Homework 5 Final Checkpoint

1)
Planned Work:

“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. “

2)

Accomplised Work:

- Dummy trades were implemented with basic trading strategy (buy when decreasing and sell when increasing (every 5^{th} trade detected)) (CHECK)

- The GUI was successfully completed and more elaborate then I had originally intended (CHECK)

- I have fully connected the backend database with MySQL calls to create tables, insert rows, and read rows (CHECK)

- I successfully displayed traded stored in the database upon a click of a button ‘show buy trades’ or ‘show sell trades’ (CHECK)

3)

What I had planned but did not fully implement:

- I planned to implement more than just a singleton class structure but had complications with accessing stock data from time to time which drove me to focus on the cryptocurrency Ethereum due its 24/7 trading hours.
- I had also planned to separate the introductory widgets from the home widget into their own .py file to separate the introduction process and the normal user process into two disctinct python functions that would work with each other.

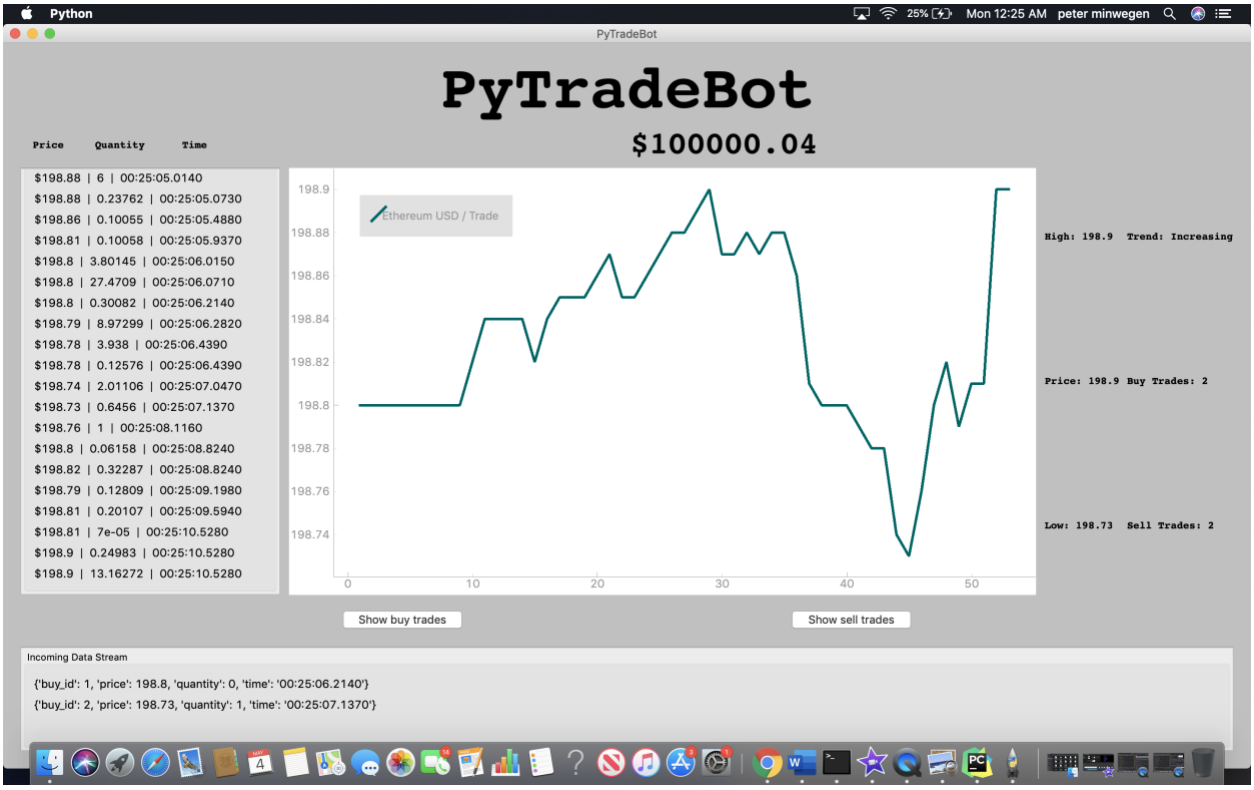
If I had more time:

- I would have liked to implement more commenting throughout but I did try to implement comments when I thought it beneficial.
- I also believe I could have made the code more efficient. One such possibility is that of using dictionaries rather than lists to alter the runtime from O(n) to O(1).

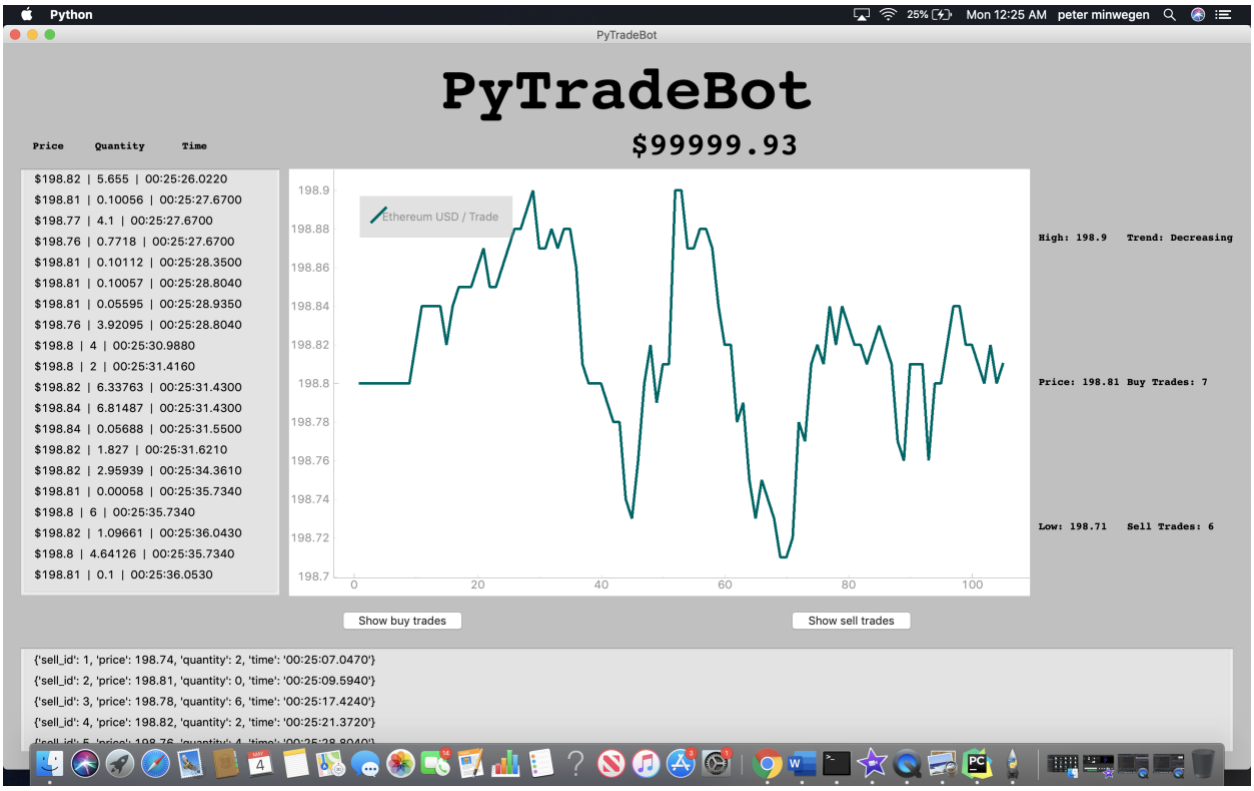
4)

Screenshots of final work:

(buys displayed from database)



(sells displayed from the database)



```
MariaDB [pytradebot]> show tables;
+----------------------+
| Tables_in_pytradebot |
+----------------------+
| account              |
| buy_trades           |
| sell_trades          |
+----------------------+
3 rows in set (0.000 sec)

MariaDB [pytradebot]> describe account;
+------------+--------------+------+-----+---------+----------------+
| Field      | Type         | Null | Key | Default | Extra          |
+------------+--------------+------+-----+---------+----------------+
| account_id | int(11)      | NO   | PRI | NULL    | auto_increment |
| phone      | varchar(11)  | NO   |     | NULL    |                |
| email      | varchar(100) | NO   |     | NULL    |                |
| wallet     | int(9)       | NO   |     | NULL    |                |
| apiKey     | varchar(100) | NO   |     | NULL    |                |
+------------+--------------+------+-----+---------+----------------+
5 rows in set (0.058 sec)

MariaDB [pytradebot]> describe buy_trades;
+----------+-------------+------+-----+---------+----------------+
| Field    | Type        | Null | Key | Default | Extra          |
+----------+-------------+------+-----+---------+----------------+
| buy_id   | int(11)     | NO   | PRI | NULL    | auto_increment |
| price    | double      | NO   |     | NULL    |                |
| quantity | int(9)      | NO   |     | NULL    |                |
| time     | varchar(50) | YES  |     | NULL    |                |
+----------+-------------+------+-----+---------+----------------+
```

```
MariaDB [pytradebot]> describe sell_trades;
+----------+-------------+------+-----+---------+----------------+
| Field    | Type        | Null | Key | Default | Extra          |
+----------+-------------+------+-----+---------+----------------+
| sell_id  | int(11)     | NO   | PRI | NULL    | auto_increment |
| price    | double      | NO   |     | NULL    |                |
| quantity | int(9)      | NO   |     | NULL    |                |
| time     | varchar(50) | YES  |     | NULL    |                |
+----------+-------------+------+-----+---------+----------------+
4 rows in set (0.003 sec)

MariaDB [pytradebot]> select * from account
    -> ;
+------------+----------+----------------------+--------+----------------------+
| account_id | phone    | email                | wallet | apiKey               |
+------------+----------+----------------------+--------+----------------------+
|          1 | xxxxxxxx | Pemi2427@colorado.edu | 100000 | bp9nhovrh5rf91tippdg |
+------------+----------+----------------------+--------+----------------------+
1 row in set (0.000 sec)

MariaDB [pytradebot]> select * from buy_trades;
+--------+--------+----------+---------------+
| buy_id | price  | quantity | time          |
+--------+--------+----------+---------------+
|      1 |  198.8 |        0 | 00:25:06.2140 |
|      2 | 198.73 |        1 | 00:25:07.1370 |
|      3 | 198.82 |        0 | 00:25:14.6600 |
|      4 | 198.73 |        0 | 00:25:17.6130 |
```



```
MariaDB [pytradebot]> select * from sell_trades;
+---------+--------+----------+---------------+
| sell_id | price  | quantity | time          |
+---------+--------+----------+---------------+
|       1 | 198.74 |        2 | 00:25:07.0470 |
|       2 | 198.81 |        0 | 00:25:09.5940 |
|       3 | 198.78 |        6 | 00:25:17.4240 |
|       4 | 198.82 |        2 | 00:25:21.3720 |
|       5 | 198.76 |        4 | 00:25:28.8040 |
```