Project Software Implementation Document

Virtual Stock Market Application

Group 9: Christopher McCracken, Darren Gabrido, Louis Brian Santos, Tai Martinez

1. Overview

The virtual stock market application prototype is a functioning desktop application that simulates an environment where users can practice and learn how to partake in trading in the U.S. Stock Market. This application prototype will not complete actual transactions of stock trading with real money but rather allows the user to mimic investing without any actual stake. The prototype shall use the World Trading Data API[1] where it will fetch live U.S. Stock Market data to determine the user's gains and losses per stock of companies. Users will be able to continually update and keep track of their trading on their chosen portfolio which is locally saved for that individual user. Users shall also be able to create multiple portfolios while using the application to help comprehend how certain trading factors will affect the user's money in the application. When inputting a stock to trade, users must input the valid company stock symbol which will help familiarize the user with companies. In their portfolios, users are able to view their net worth, net spent, net gain or loss, and the stock names for what they have traded in. This scope will help the user to learn the dynamics of the U.S. Stock Market by familiarizing themselves with how trading is completed.

2. Source Code

Github: https://github.com/Chrithtoph/StockMarketApplication

3. Test Plan

3.1 Unit Tests:

1. Function: Create a default portfolio (Main)

Seq. No	Condition To Be Tested	Test Data	Expected Result	Pass/Fail
1	Equals	Name: 'Test'	'Test'	Passed
2	Equals	netWorth: 0	0	Passed
3	Equals	sharesOwned: 0	0	Passed
4	Equals	netSpent: 0	0	Passed
5	Equals	netGainLoss: 0	0	Passed

2. Function: Get stock price from API (Main)

Seq. No	Condition To Be Tested	Test Data	Expected Result	Pass/Fail
1	Not None	Stock: 'AAPL'	True	Passed
2	Not None	Stock: 'GOOG'	True	Passed
3	Equals	Stock: 'TSLA'	(**Current market value**)	Passed

3. Function: Get portfolio info (Main)

Seq. No	Condition To Be Tested	Test Data	Expected Result	Pass/Fail
1	Not None	Portfolio: Test	True	Passed
2	Equals	Portfolio: Test	Default portfolio values with name: 'Test'	Passed

3.2 Higher-Level Tests: System Testing

1. **Use case:** Buy/Sell Stock

   Scenario: User has internet access, a portfolio set up in the database, and wants to buy shares of a stock

Seq. No	Steps	Test Data	Expected Result	Pass/Fail
1	Open up application		Application displays the main menu	Passed
2	Select Buy/Sell stock button		System prompts user to enter the portfolio to update	Passed
3	Enter a portfolio name	Portfolio = John	System prompts user to enter a stock to trade	Passed
4	Enter a stock to trade	Stock = AAPL	User is displayed the price of the stock and asks how many shares to buy/sell	Passed
5	Enter amount of shares to buy/sell	Shares = 5	Transaction complete	Passed

2. **Use Case:** View Portfolio

   Scenario: User has internet access, a portfolio set up in the database, and wants to view their portfolio

Seq. No	Steps	Test Data	Expected Result	Pass/Fail
1	Open up application		Application displays the main menu	Passed
2	Select View Portfolio button		System prompts user to enter the portfolio name	Passed
3	Enter a portfolio name	Portfolio = John	System displays the portfolio to the user	Passed

3. **Use Case:** Plot Stock History

   Scenario: User has internet access, a portfolio set up in the database, and wants to view a plot stock history for a stock

Seq. No	Steps	Test Data	Expected Result	Pass/Fail
1	Open up application		Application displays the main menu	Passed
2	Select Plot Stock History button		System prompts user to enter which stock to view	Passed
3	Enter stock name	Stock = AAPL	System prompts user to enter the number of days to view	Passed
4	Enter number of days to view	Days = 5	System prompts user to choose a chart type	Passed
5	Enter chart type	Chart Type = Candlestick	System displays candlestick chart	Passed
6	Enter chart type 2	Chart Type = Closing Prices	System displays closing prices chart	Passed

4. **Use Case:** View Detailed Stock Data

   Scenario: User has internet access, a portfolio set up in the database, and wants to view detailed stock data for a stock

Seq. No	Steps	Test Data	Expected Result	Pass/Fail
1	Open up application		Application displays the main menu	Passed
2	Select View Detailed Stock Data		System prompts user to enter which stock to view	Passed
3	Enter stock name	Stock = AAPL	System displays stock data	Passed

3.2 Testing Summary Report:

Project: Virtual Stock Market Application

Purpose: This report discusses the activities performed as part of testing our 'Virtual Stock Market' application.

Review team: Christopher, Darren, Louis, Tai

Types of testing performed: Unit tests, System testing

Testing Scope: Functions in main (create a stock portfolio, get stock from API, get portfolio info), Use cases (buy/sell stock, view portfolio, view stock data, stock history, detailed stock data)

Work products:

Unit Testing: Class Main	
Size of work product	108 LOC of Python
Review Team	Christopher, Darren, Louis, Tai
Effort (person-hours) total	2 person-hrs
Defects	0
Review Status	Accepted
Comments	Code can be improved by breaking up functions, making it easier to test. Difficulty testing live stock data.

System testing: Use cases	
Size of work product	597 LOC of Python
Review Team	Christopher, Darren, Louis, Tai
Effort (person-hours) total	2 person-hrs
Defects	0
Review Status	Accepted

Unit testing Framework:

https://docs.pytest.org

4. Updates / Modifications to SRS and SDD

Reduced Scope:

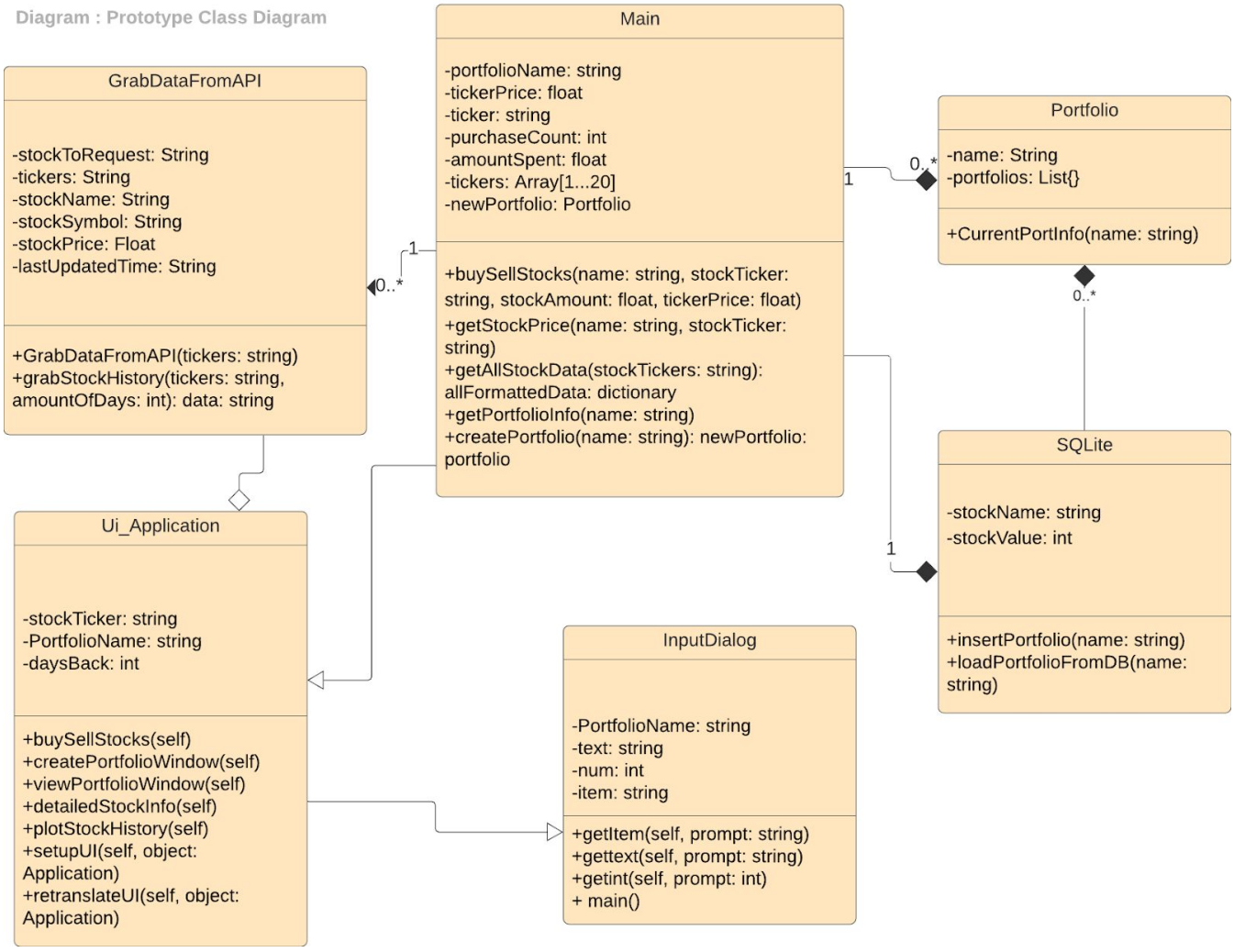
In the UI/UX design images in the project SDD, we had a search bar to search for certain stocks. We instead will have a string entry field where users must input the stock name for the stock they would like to trade.

Redesigned Prototype Architecture:

In the prototype architecture section of the project SDD, it was explained that the database element of the architecture will only support a single user rather than multiple thus being a local

database. For that reason it was also stated that the login username and password will not be encrypted for the prototype. However, due to time constraints this has been updated where the application will no longer prompt the user for a login username and password. Rather than implementing a login username and password functionality, the local database will be accessed through the user inputted portfolio name.

Updated Class Diagram:



5. User's Guide

Introduction: There are several different windows that user can view upon opening the application. This section of the user's guide will give a brief overview of each of those items and then will be followed with a step-by-step how-to. Those windows include:

Create Portfolio: This will simply prompt the user to input a string name for desired portfolio name to instantiate

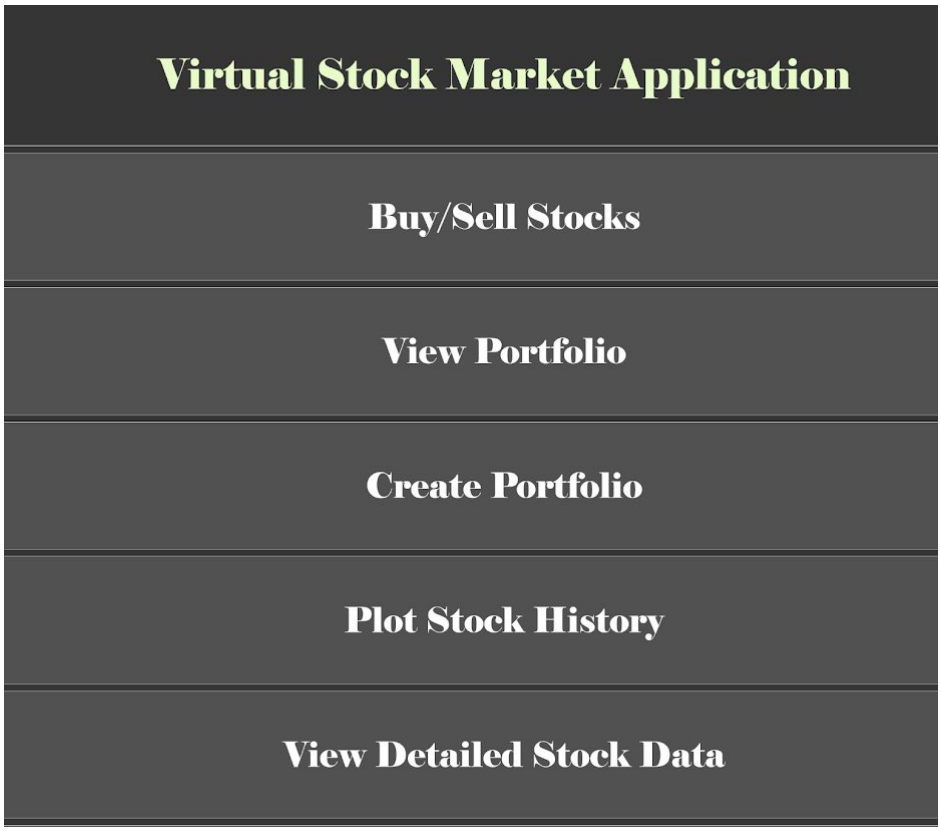
Buy/Sell Stocks: This window will prompt the user to input the portfolio that they would like to trade under and the stock name they would like to trade. It will then display the price of the stock and prompt the user for how many stocks they would like to trade and then whether they would like to buy or sell that stock.

Plot Stock History: From this window, users can input a stock name and the number of days that they would like to view the history for that stock. It will then display a graph for the closing price history or a candlestick graph with the price statistics for that stock over the days selected.

View Portfolio: This window will display the portfolio information for the inputted portfolio. That information will include the name of the portfolio, the net spent (in dollars), the net worth (in dollars), the net gain and net loss (in dollars), the shares owned, and the names of the stocks traded with the number of shares owned for that stock.

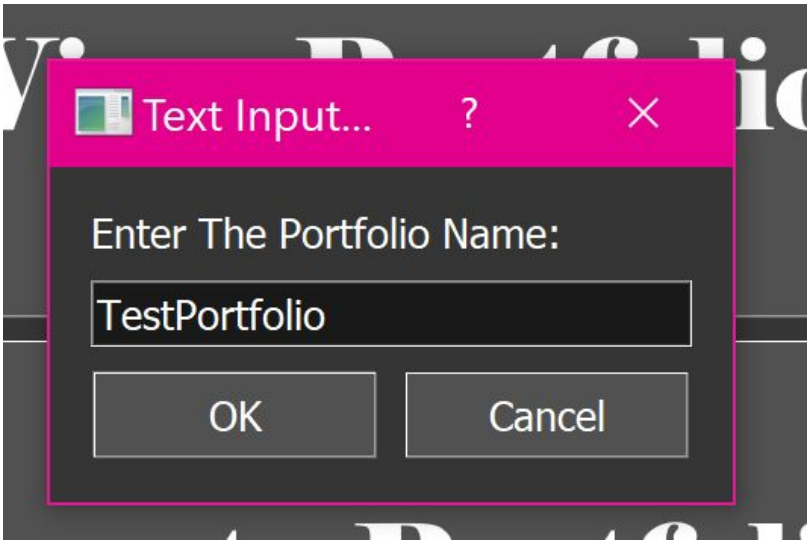
View Detailed Stock Data: From this window, users can input a stock name and it will then detailed stock data for that certain stock.

Getting Started:



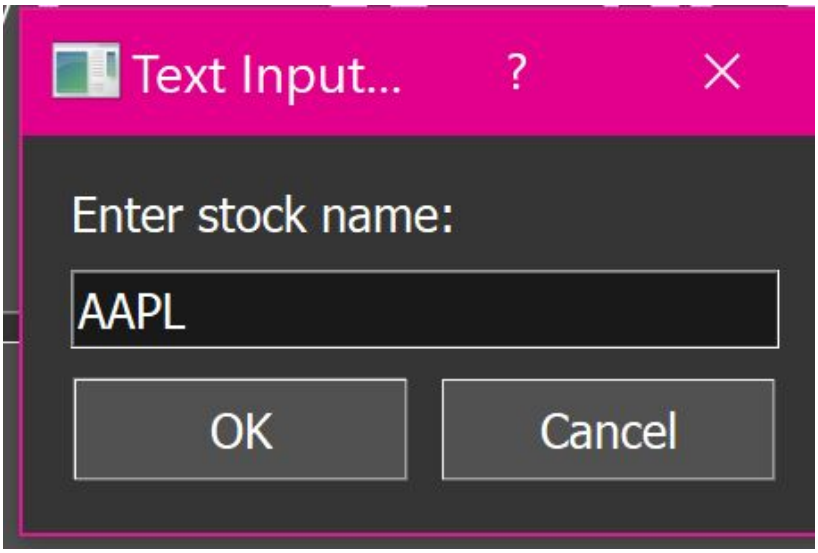
Step 1: When the user opens up the application they will see the following screen with the 5 different button options.

If users are new to using the application and thus have no portfolio, they will need to first click the button to create a portfolio.

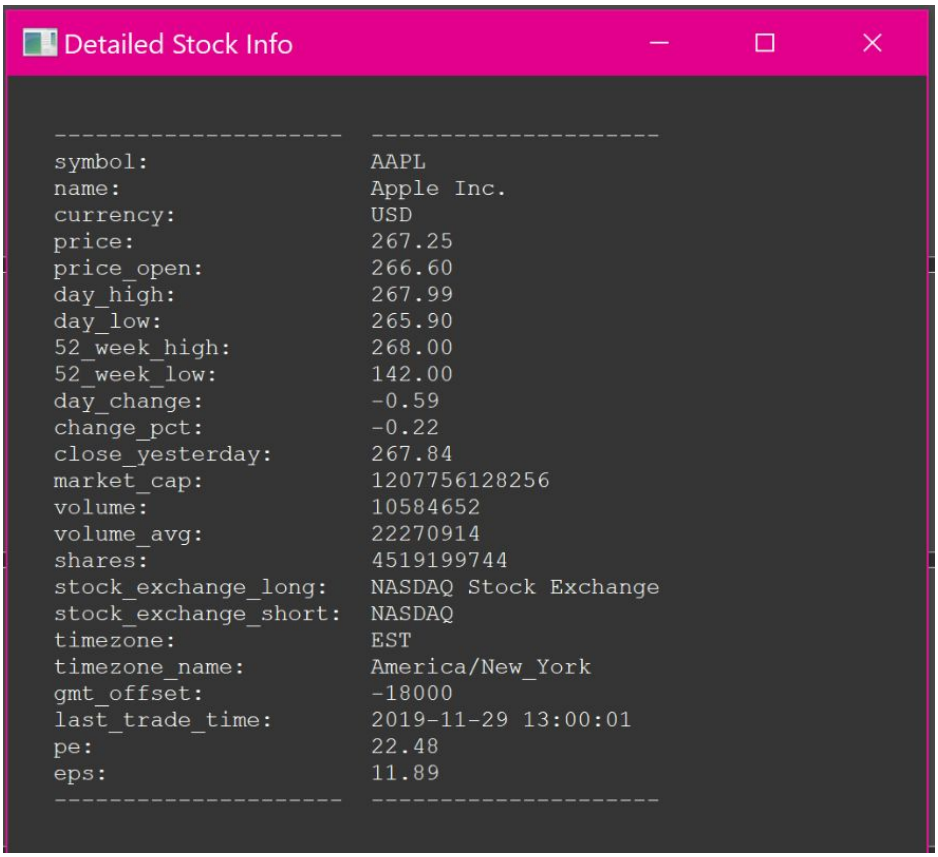


Once users click the Create Portfolio button, they will be prompted to input a name for their portfolio. After clicking the OK button, the portfolio will be created with chosen name.

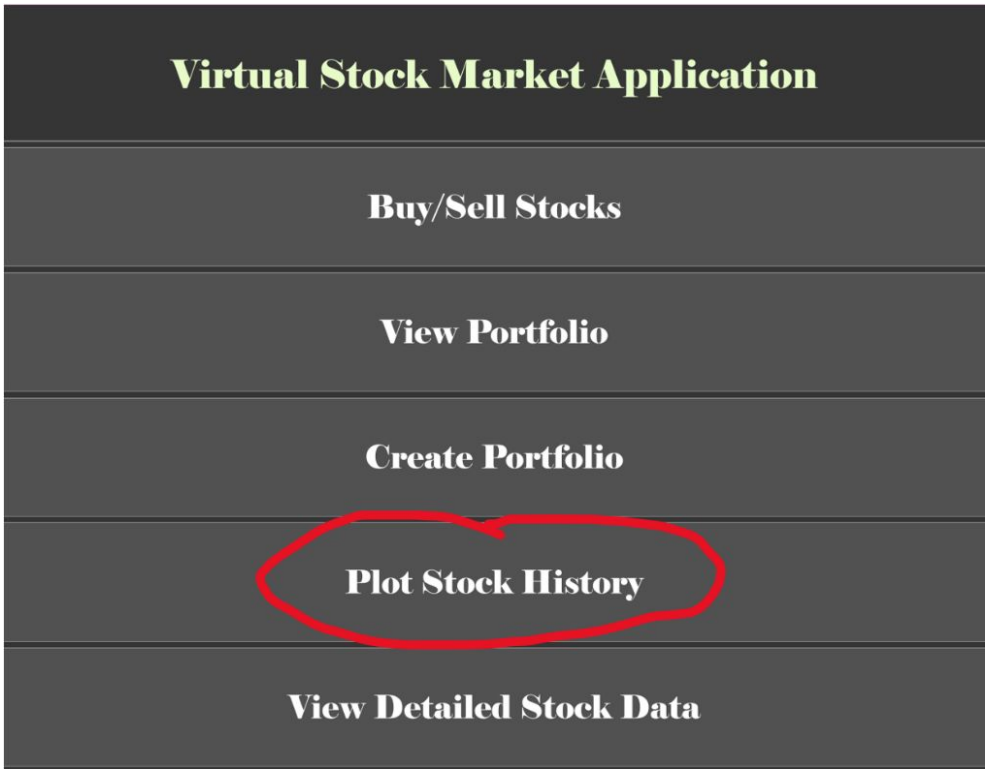
Step 2: Once users have a portfolio, they can now buy or sell stocks. A user may want to look at the history and statistics of certain stocks before investing in them (if that is not the case and users are confident in the company they are trading in, this step can be skipped). If that is the case the user will first click the View Detailed Stock Data button.



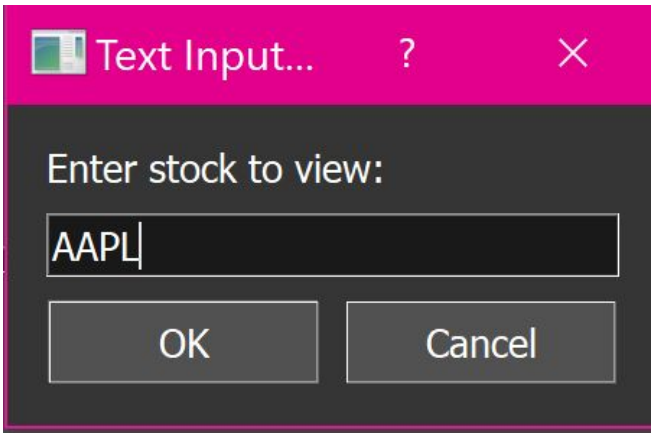
The user must then input a stock name that they would like to view the detailed data for.



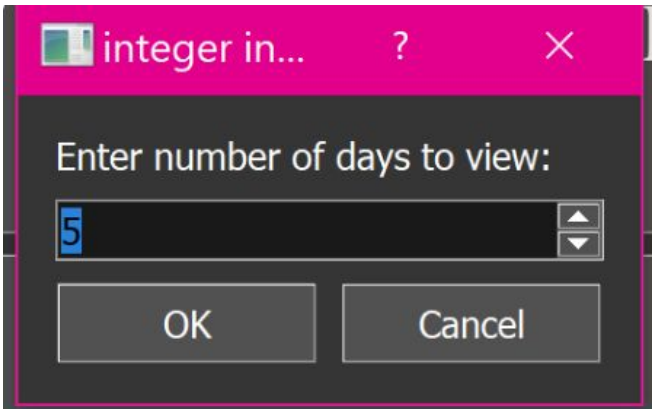
Once the users click OK, the detailed stock data will be displayed for that stock.



After users view the detailed stock data, they can also plot the stock history in graph form. To access this, users will click the Plot Stock History button.

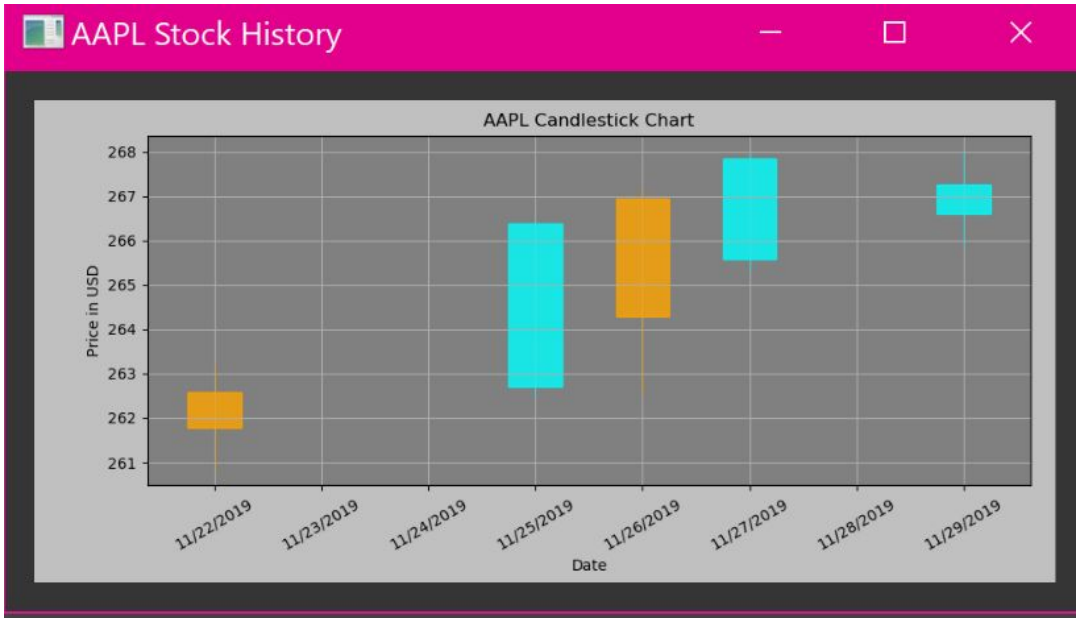


Users must then input the name of the stock that they would like to view.

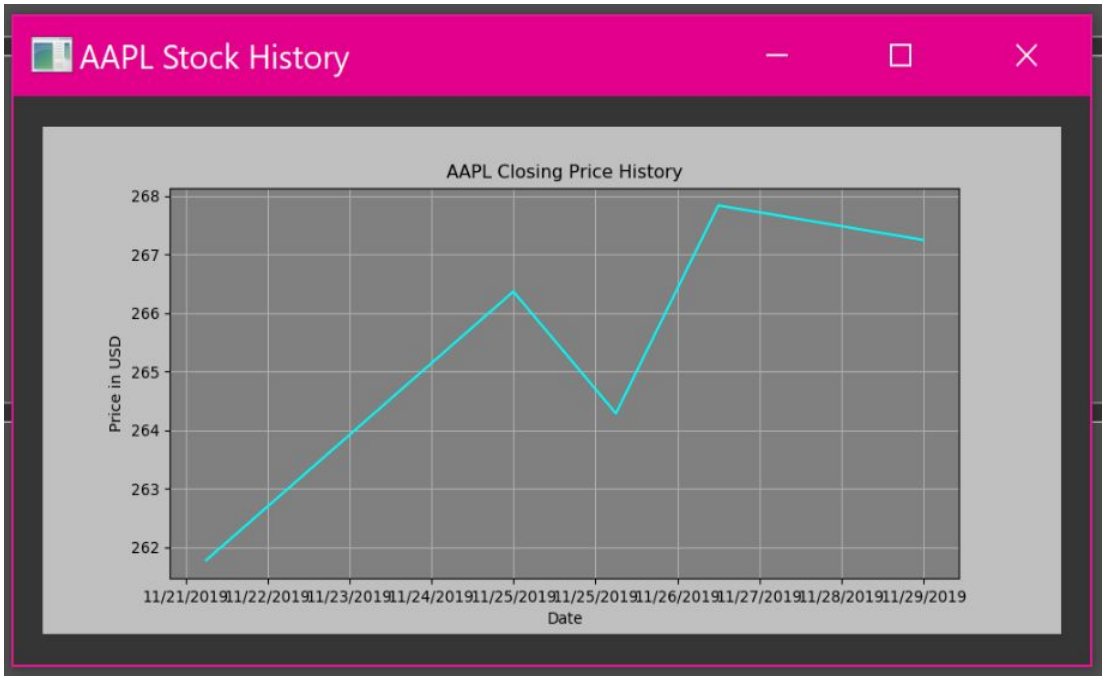


After clicking OK, users then must input the number of days that they would like the graph to display.

After clicking OK once more, users then will choose either a candlestick graph or a closing prices graph from the dropdown menu.

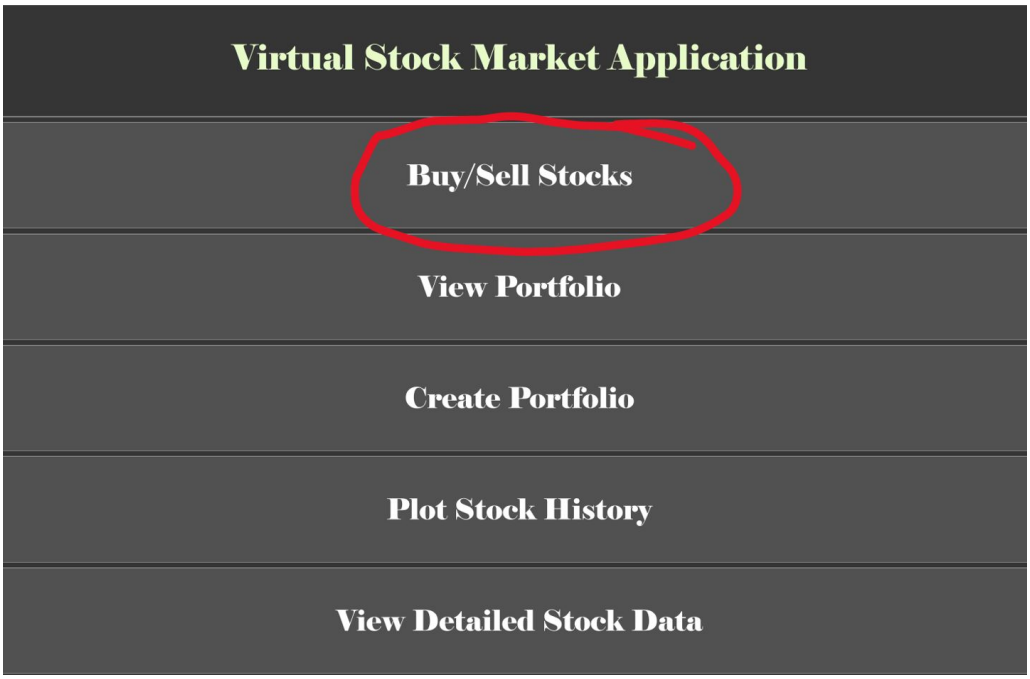


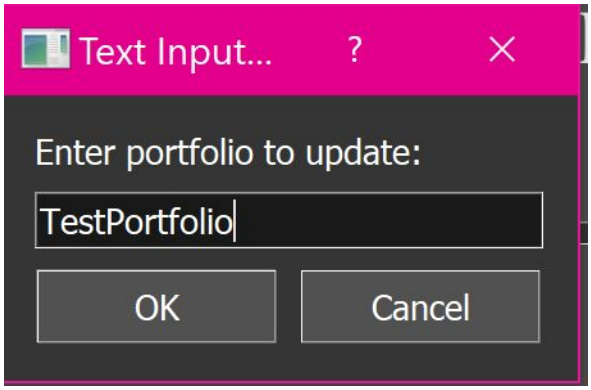
Option 1: Users choose the candlestick option and then click OK which will display a candlestick graph for the stock for the number of days.



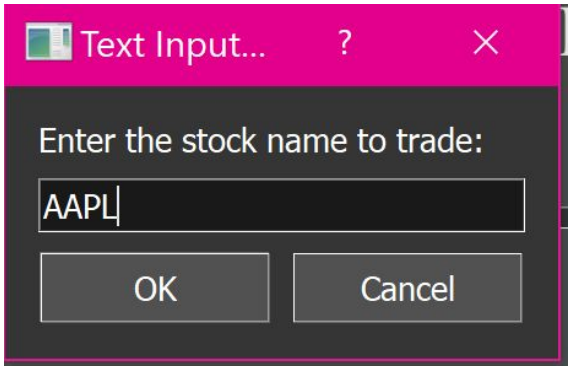
Option 2: Users choose the closing price option and then click OK which will display a closing price graph for the stock for the number of days.

Step 3: After viewing all the stock data that they wish, the next step for users is to actually trade in the stock market in the form of buying or selling stock. Users will do so by clicking the Buy/Sell Stocks button:

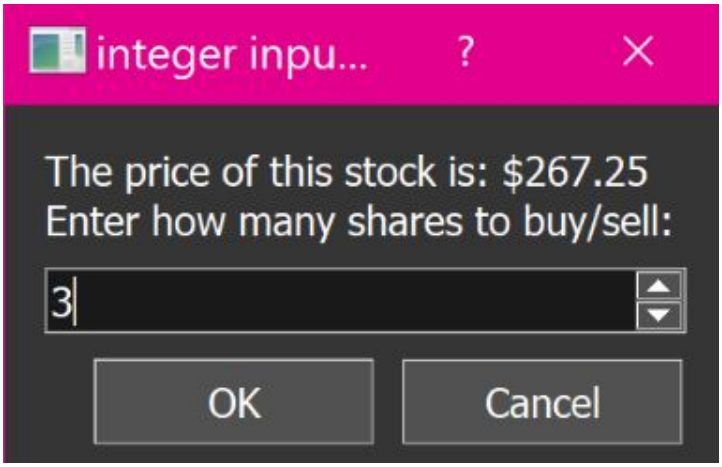




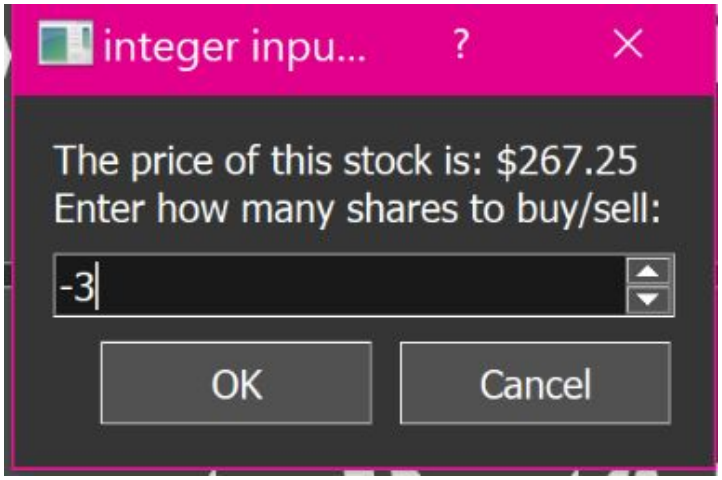
After clicking this button, the user will input a current portfolio name that they would like to trade under.



After clicking OK, users will enter the name of the stock that they would like to trade and then once again click OK

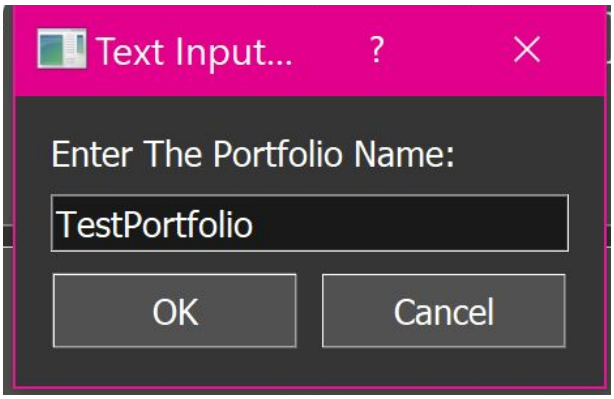


Option 1: From this point, users can buy this stock by entering a positive number for the number of shares that they would like to purchase. Users must click OK to confirm the purchase.

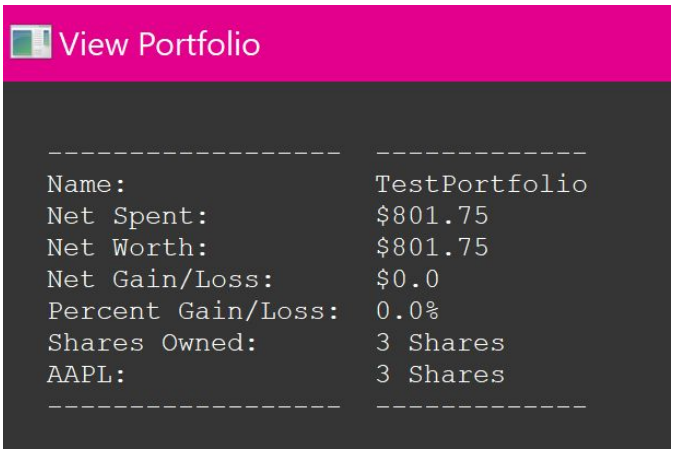


Option 2: From this point, users can sell this stock by entering a negative number for the number of shares that they would like to sell. Users must click OK to confirm the sale.

Step 4: Users can now view their portfolio and its statistics by clicking the View Portfolio button.



After clicking this button, the user must input the name of the portfolio that they would like to view.



Once the OK button is clicked, the portfolio data will be displayed

Step 5: From this point, the user know the basis for how to use the application to create portfolios, search for stock data, buy and sell stocks, and view their portfolio. Users are able to create multiple portfolios to compare stock trading choices among those portfolios. Users are also able to continually check back on any of their portfolios to track their gains and losses as each portfolio will be saved locally on a database.

6. Glossary/References

Non-Standard Acronyms

1. API: Application Programming Interface

References

1. "World Trading Data," *World Trading Data*. . "https://www.worldtradingdata.com/"