A Purposeful Walk Down Wallstreet

Exploring Advanced Data Analytics in Financial Markets



Software Requirements Specification

Version 5.0

July 30, 2020

Frino Jais Sri Padmini Jayanti William Aman Minhajul Abadeen

VERSION HISTORY

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Version #** | **Implemented By** | **Revision Date** | **Approved By** | **Approval Date** | **Reason** |
| 1.0 | William Aman  Sri Padmini Jayanti Minhajul Abadeen  Frino Jais | 06/11/2020 | William Aman  Sri Padmini Jayanti  Minhajul Abadeen  Frino Jais | 06/11/2020 | - Original copy of document submitted to Professor Seyed |
| 2.0 | Frino Jais | 07/01/2020 | William Aman  Sri Padmini Jayanti  Minhajul Abadeen  Frino Jais | 07/01/2020 | - Added images to the Software Interface section per Professor Seyed’s comments.  - Data flow diagram redesigned to show only flow of data per Professor Seyed’s comments.  - Data flow description changed. |
| 3.0 | William Aman | 07/02/2020 | William Aman  Sri Padmini Jayanti  Minhajul Abadeen  Frino Jais | 07/02/2020 | - Completely overhauled the functional requirements to make sure each is a “feature” of the application per Professor Seyed’s comments.  - Reworked the nonfunctional requirements based on online examples and comments from Professor Seyed. |
| 4.0 | Sri Padmini Jayanti | 07/11/2020 | William Aman  Sri Padmini Jayanti  Minhajul Abadeen  Frino Jais | 07/11/2020 | Newly added financial instruments’ abbreviations |
| 5.0 | Frino Jais | 07/21/2020 | William Aman  Sri Padmini Jayanti  Minhajul Abadeen  Frino Jais | 07/21/2020 | Added new FJF packages to Assumptions and Dependencies section |
| 6.0 | William Aman  Sri Padmini Jayanti  Minhajul Abadeen  Frino Jais | 07/29/2020 | William Aman  Sri Padmini Jayanti  Minhajul Abadeen  Frino Jais | 07/29/2020 | Made version history table consistent with other documents |

# Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature** | **Printed Name** | **Title** | **Date** |
|  |  |  |  |

Table of Contents

[Revision History 2](#_Toc45458286)

[Document Approval 4](#_Toc45458287)

[1. Introduction 7](#_Toc45458288)

[1.1 Purpose 7](#_Toc45458289)

[1.2 Scope 7](#_Toc45458290)

[1.3 Definitions, Acronyms, and Abbreviations 8](#_Toc45458291)

[1.4 References 10](#_Toc45458292)

[1.5 Overview 10](#_Toc45458293)

[2. General Description 10](#_Toc45458294)

[2.1 Product Perspective 10](#_Toc45458295)

[2.2 Product Functions 11](#_Toc45458296)

[2.3 User Characteristics 11](#_Toc45458297)

[2.4 General Constraints 12](#_Toc45458298)

[2.5 Assumptions and Dependencies 12](#_Toc45458299)

[3. Specific Requirements 13](#_Toc45458300)

[3.1 External Interface Requirements 13](#_Toc45458301)

[3.1.1 User Interfaces 13](#_Toc45458302)

[3.1.2 Hardware Interfaces 16](#_Toc45458303)

[3.1.3 Software Interfaces 16](#_Toc45458304)

[3.1.4 Communications Interfaces 19](#_Toc45458305)

[3.2 Functional Requirements 19](#_Toc45458306)

[3.2.1 Collect Current Stock Data 19](#_Toc45458307)

[3.2.2 Collect Current Macroeconomic Data 20](#_Toc45458308)

[3.2.3 Generate Stock Price Forecasts 20](#_Toc45458309)

[3.2.4 Generate Stock Buy/Sell Signals 21](#_Toc45458310)

[3.2.5 Macroeconomic Indicator Visualization 21](#_Toc45458311)

[3.2.6 Stock Close Price Visualization 22](#_Toc45458312)

[3.2.7 Stock Close Price Forecast Visualization 22](#_Toc45458313)

[3.2.8 Stock Buy/Sell Signal Visualization 23](#_Toc45458314)

[3.2.9 Stock Portfolio Visualization 23](#_Toc45458315)

[3.3 Non-Functional Requirements 24](#_Toc45458316)

[3.3.1 Performance 24](#_Toc45458317)

[3.3.2 Reliability 24](#_Toc45458318)

[3.3.3 Portability 24](#_Toc45458319)

[3.3.4 Availability 24](#_Toc45458320)

[3.3.5 Maintainability 24](#_Toc45458321)

[3.4 Design Constraints 24](#_Toc45458322)

[3.5 Logical Database Requirements 25](#_Toc45458323)

[3.6 Other Requirements 25](#_Toc45458324)

[4. Analysis Models 26](#_Toc45458325)

[4.1 Data Flow Diagrams (DFD) 26](#_Toc45458326)

[A.Appendices 27](#_Toc45458327)

[A.1 Project proposal 27](#_Toc45458328)

# 1. Introduction

The Software Requirements Specifications (SRS) document is set to provide an overall agenda of the project: “A Purposeful Walk Down the Wall Street” proposed by our client, Mr. Joshua Feinstein of General Motors (GM). Included below are the specific requirements and auxiliary details that both the development team and client have acknowledged are required to produce a successful project.

This specific project has been developed and added to by various teams since 2014. This project has seen the use of various machine learning techniques. However, it has finally come down to using traditional methods to predict buy/sell activities concerning a group of assets which are currently under management.

## 1.1 Purpose

The purpose of this Software Requirements (SRS) document is to provide a detailed description of the overview, functionalities, and requirements associated with the GM Fintech Project. This document will act as a contract between the development team and the client to ensure that all the required features that the client needs are accounted for, as well as provide a one-stop location to track our progress and stay on track.

## 1.2 Scope

*The scope of this project will include the following:*

1. Production of an enhanced stock market predicting software product, through the further development of existing algorithms and UI.
2. Migration of front-end client facing UI from Tableau Software to Power BI.
3. This software will provide the user with graphs and predictions to support their decisions on trading on the stock market.
4. This software will not allow the user to execute trade deals within the UI of Power BI.

## 1.3 Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| **Key** | **Definition** |
| ARIMA | Short for Auto-Regressive Integrated Moving Averages. It is a data science model that forecasts future values using its own past values. |
| CARZ | CARZ is a financial instrument index that tracks automobile manufacturing companies. |
| FCAU | FCAU is the financial instrument indicator of the Fiat Chrysler Automobiles stock |
| TM | TM is the financial instrument indicator of the Toyota Motors stock |
| F | F is the financial instrument indicator of the Ford Motor Company stock |
| HMC | HMC is the financial instrument indicator of the Honda Motor Co. Ltd. stock |
| CMA | Short for Cross-Moving Average. It is an algorithm that detects when a shorter period moving average intersects a longer period moving average. |
| EMA | Short for Exponential-Moving Average. It is a type of moving average that places a greater weight on newer data. |
| ETL | Short for Extract, Transform, and Load. A specialized form of software that allow for extraction of data from databases, applications, and systems. |
| Fintech | Short for financial technology. It is the technology and innovation that aims to compete with traditional financial methods in the delivery of financial services. |
| FRL | Short for Fibonacci Retracement Lines. They are horizontal lines overlaid on a stock chart to indicate support and resistance to a price. |
| GDP | Short for Gross Domestic Product. It is a measure of the total value of goods produced and services provided in a country during one year. |
| MACD | Short for Moving Average Convergence Divergence. It is a type of moving average that shows the relationship between two moving averages to show a stock price trend. |
| Misery Index | An economic indicator of how the average U.S. citizen is doing, calculated by adding the seasonally adjusted unemployment rate to the annual inflation rate. |
| MySQL | A relational database management system (RDBMS) we will use to store our data. |
| Power BI | A data visualization and BI tool that aids in making decisions by analyzing enterprise data for insights. |
| Quandl | An Application Programming Interface (API) that will facilitate the sourcing of various macroeconomic data from the internet. |
| S&P 500 | Short for Standard and Poor 500. A market index that measures the value of the 500 largest U.S. publicly traded companies. |
| Tableau | A data visualization and business intelligence tool that enables interaction with data in a database. |
| XGBoost | A machine learning tool in Python that uses the Gradient Boosting framework. |
| Yahoo! Finance | A source of financial data from which the application will source the stock data from. |

## 

## 1.4 References

* Quandl: <https://docs.quandl.com/docs/python>
* MySQL: <https://dev.mysql.com/doc/>
* YAHOO! Finance: <https://finance.yahoo.com/>
* Power BI: <https://docs.microsoft.com/en-us/power-bi/>

## 1.5 Overview

The remainder of this document will expand on the design and functionality of the application. It will explain the flow of data, how it is analyzed, and then displayed using tables, graphs, and images. More details will also be given about how the application can be used, what it can be used for, and any limitations there may be. Next, the document will introduce information about the application’s requirements. This includes external interface requirements, functional requirements, and non-functional requirements. All the components in this SRS will play a role in the project’s structure and overall outcome.

# 2. General Description

The following sections will guide the reader through the general description of this project. This section will not include the specifics of the project, but more of a overview. This will be useful for the reader to understand the specific requirements when they view that section later in the document.

## 2.1 Product Perspective

The GM Fintech Project is a legacy project that focuses on improving an existing application in which the main functionality is to predict the future behavior of predetermined financial indexes or instruments. The application will indicate whether one should buy, sell, or hold a portfolio of assets that are currently under management.

The application currently uses a MySQL database as the RDBMS where data is collected and then accessed to be displayed in a Tableau worksheet. The data that is stored in the MySQL tables are retrieved from Yahoo! Finance and Quandl. These two sources allow the application to analyze market movements and macroeconomic data to make proper investment decisions.

The user interface (UI) is a vital component of this application because it allows the user to interact with the data in a visualized form. The use of the Tableau software will portray the retrieved data in a clear graphical manner, where the user will be able to receive details specific to each data point. For this application, the role of Tableau is simply a UI tool that is leveraged to read and display the computations from MySQL.

The data stored in MYSQL is retrieved from the application’s financial data sources, namely Yahoo! Finance and Quandl. Python language is used to create functions and algorithms that will manipulate this data to make predictions about future market movements. Signals are given by this code to indicate the best time to buy or sell a given financial instrument. For this project, the financial instruments being assessed will be the S&P 500, GM, CARZ, XPH, PFE, ^TYX, and SPY. The use of macroeconomic variables is also implemented to create a stronger forecast measure.

## 2.2 Product Functions

The main function of this application is to predict the stock market trends to make a strong decision whether a financial instrument should be bought or sold. In the following list, the functions and features of the application are outlined:

1. Data fetch from sources Yahoo! Finance and Quandl.
   1. Python code that allows flow of live data from Yahoo! Finance and Quandl.
   2. It also establishes a connection with the MySQL server to store this incoming data into our RDBMS.
2. Data forecast of stock market trends.
   1. Leverage financial calculation concepts like CMA and EMA to predict long term market movements.
   2. Other traditional time series machine learning models like ARIMA help in accurate prediction of buy/sell movements.
   3. A relatively new approach including macroeconomic variables has also helped to predict long term market movements for specific trading instruments.
3. Data visualization.
   1. Python code runs and populates the tables in the MySQL database with the achieved results.
   2. Tableau is integrated with the MySQL database to extract results and convert them to visuals.
   3. Data is shown in charts and graphs for easy interpretation.
4. Buy/Sell signaling
   1. After results are obtained from each implemented algorithm, another python code is written to analyze the outputs and predict buy/sell signals.
   2. The outputs are compared with ideal figures to place the outputs in perfectly divided predictions.

## 2.3 User Characteristics

The user(s) of this application are expected to be individuals that have a sufficient understanding of financial markets, instruments, and investment in these areas. Users must also have sufficient knowledge of Power BI as this will be the front-end client facing UI platform to view various forecast charts and graphs. Users will also have the ability to view raw data depending on their permissions. The user should then be able to read and determine any inconsistency with the given data and information before making any decisions based on the application’s recommendations. In this case, the user should be able to understand the presented data through the UI or in its raw form.

## 2.4 General Constraints

A large component of this application is the recording of data and information in a live fetched manner. Since the application will be retrieving data from online sources, such as Yahoo! Finance and Quandl, an internet connection is required. This is also necessary for the data contained in MySQL to remain up to date as the prices of financial instruments in the market change.

Another limitation to address is the storage capacity of the MySQL database. When the capacity of this database is reached, old data will be overwritten by new data to ensure that the application continues to produce up to date results and recommendations.

Lastly, the tools and sources for this application must remain as given or there may be a chance for data misrepresentation or errors. Any change in the algorithms, data structure, or data sources must be reflected throughout the entire application for proper output.

## 2.5 Assumptions and Dependencies

The main dependencies of this application are the data sources, Yahoo! Finance and Quandl. Yahoo! Finance will provide the financial market exchange data and Quandl will provide economic data. New data sources may be added, but the backend must be changed to accommodate any new sources.

Additionally, a variety of different Python dependencies are implemented and without successful installation of all these packages, issues and errors may occur. Due to the application’s sensitive nature, the installation guide must be strictly followed to ensure that the application runs properly. These Python dependencies are:

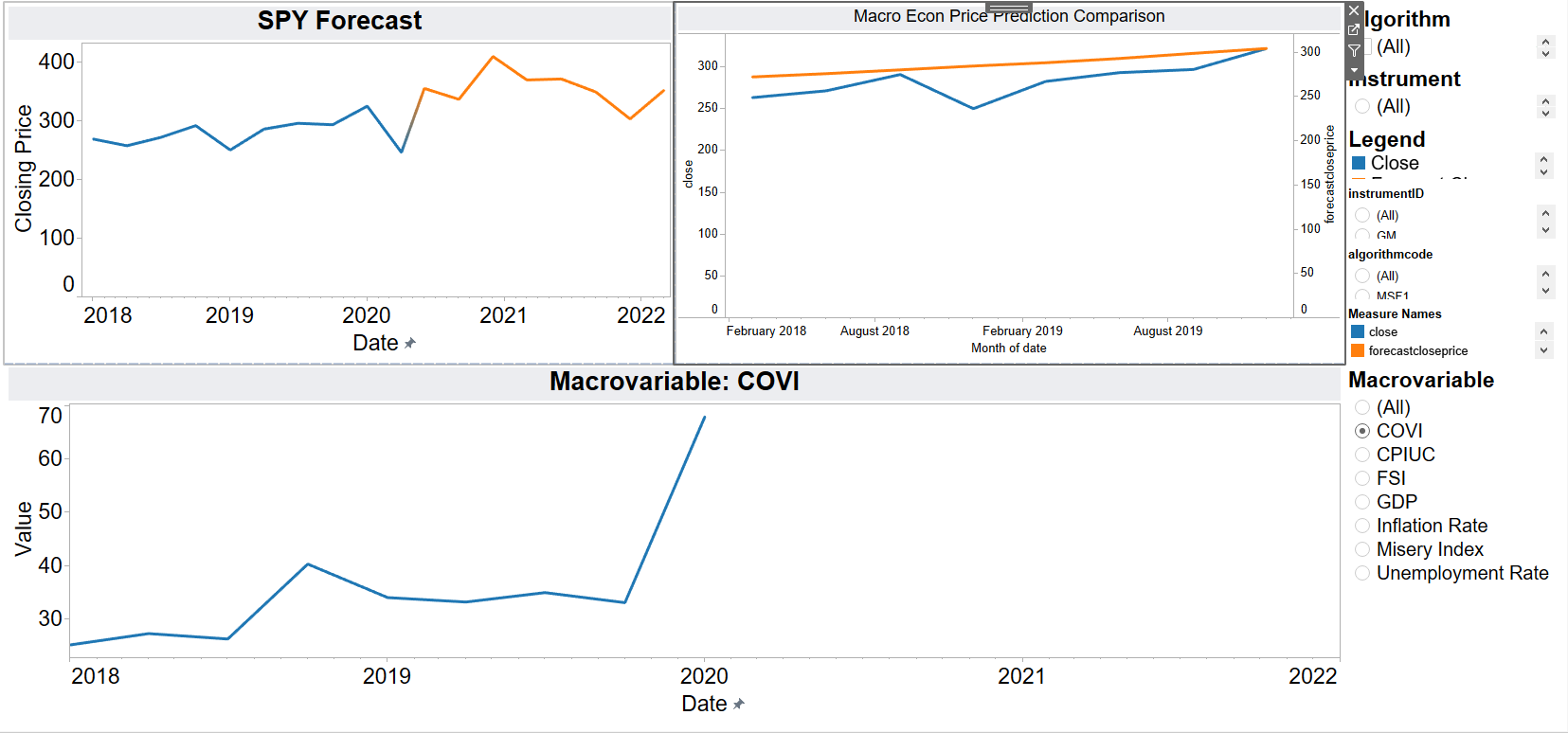
* + 1. XGBoost
    2. SQLAlchemy
    3. pandas\_datareader
    4. stockstats
    5. statsmodels
    6. sklearn
    7. Quandl
    8. fredapi
    9. pytest
    10. PyMySQL
    11. pyodbc
    12. Matplotlib
    13. Keras
    14. Tensorflow
    15. holidays

It is expected that the user will host the MySQL database and backend Python code locally. If these parts of the application are not properly hosted, the application will not run. It is also required that all data fetched from Yahoo! Finance and Quandl are properly configured with the data tables in MySQL.

# 3. Specific Requirements

## 3.1 External Interface Requirements

### 3.1.1 User Interfaces

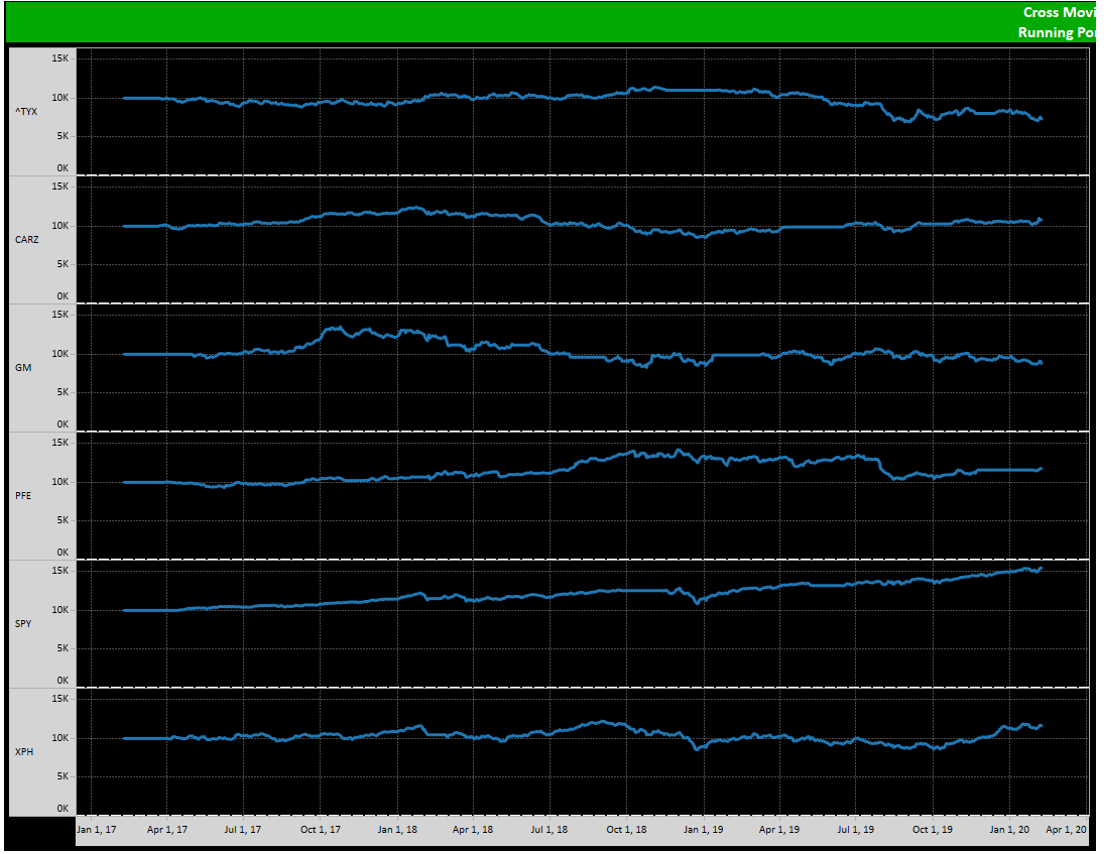
*Tableau Dashboard showing Macroeconomic Price Prediction Comparisons with respect to different financial instruments.*

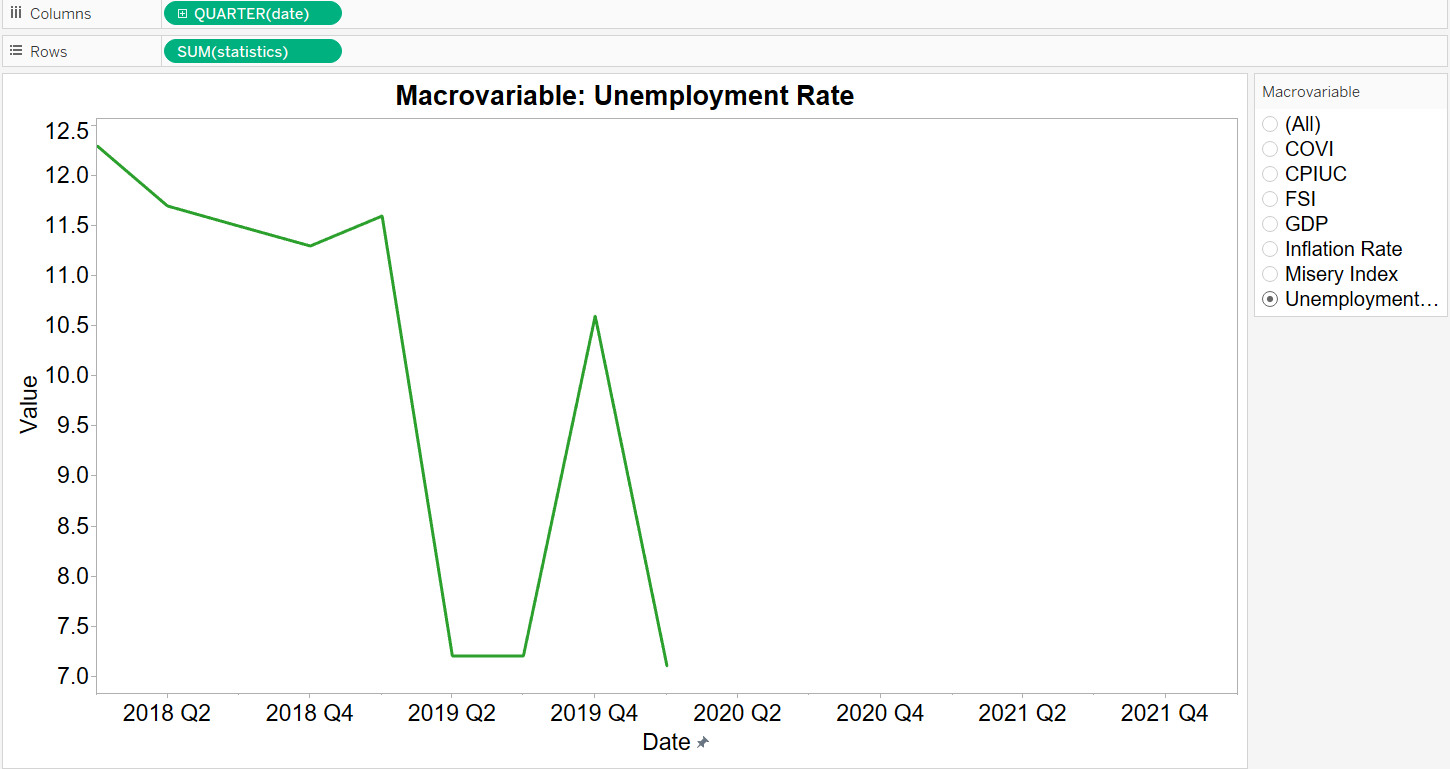
The above sheet from the Tableau dashboard shows how side-by-side comparison is possible for different financial instruments for a given algorithm. There are options such as “Algorithm” and “Instrument” which can be selected by the user. There’s also an option for selecting the “Macro variable” to allow the user to select which variables he/she wants to compare.



*Buy/sell signals comparison of EMA, MACD and ARIMA*

As mentioned in the “Product Functions” section of this document, the buy/sell forecast signals generated by the python code are an integral part of this project. The above sheet shows such buy/sell signals Tableau output generated by the python code. The three signals above are EMA, MACD and ARIMA algorithms respectively.

*Visualization of market performance by instrument*

*Visualization of macroeconomic variables (Unemployment Rate)*

The above sheet from Tableau depicts the trendline of several macroeconomic variables over time. There are several user selectable options to change the time, variable, and statistics shown on the graph.

The images shown above are the Tableau dashboard before migration to Power BI. A collection of charts and graphs explaining the market trends and predictions are given, as well as a layout of the buy/sell interface for CMA and FRL. The tracking of stock performance is reflective of live data queried from Yahoo! Finance and Quandl. The depiction of macroeconomic variables are comprised of the GDP, Misery Index, Unemployment Rate, Inflation Rate, etc. The application assesses these values to find correlations that can be used in market prediction and future economic states as well.

The Tableau interface is surely easy on the developers and is one of the best data visualization tools on the market. However, it does have a high monetary price to be paid for several users to work on it. Hence, our client is looking for more affordable options, and Power BI is the top pick now. We are planning to migrate a share of the reports created on Tableau to Power BI in this semester. We have already tested some functionalities of Power BI and it does offer all the functionalities that Tableau has that our client requires. It is also quick in response time, query analysis time is faster, and produces great visuals.

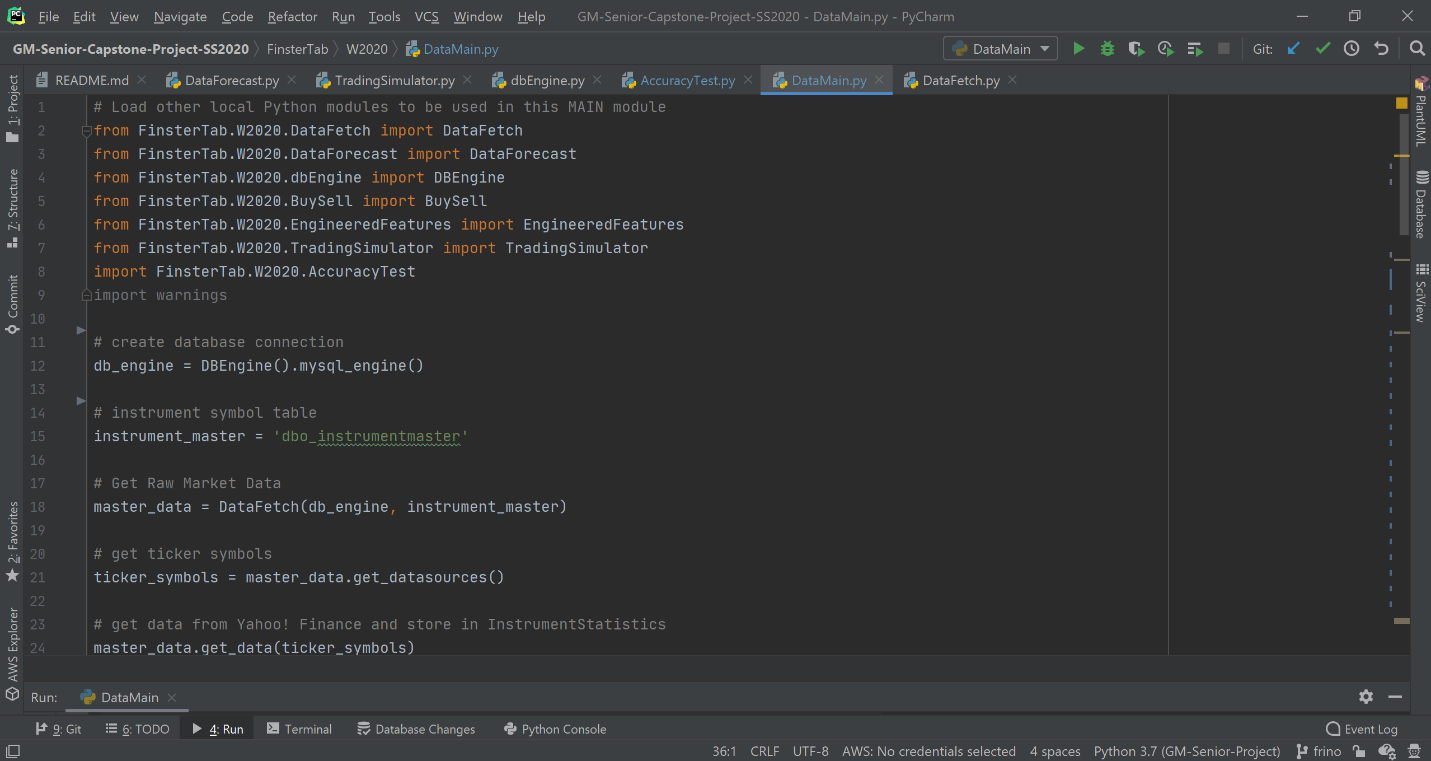
### 3.1.2 Hardware Interfaces

The GM Fintech application is developed for computer machines with current generation processors, 5GB of free disk storage space, and 4GB of RAM.

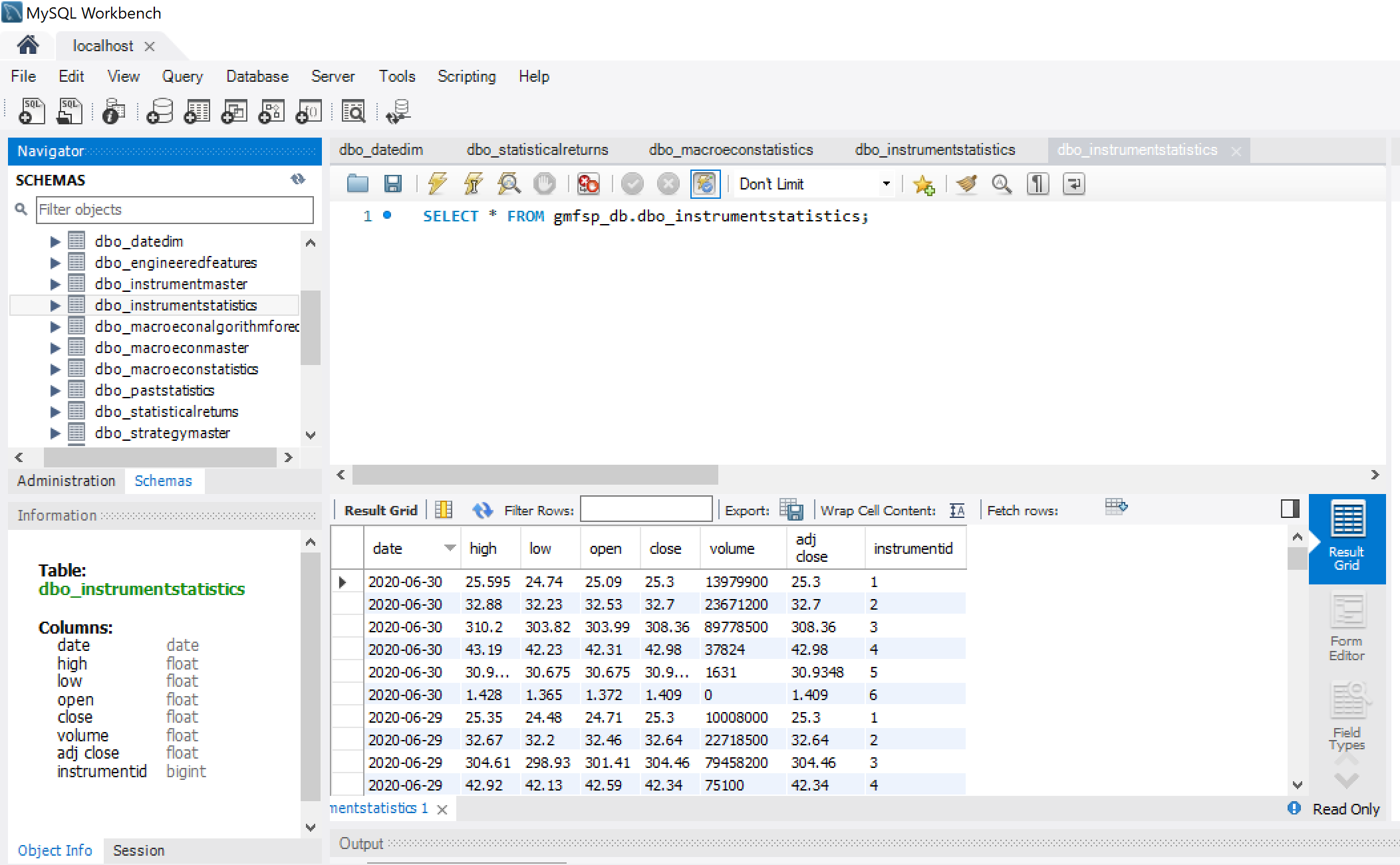
### 3.1.3 Software Interfaces

The GM Fintech application requires a system with that meet the following requirements:

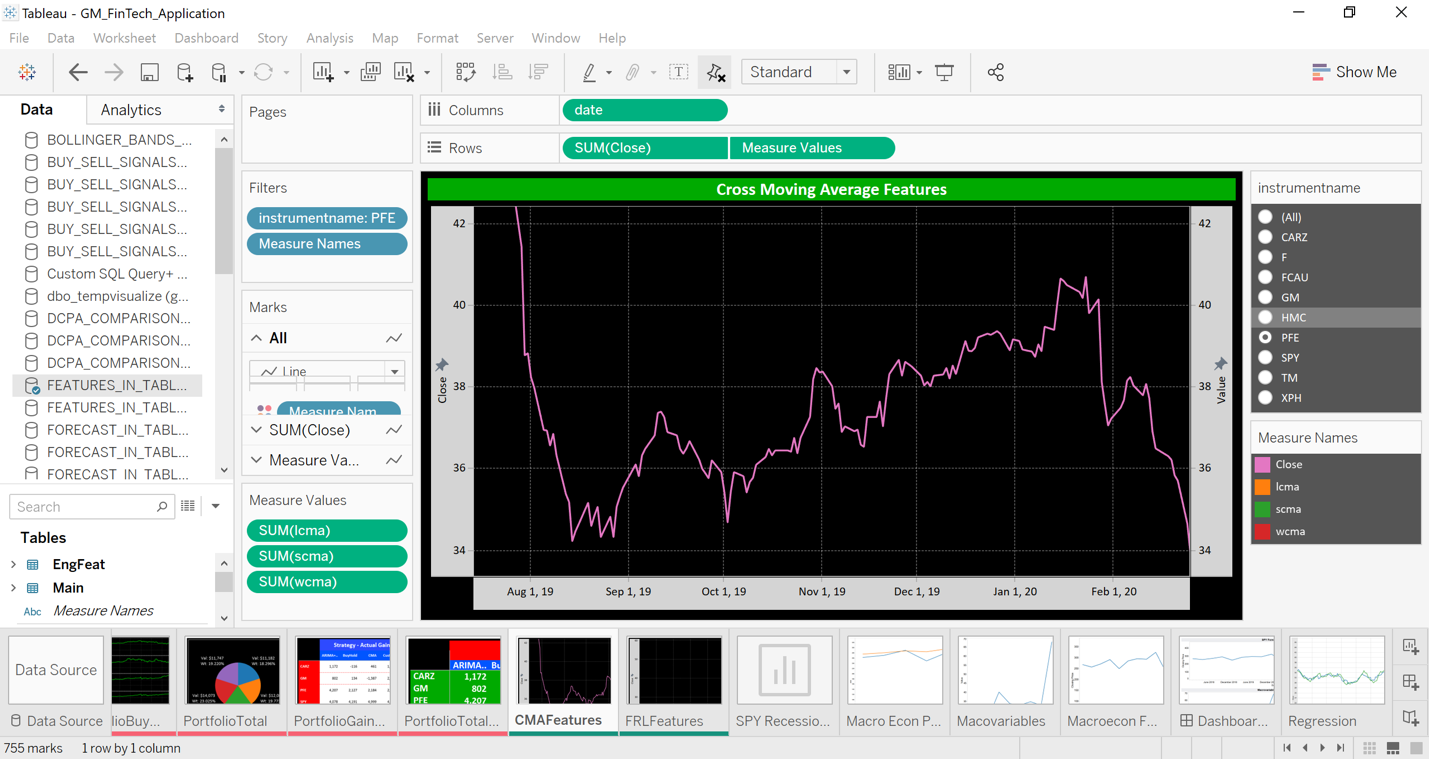
1. PyCharm Professional Edition IDE



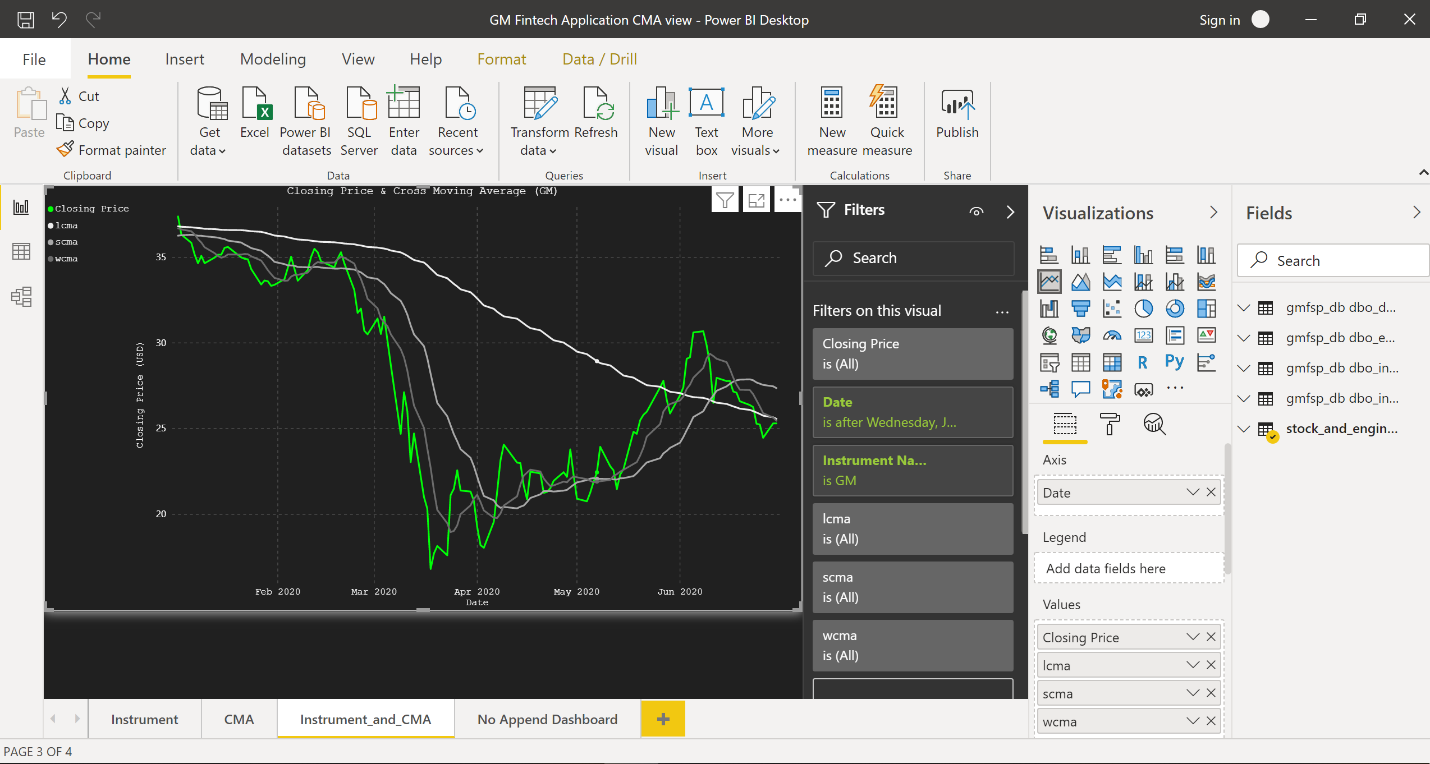
1. MySQL Relational Database Management System



1. Tableau



1. Power BI (after migration)



1. Windows Operating System

A screen shot of a computer

Description automatically generated

### 3.1.4 Communications Interfaces

The GM Fintech application will require a stable internet connection so that the application’s network server that is running on the user’s machine has access to the database. The server’s main responsibility is to collect the requested data from the sources Yahoo! Finance and Quandl to populate the tables in MySQL. The application will retrieve this stored information and analyze it through the various algorithms in the Python code. Tableau will represent and portray a visualization for the application’s findings using the updated data from the database, processed by the Python code.

## 3.2 Functional Requirements

### 3.2.1 Collect Current Stock Data

|  |  |
| --- | --- |
| **ID** | FR-1 |
| **Title** | Provide Current Stock Data |
| **Description** | The user will have the feature of retrieving current stock data for GM, F, FCAU, TM, HMC, PFE, SPY, XPH, CARZ, and ^TYX that will then be stored in the database. |
| **Priority** | High |
| **Backward Dependencies** | N/A |
| **Forward Dependencies** | FR-3, FR-4, FR-6, FR-7, FR-8, FR-9 |
| **Input(s)** | The user calls get\_data from DataFetch.py |
| **Processing** | This function makes a call to the Yahoo! Finance API, which returns the most recent close price statistics for each instrument listed in dbo\_instrumentmaster. |
| **Output(s)** | The table dbo\_instrumentstatistics is populated with stock closing price data for GM, F, FCAU, TM, HMC, PFE, SPY, XPH, CARZ, and ^TY. |
| **Error Handling** | N/A |

### 

### 3.2.2 Collect Current Macroeconomic Data

|  |  |
| --- | --- |
| **ID** | FR-2 |
| **Title** | Provide Current Macroeconomic Data |
| **Description** | The user will have the feature of retrieving current macroeconomic data for GDP, UR, IR, MI, TYX, COVI, CPIUC, and FSI that will then be stored in the database. |
| **Priority** | High |
| **Backward Dependencies** | N/A |
| **Forward Dependencies** | FR-3, FR-5, FR-7 |
| **Input(s)** | The user calls macroFetch from DataFetch.py |
| **Processing** | This function loops through all the macroeconomic indicators from dbo\_macroeconmaster and makes retrieves their most recent data using the Quandl API. |
| **Output(s)** | The new data is inserted into the table dbo\_macroeconstatistics for the indicators GDP, UR, IR, MI, TYX, COVI, CPIUC, and FSI. |
| **Error Handling** | N/A |

### 

### 3.2.3 Generate Stock Price Forecasts

|  |  |
| --- | --- |
| **ID** | FR-3 |
| **Title** | Generate Stock Price Forecasts |
| **Description** | The user will have the feature of generating stock forecasts calculated by the following algorithms: Polynomial Regression, ARIMA, Random Forest Regression, SVM, MSF, FJF, ARS, SJF, MAF, and XGB that will then be stored in the database. |
| **Priority** | High |
| **Backward Dependencies** | FR-1, FR-2 |
| **Forward Dependencies** | FR-7 |
| **Input(s)** | The user runs DataForecast.py |
| **Processing** | This class then calls calculate\_forecast, calculate\_arima\_forecast, calculate\_svm\_forecast, calculate\_xgboost\_forecast, calculate\_regression, MSF1, MSF2, MSF3, FJF, ARS, SJF, and MAF which all perform their own algorithms. |
| **Output(s)** | The calculations of calculate\_forecast, calculate\_arima\_forecast, calculate\_svm\_forecast, calculate\_xgboost\_forecast, calculate\_regression, and ARS are inserted into the table dbo\_algorithmforecast. The calculations of MSF1, MSF2, and MSF3 are inserted into dbo\_macroeconalgorithmforecast. |
| **Error Handling** | N/A |

### 3.2.4 Generate Stock Buy/Sell Signals

|  |  |
| --- | --- |
| **ID** | FR-4 |
| **Title** | Generate Stock Buy/Sell Signals |
| **Description** | The user will have the feature of generating buy/sell signals calculated by the following algorithms: CMA, EMA, FRL, and MACD that will then be stored in the database. |
| **Priority** | High |
| **Backward Dependencies** | FR-1 |
| **Forward Dependencies** | FR-8 |
| **Input(s)** | The user calls calculate from the class EngineeredFeatures.py |
| **Processing** | For each instrument in dbo\_macroeconmaster, the function calculate generates CMA, EMA, FRL, and MACD data using their own algorithms. |
| **Output(s)** | The calculations generated by CMA, EMA, FRL, and MACD are insterted into the table dbo\_engineeredfeatures for GM, F, FCAU, TM, HMC, PFE, SPY, XPH, CARZ, and ^TY. |
| **Error Handling** | N/A |

### 3.2.5 Macroeconomic Indicator Visualization

|  |  |
| --- | --- |
| **ID** | FR-5 |
| **Title** | Macroeconomic Indicator Visualization |
| **Description** | The user will have the feature of selecting and viewing line charts for the values of GDP, UR, IR, MI, TYX, COVI, CPIUC, and FSI in Power BI. |
| **Priority** | High |
| **Backward Dependencies** | FR-2 |
| **Forward Dependencies** | N/A |
| **Input(s)** | The user selects the dashboard named Macro Indicator in the Power BI template file named GM\_Fintech\_SS2020.pbit |
| **Processing** | Power BI selects the field “statistics” from the table dbo\_macroeconstatistics as the data source, the field “date” from the table dbo\_datedim as the x-axis, and the field “macroeconname” from the table dbo\_macroeconmaster as the filter. |
| **Output(s)** | A line chart is generated showing the historical data for the macroeconmic indicators GDP, UR, IR, MI, TYX, COVI, CPIUC, and FSI |
| **Error Handling** | N/A |

### 3.2.6 Stock Close Price Visualization

|  |  |
| --- | --- |
| **ID** | FR-6 |
| **Title** | Stock Close Price Visualization |
| **Description** | The user will have the feature of selecting and viewing line charts for the closing prices of GM, F, FCAU, TM, HMC, PFE, SPY, XPH, CARZ, and ^TYX in Power BI. |
| **Priority** | High |
| **Backward Dependencies** | FR-1 |
| **Forward Dependencies** | N/A |
| **Input(s)** | The user selects the dashboard named Instrument Close in the Power BI template file named GM\_Fintech\_SS2020.pbit |
| **Processing** | Power BI selects the field “close” from the table dbo\_instrumentstatistics as the data source, the field “date” from the table dbo\_datedim as the x-axis, and the field “instrumentname” from the table dbo\_instrumentmaster as the filter |
| **Output(s)** | A line chart is generated showing the close price history for GM, F, FCAU, TM, HMC, PFE, SPY, XPH, CARZ, and ^TYX. |
| **Error Handling** | N/A |

### 3.2.7 Stock Close Price Forecast Visualization

|  |  |
| --- | --- |
| **ID** | FR-7 |
| **Title** | Stock Close Price Forecast Visualization |
| **Description** | The user will have the feature of selecting and viewing line charts for the forecasted close prices of GM, F, FCAU, TM, HMC, PFE, SPY, XPH, CARZ, and ^TYX in Power BI. |
| **Priority** | High |
| **Backward Dependencies** | FR-1, FR-3 |
| **Forward Dependencies** | N/A |
| **Input(s)** | The user selects the dashboard named Instrument Forecast in the Power BI template file named GM\_Fintech\_SS2020.pbit |
| **Processing** | Power BI selects the fields “forecastcloseprice” from the table dbo\_algorithmforecast and “foreecastprice” from the table dbo\_macroeconalgorithmforecast as the data sources, the field “date” from the table dbo\_datedim as the x-axis, and the fields “algorithmname” from the table dbo\_algorithmmaster and “algorithmcode” from dbo\_macroeconalgoreithmforecast as the filters |
| **Output(s)** | A line chart is generated showing forecasted close prices for GM, F, FCAU, TM, HMC, PFE, SPY, XPH, CARZ, and ^TYX. |
| **Error Handling** | N/A |

### 3.2.8 Stock Buy/Sell Signal Visualization

|  |  |
| --- | --- |
| **ID** | FR-8 |
| **Title** | Stock Buy/Sell Signal Visualization |
| **Description** | The user will have the feature of selecting and viewing line charts for the buy/sell signals of GM, F, FCAU, TM, HMC, PFE, SPY, XPH, CARZ, and ^TYX in Power BI. |
| **Priority** | High |
| **Backward Dependencies** | FR-1, FR-4 |
| **Forward Dependencies** | N/A |
| **Input(s)** | The user selects the dashboard named Buy/Sell Signals in the Power BI template file named GM\_Fintech\_SS2020.pbit |
| **Processing** | Power BI selects the the fields for CMA, EMA, FRL, and MACD from the table dbo\_engineeredfeatures and the field “close” from the table dbo\_instrumentstatistics as the data sources, the field “date” from the table dbo\_datedim as the x-axis, and all of the fields for CMA, EMA, FRL, and MACD as well as the field “instrumentname” from the table dbo\_instrumentmaster as the as the filters. |
| **Output(s)** | A line chart is generated showing the relationship between the closing prices of GM, F, FCAU, TM, HMC, PFE, SPY, XPH, CARZ, and ^TYX against the engineered features CMA, EMA, FRL, and MACD. |
| **Error Handling** | N/A |

### 

### 3.2.9 Stock Portfolio Visualization

|  |  |
| --- | --- |
| **ID** | FR-9 |
| **Title** | Stock Portfolio Visualization |
| **Description** | The user will have the feature of selecting and viewing simulated portfolio values of owning $10,000 worth of the following stocks: GM, F, FCAU, TM, HMC, PFE, SPY, XPH, CARZ, and ^TYX in Power BI with the output being shown using the engineered features CMA, EMA, FRL, and MACD. |
| **Priority** | Medium |
| **Backward Dependencies** | FR-1, FR-4 |
| **Forward Dependencies** | N/A |
| **Input(s)** | The user selects the dashboard named Buy/Sell Portfolio in the Power BI template file named GM\_Fintech\_SS2020.pbit |
| **Processing** | Power BI selects the field “portfoliovalue” from the table dbo\_statisticalreturns as the data source, the field “date” from dbo\_datadim as the x-axis, and the fields “instrumentname” from the table dbo\_instrumentmaster and “the fields corresponding to CMA, EMA, FRL, and MACD from the table dbo\_engineeredfeatures as the filters. |
| **Output(s)** | Four line charts are generated, showing visualizations of simulated portfolio values for the instruments GM, F, FCAU, TM, HMC, PFE, SPY, XPH, CARZ, and ^TYX in relation to the engineered features of CMA, EMA, FRL, and MACD. |
| **Error Handling** | N/A |

## 3.3 Non-Functional Requirements

### 3.3.1 Performance

Our software will perform the initial population of the database within two hours, and within thirty minutes for each run after that.

### 3.3.2 Reliability

Our software will update the database with accurate data and provide accurate data visualization 99% of the time it is ran.

### 3.3.3 Portability

Our software will run on 99% of laptops and desktops which are running Windows.

### 3.3.4 Availability

Our software will be available to use around the clock. It will be available to receive up-to-date data during stock market hours.

### 3.3.5 Maintainability

Our software will receive daily maintenance and updates through 07/30/2020.

## 3.4 Design Constraints

Design constraints include the following:

1. Python version 3.7 should be used when installing dependency packages to ensure that they are installed correctly.
2. MySQL RDBMS should be used instead of any other database due to the project’s current integration with MySQL.
3. A connection with the database server should be established before running the application.
4. GitHub must be used as the version control system for this project as the legacy project’s existing documentation is logged in GitHub.
5. If data sources Yahoo! Finance and Quandl are no longer available, the code will need to be changed to accommodate new data sources.
6. The user must have Tableau or Power BI (after migration) installed on their local machine to open the files for visualization.
7. The act of buying/selling stocks must be executed by the user, not the application.

## 3.5 Logical Database Requirements

As mentioned previously in this document, the logical database used for this application is MySQL. MySQL Workbench along with MySQL Server needs to be installed in the first place to get the application running. The creation of schema with the included tables needs to be done without errors to successfully run the program. The python code files also rely on MySQL database to run. Input from the database tables “Instrument master” and “Macroeconomic variables master” is required for the python code to run. Output predictions, buy/sell signals and other statistics from the python code also need to be stored in the database. So, establishing this connection is vital for producing outputs as well.

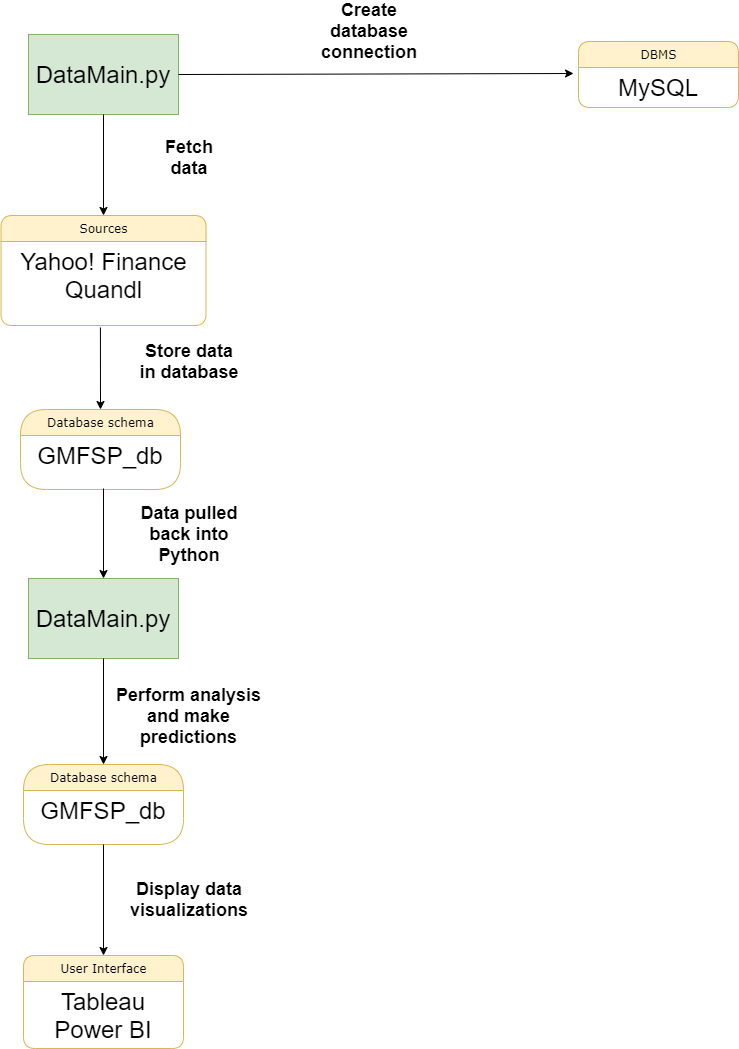
## 3.6 Other Requirements

A basic understanding of economics, stocks, and assets is required to take up this project. Analyzing how the stock market works with a variety of indices affecting it day-by-day is necessary once you have started working on this project.

A strong research mentality is needed to understand the logic behind the macroeconomic variables, formulae used in calculations in the python code, and to view the output graphs on UI which include financial instruments.

# 4. Analysis Models

## 4.1 Data Flow Diagrams (DFD)



This data flow diagram maps out the flow of data and information for all processes in the application from end to end. A database connection must be established with MySQL in order for the data to be pushed and fetched from their respective tables. The python code will fetch the statistics for each instrument that are currently being reviewed by the GM Fintech Application from its data sources: Yahoo! Finance and Quandl. This data will be stored in our database, gmfsp\_db. From there, the data will be pulled back into Python to perform further calculation and analysis to make predictions on the datasets. These results will be pushed back to gmfsp\_db. Finally, the UI (Tableau and/or Power BI) will make use of the stored data in gmfsp\_db and output views of this data in a graphical manner.

# Appendices

## A.1 Project proposal

**A Purposeful Walk Down Walk-Street – Exploring Advanced Data Analytics in Financial Markets**

**WSU COURSE:**

Senior Project Design and Ethics. Summer-Spring 2020

**Description:**

General Motors, a fortune 10 company, is currently one the leaders in Advanced Analytics implementations. Today, at GM, we leverage Advanced Analytics technologies to run our core Global Data, AI and Analytics Services Platform. Looking out over the next several years, the United States will need to continue to produce highly talented computer engineers and data scientists who are well versed in data analytics and data science. GM continues to partner with WSU to create real-world projects that will enable senior year students to learn these critical skills and technologies. The Use-Case chosen for this project is as follows:

*Today’s global markets and pricing of financial instruments are impacted week-by-week and day-by-day as a result of real-world events. In this project we will leverage fintech (financial technologies) to attempt to track and predict the positive or negative pricing movements within global financial markets. The students will design and implement key enhancements to our system that measures correlations with regards to financial instruments, trading strategies, and pricing. Ultimately, the WSU students will put their team efforts into improving our framework for descriptive and predictive analytics. They will elevate our platform to the next level and enable execution of highly accurate ‘buy’ and ‘sell’ opportunities for financial portfolios.*

Through the viewpoint of GM, the importance of advance analytics plays a huge part in the overall company structure of GM. Financial Technology plays a huge role in predicting outcomes when it comes to the financial markets. Any form of progress in this Capstone will be used in feature versions of this Capstone project to enable students and GM to learn.

**Project Deliverables:**

* 1. Improve the accuracy of the previous semester’s software to have a better predictive model.
  2. Implement a model that utilizes financial instruments and trading methodologies to a predictive model that predicts the right time to buy and sell amidst many economic factors in the stock market, while enhancing the profit margin of investments.
  3. Create a new algorithm that can be used alongside the current algorithms to aid in the decision making process of trading stocks.
  4. Create a front end UI through a new platform (Power BI) that will give the user the ability to analyze the data through a different array of graphs, charts, and models.
  5. Front end migration of the old platform (Tableau) to the new platform (Power BI) that will take the existing front end layer and transform it into a new, well performing, accurate, interface.

**STUDENT TEAM:**

All members of the GM FinTech project can choose to work on any of the given deliverables mentioned above.

**Technologies**

* RDBMS (Relational Database Management Systems) such as MySQL Workbench
* Operating system environment, namely Windows 10
* Financial algorithms and models that incorporate AI/ML based libraries and technologies
* Power BI as the front-end user interface
* Financial trading techniques, predictive financial algorithms
* ETL tools used to extract raw data from the database to the front end
* Programming languages: Python, SQL, DAX(Data Analysis Expression used for Power BI)

**GENERAL MOTORS CONTACT INFORMATION:**

Joshua Feinstein of Global Data,

AI & Analytics Services

Financial Analytics Lead -

General Motors

Joshua.feinstein@gm.com

**WAYNE STATE UNIVERSITY CONTACT INFORMATION:**

Professor Seyed Mousavi

Wayne State University

Department of Computer Science

5057 Woodward Avenue, Suite 200.44, Floor 14

Phone: 313-577-0722