# IT 4320 Project 3

Aidan Engbert – Scrum Master

Pranav Malatkar

Grant Wiedeman

# Table of contents

# Use Case Description

### Stock Visualization App – Use Case Descriptions

**UC-01: Configure API Key**
- User makes sure the app has the Alpha Vantage API key so it can pull stock data.

**UC-02: Enter Stock Symbol**
- User types in the stock ticker they want (ex: AAPL, TSLA, MSFT).

**UC-03: Select Time Series**
- User chooses whether they want daily, weekly, or monthly stock data.

**UC-04: Select Chart Type**
- User picks chart style (line chart, candlestick, etc.).

**UC-05: Enter Date Range**
- User gives start + end dates to control the time period they want to see.

**UC-06: Validate Inputs**
- App checks that dates are valid and end date isn't before start date.

**UC-07: Request Stock Data**
- App sends the request to Alpha Vantage using the inputs the user gave.

**UC-08: Filter & Prep Data**
- App cleans the data and filters it so only the selected date range shows up.

**UC-09: Generate Chart**
- App builds the chart based on the user's selection.

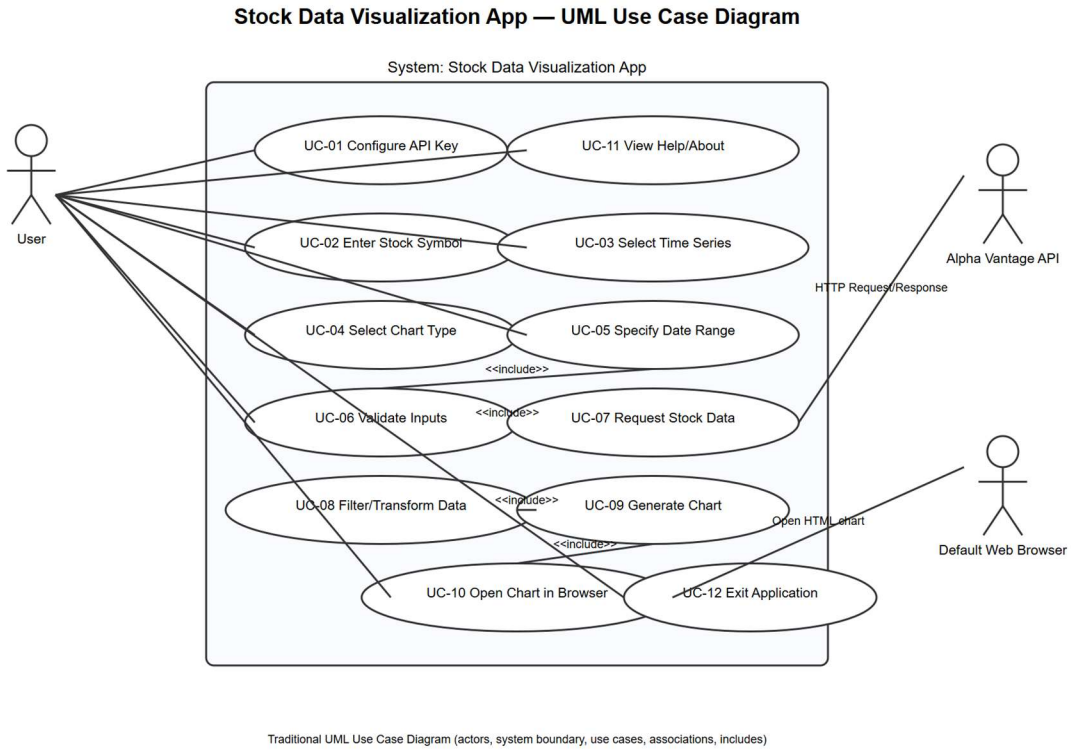
**UC-10: Show Chart**
- App opens the chart in the user's default browser so they can see it.

**UC-11: View Help**
- User can pull up simple help info if needed.

**UC-12: Exit App**
- User closes the program. Nothing complicated.

Overall: user picks a stock, picks how they want it shown, and the app makes it happen without breaking or confusing people.

# User Diagram




Traditional UML Use Case Diagram (actors, system boundary, use cases, associations, includes)

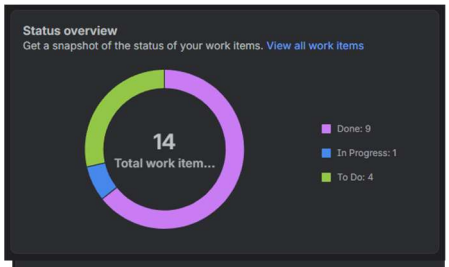
# Work Distribution

The work distribution was based on both skill and availability while also keeping in mind a even balance between all members. Some simple task put others above with task but generally the difficult task were evenly distributed among group members.

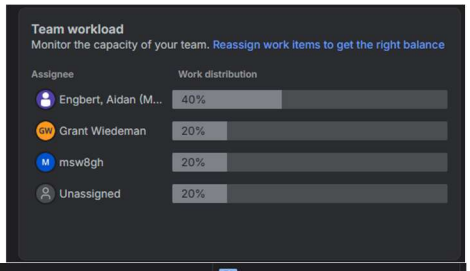
# Communication Plan

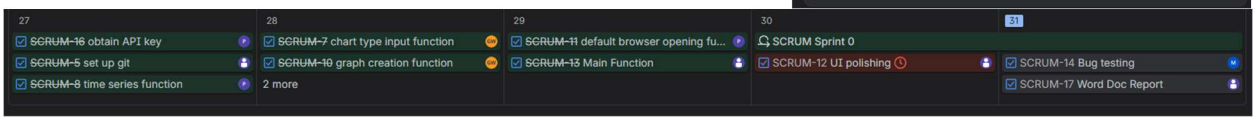
The teams communication plan was simple using Discord as a primary form of communication with email being a secondary form. The basic task were assigned via Jira and communication was sent out via discord to ensure all members knew assignments and expectations. Should project parts not be finished on schedule, the member assigned that task was expected to communicate with the scrum master to ensure the group could stay on target.

Here we have a screenshot of our total number of task (14) during the work progress. We made sure to lable all of our task with priority to make sure we understood what jobs were needed first in case group members fell behind. The goal was to outline every job from something as simple as creating a word document and sharing it to as complex as the actual programing itself. We ended up with a 40/20/20/20 distribution at the end primarily because of simple task such as setting up git and sharing word documents. These task were all set to be due the same week a the end of the project to ensure it was done on time.

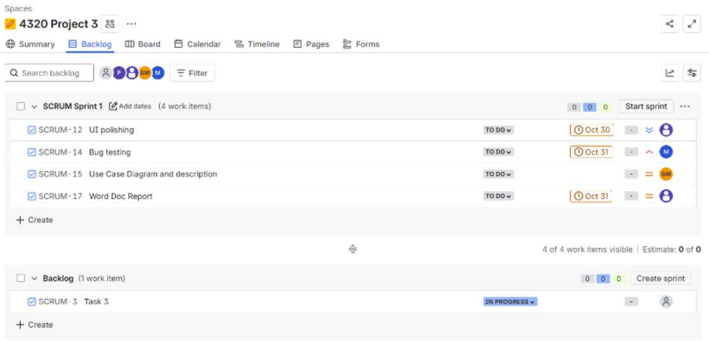







## Lessons Learned:




During the process of developing this project, we still struggle to split the work evenly and get our required processes completed before the deadline. We experience this type of procrastination from our last projects as well. We all need to spend some time on this project before the deadline. That being said, we have completed the project with most bugs fixed and published. This was my (Pranav) first time obtaining an API key and it worked well when testing the functions. In addition, this experience helped us better understand the importance of planning ahead, maintaining consistent communication, and setting internal milestones to stay on track. Throughout development, we learned how crucial time management and task delegation are in a team-based environment, especially when

multiple features are being built in parallel. Although we faced challenges balancing responsibilities and coordinating our workload, we adapted by scheduling more regular check-ins and dividing tasks based on each member's strengths. Moving forward, we aim to apply these lessons to future projects by implementing clearer timelines, utilizing task tracking tools more effectively, and starting earlier to reduce last-minute pressure. Overall, this project strengthened our teamwork skills, technical problem-solving abilities, and understanding of the software development lifecycle.

https://github.com/aheg7b/IT4320-Project3