Homework 4

1)
Planned Work:

“ Previous checkpoint: At this stage I would like to really hone in and improve the trading strategy (optimization). I would like to analyze data capture in the previous stage (Simple Moving Averages, highPrice, lowPrice) to determine relative lows and relative highs. I would also like to use the previously mentioned data to determine a direction of trend. I will determine entry/exit strategies for the investment strategy at this time as well. At this stage relative lows and highs and trend direction will be available to view once the program is executed.

Adjustment of planned work: Elaborating on relative highs and lows, I would like to utilize the polynomial equation that is formed in the plots to determine relative highs and lows. I would like to implement a log of relative lows and highs. I would also likely adjust the time interval of 15 seconds down to 5 seconds in hopes of creating a quicker reacting bot (unless I find way to keep thread open constantly). I would also like to complete the structural design of the algorithm and its class construction through the factory method to have a more object-oriented design. I will remove the current global variables. **“**

– last checkpoint

2)
Accomplished Work:

- Determine relative lows and highs (CHECK) (I have implemented a method to calculate derivative of a polynomial which reveals its highs and lows)
- Calculate simple moving averages (CHECK) (currently commented out in code but functional)
- Hone in and improve trading strategy (CHECK) (I have really spent a lot of time evaluating a trading strategy. A lot of this time was spent researching and unfortunately that sort of progress isn’t visually noticeable)
- Determine trend (CHECK) (currently commented out in code but functional)
- Determined exit strategy (CHECK) (a generic 1% less than entry price) (But not yet implemented)
- Highs, lows, and average simple moving averages are generated (CHECK)
- Change frequency from 15 to 5 seconds (not yet implemented, might change to 10)
- Complete structural design of the algorithm (not yet complete) (but global variables have been removed)
- Factory method (Initiated not yet completed)
- Created and connected to the backend MySQL database with MariaDB
- Made Significant progress on the PyQt5 GUI for the program
- Implemented a view process dependent on new or returning users (user experience) motivated from feedback from the last checkpoint (api key feedback)

I chose to focus on PyQt5 Gui and backend design for this checkpoint as I believed them to be my largest hurdles. I was unsure how Qt and pymysql might play into my algorithm design and for that reason I chose to postpone completing the structural design of the algorithm until I had a more concrete understanding of the design.

3)

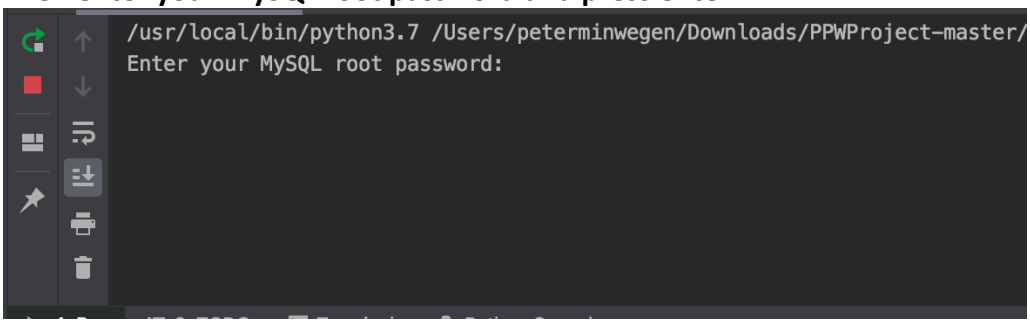
Planned for next deadline: At this stage, I would like to execute dummy trades and store them in a dummy portfolio. Theoretical gains and loss should be available. At this stage, the user should be able to execute the python program and view output that trades have been made. If no trades have been made, no output will be visible. The backend will become connected through PyMySQL and MariaDB which will store executed trades. Construction of the GUI through PyQT will begin.

UPDATED Plan for next deadline: Dummy trades are not the primary focus but will be implemented at the very end. Theoretical gains and losses will also be implemented at the very end. I have already begun construction of the GUI through PyQT5 and I have made quite a bit of progress. I have also already connected the backend through PyMySQL and MariaDB. I need to implement the proper tables for the created database and then populate one such table "Account" when a user confirms. I will aim to complete the home page of the GUI (where I left off, connecting the home page to the commented-out portion of code from the previous checkpoint). I would like to change the proposed GUI home page. I would like the GUI to show a live display of data from the Finnhub.io API feed. I believe the live feed will really demonstrate that the bot is processing live data. I am debating changing the pie graph to a line graph or just having both (option to change) as the center of the home page. I believe it would it could provide more rewarding information to users. I would also like to show a display of trades that have been stored in the database.

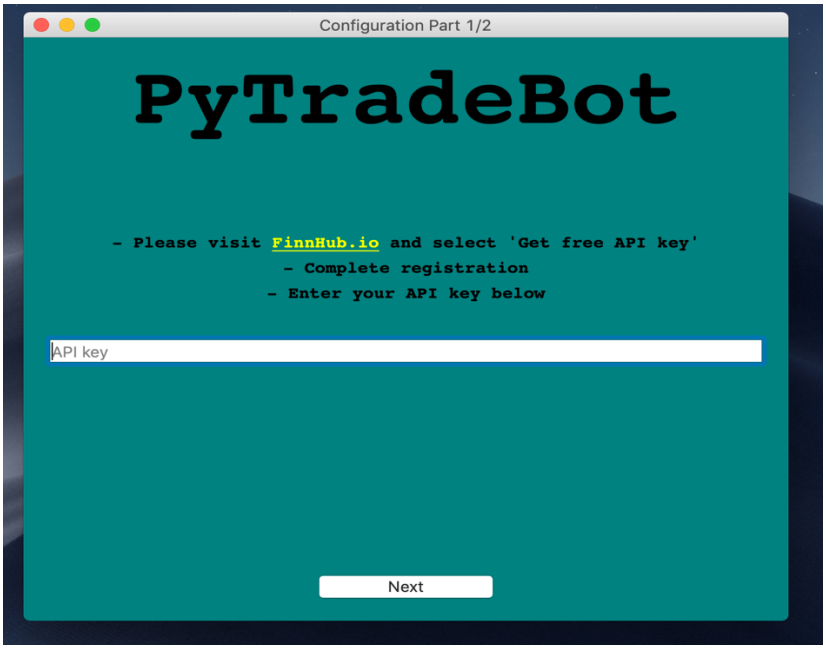
4)

Screenshots:

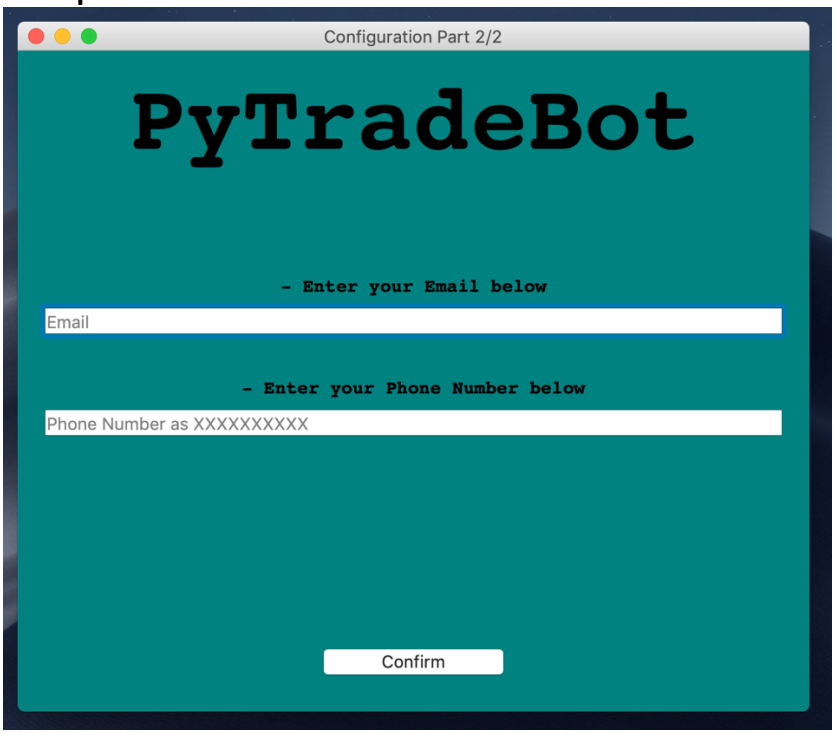
If new user (PyTradeBot.txt does not exist in directory yet)
Then enter your MySQL root password and press enter



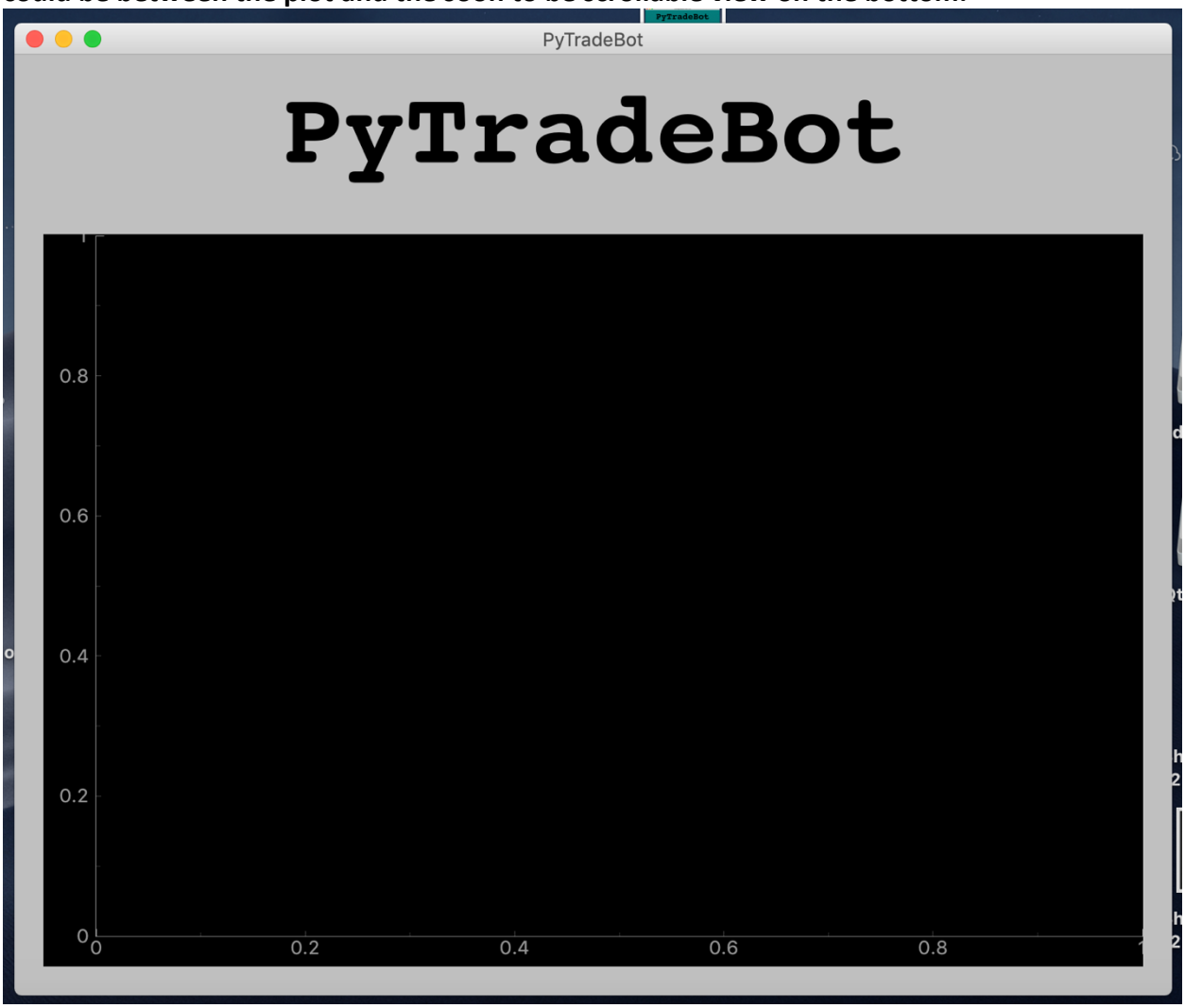
Yellow hyperlink works, can easily register an account and paste api key (key saved in database)
Then click ‘next’



Enter email and phone number (saved in database, to be used for alerting)
Then press ‘confirm’



This is the home page of the application. The plot is not fit to any data. The plot size is too large and needs to be adjusted to a smaller size. Beneath it I would like to place a scrollable data view for the incoming live data. I think a good place for any displayed or labeled data could be between the plot and the soon to be scrollable view on the bottom.



PyTradeBot database created from within script once 'confirm' was clicked in configuration part 2/2

```
MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| Homework4          |
| PyTradeBot         |
| information_schema |
| mysql              |
| performance_schema |
| test               |
+--------------------+
6 rows in set (0.003 sec)

MariaDB [(none)]> Use PyTradeBot;
```

PyTradeBot.txt is also created from within the script once 'confirm' was clicked in configuration part 2/2. This important txt file allows the application to determine if a user is new or not. For returning users, it is used to remember their MySQL root password so they are not constantly asked.