, due to its extensive libraries, as well as the relative ease with which it allows for the creation of GUI interfaces, and because efficiency is not the highest priority. To allow for easy distribution and execution of the software, I will "freeze" the python project into an executable using a library such as PyInstaller. If I have extra time, I will focus some of my efforts on distribution and marketing, as I believe that this tool can be useful to many people.

In order to complete this project in the way that I want, I have to give myself specific, measurable, attainable, and relevant goals to strive for. My major goal is to ship a version of MaxTrade out to at least ten customers, and get their feedback on it. This goal is attainable because I already have an email list of around 100 potential customers who are interested in stock analysis, so it will not be too unreasonable to convince one of them to use my software. Additionally, the software does not have to be fully complete when I send it, it just has to have the core features.

Another major goal which I have for this project is for MaxTrade to become the trading software which I use day to day, specifically the trading bot aspect of it. I have wanted to use a trading bot with my broker so I could automate some trading processes, and I could never find one, so creating and utilizing this feature will be very helpful to me. This goal is measurable, because I can see how often I use my software in day to day life. It is attainable because I will just need to focus on implementing the trading bot feature. This goal is relevant because it is a good indicator of the progress of the software if I begin to use it myself, and is an important milestone to reach.

A minor goal I have for MaxTrade is for its GUI design to look modern and visually appealing. While this aesthetic goal will not affect performance, customers are much more likely to use software if it appears to be professional. I can accomplish this by using python libraries which are effective at creating high quality interfaces, buttons, and text fields. A strong gui toolkit I've found is PyQT.

The two major components of the analysis aspect of the software are the UI and the backend. Initially I will host the backend on my computer, but as I expand, I will move to AWS or Azure. For pulling data for technical indicators and prices, the major source of stock data which I will use is the pandas datareader library, which will allow me to access data from numerous sources such as Yahoo Finance, Google, or Morningstar. Using the basic data I receive from here I can calculate all technical indicators, and store them on the server where they can then be fetched by clients. The live news feed will be compiled from pandas datareader, and other apis such as stocknewsapi, or bitvore.

The second major component of the analysis is the UI, which will take the information collected by the backend and display it in a visually appealing and useful manner. The main GUI frameworks I am considering using are Tkinter, PyQT, or Kivy. The GUI will be responsible for embedding the news feed for each ticker, creating complex graphs, and allowing for customization of search parameters. The way which I plan on structuring the GUI is using three tabs. One tab will contain the analysis portion, with a search bar where the user can find information on available stock tickers. This will include a main graph window, where the user can add different overlays on the price graph. There will also be a news feed portion next to the graph. The second tab is the stock scanner window, which will allow the user to choose certain parameters to scan for, and after starting scanning will run in the background, creating lists of potential stocks.

Finally, the third tab will allow the user to connect their Robinhood (a popular broker) account to the software to allow for automatic trades. It will then display all positions in an easy to read manner. The operations I have selected to initially implement will be automatic selling positions at specific profit/loss %, as well as buying stocks when they reach a certain price. The way this will be implemented is by using the Robinhood API.

Materials Covered: The major materials covered in this project are very important in computer science. They include database management, network security, general purpose server management and programming, development of GUIs, using APIs, and packaging programs into easy to run executables.

For the server aspect of this project, I will need to learn how to handle user requests to query a mysql database which I set up. The main aspects of this will be connecting the client program to the backend, which will have to supply stock data and other information to the client securely. Since most operations will be done client side, managing the server should not be that difficult, its main purpose will simply be to supply data. I will need to ensure however that all requests are secure, and no client information can be leaked. The most difficult part of this will be ensuring the database is set up in an efficient manner. Since the server will constantly have to supply information and simultaneously update its own database, it will be crucial to ensure it runs efficiently.

The development of a strong GUI is critical to ensuring MaxTrade is successful commercially, and that it is easy to learn how to use. I will need to make custom buttons, layouts, and field entries to make the entire GUI feel natural, and behind the scenes make it work seamlessly with the backend. For python, creating GUIs is not that hard, as there are extensive open source frameworks which I can use, the primary one I've found being PyQT.

Finally, the last aspect which my project will focus on is the client side aspect. This will focus on fetching data from my server using mysql. In python this can be easily done using the mysql.connector module. I will have to learn to integrate the data with the GUI interface to provide the user the best experience possible.

Organization:

Due Date	Task	How to measure
9/26/2020	1. Setup Robinhood API to connect to user accounts. 2. Find all user positions and display them (not gui) 3. Allow for manual input of stop losses and take profit % on each position w/ loop. 4. Sell if less than or equal to stop loss or greater than or equal to take profit.	Connect to my account, and practice with some positions, try 1% stop loss and take profit and see if it places a sell order. Verify that this will work with other accounts.
10/3/2020	1. Transition trading bot into GUI. 2. Add pyQT / qt to project virtual env. 3. Display positions in table format. 4. Allow users to select position from table and set loss and gain,	Confirm with practice trade that it will sell position at certain points. Must have the same functionality as non GUI interface.

	as well as for multiple positions. 5. Show that bot is running with a loading loop. 6. Text box with bot output (“Placed sell order for this position”)	
10/10/2020	1. Add the ability to buy stock/option when the asset reaches a certain price. 2. First do this without gui, create “watchlist” and query prices from yahoo each loop. When it reaches the specified price place buy order.	Practice with tickers like AAPL or MSFT. Place buy order when it reaches a certain price, and see if it executes. Try with multiple tickers as well.
10/17/2020	1. Add bot entering at specific price to GUI. 2. Add text entry box where stock ticker is entered, and additional box for price of entry, as well as whether to buy options or stock.	Confirm that the GUI version maintains the same functionality as before. Place orders and see if it works.
10/24/2020	1. Begin working on the scanner aspect. Add different things which the user can select (Volume, price, IV percentile, RSI, MA) 2. Query yahoo finance to get price data for each ticker, begin with most exclusive factors like price. 3. Print output as “watchlist”	Try a variety of different settings and see how the scanner reacts and how long it takes it to scan each time. Verify all output as being correct.

11/1/2020	1. Add GUI interface to scanner. 2. Add Scanner tab to application. 3. Add sliders for each factor, as well as a text box. 4. Add a scan button, with loading gif. 5. Add output box for results, and option to save to file.	Verify saving feature, as well as all functionality from non-GUI version being maintained.
11/8/2020	1. Begin working on the analysis portion. 2. Set up AWS server 3. Add mysql database to it 4. Begin pulling and storing data in database	Practice querying the database from the client side, and verify that the information is correct.
11/15/2020	1. Create another tab in GUI for analysis. 2. Create a basic graph using a ticker and queried data.	Practice making a graph from the client side and compare it to external sources for accuracy.
11/22/2020	1. Add overlaying MA to graph. 2. MA will be calculated client side.	Practice making a graph from the client side with a MA and compare it to external sources such as TOS for accuracy.
11/29/2020	1. Add Volume subgraph underneath current graph. 2. Add RSI graph option to graph	Confirm accuracy of all graphs with external sources.
12/6/2020	1. Pull news feed from morningstar and place it on GUI interface.	Confirm with practice tickers that news feed is live.

Assessment

The first assessment for MaxTrade is that the trading bot should automatically sell specific positions when they reach a stop loss or gain percentage as specified by the user. I want the trades to execute within one minute, and I can test this feature by setting the stop loss to 0% and seeing how long it takes for the trade to get filled. Out of 5 trades, I expect the average time it will take for them to fill the sell requests will be under a minute.

The next assessment for MaxTrade is that I expect the stock scanning feature to run in under 15 minutes, and produce a list of stocks and their expected movement based on user set criteria. This will be indicative of performance. Additionally, I will verify if the output matches up with other trading platforms data on those stocks. For each stock on the list, I will measure whether or not the scanner accurately predicted the price movement over the course of a week, and create a percentage. My goal is for the accuracy to be between 70-80%.

The final assessment will be checking whether or not the analysis feature is accurate with the data from other stock analysis software. I will pick a ticker like AAPL, add an overlaying volume chart, and moving averages to the graph, and compare its accuracy to the Thinkorswim platform’s graphs. This goal will be met if the moving averages contain the same numbers, the prices match up, and the volume is the same. I want my analysis feature to be very accurate.