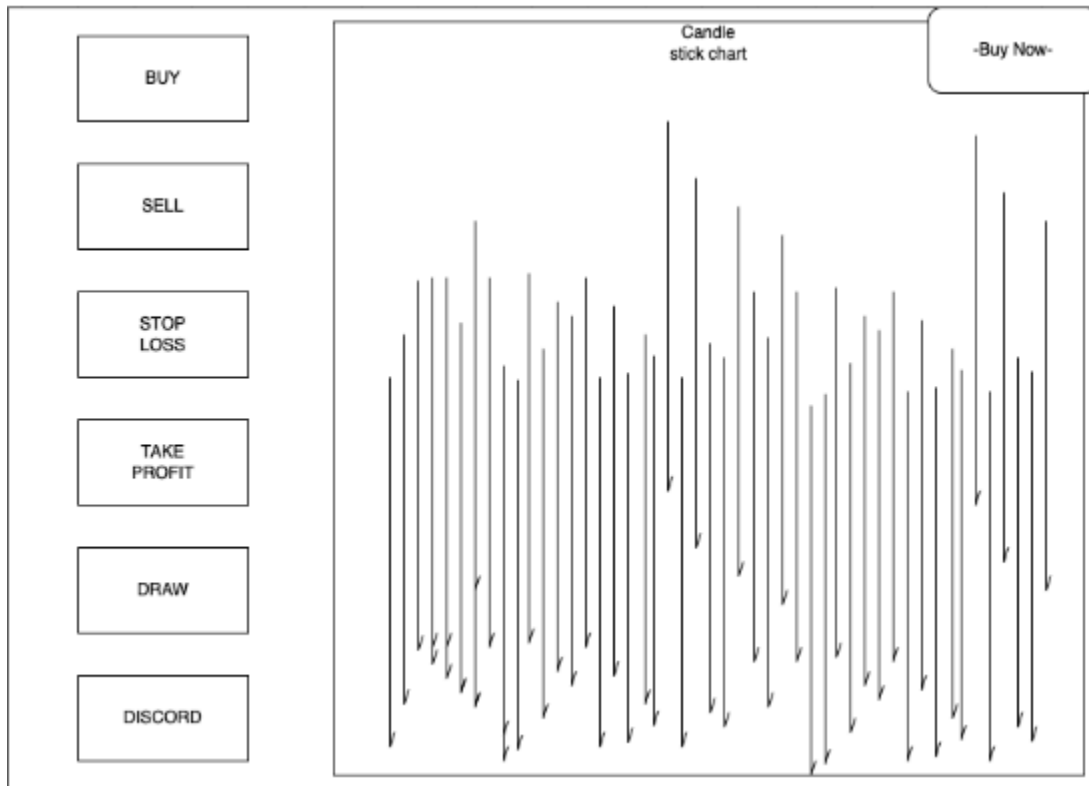Imaar Chauthani

Computer Science

Mr. Beradino

**CRIT B**

**Initial GUI:**



**{Figure 1 - Initial GUI)**

The image above is my preliminary GUI. It is a full window of live stock charts which also has

the ability to buy and sell through that program. I did not build upon this initial idea further due

to issues with money security. If I did continue with the initial idea of having the user be able to

buy and sell through this program, the user would have to put money into the program which

would cause concerns and issues. My client did not want to put their money into a third party

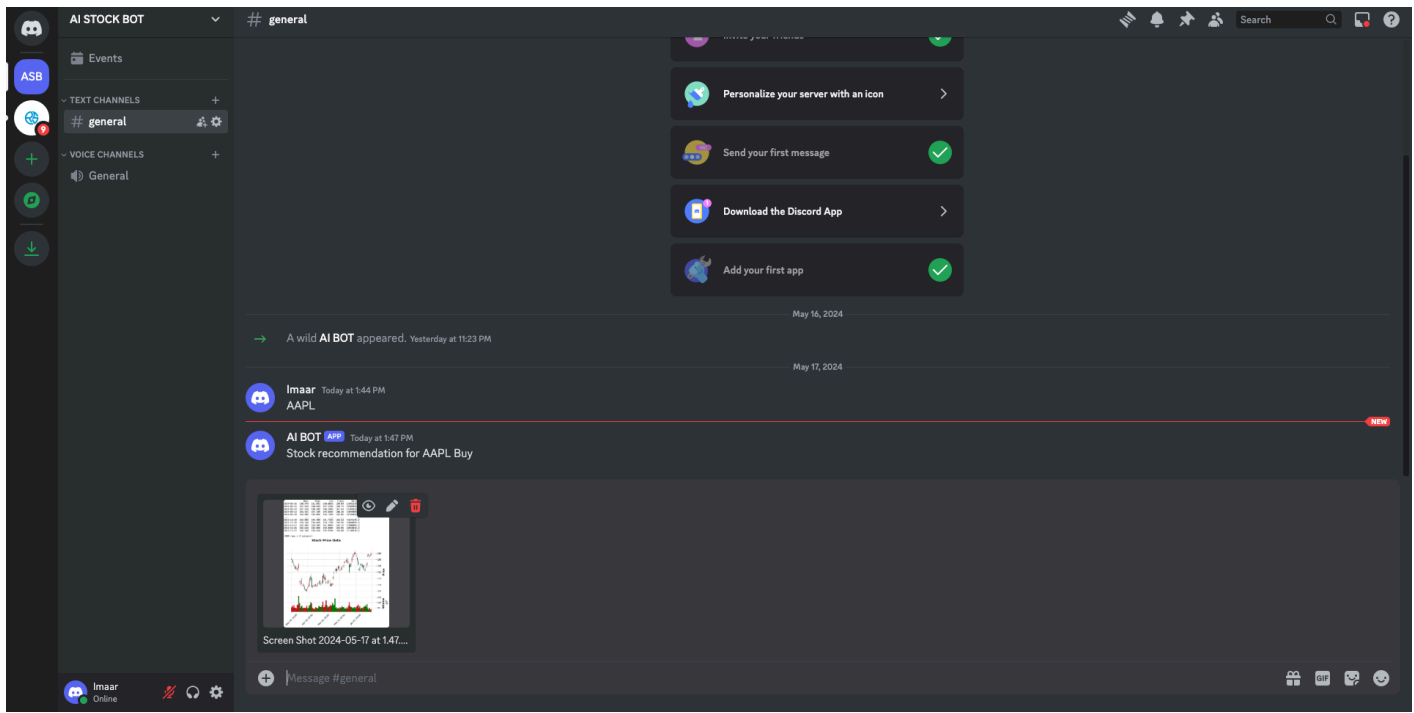program. We came to the conclusion to scrap this idea.

**Initial Flow Chart:**

I started fresh with a flowchart so I can fully develop the new idea before creating the graphical

user interface.
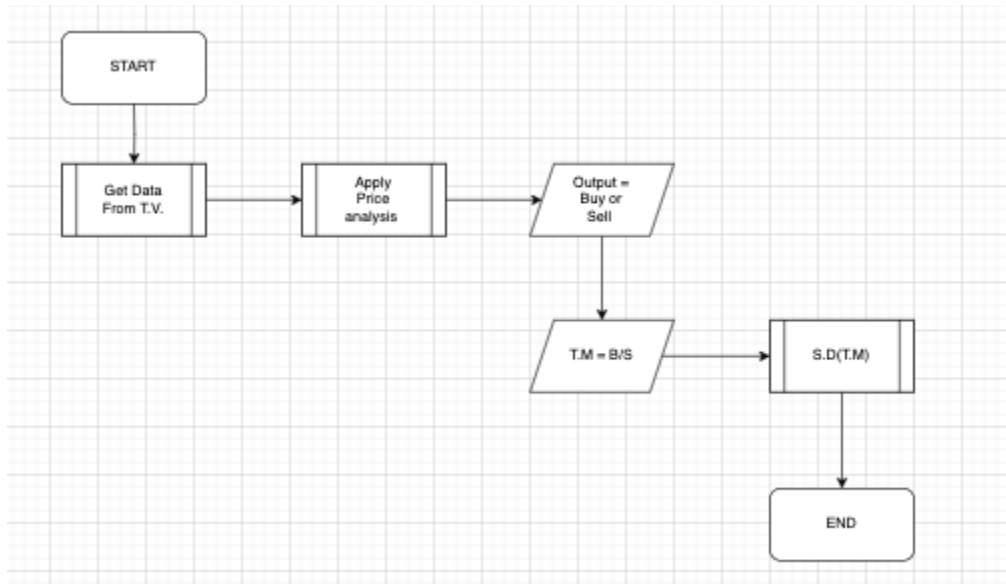


**{Figure 2 - Preliminary flowchart}**

It starts with retrieving data from a TradingView (API) and then applying given price analysis to

that data. It then sends an output of whether to buy or sell that stock. That is the basic rundown

of the flowchart for the program. After creating the flowchart I was able to comprehend and

create a GUI of what the user will see

**New Graphical User Interface:**



**{Figure 3 - Final GUI}**

```
             Open     High      Low   Close     Volume
2024-05-16  190.470  191.095  189.6601  189.84  52845230.0
2024-05-15  187.910  190.650  187.3700  189.72  70399988.0
2024-05-14  187.510  188.300  186.2900  187.43  52393619.0
2024-05-13  185.435  187.100  184.6200  186.28  72044809.0
2024-05-10  184.900  185.090  182.1300  183.05  50759496.0
...             ...      ...       ...     ...         ...
2023-12-29  193.900  194.400  191.7250  192.53  42672148.0
2023-12-28  194.140  194.660  193.1700  193.58  34049898.0
2023-12-27  192.490  193.500  191.0900  193.15  47899806.0
2023-12-26  193.610  193.890  192.8300  193.05  28919310.0
2023-12-22  195.180  195.410  192.9700  193.60  37149570.0

[100 rows x 5 columns]
```



**Stock Price Data**

**{Figure 4 - Expanded Chart GUI}**

The image above is what my program should look like to the user. In a discord channel, the user will input a stock ticker and then a discord bot connects to the program and outputs whether to buy or sell and provides a stock chart. At this point I did research and decided to use alpha vantage APIs instead of TradingView APIs. Then I created my final flowchart.

**Final Flow Chart:**
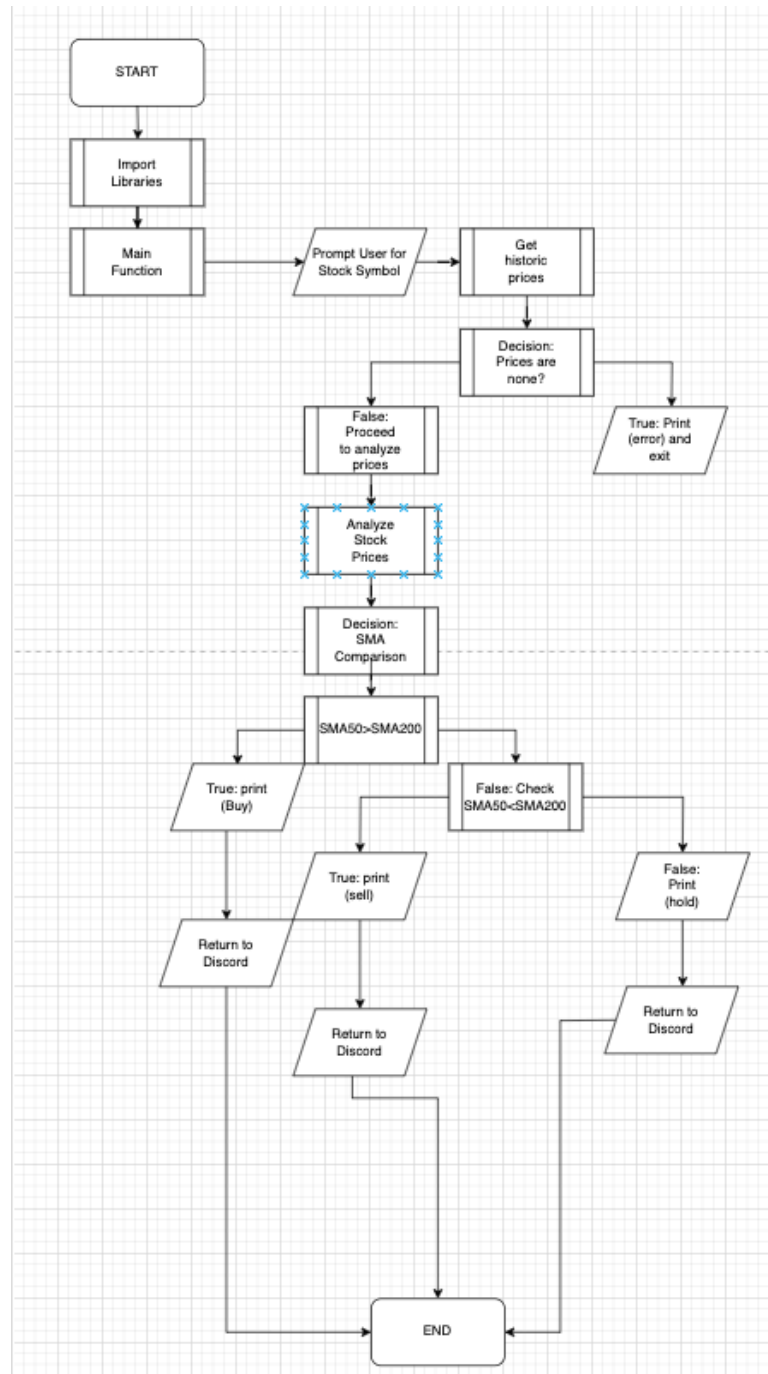


{**Figure 4 - Final Flowchart**}

In this flowchart I finalized the process of which my code will run and how it will output it to the

user.

**Pseudo Code:**

```
 2    // Import necessary libraries
 3    IMPORT tradingview_api_library
 4    IMPORT graph_generation_library
 5    IMPORT discord_api_library
 6    IMPORT price_analysis_library
 7
 8    METHOD MAIN()
 9        TRY
10        // Define stock symbol and Discord channel ID
11        DEFINE stock_symbol AS "AAPL"
12        DEFINE discord_channel_id AS "YOUR_DISCORD_CHANNEL_ID"
13
14        // Retrieve live stock market data
15        DEFINE stock_data AS RETRIEVE_LIVE_STOCK_DATA(stock_symbol)
16
17        // Perform price analysis
18        DEFINE analysis_result AS PERFORM_PRICE_ANALYSIS(stock_data)
19
20        // Generate price graph
21        DEFINE price_graph AS GENERATE_PRICE_GRAPH(stock_data)
22
23        // Send analysis result and graph to Discord channel
24        SEND_TO_DISCORD(discord_channel_id, price_graph, analysis_result)
25        END TRY
26        EXCEPT
27            OUTPUT: "an error occured"
28
29    END METHOD
30
31    METHOD RETRIEVE_LIVE_STOCK_DATA(symbol)
32        TRY
33        // Use TradingView API library to get live market data
34        DEFINE live_data AS tradingview_api_library.get_live_market_data(symbol)
35        RETURN live_data
36        END TRY
37        EXCEPT
38            OUTPUT "an error occured"
39    END METHOD
40
41    METHOD PERFORM_PRICE_ANALYSIS(data)
42        TRY
43        // Use price analysis library to analyze the stock data
44        DEFINE analysis AS price_analysis_library.analyze(data)
45        RETURN analysis
46        END TRY
47        EXCEPT
48            OUTPUT "an error occured"
49    END METHOD
50
51    METHOD GENERATE_PRICE_GRAPH(data)
52        TRY
53        // Use graph generation library to create a graph
54        DEFINE graph AS graph_generation_library.create_graph(data)
55        RETURN graph
56        END TRY
57        EXCEPT
58            OUTPUT "an error occured"
59    END METHOD
60
61    METHOD SEND_TO_DISCORD(channel_id, graph, analysis)
62        TRY
63        // Use Discord API library to send the graph and analysis to the specified channel
64        discord_api_library.send_image(channel_id, graph)
65        discord_api_library.send_message(channel_id, analysis)
66        END TRY
67        EXCEPT
68            OUTPUT "an error occured"
69    END METHOD
```

The program takes an input from the user for a ticker symbol then fetches that ticker symbol's data and performs a price analysis using moving averages. It then sends a buy or sell recommendation along with a graph to discord.

**Test Plan:**

| Action Tested | Nature of Test | Planned Result |
|---|---|---|
| Check if the program runs | Start program | All libraries should download and a message asking for a ticker symbol should appear |
| User can enter ticker symbol | User enters ticker symbol | Program retrieves data for that symbol. |
| Buy or Sell is indicated and graph is shown | Program runs | Program outputs a graph and a buy or sell recommendation through discord. |
| If incorrect ticker symbol is inputted, program will display message | User inputs a wrong ticker symbol | Through discord, the program outputs a message that says incorrect ticker, try again. |
| If market is not open, it will still recommend a buy or sell | User runs program while market is closed | Program outputs a buy or sell recommendation and a graph. |

Word Count: 322