### **Part 1. Time Series Application Architecture 50pts**

Many web applications follow MVC schema.

* **Model**: The backend that contains all the data logic
* **View**: The frontend or graphical user interface (GUI)
* **Controller**: The brains of the application that controls how data is displayed

You should decide and specify

1. where your data is stored: cloud server (e.g. firebase), stand-along - 5pts

The data will be called through the Yahoo Finance API so there is no need to store the data. Since the project revolves around stock and ETF (Exchange-Traded Fund) analysis, it is strongly preferred to have up-to-date information each time the user makes a request.

We do have local files stored for the previous Anomaly detection assignment, but the dashboard will be making API calls using the quantmod package’s [getSymbols() function](https://www.rdocumentation.org/packages/quantmod/versions/0.4.20/topics/getSymbols) to gather the time series data.

The team agreed to use Github to work collaboratively on the project.

1. what languages will be used to build the back-end: python, R, javascript etc -5pts

The language of choice to build the back-end will be R. The concepts covered in this course were primarily used in R, and this programming language is user-friendly with an active community.

1. how you will be accessing data and web app: how your app is connected to data and is it secure (e.g. you have admin privileges, login, and users cannot modify stored data etc) - 5pts

As previously mentioned, the app will be retrieving the data through the Yahoo Finance API. The data is publicly accessible and open-sourced, and therefore security is not an issue. Regardless, users will not be able to access the raw data to tamper with it given the setup of our project.

1. what will you use to create a front-end layout (HTML, CSS, js). - 5pts

Since the programming language for this project is R, Shiny is the best choice to create the front-end layout. Shiny allows developers to create interactive dashboards without knowing HTML, CSS, and JavaScript.

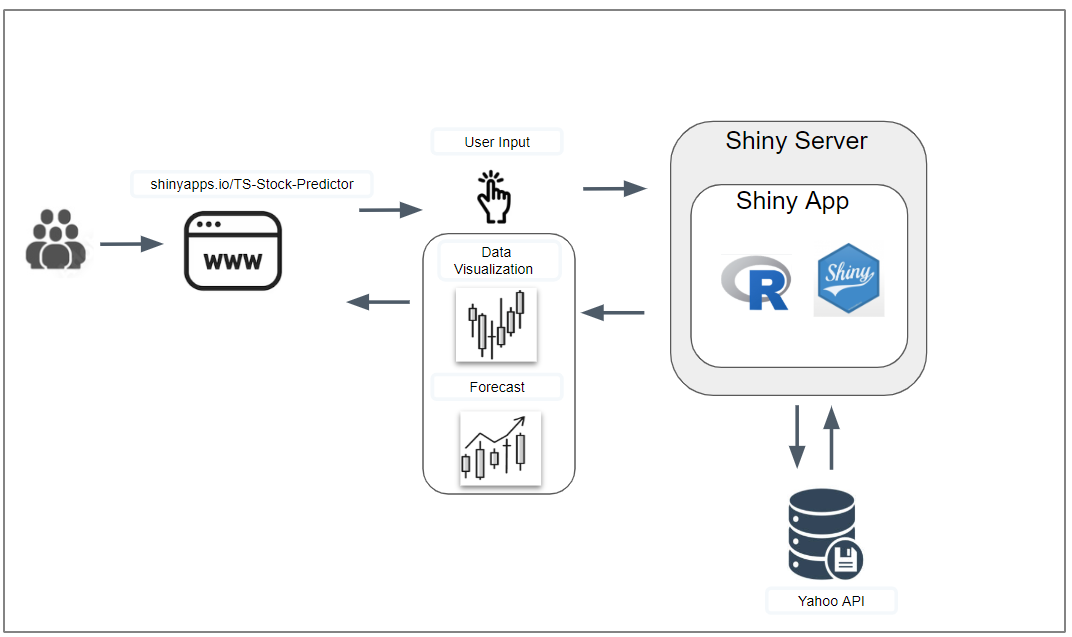
1. where your application is deployed: shiny server, firebase, heroku, pythonenywhere ... - 5pts

The application will be deployed through Shinyapp.io, given that we are using Shiny as the front-end layout.

1. how you will provide interactivity for your app. Note - users should be able to click, select, view etc - 5pts

Through the library package Shiny, Quantmod, and Forecast, the team will be able to produce an interactive dashboard for users. Please refer to the screenshots below to see the prototype dashboard.

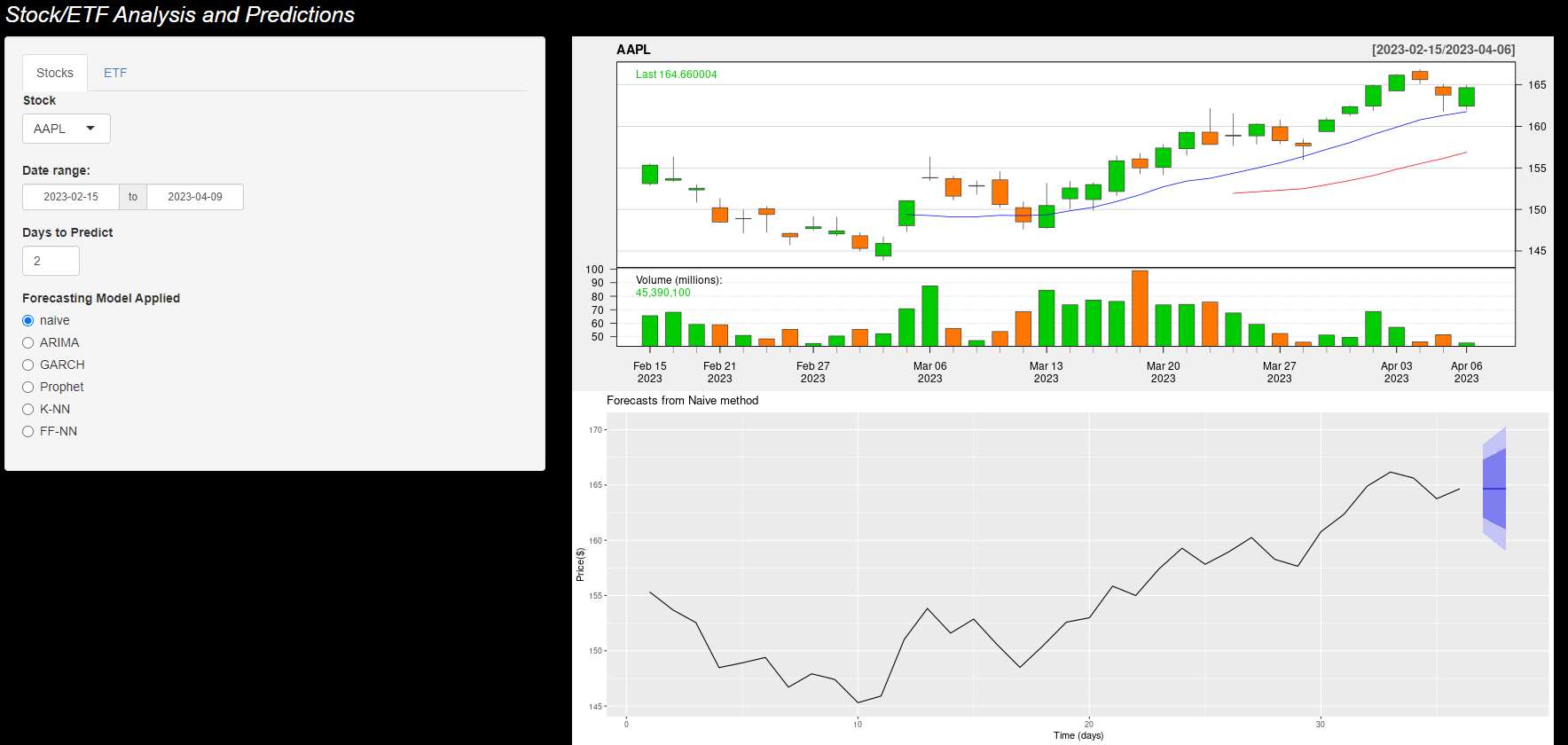
1. draw a schema with the web app architecture (see an example) - you can use ppt , draw.io, figma or other tools to sketch - 20pts [add a screenshot to your word document]



#### **Part 2. App Layout 40pts**

* What is the initial layout (when a user sees your app first)?

The user should be able to see the entire dashboard with a preloaded stock loaded up. The historical data, volume, and forecast will be displayed on the right, whereas the left side will contain all the interactive buttons and drop-down menus. Interactive buttons and menus include an option to change symbols or tickers, date range for analysis, forecast period, and forecast method.



* Where is the menu panel?

The menu panel will be on the left side of the screen as that location would be the most optimal.

* How many pages do you need? Or will you be using Tabs?

The team agrees to use tabs as the dashboard will update when the user selects the choice of Stock/ETF and the forecasting approach.

Tab 1 will be designed for Stocks analysis and Tab 2 will be designed for ETFs analysis.

* What is the color schema?

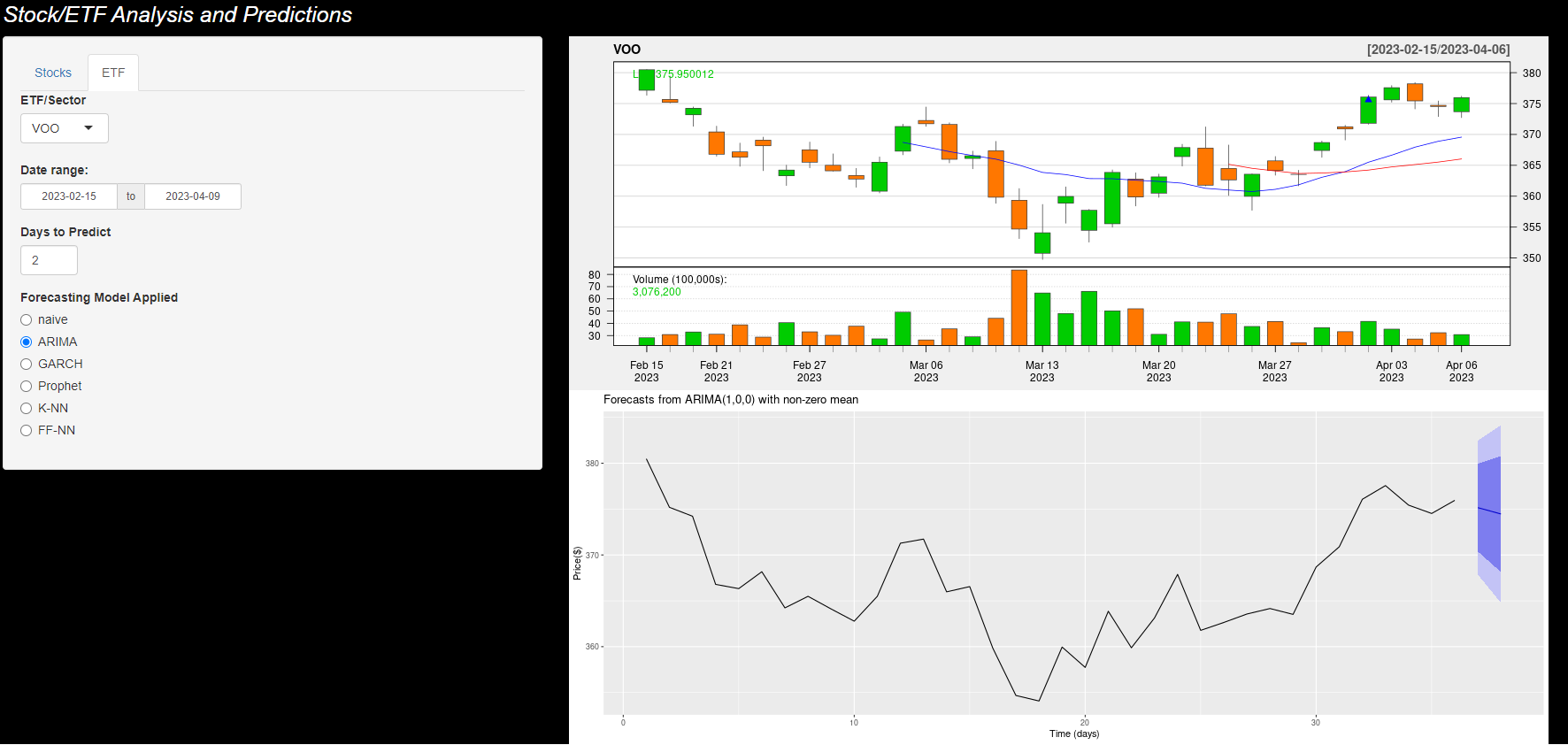
Black will be our choice for the background color as it provides a sleek look. Having green for positive daily change values and red for negative values will allow users to see how a stock or ETF performs at a glance. These colors are also the standard in candlestick charts for price shifts in securities.

* What each page or Tab will display?

As previously stated, the time series plot will be displayed on the right side, whereas the left side will have the criteria for the user to select. The user can select the stock/ETF of choice, the timeframe of the data, the forecasting method, and how many days the forecasting method will predict.

The plots on the right will not only show the candlestick style for each datapoint, but also the volumes associated with each day. The forecasting plot will be displayed below the historical data.

Two tabs will be present at the top so that the user can select between Stocks and ETFs, with the ETF tab displayed below.



Near future plans could be that the team add additional graphs such as gg\_season or gg\_subseries. Another plan is that the team may want to set up a different format so that instead of rectangles, users get the option of picking a continuous line as well. If we want to display more tabs, we could use other investment securities like bonds.

#### **Part 3. Team Work - 10pts**

* If you are working individually, just describe your planning for building an app
* Teams:
  + Describe your planning and work distribution. We need to make sure everyone is participating and work is equally distributed. While the grade is initially provided for the entire team, we reserve the right to change the grade for any individual students who did not fully participate and fulfill their obligations/responsibilities in the project.
  + Let us know if someone is not responding/participating - we will reassign that person to an individual group

Work Distribution:

Mohan: Technical Lead, Leading the dashboard project, Editor

Brian: Team Lead for submissions, Follow-up Developer, Writer

Brendan: Follow-up Developer, Writer, Editor

Everyone is communicating in a group chat in a timely manner.