FINAL SYSTEM DESIGN REPORT (COMP-195)

ADK Financial Service

(GitHub)

Antonio Barrios - *a_barrios5@u.pacific.edu*
Dalton Coppage - *d_coppage@u.pacific.edu*
Kunal Babbar - *k_babbar@u.pacific.edu*

Last Revised: May 5, 2022

System Architecture

Due to the fact that our team chose to recreate the well-established, classic terminal of trading, it would make sense to review the essential features that will make this Stock Terminal so useful to the users.
The main architectural style of our project is application based. We will be using an application to receive user input. Once inputted, our program will run Python code to scrape, collect, and perform analysis on the given company. The user will only be interacting with PyQt5. The application, however, will perform data analysis on the back-end to provide information.
By reviewing the overall trading system and the latest news in the market, we will be able to translate the key controls as well as put our own spin to the idea to add originality.
We will be using Yahoo Finance to scrap our data for the stock terminal to provide us with the latest financial news, data and commentary including stock quotes, press releases, financial reports, and original content. In addition to posting partner content from other web sites, it posts original stories by its team of staff journalists.
The Python3 Library PyQt5 will be very useful in our code as it is a comprehensive set of Python bindings for Qt v5. It is implemented as more than 35 extension modules and enables Python to be used as an alternative application development language to C++ on all supported platforms including iOS and Android.
The application opens up with a menu for the user to choose from, that allows for the user to enter a company ticker, scrape articles, get data, or exit the application. Once a user clicks from one of the many options it will open another screen that can display statistical information or articles of a company. The Menu screen will be the main screen that will link all other classes together.

- Front-End
    - The front end consists of what the user sees and interacts with. Our front end will consist of the PyQt5 library in Python3. Therefore, Python will be the only main language used to complete the application.
- Back-End
    - The back end will consist of a mix of Python3 libraries and modules like Matplotlib and Pendulum.

Hardware, Software and System Requirements

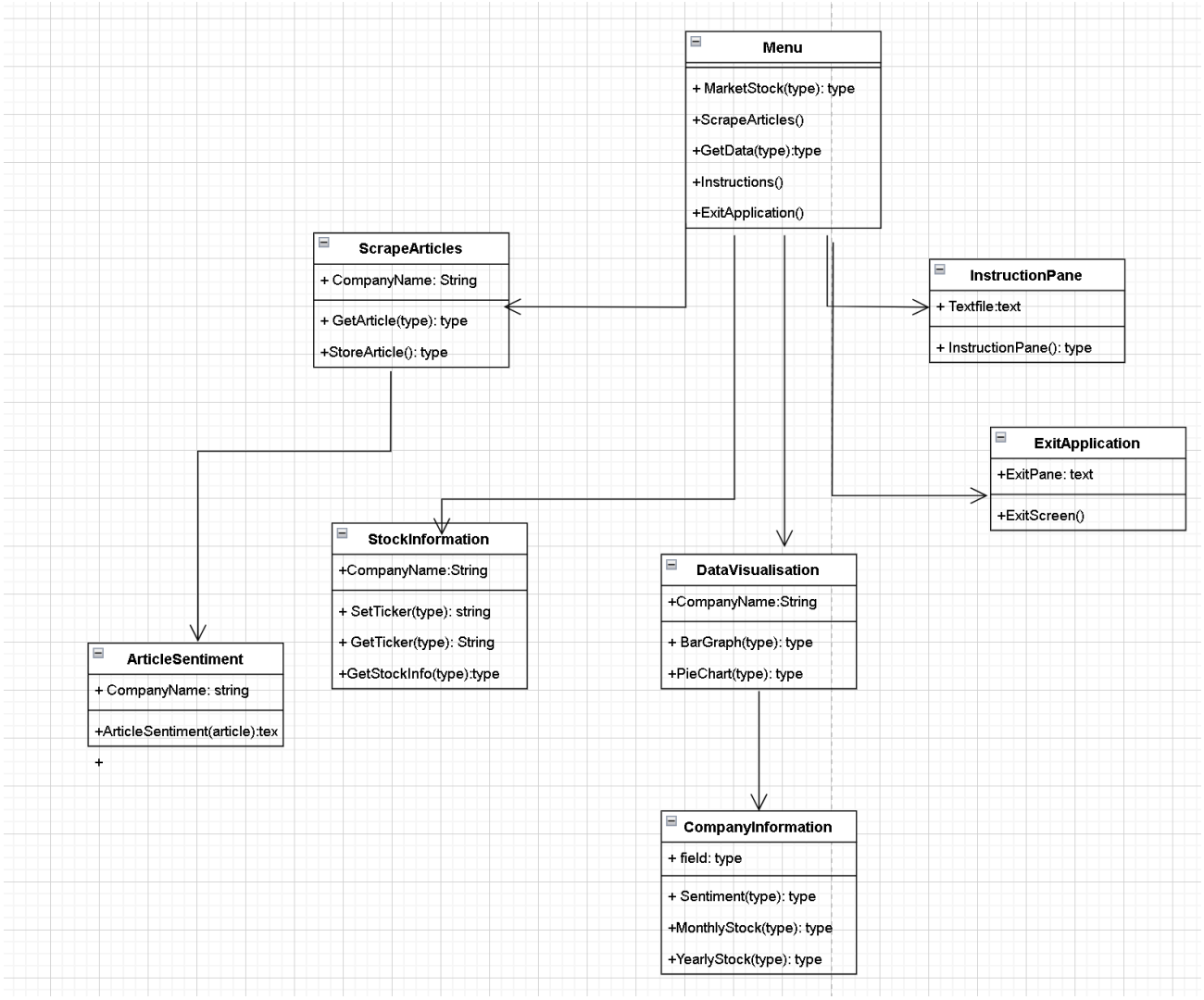
- Display
    - 800 x 600 pixels required
- Storage
    - At least 10gb for this program
- A keyboard and mouse for input
- Any processor

    - Intel i3 processor or above

- At least 4 gigabyte of RAM
- Any graphics processor
- Any Operating System
    - Intel i3 processor or above
- PyCharm's Latest version of Community Edition
    - Python3 or above Interpreter
    - PyQT5 library
    - Yahoo Finance library
    - Matlibplot library

Software Design

Class Diagram



Class Specifications

- Public Class **Menu**
  - MarketStockInfo**()**: Finds market stock information from the yahoo API.
  - ScrapeArticle**()**: Scrape Google and Yahoo news articles to return information about a company.
  - GetData**()**: Gets data to the user about a specific company.
  - Instructions**()** : Shows the user how to work around the stock terminal.
  - ExitApplication**()**: Will exit the program when user chooses to "Close"

- Public Class **CompanyInformation**
  - SetTicker**()**: Saves the company ticker.
  - GetTicker**()**: Gets the save ticker.
  - GetStockInfo**()**: Gets the information that was returned from the user entered ticker.

- Public Class **StockInformation**
  - Sentiment**()**: Gets the sentiment of each scraped news article
  - MonthlyStock**()**: Shows the company stock performance for a month
  - YearlyStock**()**: Shows the company stock performance for a year

- Public Class **DataVisualize**
  - BarGraph**()**: Displays to the user statistical information about company in Bar graph form
  - PieChart**()**: Displays to the user statistical information about company in Pie chart form

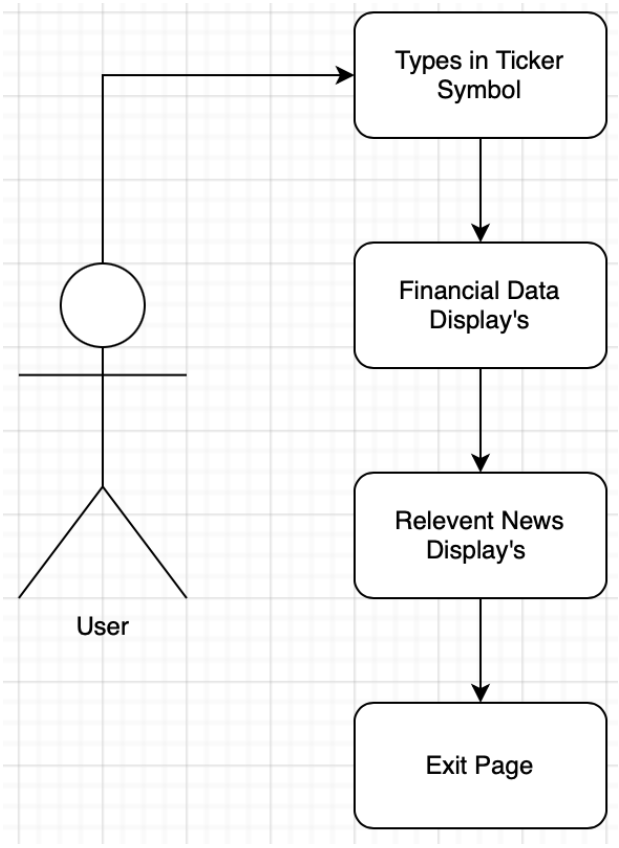
- Public Class **ScrapeArticles**
  - GetArticle**()**: Gets the computed value from ArticleSentiment()
  - StoreArticle**()**: Store the sentiment of the computed value in ArticleSentiment**()**

- Public Class **ArticleSentiment**
  - ArticleSentiment**()**: Computes the value of each scraped article and applies a positive,negative, or neutral value.

- Public Class **InstructionsPane**
  - Instructions**()**: Displays the instructions on how to use the application.

Interaction Diagrams

1. Welcome screen will appear with instructions on how to search a company
2. User clicks on text box and types in a company's ticker symbol
3. The company screen will appear with:
   a. Financial Data - open, close, volume, market cap, beta, and other ratios
   b. News Articles - Professional news article about performance outlook
4. Page Close
   a. Page will close and user is sent back to welcome screen

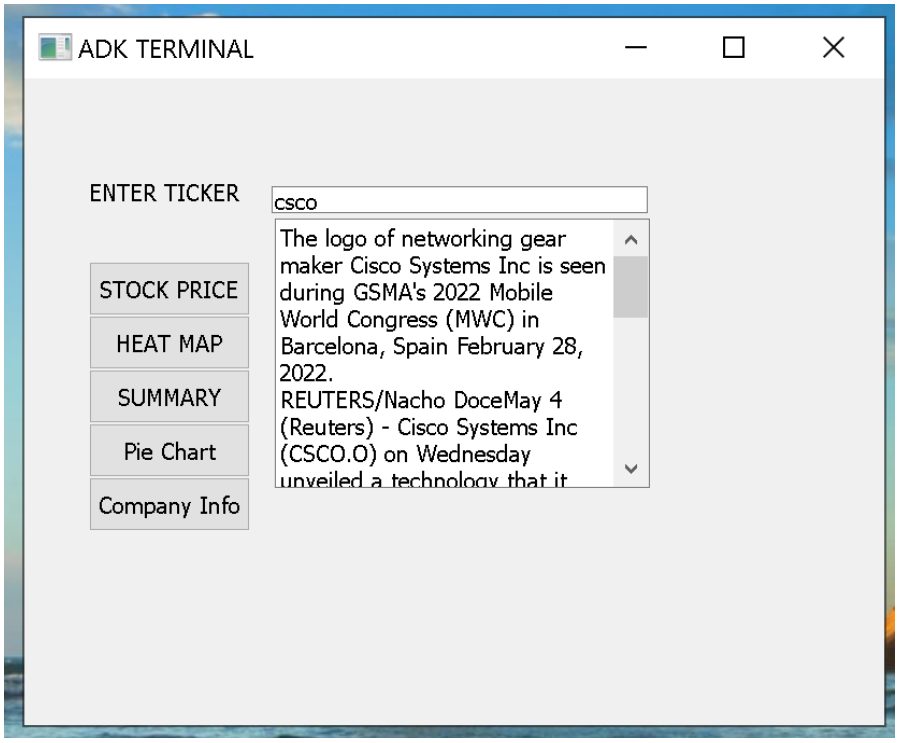


Design Considerations

- For the design requirements of this terminal used something called PyQt Framework - a library that will let us use the Qt GUI framework from Python.
- For all the data gathered using tickers, if we want to plot it, we'll have to install some additional Python libraries to help us out — matplotlib to plot and pendulum to do some easy time conversions for us.
- This GUI will be built to run on a single machine and will not require any external interfaces.

User Interface Design

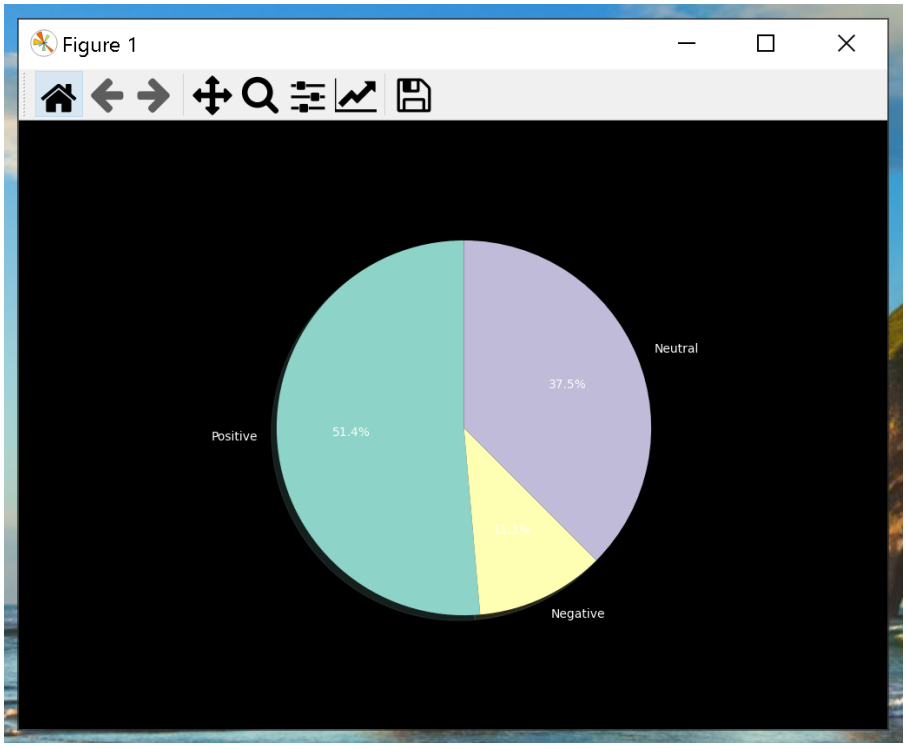
- Our project displays a search bar similar to the image below. As we can see from this snapshot 'csco' is the ticker symbol used to search Cisco.



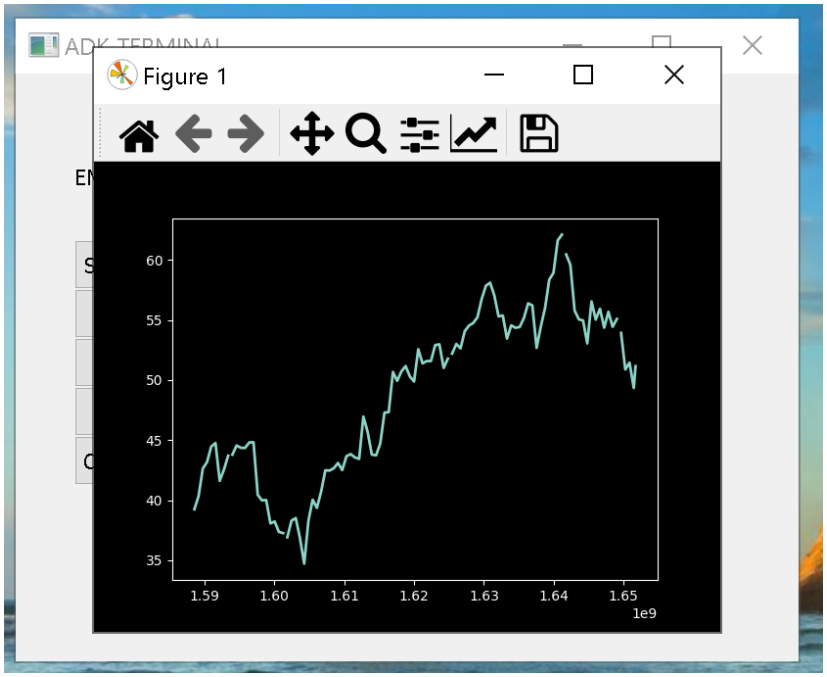
- A page will display different analytical information about Cisco similar to this screenshot. We will provide different charts and news articles for the user to analyze.

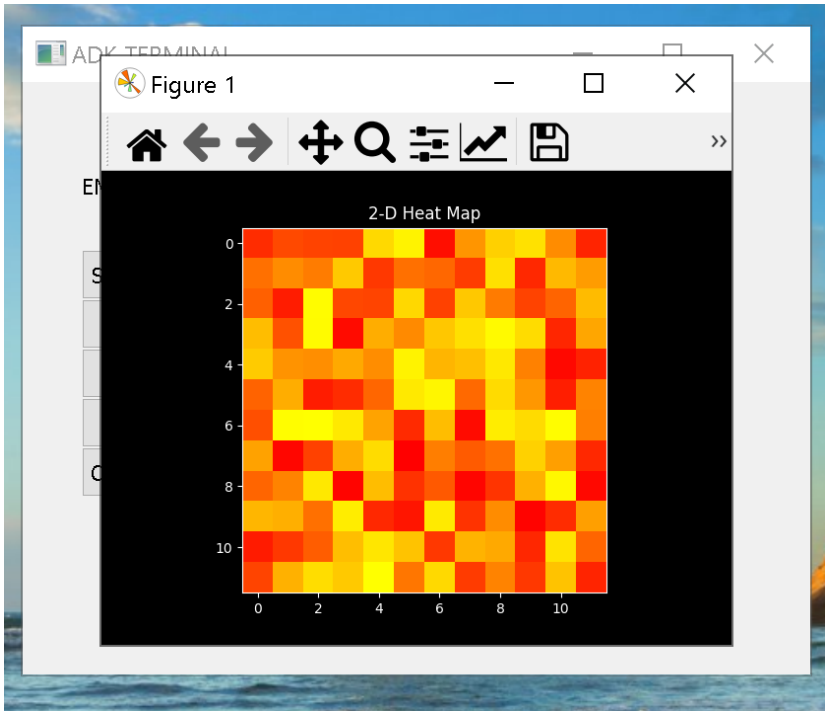
- Through this application the user enters the ticker of the company and can see how the stock market is for that particular company.

- Our program will analyze those scrapped sentiments based on “Positive”, “Negative” and “Neutral” feedback and give a pie chart in the output as displayed.



- It will then give a detailed HeatMap and a graphical representation for the stock prices of that company over the years with respect to the volume of trading as shown below.





Glossary of Terms

- Yahoo Finance Library : **yfinance** is not affiliated, endorsed, or vetted by Yahoo, Inc. It's an open-source tool that uses Yahoo's publicly available APIs, and is intended for research and educational purposes.

- Python – Pendulum Module : The pendulum is one of the popular Python DateTime libraries to ease DateTime manipulation. It provides a cleaner and easier to use API. It simplifies the problem of complex date manipulations involving time zones which are not handled correctly in native datetime instances. It inherits from the standard datetime library but provides better functionality.

- Matlibplot Library : **matplotlib.pyplot** is a collection of functions that make matplotlib work like MATLAB. Each pyplot function makes some change to a figure: e.g., creates a figure, creates a plotting area in a figure, plots some lines in a plotting area, decorates the plot with labels, etc.

- PyQt Library : Qt itself is written in C++. By using it from Python, we can build applications much more quickly while not sacrificing much of the speed of C++. PyQt5 refers to the most recent version 5 of Qt. We may still find the occasional mention of (Py)Qt4 on the web, but it is old and no longer supported.

- Financial Data Displays : Financial data visualizations involve the visual representation of all information in the balance sheet, profit and loss and cash flow statements so the finance team can easily analyze the information and share the trends, exceptions and opportunities.

- Bid : The highest price that a buyer is willing to pay for a particular stock.

- Ask : Lowest price at which an owner agrees to sell the shares.

- Volume : The total number of shares or contracts traded on national and regional exchanges in a stock, bond, commodity, future, or option on a certain day.

- Price to Earnings Ratio : Shows the multiple of earnings at which a stock sells. Determined by dividing current stock price by current earnings per share(adjusted for stock splits).

- Beta : It is a measurement of the relationship between stock price of any particular stock and the movement of the whole market.

References (APA format)

1. *Creating a stock research terminal with python*. Coding and Fun. (2021, June 15). Retrieved February 7, 2022, from https://codingandfun.com/creating-a-stock-research-terminal-with-python/

2. Yahoo! *Cisco Inc. (AAPL) stock price, news, Quote & History*. Yahoo! Finance. Retrieved February 7, 2022, from https://finance.yahoo.com/quote/AAPL?p=AAPL&.tsrc=fin-srch

3. Tang, K. (2021, August 1). *The easiest way to pull stock data into your python program: Yfinance*. Retrieved February 7, 2022, from https://towardsdatascience.com/the-easiest-way-to-pull-stock-data-into-your-python-program-yfinance-82c304ae35dc

4. *PYQT5*. PyPI. (n.d.). Retrieved February 7, 2022, from https://pypi.org/project/PyQt5/

5. *Python - pendulum module*. GeeksforGeeks. (2020, December 17). Retrieved February 7, 2022, from https://www.geeksforgeeks.org/python-pendulum-module/

6. *Diagrams.net - free flowchart maker and diagrams online*. Flowchart Maker & Online Diagram Software. (n.d.). Retrieved February 7, 2022, from https://app.diagrams.net/