

Stock Data Visualizer App

<https://github.com/bo-nzi/Stock-Data-Visualizer>

Team Members

Dylan May

Farouk Hamdan

Emma Boyd

Bo Nzikamira

Table of Contents

Contents

Table of Contents.....	2
Work Division	3
Use Case Diagram:	4
Use Case Description:.....	5
Team Communication Summary	10
Screenshots	11
Lessons Learned.....	13

Work Division

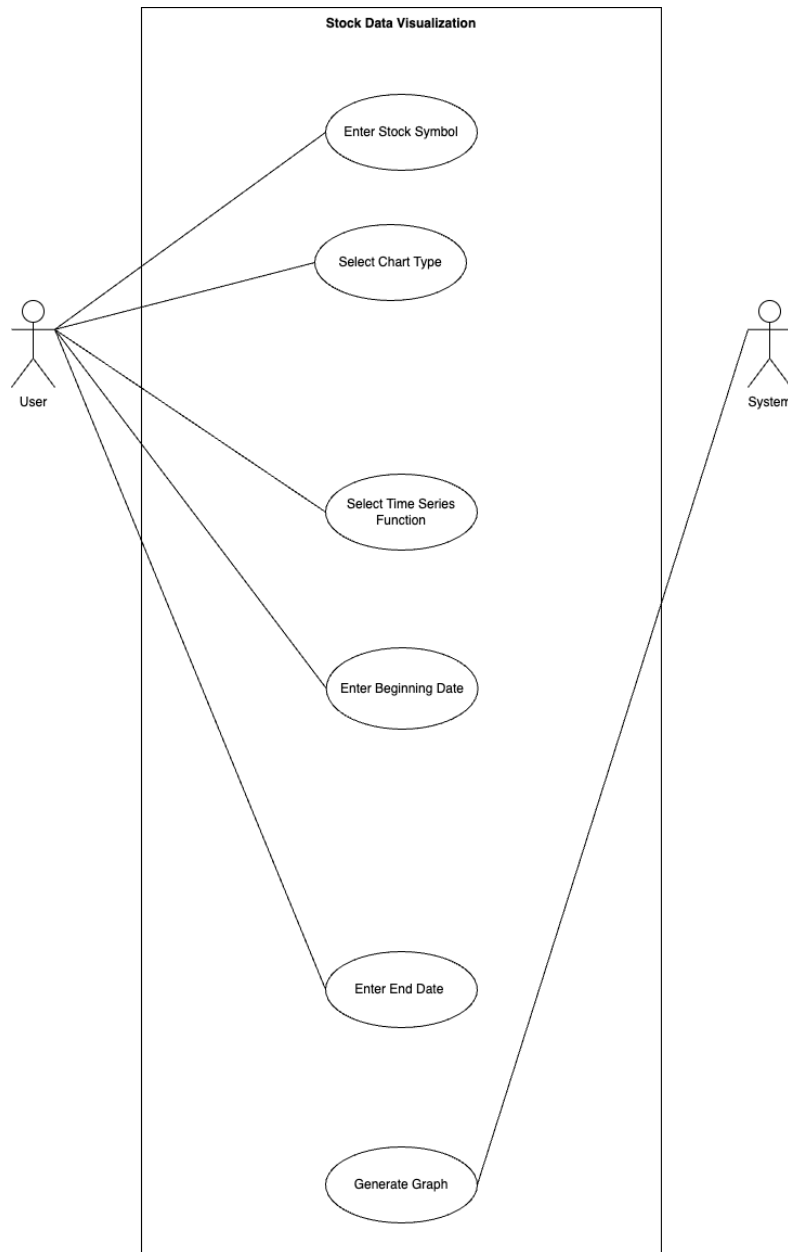
Dylan May – Scrum board, stock symbol function, chart type function, table of contents page, screenshots of scrum page, work division page, table of contents page, lessons learned page

Farouk Hamdan – team communication plan page, Graph generation function

Emma Boyd – Use case diagram, use case description pages, time series function, start date function, end date function

Bo Nzikamira – GitHub page, error handling in program

Use Case Diagram:



Use Case Description:

Use Case Description:

Enter Stock Symbol Use Case

- **Use Case 1:** User is entering a stock symbol for the company they want data for
- **Primary Actor:** App User
- **Use Case Overview:** The user is prompted to enter a field for a stock symbol and the system accepts the entered field, then prompts the user for the next question
- **Trigger:** The application is running and user enters a valid stock symbol
- **Preconditions:** The application is running
- **Postconditions:** If the stock symbol is a valid input and is accepted by the system, the next prompt is asked

Enter Stock Symbol Basic Flow

Description: This scenario describes when a user successfully enters a stock symbol and is prompted to enter the next selection

1. User is prompted to enter a stock symbol
2. The system verifies the stock symbol
3. The system continues with the next part of the application

Termination Outcome (Success): User is taken to the next step of the application

Alternative Flows 2A: Invalid Stock Symbol Input

- **1A1:** The user inputs an invalid stock symbol
- **1A2:** The system displays an error indicating that there is no matching stock data
- **1A3:** The system re-prompts the user to enter a valid stock symbol

Termination Outcome (Success): User is allowed another attempt to enter a valid stock symbol

Select Chart Type Use Case

- **Use Case 2:** User is entering a chart type

- **Primary Actor:** App User
- **Use Case Overview:** The user is prompted to enter either 1 or 2 (1 is a Bar chart and 2 is a Line chart)
- **Trigger:** User has successfully entered a Stock Symbol
- **Preconditions:** The application is running, and the user has entered a Stock Symbol
- **Postconditions:** If the chart selection is a valid input and is accepted by the system, the next prompt is asked

Select Chart Type Basic Flow

Description: This scenario describes when a user successfully selects a chart type and is prompted to enter the next selection

1. User is prompted to enter a 1 (bar) or 2 (line)
2. The system verifies the number entered
3. The system continues with the next part of the application

Termination Outcome (Success): User is taken to the next step of the application

Alternative Flows 1A: Invalid Chart Selection

- **1A1:** The user inputs a value that is not 1 or 2
- **1A2:** The system re-prompts the user to enter a chart type

Termination Outcome (Success): User is taken to the next step of the application

Time Series Function Use Case

- **Use Case 3:** User is entering a time series function
- **Primary Actor:** App User
- **Use Case Overview:** The user is prompted to enter a number between 1-4
- **Trigger:** User has successfully entered a Stock Symbol, and Chart Type
- **Preconditions:** The application is running, and the user has entered a Stock Symbol and Chart Type
- **Postconditions:** If the time series function selection is a valid input and is accepted by the system, the next prompt is asked

Time Series Function Basic Flow

Description: This scenario describes when a user successfully selects a time series function and is prompted to enter the next selection

1. User is prompted to enter a number between 1 and 4
 - a. 1 = Intraday
 - b. 2 = Daily
 - c. 3 = Weekly
 - d. 4 = Monthly
2. The system verifies the number entered
3. The system continues with the next part of the application

Termination Outcome (Success): User is taken to the next step of the application

Alternative Flows 1A: Invalid Time Series Function Selection

- **1A1:** The user inputs a value that is not 1-4
- **1A2:** The system re-prompts the user to enter a time series function

Termination Outcome (Success): User is taken to the next step of the application

Beginning Date Use Case

- **Use Case 4:** User is entering a beginning date
- **Primary Actor:** App User
- **Use Case Overview:** The user is prompted to a date in YYYY-MM-DD format
- **Trigger:** User has successfully entered a Stock Symbol, Chart Type, and Time Series Function
- **Preconditions:** The application is running, and the user has entered a Stock Symbol, Chart Type, and Time Series Function
- **Postconditions:** If the beginning date is a valid input and is accepted by the system, the next prompt is asked

Beginning Date Type Basic Flow

Description: This scenario describes when a user successfully enters a beginning date and is prompted to enter the next selection

1. User is prompted to enter a beginning date in YYYY-MM-DD format
2. The system verifies the date entered
3. The system continues with the next part of the application

Termination Outcome (Success): User is taken to the next step of the application

Alternative Flows 2A: Invalid Date Entered

- **1A1:** The user enters an invalid date

- **1A2:** The system re-prompts the user to enter a beginning date

Termination Outcome (Success): User is taken to the next step of the application

Beginning Date Use Case

- **Use Case 5:** User is entering a end date
- **Primary Actor:** App User
- **Use Case Overview:** The user is prompted to a date in YYYY-MM-DD format, that is after the beginning date
- **Trigger:** User has successfully entered a Stock Symbol, Chart Type, Time Series Function, and Beginning Date
- **Preconditions:** The application is running, and the user has entered a Stock Symbol, Chart Type, Time Series Function, and Beginning
- **Postconditions:** If the end date is a valid input and is accepted by the system, the next prompt is asked

Beginning Date Type Basic Flow

Description: This scenario describes when a user successfully enters an end date and is prompted to enter the next selection

1. User is prompted to enter a end date in YYYY-MM-DD format, that is after the beginning date
2. The system verifies the date entered
3. The system continues with the next part of the application

Termination Outcome (Success): User is taken to the next step of the application

Alternative Flows 1A: Invalid Date Entered

- **1A1:** The user enters an invalid date
- **1A2:** The system re-prompts the user to enter an end date

Alternative Flows 1B : End Date is before Beginning Date

- **1B1:** The user enters a date that is before the beginning date
- **1B2:** The system re-prompts the user to enter an end date

Termination Outcome (Success): User is taken to the next step of the application

Generate Graph Use Case

- **Use Case 6:** The system creates a graph that matches the conditions entered by the user and displays it
- **Primary Actor:** System
- **Use Case Overview:** The system displays the graph that the user made
- **Trigger:** User has successfully entered a Stock Symbol, Chart Type, Time Series Function, Beginning Date, and End Date
- **Preconditions:** The application is running, and the user entered a valid Stock Symbol, Chart Type, Time Series Function, Beginning Date, and End Date
- **Postconditions:** The system asks if the user wants to view more stock data

Generate Graph Basic Flow

Description: This scenario describes when the system matches

1. System displays a graph that fits the user inputs criteria
2. The graph is displayed on a browser tab by the system
3. The system asks user if they want to view more stock data

Termination Outcome (Success): The Program is shut down or restarted based on user input

Team Communication Summary

Our team primarily communicated through Discord, which served as our central hub for daily messaging, quick updates, and file sharing. This platform allowed for efficient, real-time collaboration and helped us stay connected throughout the week. In addition to asynchronous communication on Discord, we held weekly synchronous meetings every Thursday from 2:00 to 3:15 PM via Zoom. These meetings provided a dedicated time to discuss progress, align on tasks, resolve any blockers, and plan ahead. This combination of tools ensured consistent and effective communication, supporting our team's productivity and cohesion.

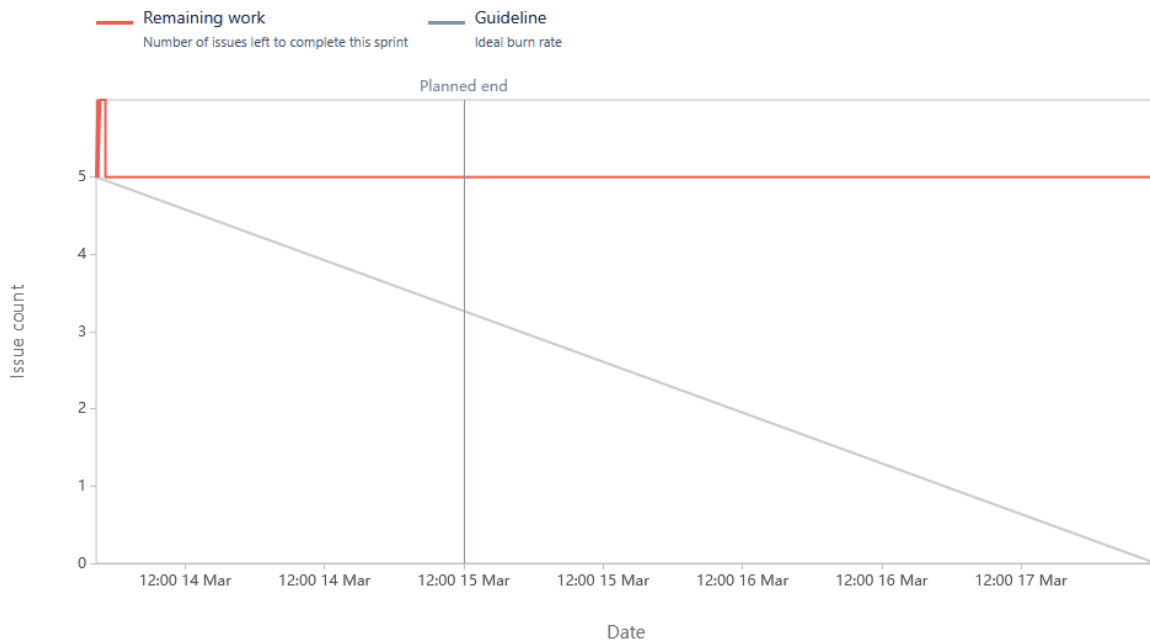
Screenshots

Initial sprint that tasks the group to install python libraries, create GitHub page and connect to it, get API key, and other requirements to start work on the project.

■	P3SDV-21	Create Scrum project	DONE	🔄 Getting Started	DM Dylan May			Mar 13, 2025	Mar 13, 2025
■	P3SDV-22	Email professor	DONE	🔄 Getting Started	DM Dylan May			Mar 13, 2025	Mar 13, 2025
■	P3SDV-1	Ensure everyone has Pygal installed	DONE	🔄 Getting Started				Mar 13, 2025	Mar 17, 2025
■	P3SDV-2	Ensure everyone has Lxml installed	DONE	🔄 Getting Started				Mar 13, 2025	Mar 17, 2025
■	P3SDV-3	Ensure everyone has Requests installed	DONE	🔄 Getting Started				Mar 13, 2025	Mar 17, 2025
■	P3SDV-5	Everyone acquire API	DONE	🔄 Getting Started				Mar 13, 2025	Mar 17, 2025
■	P3SDV-4	Create GitHub repository	DONE	🔄 Getting Started	BN Bo Nzikamira			Mar 13, 2025	Mar 17, 2025

Date - March 13th, 2025 - March 15th, 2025

Sprint goal - Ensure everyone has requirements needed to work on project, and has access to scrum project and GitHub repository.

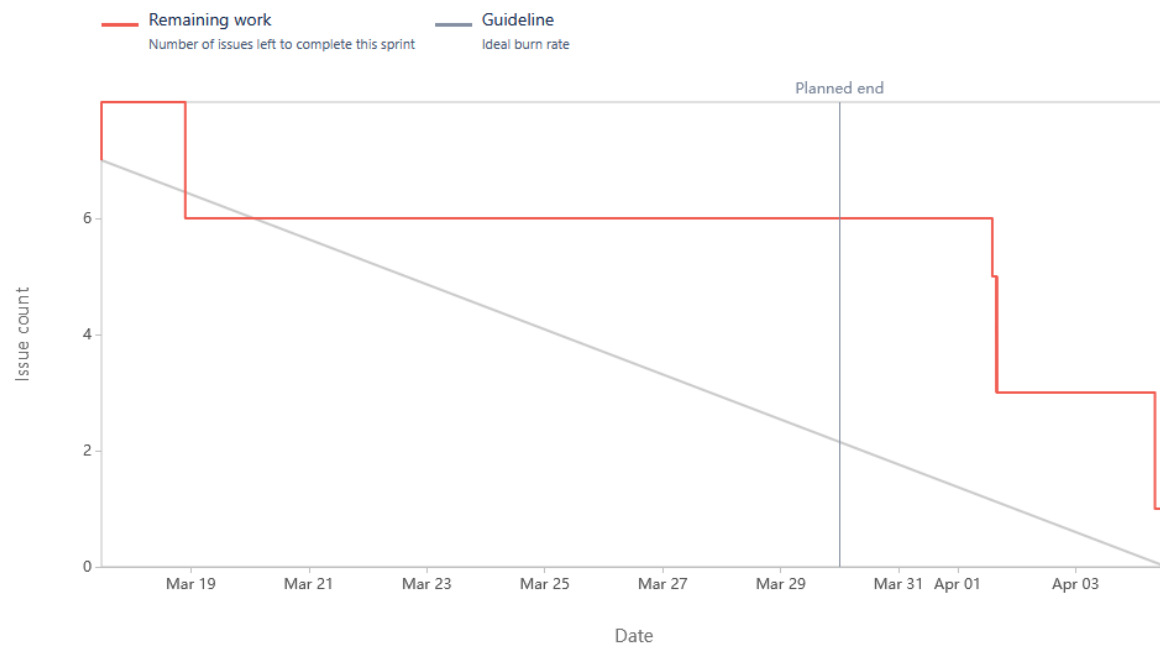


Second Sprint was the majority of the work. It included creating the use case diagram, writing the program, and error checking the program.

<input type="checkbox"/>	<input checked="" type="checkbox"/>	P3SDV-6	Create function asking for stock symbol	DONE	Creating Program	Dylan May		Mar 13, 2025	Mar 18, 2025
<input type="checkbox"/>	<input checked="" type="checkbox"/>	P3SDV-7	Create function asking for chart type	DONE	Creating Program	Dylan May		Mar 13, 2025	Apr 1, 2025
<input type="checkbox"/>	<input checked="" type="checkbox"/>	P3SDV-23	Complete Use Case	DONE	Creating Program	Emma Boyd		Mar 17, 2025	Apr 1, 2025
<input type="checkbox"/>	<input checked="" type="checkbox"/>	P3SDV-8	Create functions to get start and end date	DONE	Creating Program	Emma Boyd		Mar 13, 2025	Apr 1, 2025
<input type="checkbox"/>	<input checked="" type="checkbox"/>	P3SDV-19	error handling for chart type	DONE	Creating Program	Bo Nzikamira		Mar 13, 2025	Apr 4, 2025
<input type="checkbox"/>	<input checked="" type="checkbox"/>	P3SDV-18	error handling for stock symbol function	DONE	Creating Program	Bo Nzikamira		Mar 13, 2025	Apr 4, 2025
<input type="checkbox"/>	<input checked="" type="checkbox"/>	P3SDV-20	error handling for start and end date function	DONE	Creating Program	Emma Boyd		Mar 13, 2025	Apr 1, 2025
<input type="checkbox"/>	<input checked="" type="checkbox"/>	P3SDV-24	error handling for time series function	DONE	Creating Program	Emma Boyd		Apr 1, 2025	Apr 1, 2025
<input type="checkbox"/>	<input checked="" type="checkbox"/>	P3SDV-25	Create function asking for time series	DONE	Creating Program	Emma Boyd		Apr 1, 2025	Apr 1, 2025
<input type="checkbox"/>	<input checked="" type="checkbox"/>	P3SDV-9	Create function to generate graph in browser	DONE	Creating Program	Farouk Hamdan		Mar 13, 2025	Apr 4, 2025

Date - March 17th, 2025 - March 30th, 2025

Sprint goal - Create the program



The last sprint is to ensure all the documentation is complete for the project.

Finishing Project

Finish documentation on project



Start stand-up

Complete sprint



GROUP BY None ▾

TO DO 1

Create lessons learned page
P3SDV-17 BN

+ Create issue

IN PROGRESS 1

Create work division page
P3SDV-13 DM

DONE 5 ✓

Create cover page
P3SDV-11 ✓ DM

Create table of contents
P3SDV-12 ✓ DM

Take screenshots of scrum through sprints
P3SDV-15 ✓ DM

Create burndown charts
P3SDV-16 ✓ DM

Create summary of teams communication plan
P3SDV-14 ✓ FH

Lessons Learned

The team learned many lessons throughout the project by using services like Jira and GitHub in a team setting for the first time. It was a learning experience for all of us on how to commit changes and create branches in GitHub as we all just pasted our work into it before. We also learned and utilized some new features in Jira Scrum like the list view, attaching documents to tasks that are completed, and setting multiple sprints. Some members of the group also worked with and learned about how to utilize APIs for the first time. One of the largest lessons to take away from this project is to keep after fellow group members and make sure they are completing tasks on schedule.