General Motors FinTech

Product Design Specification

v2.0

11/3/2019

Mohamad Saab

Aalem Singh

John Gettel

Abdul Ahmad

Document Version History

Version	Date	Author	Reviewer	Approved by	Description
0.1	10/10/2019	Abdul Ahmad	N/A	N/A	Created the new document, formatted, wrote new sections starter text in each section
0.2	10/16/2019	Aalem Singh	Abdul Ahmad	N/A	Format, editing, worked on section 1, 2.1
0.3	10/17/2019	Abdul Ahmad	N/A	N/A	Edited sec 1 Introduction, edited and organized the entire document
0.4	10/18/2019	Abdul Ahmad	N/A	N/A	Rewrote sec 2.1, wrote sec# 2.2, 2.3, 2.4, 3.1, 3.2, 3.3, 3.4, 3.5. Created hardware, software and security architecture diagrams. Created use cases# 1, 2, 3, 4
0.5	10/19/2019	Abdul Ahmad	N/A	N/A	Wrote use cases# 5, 6, 7, 8. Traceability Matrix
0.6	10/20/2019	Abdul Ahmad	N/A	N/A	DFD, system level seq diagram, ARIMA seq diagram, database diagram, wrote sec 4.1, 4.2, 4.3, 4.5, 4.6, 4.7, 5 and appendix A, B
1.0	10/21/2019	Abdul Ahmad	N/A	John Gettel; Aalem Singh; Mohamad Saab;	Review, proofread, updated use cases, v1.0 release
1.1	11/2/2019	Abdul Ahmad	N/A	N/A	Section 2 minor changes, edited use cases
2.0	11/3/2019	Abdul Ahmad	N/A	John Gettel; Aalem Singh; Mohamad Saab;	Sec 4.1 use cases text and priority modified, sec 4.5 class diagram redesigned, sec 3.2 SW arch diagram modified, multiple seq diagrams added, sec 4.4 DB diagram fixed, numbering added to sequences, reviewed, proofread, v2.0 release
2.0	12/7/2019	Abdul Ahmad	N/A	John Gettel; Aalem Singh; Mohamad Saab;	Final updates, added logo, TOC, hyperlinks, ERD updated

TABLE OF CONTENTS

1 INTRODUCTION

1.1 PURPOSE OF THE PRODUCT DESIGN SPECIFICATION DOCUMENT

The purpose of this document is to provide developers with fundamental information about the system design and architecture. It will serve as a development framework for new developers and assist with future enhancements of the product. It is created during the planning phase of the project. It's intended to be utilized by project managers, project teams, and developers. Sections that discuss the user interface can also be used to inform the end-users about the setup of the application. This document tracks the necessary information required to effectively define the system architecture and design.

2 GENERAL OVERVIEW AND DESIGN GUIDELINES/APPROACH

2.1 ASSUMPTIONS

- Tableau software is installed on the user's machine
- Data Analyst has familiarity with data analysis as well as the concepts surrounding the stock prices, adjusted closings and predictions
- Analyst's PC has a working internet connection
- Analyst's machine is configured properly to connect to MySQL database using MySQL Workbench
- The admin has familiarity with Tableau Workbooks

2.2 TECHNICAL CONSTRAINTS

- Database Management : MySQL 8.0 or later
- Language : Python should be used for the backend
- Pandas : Structures data for analysis
- Pandas-DataReader : Reads pandas data structures
- FuncTools : Wraps a callable object to extend its functionality and callability
- Python statmodels : Run ARIMA, RandomForest, SVM and other data models
- NumPy : Array processing for data variables
- Math : Fundamental mathematical operations
- Statistics : Statistical Analysis operations (i.e. standard deviation)
- Python-DateUtil : Date and time module
- TabPy : Tableau Python Client/Server module

- **SQL Engine**
  - SqlAlchemy : Database Abstraction for SQL
  - pyMySQL : MySQL API Module

- **Testing**
  - PyTest : Testing module

2.3 BUSINESS CONSTRAINTS

- Connectivity issues with the internet
- Issues with the database connection
- The data source is not available or compromised
- Data analyst is not familiar with Tableau or have never use any business intelligence tool

2.4 DESIGN CONSTRAINTS

- Data must be loaded into newly structured tables
- Provide a comprehensive way to display the data
- Data integrity standards must be maintained, close prices must be validated during design, these can also be validated by the end-user at any stage
- Provide an agnostic platform for financial instruments

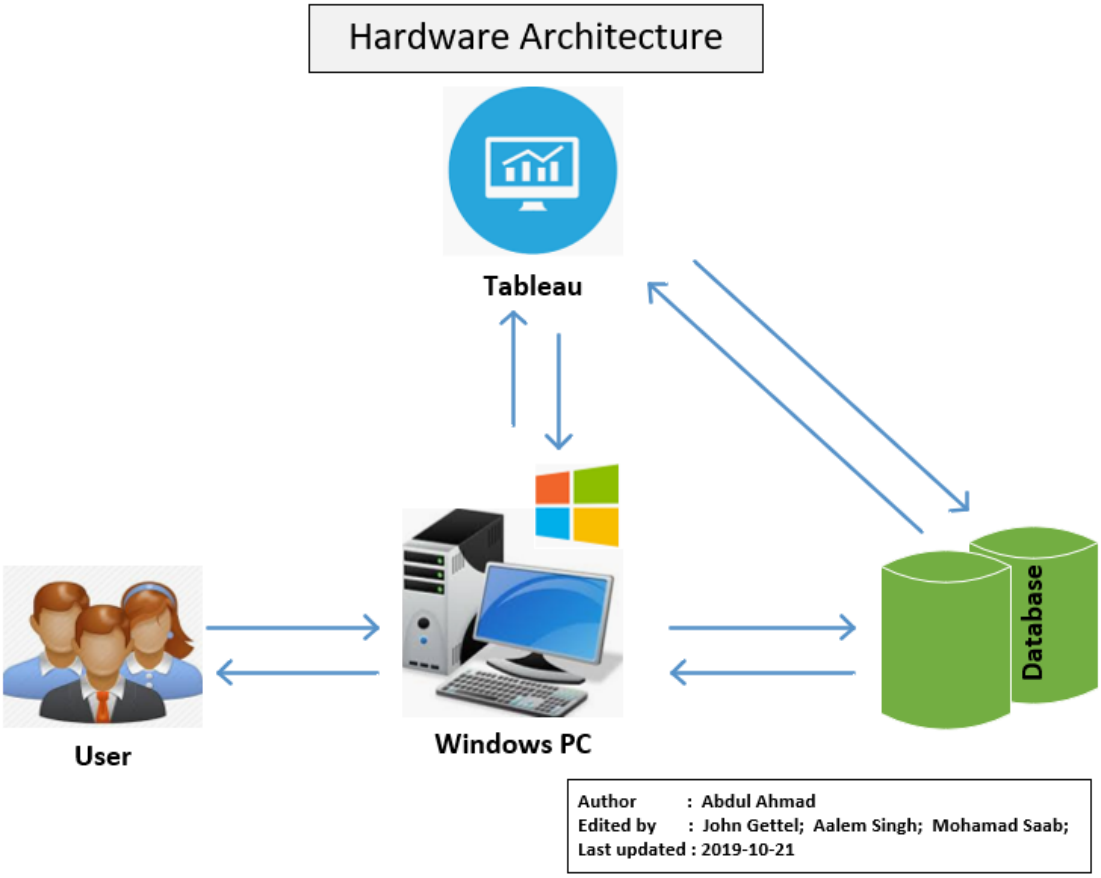
3 ARCHITECTURE DESIGN

3.1 HARDWARE ARCHITECTURE

There are three main components of the hardware architecture. The first component is the local machine that the user is using for this application. The second component is the database server. The last component is the external data source (YAHOO) from where the basic statistical data is being obtained from. The fourth physical component of the system is the user and their interaction with the front-end. The end-user will interact with the Tableau dashboard that displays all the required information. Developers will interact with the Python classes in PyCharm and run the main file which will pull the most up-to-date data. This information is handled by the application before it is stored on the database server. In turn, Tableau will fetch the information from the database and display it for the user. The following are the hardware requirements to install Tableau, MySQL 8.0, and PyCharm Professional Edition respectively:

Tableau:

- Disk Space : Minimum 5 GB (entire 5GB will not be needed)
- Memory : 4 GB RAM
- Screen Resolution : Minimum screen resolution of 1366 x 768
- Processor : Intel 10th generation or AMD 7th generation processor



MySQL 8.0:

- Disk Space : Minimum 5 GB
- Memory : 4 GB RAM
- Processor Type : 64-bit x 64-compatible AMD or Intel processor
-

Windows Server: Ideal (minimum requirement):

- Processor : Intel 10th generation with 16 cores or AMD 7th generation
- Storage : 250GB SSD (40 GB Free)
- Memory : 16GB (4GB)
- Network Connection : Ethernet 10/100/1000baseT (min)
- Active Directory : In use (optional)

PyCharm Professional Edition 2019.2:

- 4 GB RAM minimum, 8 GB RAM recommended
- 1.5 GB hard disk space + at least 1 GB for caches
- 1024x768 minimum screen resolution

3.2 SOFTWARE ARCHITECTURE

The software architecture follows the structure of a front-end, middle-layer, and backend(database). The front end is Tableau, the middle layer is Python scripts, and the backend is MySQL.

Front-End:

The user interface will be displayed in a Tableau Dashboard. This UI will provide a detailed analysis of multiple trading algorithms and strategies. These algorithms and strategies will be acting on the GM stock, the CARZ index, the PFE stock, the XPH index, and the S&P 500 index, more stocks can be added by using a SQL script to insert the symbol name into the 'InstrumentMaster' table . Tableau can run concurrently with the application, it will automatically populate by pulling data from MySQL database. The data retrieval and manipulation will be carried out by Python and SQL Stored Procedures.

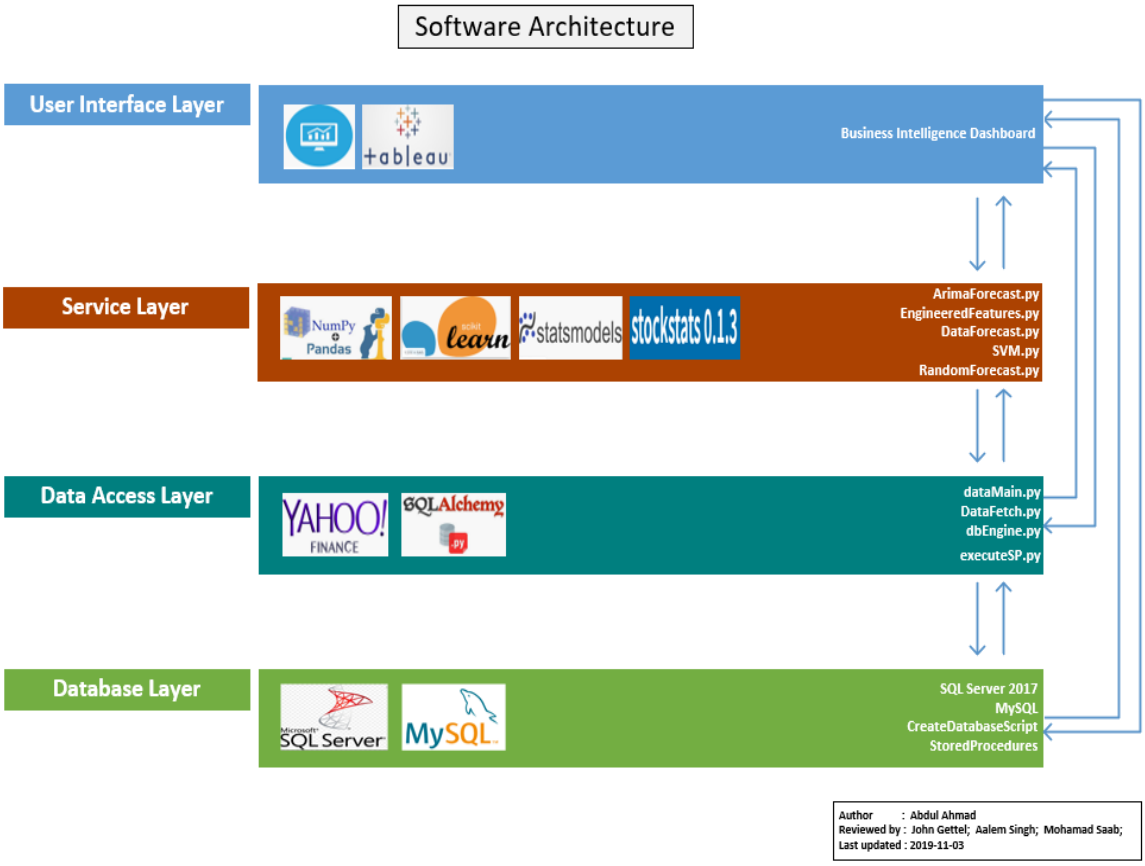
Middle-Layer:

The middle layer of Python provides the logic, which includes buy/sell signaling, algorithm-based predictions and any other manipulation to the incoming stock data statistics. It also communicates directly with the MySQL database. Python will use the Pandas Library (pandas_datareader) to communicate with the external data source (YAHOO finance data exchange server) to pull in the stock data. In addition, Python requires Anaconda (an open-source distributor) installed as well to include extra dependencies that are crucial to retrieve and manipulate the data. More precisely Python will use the Pandas, Pandas-DataReader, NumPy, SqlAlchemy, StockStats, SkLearn, StatsModels libraries.

Back-End:

MySQL will further manipulate the data as it is sent from Python, through SQL Stored Procedures. SQLAlchemy, pyODBC and, pyMySQL will orchestrate the communication between the application and the MySQL Database. The implementation of all data manipulation is currently executed through the stored procedures. The implementation of CMA and FRL is currently executed through the stored procedures.

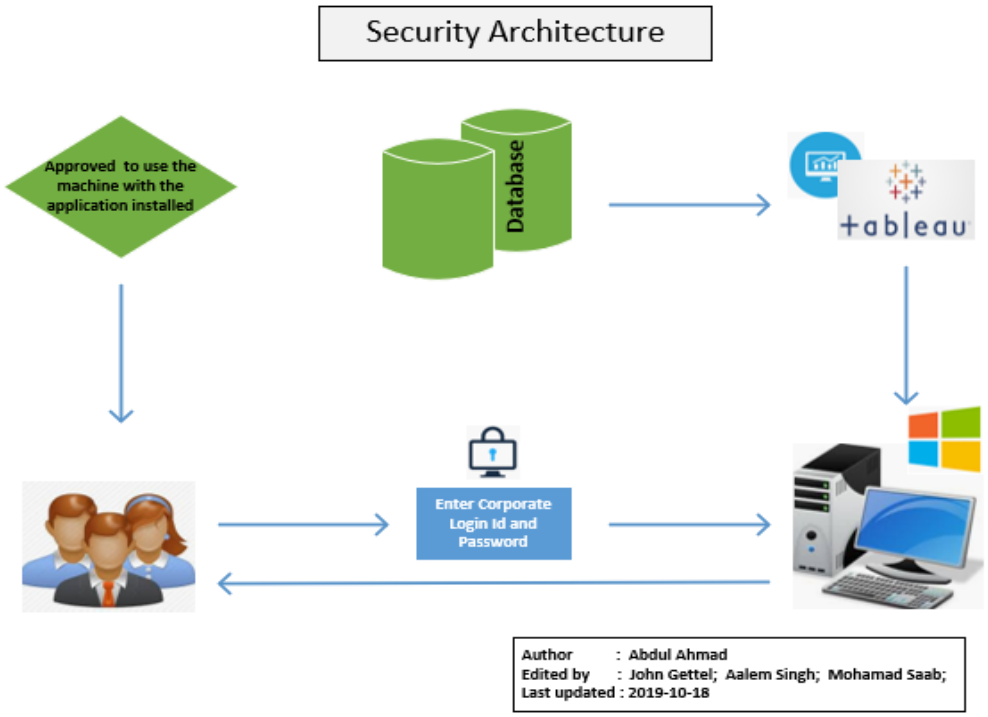
Tableau has the capability to run Python scripts, a Tableau-Python environment could be setup in Anaconda to facilitate that feature. If there is a need for two-way communication between the application, Tableau, and the database, the TabPy-Client library can be utilized (currently not used).



3.3 SECURITY ARCHITECTURE

]Application security is not a direct operational responsibility because its only accessible on the machine that has all the required Software installed, scripts loaded, and proper admin privileges given. Security will not be a responsibility of the developers. The application is designed to be installed on any Windows PC.

The application is not designed for IoT. It will only be installed and used within the intranet of a corporate department. The organization will decide which PC the application should be installed on and then anyone with access to that PC will have access to the application.



3.4 COMMUNICATION ARCHITECTURE

The communication architecture is based on the internet connection and the database connection. The internet connection is necessary to pull the latest data from YAHOO trader via Pandas API. When the main file is run, it is essential to have the internet connection to pull the latest data from the various financial instruments. Additionally, the database connection is necessary to store 3 years-worth of data in local database. Within this database there is storing and displaying of the data.

3.5 PERFORMANCE

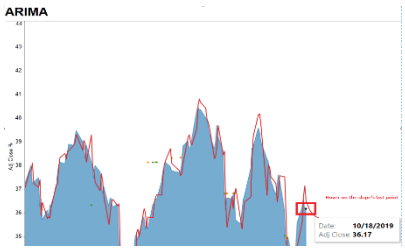
Tableau will not have updated information until the user runs the necessary scripts to pull the most up-to-date data. The speed of execution within Python will depend heavily on the connection to the internet and database connection, however, under normal circumstances a complete execution will take less than one minute.

The database will require enough storage to support the application. Since the historical data tables are replaced daily, the storage requirements will remain mostly static after the first execution for that day. Storage needs will be met by meeting the Hardware Requirements as defined above.

Once the main file has been run (without error), Tableau will display the various forecasts and calculations of the outlined financial instruments. The application is designed to be run independent of other instances, so it is possible to install the application on multiple user machines on the same network, or domain, and set them to run for separate financial instruments.

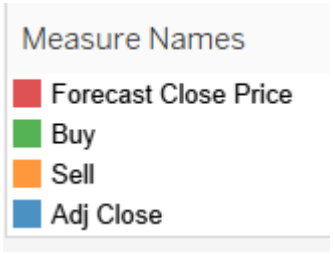
4 SYSTEM DESIGN

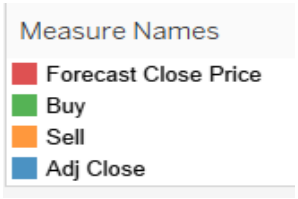
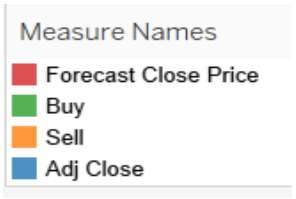
4.1 USE CASES

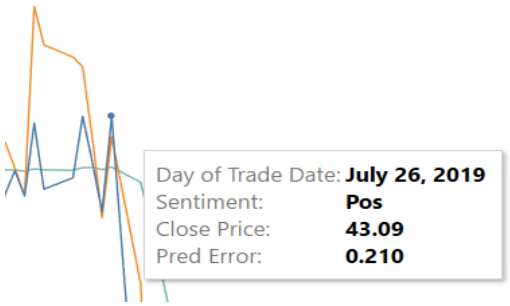
Use Case ID	**UC_1.0**		
Name	Availability of updated data in Tableau and in the database		
Created By	Abdul Ahmad	**Date Created**	10-18-2019
Description	Application must use new data every time it is opened. While the application is being opened, the data should already be pulled and up-to-date so the user is greeted with the most recently available information.		
User	Data Analyst or Developer		
Frequency	Anytime the application opens		
Primary Actor	Application User		
Preconditions	Users must have all dependencies downloaded (See Appendix A) and have the required version of the software pulled to their local machine.		
Postconditions	GMFSP_db database will be updated with lasted trading info and stock calculations. Tableau will have visualizations for trading information, algorithmic calculations, and trade simulations up to one business day before the current day.		
Steps	1 – Double click the Tableau application GM FinTech.twb from the shortcut on the desktop 2 – Click on the ARIMA tab 3 – Hover over the blue graph line, the last data point mush show Date value 4 – Check if that Date values is yesterday's date, if yes the data is updated 		
Expected Result	The new data is pulled from external data source. The data is processed, and trading calculations are made. Database is updated with latest trading info and algorithmic calculations. Tableau is updated and user can view latest trading data.		
Success Scenario	Data is updated and checked by the user following the steps listed in this use case		
Priority	**High**	**Priority Reason**	**Important to have updated data**
Priority Sequence	**H-1**		

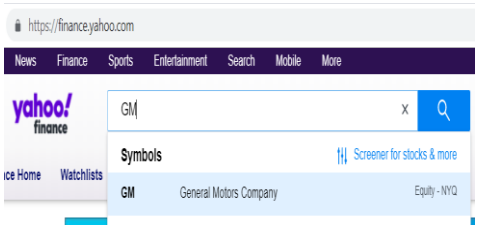
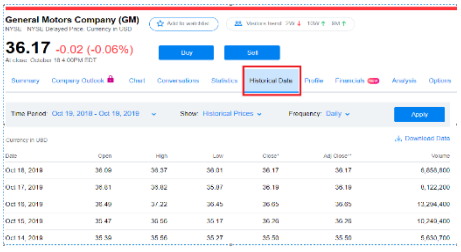
Use Case ID	**UC_2.0**		
Use Case Name	Tableau shows calculations of all predicting algorithms		
Created By	Abdul Ahmad	**Date Created**	10-18-2019
Description	Along the bottom of the Tableau dashboard, the graphs titles for the pricing data collected for each financial instrument will be displayed. Their labels will also indicate which algorithm is being used for predictions/calculations.		
User	Data Analyst or Developer		
Frequency	Anytime the application opens		
Primary Actor	Application User		
Preconditions	UC_1.0		
Postconditions	All graphs in tableau will be connected to tables in the database and calculations for the individual algorithms will also be displayed along with trading data.		
Steps	1 – On the Tableau application GM FinTech.twb look at the tabs 2 – Pick any tab and click on it 3 – Look at the data displayed 4 – It should be updated with latest data, UC_1.0 5 – The data analyst should be able to see if the data is displayed according to the expected calculations as indicated in the tab name or the title of the graph		
Expected Result	Values of related algorithms and strategies are showing as expected and designed		
Success Scenario	If the data analyst agrees with the expected calculations		
Priority	**High**	**Priority Reason**	**Validity of prediction is important**

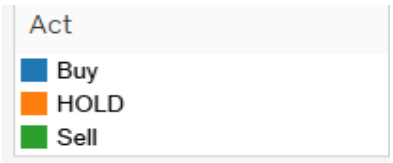
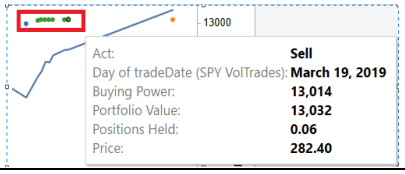
Use Case ID	**UC_3.0**		
Use Case Name	Use the zoom feature in Tableau		
Created By	Abdul Ahmad	**Date Created**	10-18-2019
Description	When in any tab in Tableau, a graph with multiple lines and or other visuals will be shown to the analyst. The analyst should be able to adjust the visuals by zooming-in or out of it. The values are sometimes small so zoom-in will help the analyst in looking at the information closely.		
User	Data Analyst or Developer		
Frequency	Anytime the application opens		
Primary Actor	Application User		
Preconditions	Users must have all dependencies downloaded (See Appendix A) and have the latest version of the software pulled to their local repository.		
Postconditions	GMFSP_db database will be updated with lasted trading info and stock calculations. Tableau will have visualizations for trading information, algorithmic calculations, and trade simulations up to one business day before the current day.		
Steps	Following are the steps to check up to date data 1 – When on any graph in Tableau, double click the left mouse button to zoom-in towards any section 2 – User can also use the wheel in the middle of the mouse, if the mouse is equipped with it 3 – User can also use a pinch action, as done on a smartphone screen to zoom-in or zoom-out of the interface		
Expected Result	The user should be able to move inwards toward the graph or white-space or move outwards from it.		
Success Scenario	User is able to zoom-in or zoom-out		
Priority	**Medium**	**Priority Reason**	**Visualizations of data is important**
Priority Sequence	**M-1**		

Use Case ID	**UC_4.0**		
Use Case Name	View algorithmic predictions for each stock symbol and strategy. This is the extended line that must show future dates.		
Created By	Abdul Ahmad	**Date Created**	10-18-2019
Description	The user will be able to hover their mouse over future predicted days and view the predicted closing price that the algorithm gives.		
User	Data Analyst or Developer		
Frequency	Anytime the application opens		
Primary Actor	Application User		
Preconditions	UC_1.0, UC_2.0		
Postconditions	The 'adjusted closed' price for future days will be shown		
Steps	1 – The analyst should go to any tab in Tableau 2 – Look to the right of the user interface look for a box that displays  3 – Hover over the red line because the 'Forecast Close Price' is shown with a red line 4 – Check if the line shows future date values		
Expected Result	The red line should have values for future days		
Success Scenario	The red 'Forecast Close Price' line must be extended beyond today's date and it shows predicted values		
Priority	**High**	**Priority Reason**	**Predictions are the most important component of the project**
Priority Sequence	**H-2**		

Use Case ID	**UC_5.0**		
Use Case Name	Visual Representation of buying and selling signals		
Created By	Abdul Ahmad	**Date Created**	10-19-2019
Description	Each tab related to an algorithm/strategy will show color coded buy/sell signal dots along with the forecast and prediction lines. The trade simulator shows all points of buying and selling and holding decisions and in an easy to view/color coordinated fashion.		
User	Data Analyst or Developer		
Frequency	Anytime the application opens		
Primary Actor	Application User		
Preconditions	UC_1.0, UC_2.0, UC_4.0		
Postconditions	A graph showing closing price, forecast and trade decisions will be shown. Color coded legends will appear on the right side of the UI for easy understanding and differentiating different lines and markers. 		
Steps	1 – The analyst should go to any tab in Tableau 2 – Check if the tab is designed to show Buy/Sell values 3 – If yes to step 2, look to the right of the user interface look for a box that displays  3 – See if the graph in the UI shows ‘Buy’ in green color and ‘Sell’ in light orange color 4 – Certain Buy/Sell strategies are not designed to show the values		
Expected Result	1 - Tableau shows updated data with graphs showing signal values 2 – Color coding of the ‘Buy’ and ‘Sell’ must be clear and valid according to the legend on the right side		
Success Scenario	The color-coded ‘Buy’ and ‘Sell’ values appear on the graph in the UI		
Priority	**High**	**Priority Reason**	**Buy Sell signaling is an important component of the project**
Priority Sequence	**H-3**		

Use Case ID	**UC_6.0**		
Use Case Name	View margin of error between actual close prices and predicted closed prices.		
Created By	Abdul Ahmad	**Date Created**	10-19-2019
Description	To visually see the effectiveness of the algorithms a margin of error calculation will be done with the known stock data and forecasted stock data. The forecast algorithm will assume to not know the stock closing price of prior days(adjustable by developer based on the algorithm). It will forecast the closing price of the stock. The forecasted closing price and the actual closing price will be put into a margin of error algorithm and this value will be shown in the graph once hovering over the specific graph in Tableau(does not apply to all tabs/algorithms).		
User	Data Analyst or Developer		
Frequency	Anytime the application opens		
Primary Actor	Application User		
Preconditions	UC_1.0, UC_2.0, UC_4.0		
Postconditions	User will see a margin of error value for each trade date		
Steps	1 – User will open the Tableau .twb application 2 – User will hover over any datapoint in the algorithmic graphs 3 – A Pred Error value should appear, it may appear as a line graph on few tabs		
Expected Result	A margin of error calculation will be displayed in a box labeled as Pred Error. 		
Success Scenario	The value does not appear on each data marker on the graph in the UI		
Priority	**Medium**	**Priority Reason**	**Clean and clear visualization are important**
Priority Sequence	**M-2**		

Use Case ID	**UC_7.0**		
Use Case Name	Show validated base statistics values by pulling data records from the database table and comparing it to https://finance.yahoo.com/		
Created By	Abdul Ahmad	**Date Created**	10-19-2019
Description	Validated values with ability to perform quality check of basic statistical values pulled from the YAHOO's financial data exchange. Sometimes the exchanges has issues and it is a good practice to validate the data on routine basis.		
User	Data Analyst or Developer		
Frequency	Anytime the application opens		
Primary Actor	Application User		
Preconditions	UC_1.0		
Postconditions	Analyst will validate the values in the 'InstrumentStatistics' table. This table holds the 'Adj Close' price value as well which is the most important for our project.		
Steps	1 – Note down the 'Adj Close' value from a particular day 2 – Open a web browser(Firefox, Chrome, Internet Explorer) 3 – Go to the website https://finance.yahoo.com/ 4 – Enter the stock, symbol you are validating values for in the text box and press the blue search button(magnifying glass)  5 – Click on the 'Historical Data' link 6 – Compare the values shown in the SQL query results from step 1 to the numbers that appear on this page. Look at the data and values. 		
Expected Result	All values in our database table 'InstrumentStatistics' will match with the YAHOO's online values.		
Success Scenario	**Passed :** The values match between database and YAHOO online **Failed :** The value appears on each data marker on the graph in the UI		
Priority	**Medium**	**Priority Reason**	**Valid base statistics drive all other calculations**
Priority Sequence	**M-3**		

Use Case ID	**UC_8.0**		
Use Case Name	View volumetric trading information in trade graph		
Created By	Abdul Ahmad	**Date Created**	10-19-2019
Description	The analyst will be able to go to the trade graphs at the bottom of tableau and hover over each trading decision and see how much of the money the algorithm has decided to invest based on the confidence indicated by the trading algorithms/strategies.		
User	Data Analyst or Developer		
Frequency	Anytime the application opens		
Primary Actor	Application User		
Preconditions	UC_1.0		
Postconditions	User will be able to view how much stock is traded on each trade date		
Steps	1 – On the Tableau application GM FinTech.twb look at the tabs labeled 'Vol Trade' 2 – The 'Buy', 'Sell' and, 'Hold' values are color-coded  3 – Look at the graph displayed on the UI 4 – It should be updated with latest data, UC_1.0 5 – The data analyst should be able to see if the data is displayed according to the expected calculations and 'Buy', 'Sell' or 'Hold' signals resonate with the market. 6 – Analyst can hover over the bubble shaped color-coded markers to see 'Buying Power', 'Portfolio Value', 'Position Held' and, 'Price'. 		
Expected Result	1 – Analyst opens Tableau 2 – Analyst clicks on tabs containing the word 'Vol Trade' 3 – Analyst is visually able to see how much stock is sold, bought, or held for each trade date.		
Success Scenario	**Passed :** The values resonate with the analyst **Failed :** The value appears to be out of sorts according to the analyst		
Priority	**Low**	**Priority Reason**	**Low priority in the requirements**
Priority Sequence	**L-1**		

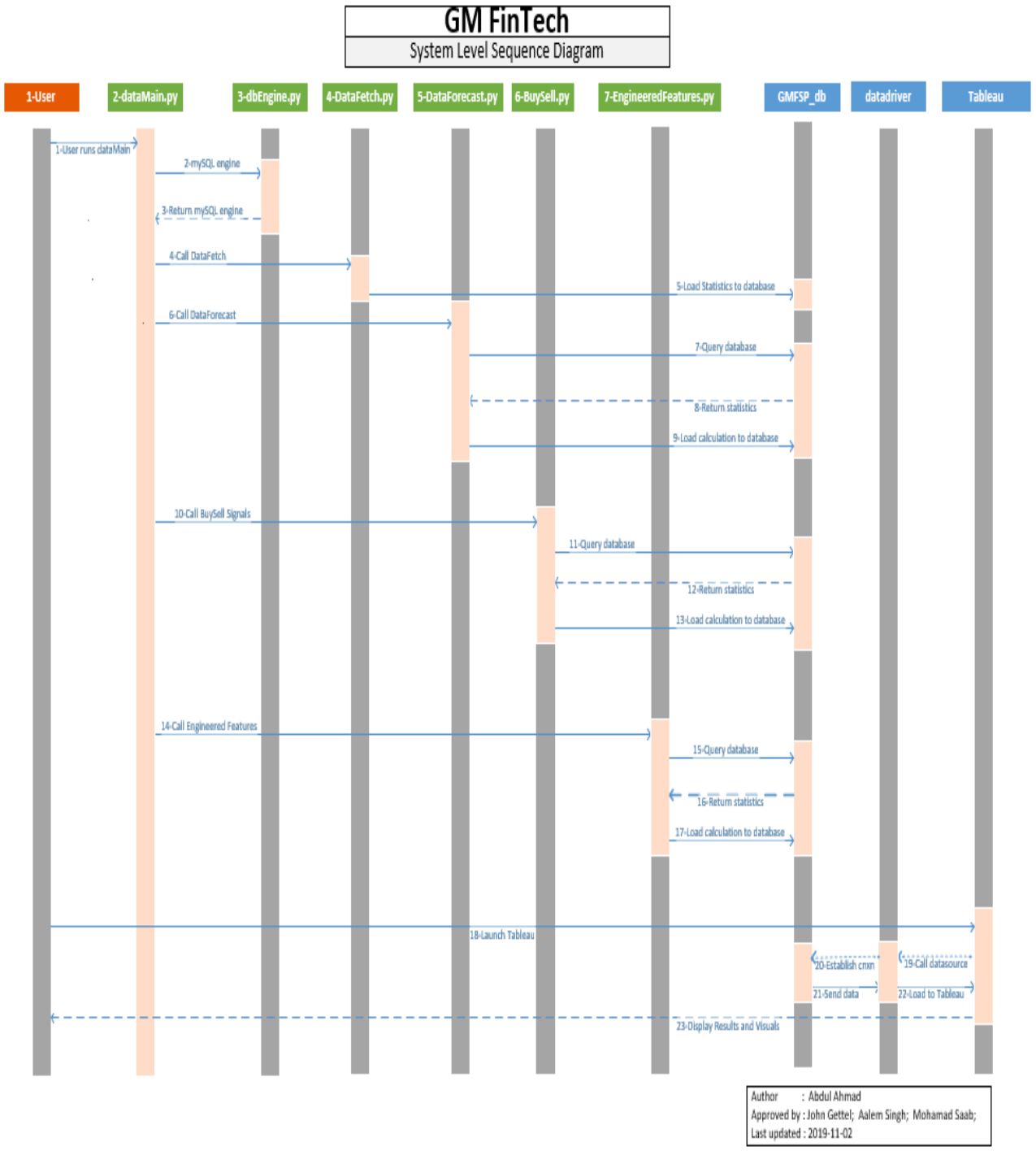
4.1.1 TRACEABILITY MATRIX

Functional Requirement ID	Functional Requirement Name	Use Case ID	Use Case Name
FR_0.1	Login Information	**UC_1.0**	Availability of updated data in Tableau and in the database
FR_1.0	Database must be restructured with a completely new optimized design	**UC_1.0**	Availability of updated data in Tableau and in the database
		UC_7.0	Validate base values by pulling data records from the database table and comparing it to values found at https://finance.yahoo.com/
FR_2.0	Database Management System needs to be converted over to MySQL from currently locally deployed MS SQL Server	**UC_1.0**	Availability of updated data in Tableau and in the database
		UC_2.0	Tableau shows calculations of all predicting algorithms
FR_3.0	Functionality of existing prediction techniques must be changed and improved	**UC_6.0**	View margin of error between actual closed prices and predicated close prices.
FR_4.0	Introduce new stock symbols such as GM, CARZ and more as needed by client with least amount of manual effort by the user	**UC_1.0**	Availability of updated data in Tableau and in the database
		UC_7.0	Show validated base statistics values by pulling data records from the database table and comparing it tohttps://finance.yahoo.com/
FR_5.0	Introduce new algorithms for predictions using adjusted close price data label	**UC_4.0**	View algorithmic predictions for each stock symbol and strategy. This is the red extended line that must show future dates.
		UC_6.0	View margin of error between actual closed prices and predicated close prices.
FR_6.0	Data in Tableau must be up to date	**UC_1.0**	Availability of updated data in Tableau and in the database
		UC_2.0	Tableau shows calculations of all predicting algorithms
FR_7.0	Tableau shows all updated and relevant graphs	**UC_1.0**	Availability of updated data in Tableau and in the database
		UC_2.0	Tableau shows calculations of all predicting algorithms
		UC_4.0	View algorithmic predictions for each stock symbol and strategy. This is the red extended line that must show future dates.
		UC_5.0	Visual Representation of buying and selling signals
FR_8.0	Tableau shows CMA, FRL and EMA(cross moving average and	**UC_1.0**	Availability of updated data in Tableau and in the database

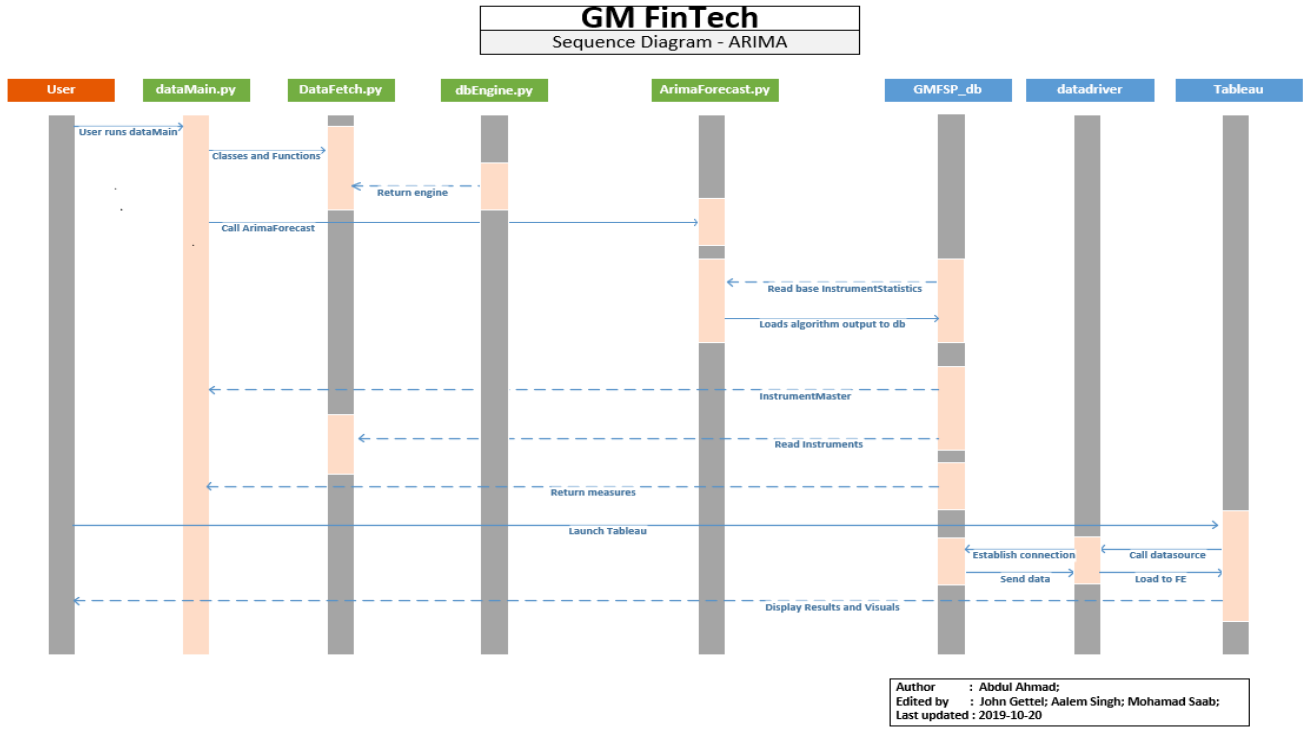
	Fibonacci retracement, and exponential moving average) graphs.	**UC_4.0**	View algorithmic predictions for each stock symbol and strategy. This is the red extended line that must show future dates.
FR_9.0	User can view algorithm predictions within the graphs and select which specific algorithm would like to be viewed	**UC_1.0**	Availability of updated data in Tableau and in the database
		UC_4.0	View algorithmic predictions for each stock symbol and strategy. This is the red extended line that must show future dates.
FR_10.0	Visual Representation of buying and selling in portfolio	**UC_3.0**	Use the zoom feature in Tableau
		UC_4.0	View algorithmic predictions for each stock symbol and strategy. This is the red extended line that must show future dates.
		UC_5.0	Visual Representation of buying and selling signals
FR_11.0	Display the variables in the CMA calculation and the trade signal	**UC_2.0**	Tableau shows calculations of all predicting algorithms
FR_12.0	Display the projected calculation for week CMA, short-term CMA, and long-term CMA	**UC_2.0**	Tableau shows calculations of all predicting algorithms
FR_13.0	Display the variables in the FRL calculation and the trade signal	**UC_8.0**	View volumetric trading information in trade graph
FR_14.0	Display the projected calculation for FRL and the natural numbers in the formula (23%, 38.2%, 61.8%, 79.4%)	**UC_1.0**	Availability of updated data in Tableau and in the database
		UC_2.0	Tableau shows calculations of all predicting algorithms
FR_15.0	Display the variables in the EMA calculation and the trade signal	**UC_2.0**	Tableau shows calculations of all predicting algorithms
FR_16.0	Display the projected calculation for short EMA, medium EMA, and long term EMA	**UC_2.0**	Tableau shows calculations of all predicting algorithms
FR_17.0	Navigate through the different graphs (CMA, FRL, Trade) for the different markets	**UC_2.0**	Tableau shows calculations of all predicting algorithms
		UC_3.0	Use the zoom feature in Tableau
FR_18.0	View data values on Tableau from the database	**UC_1.0**	Availability of updated data in Tableau and in the database
		UC_2.0	Tableau shows calculations of all predicting algorithms
FR_19.0	Tooltip Windows	**UC_8.0**	View volumetric trading information in trade graph
FR_20.0	Zoom Feature	**UC_3.0**	Use the zoom feature in Tableau

4.2 SEQUENCE DIAGRAMS

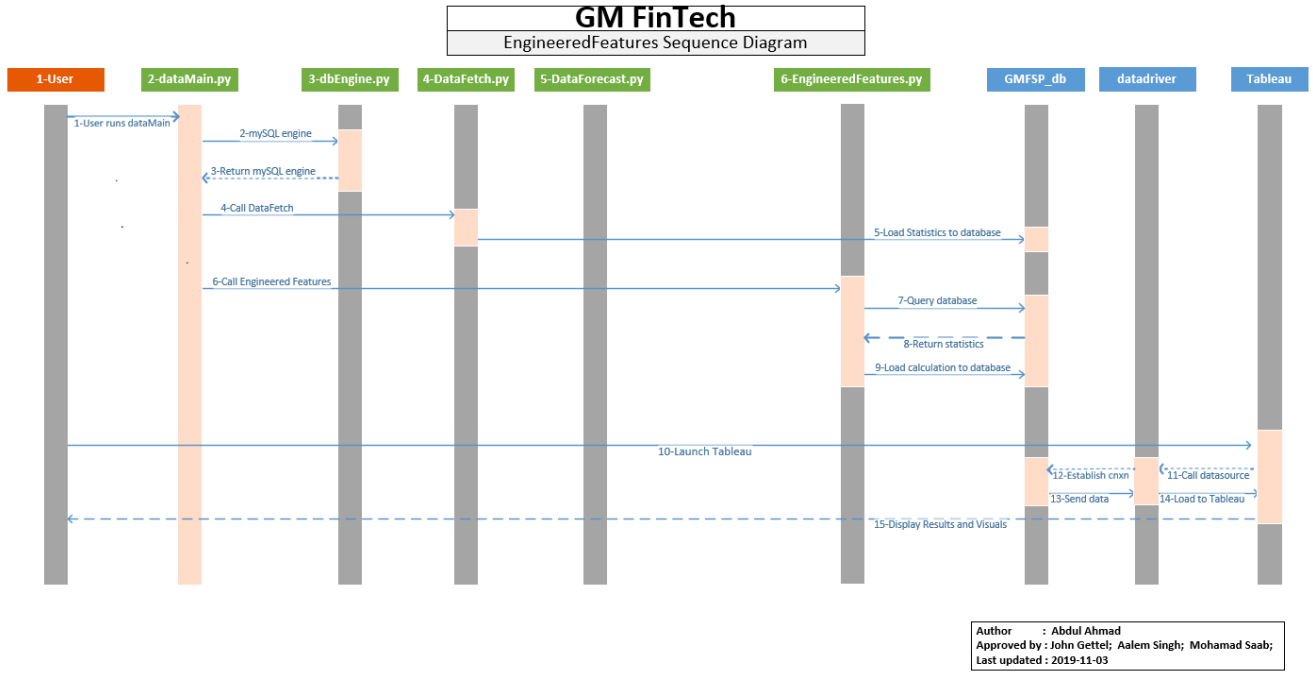
4.2.1 SYSTEM LEVEL DIAGRAM



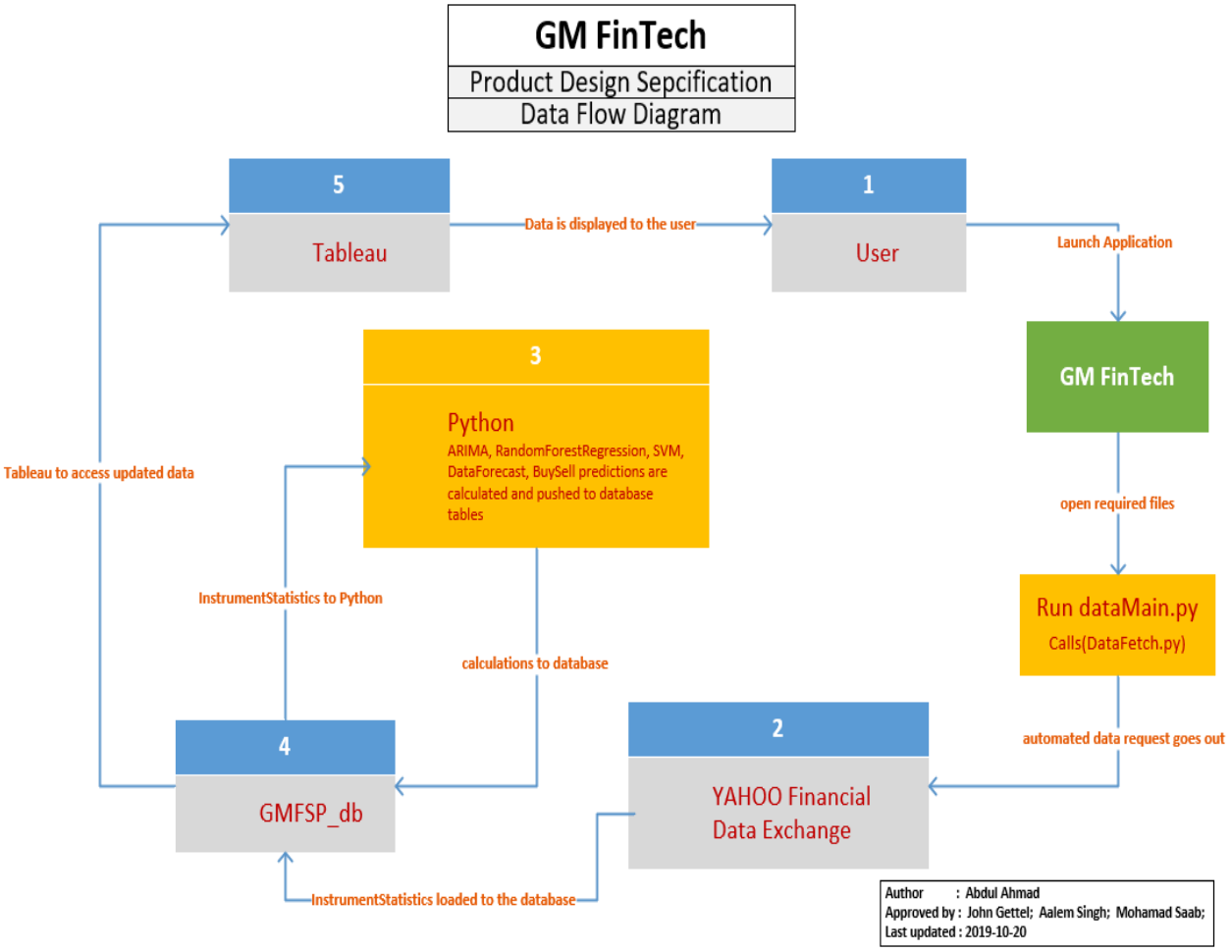
4.2.2 ARIMA SEQUENCE DIAGRAM



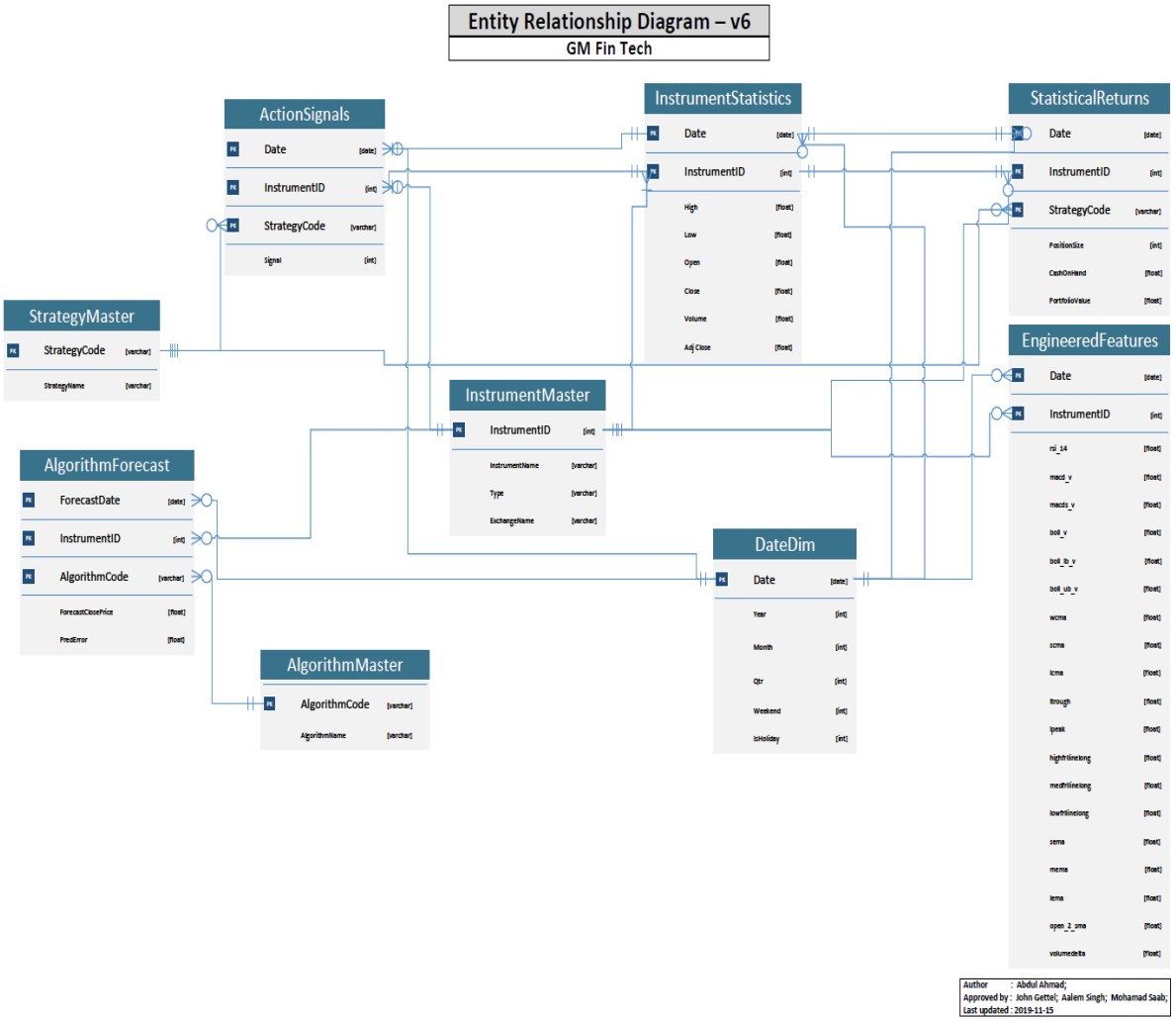
4.2.3 ENGINEERED FEATURES SEQUENCE DIAGRAM



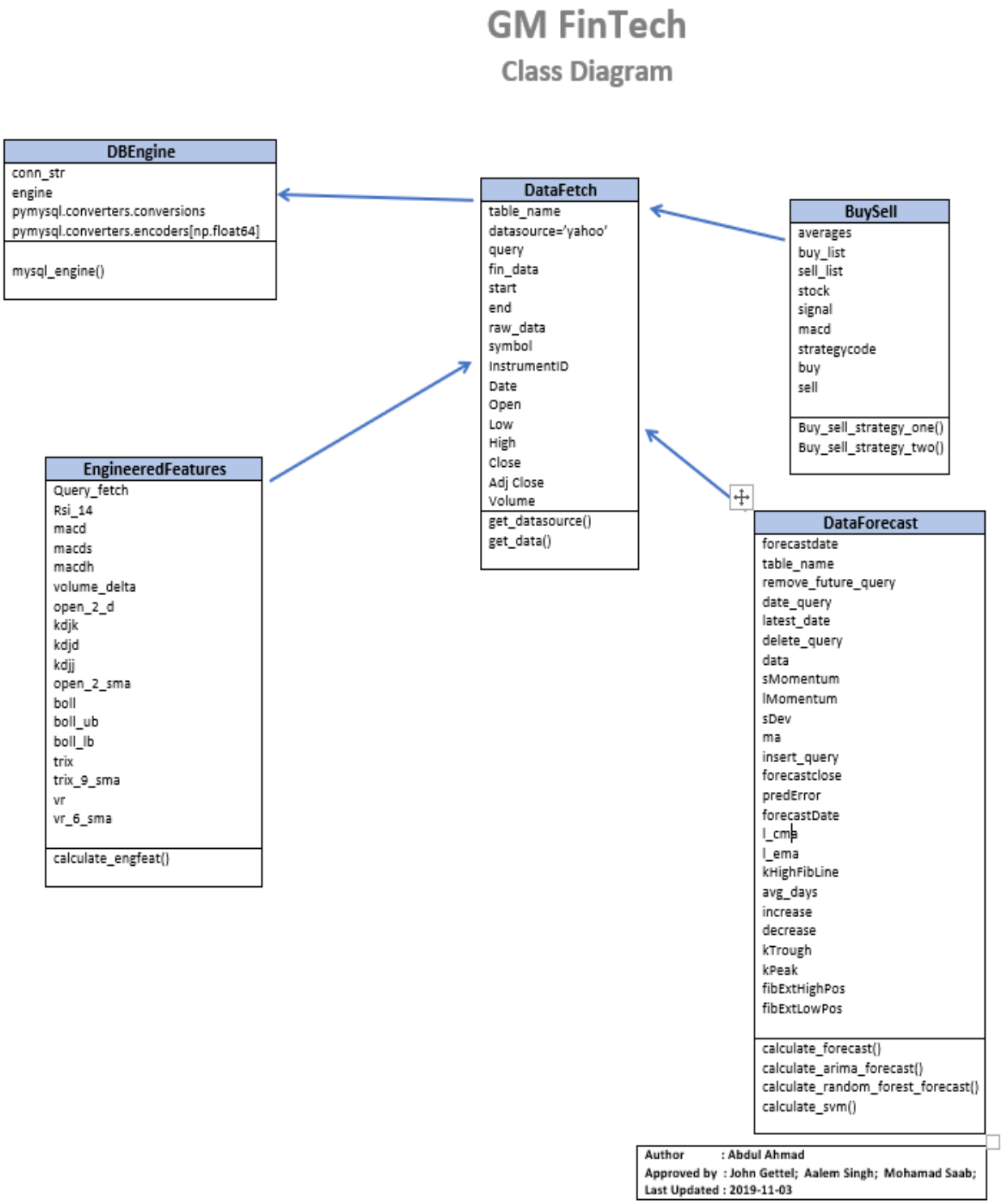
4.3 DATA FLOW DIAGRAM



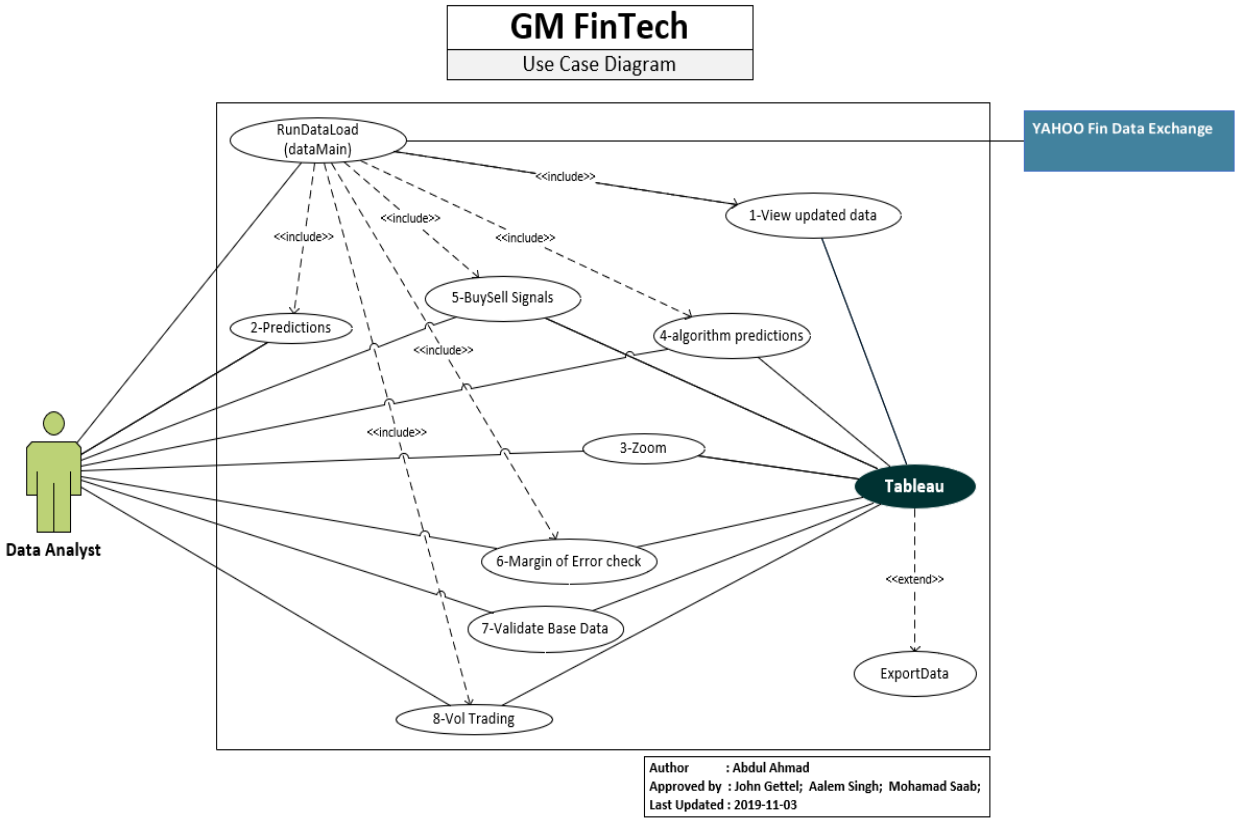
4.4 DATABASE DESIGN



4.5 CLASS DIAGRAM



4.6 USE CASE DIAGRAM



4.7 APPLICATION PROGRAM INTERFACES

Pandas
We will be using Pandas API for connection to our external data source of YAHOO financial data exchange server. Documentation of the Panda's API can be found on their website https://pandas.pydata.org/pandas-docs/stable/reference/index.html.

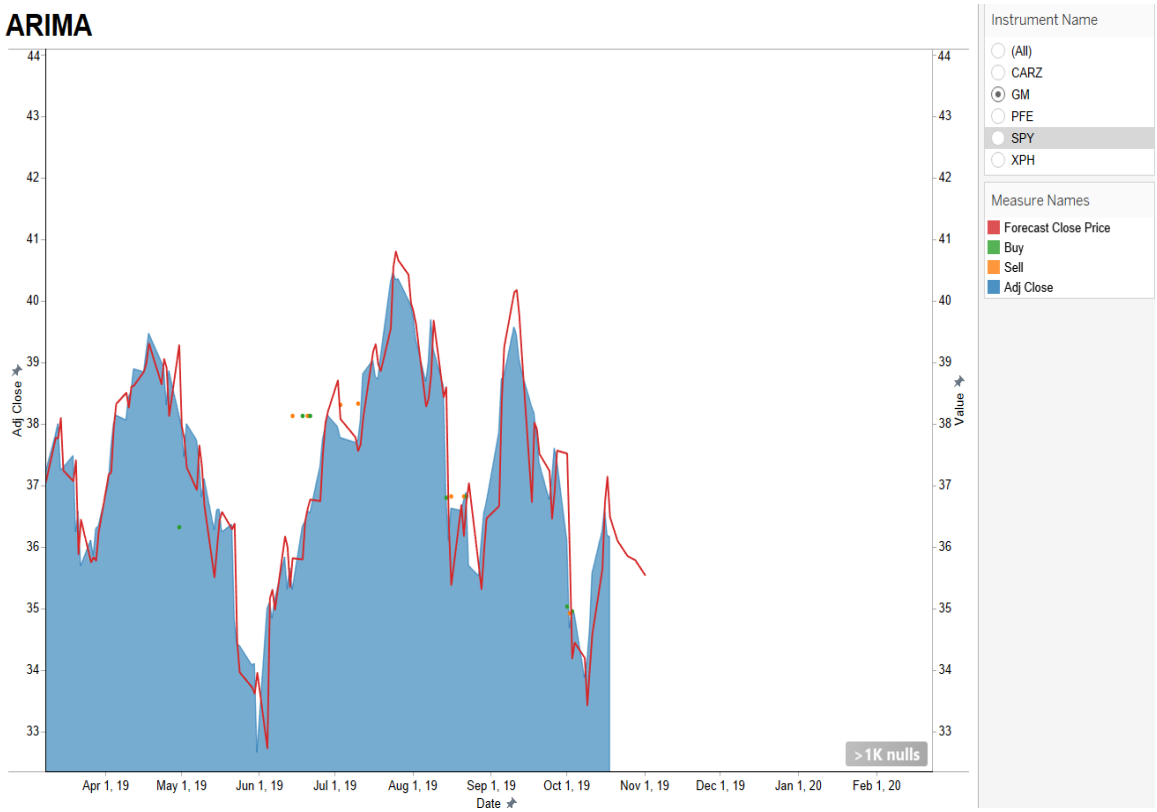
YAHOO Financial Date Exchange
Following data elements will be fetched and stored in the database's 'InstrumentStatistics' table: Date, High, Low, Open, Close, Volume and, Adj Close. InstrumentID columns will be appended using the 'DataFetch' class.

YAHOO finance provides informational purposes only and is not intended for trading or investing purposes. Here you can find all exchanges and markets that YAHOO finance covers. Each row includes the exchange suffix (which you can add to the end of the instrument ticker to look up the quote as traded on that exchange), the time delay between the exchange and Yahoo Finance, and the data provider. YAHOO finance provides all information AS-IS and is not responsible for any inaccuracy. US quotes are real-time for NASDAQ, NYSE, and NYSE American when available from Nasdaq Last Sale and if not available it will appear delayed from the consolidated tape. See delay times for other exchanges below. Quotes are updated automatically but will be turned off after 25 minutes of inactivity.

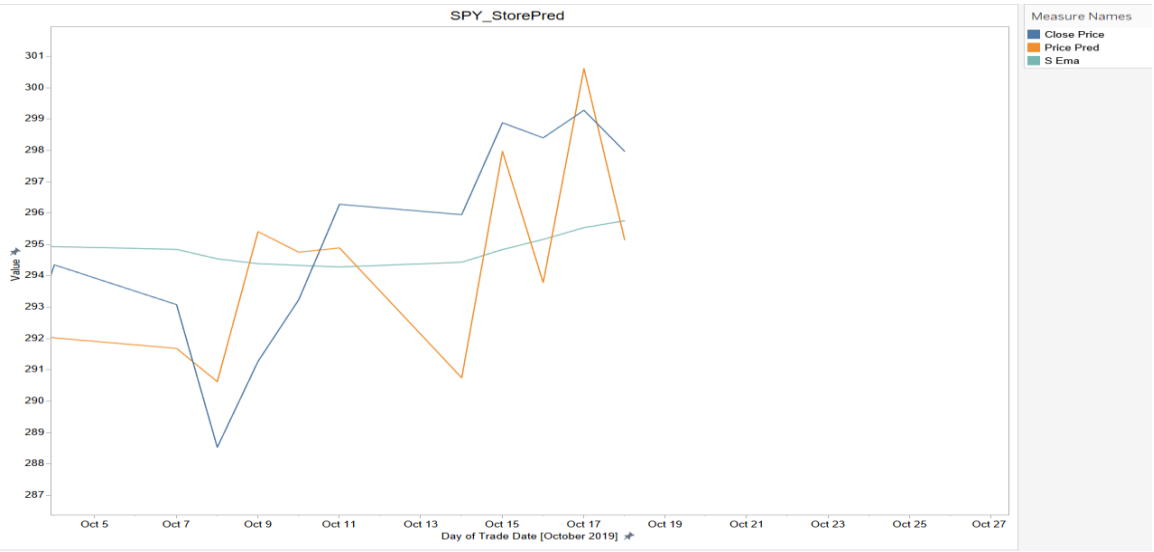
SqlAlchemy
We will be using SQLAlchemy for ease of connection and manipulation of stock data in the database within Python. Documentation of SQLAlchemy can be found on their website:
https://www.sqlalchemy.org/

4.8 USER INTERFACE DESIGN

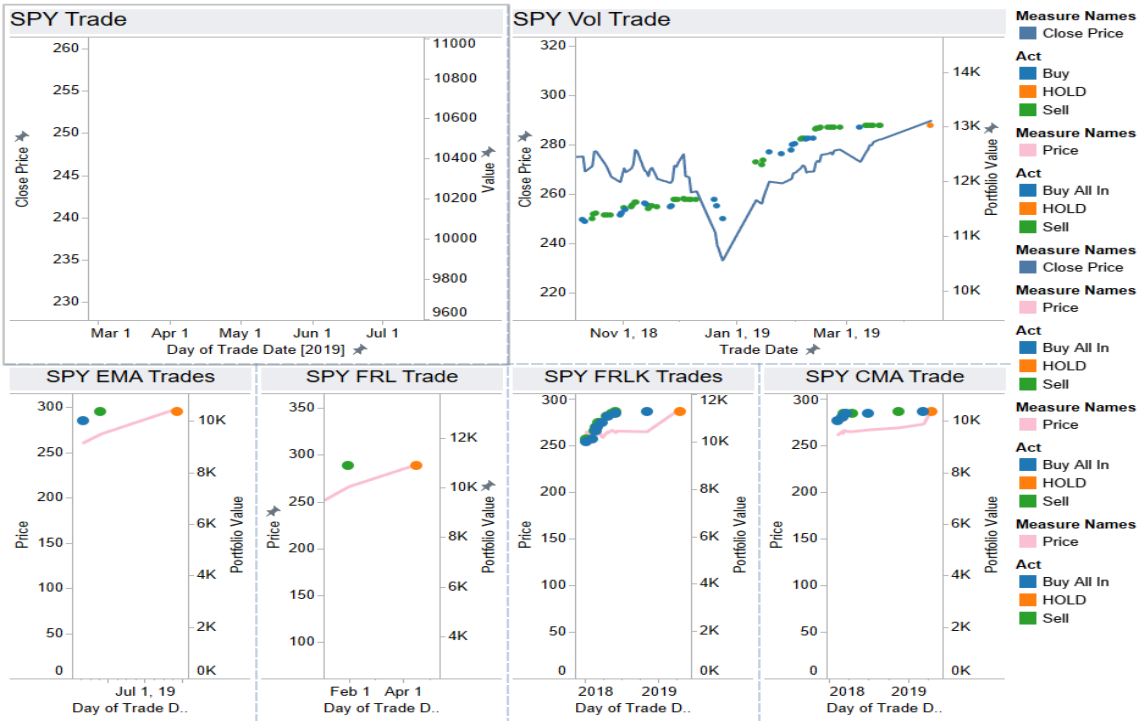
The User Interface(UI) is built using Tableau. Tableau is a popular business intelligence and dashboard tool. The UI will be made available on a local PC/server. All the visuals presented in the user interface will be supported by a database that obtains information from YAHOO financial data exchange. The user interface will offer informative interaction, users will be able to 'hover over' data points on the graphs to receive information pertaining to a data point. In addition, users will be able to zoom in and out. Tableau will connect to the database for visualization purposes, it will not directly have access to the database to write changes. Analyst will be able to switch between the different workbooks or graphs using the tabs at the bottom of the Dashboard.



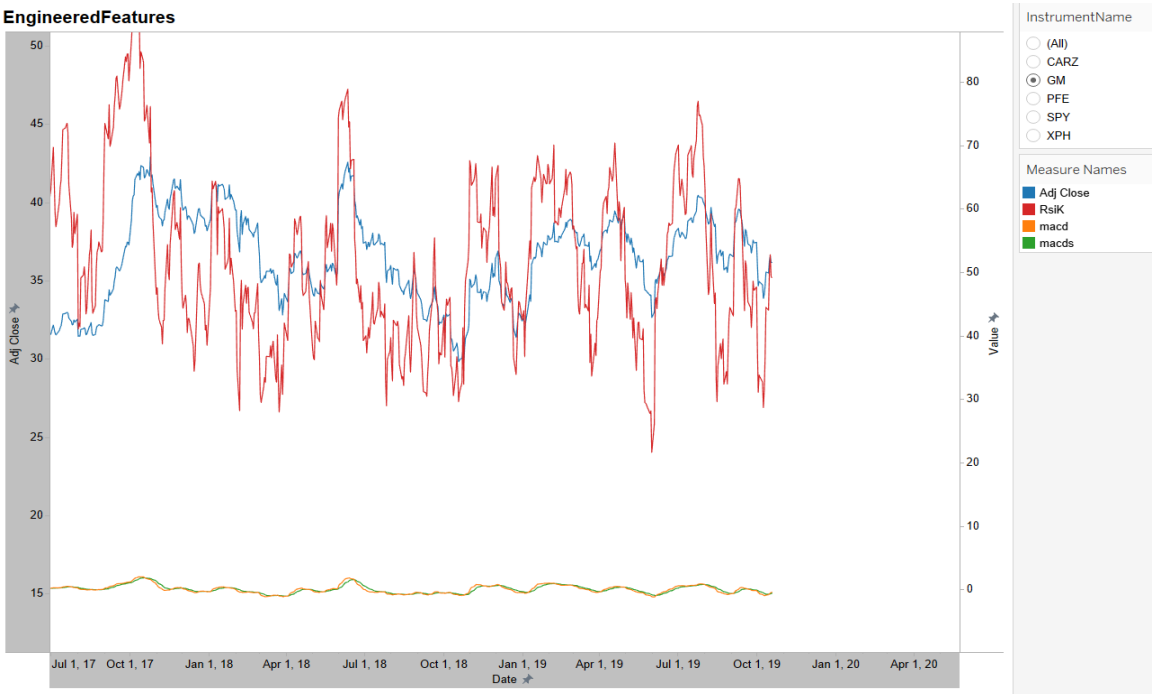
Each algorithm will have its own graphs for the respective forecast and buy/sell strategy. Above screenshot is for ARIMA algorithm interface. Analyst can use the toggle radio button on the right side to pick any desired stock/instrument and see the results of ARIMA forecast.



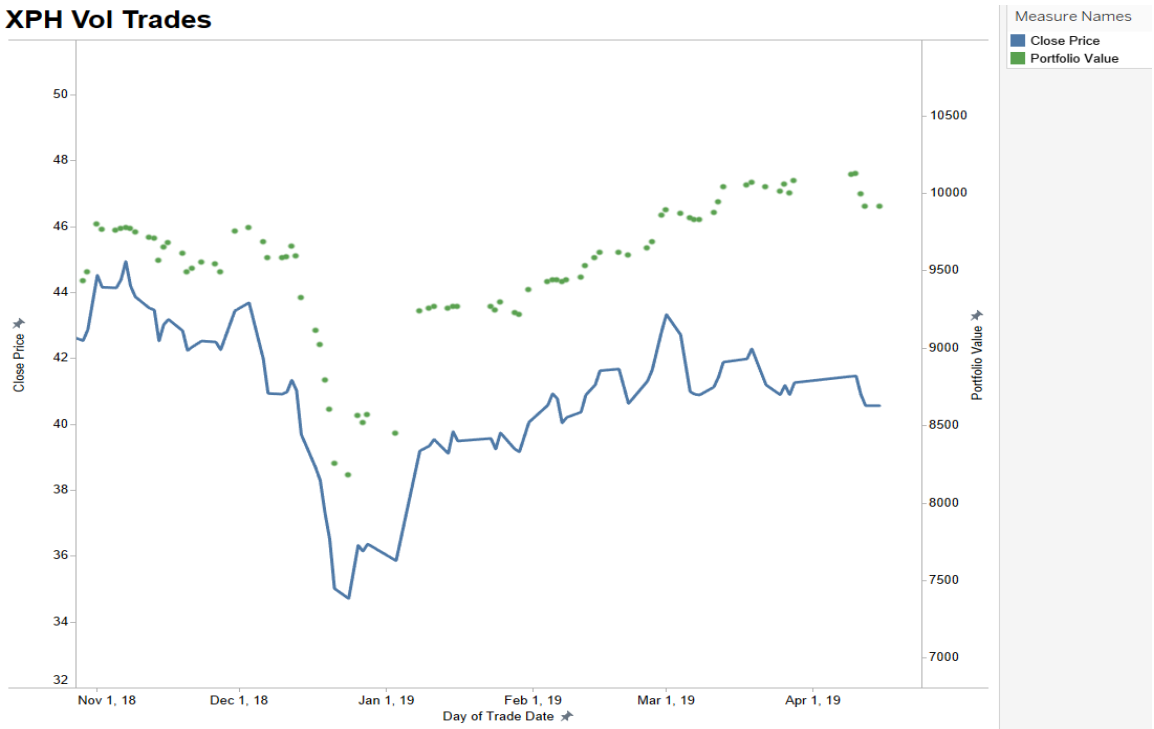
Each stock/instrument will have its own graphs for the respective forecast and buy/sell strategy.



There will be views available that combine different algorithms and strategies on to one single sheet.



A UI utilizing the StockStats library to implement Buy/Sell signaling and other informative measure. This will facilitate the analyst in making actionable decisions.



Each market will also have a simulated trading portfolio with an initial buying power of $10000 with the simulation beginning from 3 years go to present. This graph will show all the buy and sell decisions that the algorithm made as well as the total value of the portfolio.

5 PRODUCT DESIGN SPECIFICATION APPROVAL

The undersigned acknowledge that they have reviewed the GM FinTech **Product Design Specification** document and agree with the documented design and architecture. Any changes in design or requirements mentioned in this Product Design Specification document will be coordinated with and approved by the undersigned or their designated representatives.

Team Lead, Front End Lead, Back End Lead

Signature		**Date**	
Print Name			
Title			
Role			

Signature		**Date**	
Print Name			
Title			
Role			

Signature		**Date**	
Print Name			
Title			
Role			

APPENDIX A: REFERENCES

- Anaconda Enviornment
- pyODBC
- SQLAlchemy
- Pandas
- Pandas-datareader
- DateTime
- Statistics
- Math
- NumPy
- pyUse

SOFTWARE LIST

MySQL 8.0 download
https://www.mysql.com/downloads/

MySQL 8.0 documentation
https://dev.mysql.com/doc/

Anaconda Environment
https://www.anaconda.com/download/#windows

PyCharm free student
https://www.jetbrains.com/student

Tableau free student
https://www.tableau.com/academic/students

GitLab
https://about.gitlab.com

APPENDIX B: KEY TERMS

Term	Description
Anaconda	Software used to call Python libraries
API	Application Program Interface. A set of routines, protocols, and tools for building software applications.
ARIMA	Auto-Regressive Integrated Moving Average
Back-End	Database that supports the middle layer and front-end
CMA	Cross Moving Average
EMA	Exponential Moving Average
Financial Instrument	Financial instruments are monetary contracts between parties. They can be created, traded, modified and settled. They can be cash, evidence of an ownership interest in an entity, or a contractual right to receive or deliver cash.
FRL	Fibonacci Retracement Line
Functional Requirement	A requirement that is deemed part of important and critical functions of a Software
GMFSP_db	Name of the database used to store data
IoT	Internet of Things. A relatively new term for the internet.
Middle-Layer	Python code that runs the algorithms, data fetch and, data loads
Random Forest Regression	An algorithm for making predictions on the data using the learning method of classification.
Tableau	Software used to visualize data
Use case	Steps containing actions to use a functionality of the Software
Ticker	Stock symbol or instrument