ML-Stock Prediction

Report 1
Software Engineering

The Haystack Group *(Team):*
Alex Danielle Basden
Keita Sakurai
Curtis Hill
Taylor Davis
Lisa Thoms
Brian Luing

Website / GitHub
thepressq.com (Hidden Currently)
GitHub incoming:

All Team Members contributed equally to this Report.

Version 1.01
*Cleaned up wording/grammar
Added details on figures
Fixed an issue with the traceability matrix
Group had sent Version 1.00 on Report 1 Part 2 on Tuesday, which had combined and fixed multiple issues prior.

Table of Contents

Customer Problem Statement

a. Problem Statement

The stock market, particularly the New York Stock Exchange (NYSE), Nasdaq, Shanghai Stock Exchange, Euronext, and Japan Exchange Group, hold significant importance in the global economy today. While the NYSE and Nasdaq primarily operate within the United States, they serve as vital indicators of the private sector's health and consumer confidence. Consequently, it's unsurprising that an increasing number of individuals aspire to participate in these markets with the aim of enhancing their personal wealth.

Nonetheless, a significant obstacle to entry exists for numerous individuals, regardless of age, who wish to engage in these markets. That's why The Haystack Group is keenly interested in developing a ML-Stock Prediction application that facilitates interaction with the American markets and offers educational tools to dismantle these barriers. Users should encounter a seamless registration process, enabling them to swiftly initiate their participation. Within this platform, they will have the opportunity to select multiple stocks from various stock markets and receive predictions about the future value of these stocks. These stock predictions aim to closely resemble actual market value that will occur in the future.

To enhance users' comprehension of the financial markets, it's essential to provide them with access to market metrics. These metrics can be delivered through news updates related to the companies within their investment portfolio, interactive charts for visual analysis, as well as information about the market's opening and closing activities throughout the trading day.

The complete user experience should be consistent and seamless across mobile devices, tablets, and desktop computers. When combined with the aforementioned features, this unified approach aims to deliver an engaging and captivating core experience for users to gain knowledge about the stock market performance and predictions.

Current problems include the inability to show real-time stock prices via the APIs, and makes it difficult to analyze because of not showing various indicators. Also, automated system trading is not possible, so we would like to implement an algorithm to automate trading by developing an AI model and an APIs.

We also want to provide a user-friendly interface. It is a simple, easy-to-understand interface that allows even beginners to trade stocks. It has a wide range of user-friendly trading tools. For example, in addition to regular trading, it allows the user to place limit orders, as well as pre-orders.

In addition, we will implement robust security measures from time to time to ensure that users can trade stocks safely. There are two main parts of security measures: one is the establishment of a security system to safely manage assets, and the other is countermeasures against cyber-attacks. For the security system, we are considering adopting a multi-signature system. For cyber-attack countermeasures, for example, we will take measures against viruses and hacking, phishing e-mails, and phishing sites.

Our project aims to develop a Machine Learning (ML) algorithm that predicts stock prices. This ML-based stock predictor will leverage historical stock data, market indicators, and other relevant factors to provide users with accurate stock price predictions. The system will cater to both novice investors and seasoned traders seeking data-driven insights for their investment decisions.

b. Glossary of Terms

Stock − A stock is a share of an ownership of a company. This allows one to have a claim on a company's earnings and assets depending on how many stocks they own. Stocks have risen or fallen throughout the day depending on how the business is doing. This in turn will lead to selling or purchasing of stock as it cycles through its fiscal period.

Stock Ticker − A stock ticker is a tool that will be used to track the rise and fall of a stock throughout the day. Information will be shown to the user so they may view what a stock is currently at and the changes that will be made to the stock.

Stock Graph − The stock graph will be a graph made up of data/values collected throughout the day to show the rise and fall of the stock prices to the user while also predicting where the stocks may go based on the data that was collected for that day.

scikit-learn − Or sklearn, is a popular machine learning library in Python. It provides a wide range of tools and algorithms for tasks such as classification, regression, clustering, dimensionality reduction, and more. Scikit-learn is known for its user-friendly and consistent API, making it accessible for both beginners and experienced machine learning practitioners. It also offers utilities for data preprocessing, model selection, and evaluation, making it a valuable resource for building and deploying machine learning models.

Linear regression − Linear regression is a simple and widely used machine learning algorithm. It is commonly used in tasks such as predicting prices, estimating trends, and analyzing relationships between variables in various fields, including economics, finance, and science.

Tick Indicators - Used to identify those stocks whose last trade was either a downtick or an uptick. This is used as an indicator for determining the market’s trend regarding market sentiment.

Simple Moving Average − SMA is a commonly used technical indicator that calculates the average price of an asset over a specific period of time. It helps smooth out price fluctuations and is used to identify trends in the price movement of an asset.

Exponential Moving Average − EMA is similar to SMA but gives more weight to recent prices, making it more responsive to recent price changes. It is used to identify short-term trends and react more quickly to market movements.

Bollinger Bands − Bollinger Bands consist of a middle band (SMA) and two outer bands that are standard deviations away from the middle band. They help traders identify volatility and potential price reversals.

Ichimoku Kinko Hyo (IKH) − A Japanese charting system that provides information about support and resistance levels, trend direction, and potential buy/sell signals.

Volume − Volume is a measure of the number of shares or contracts traded in a financial asset within a given time period.

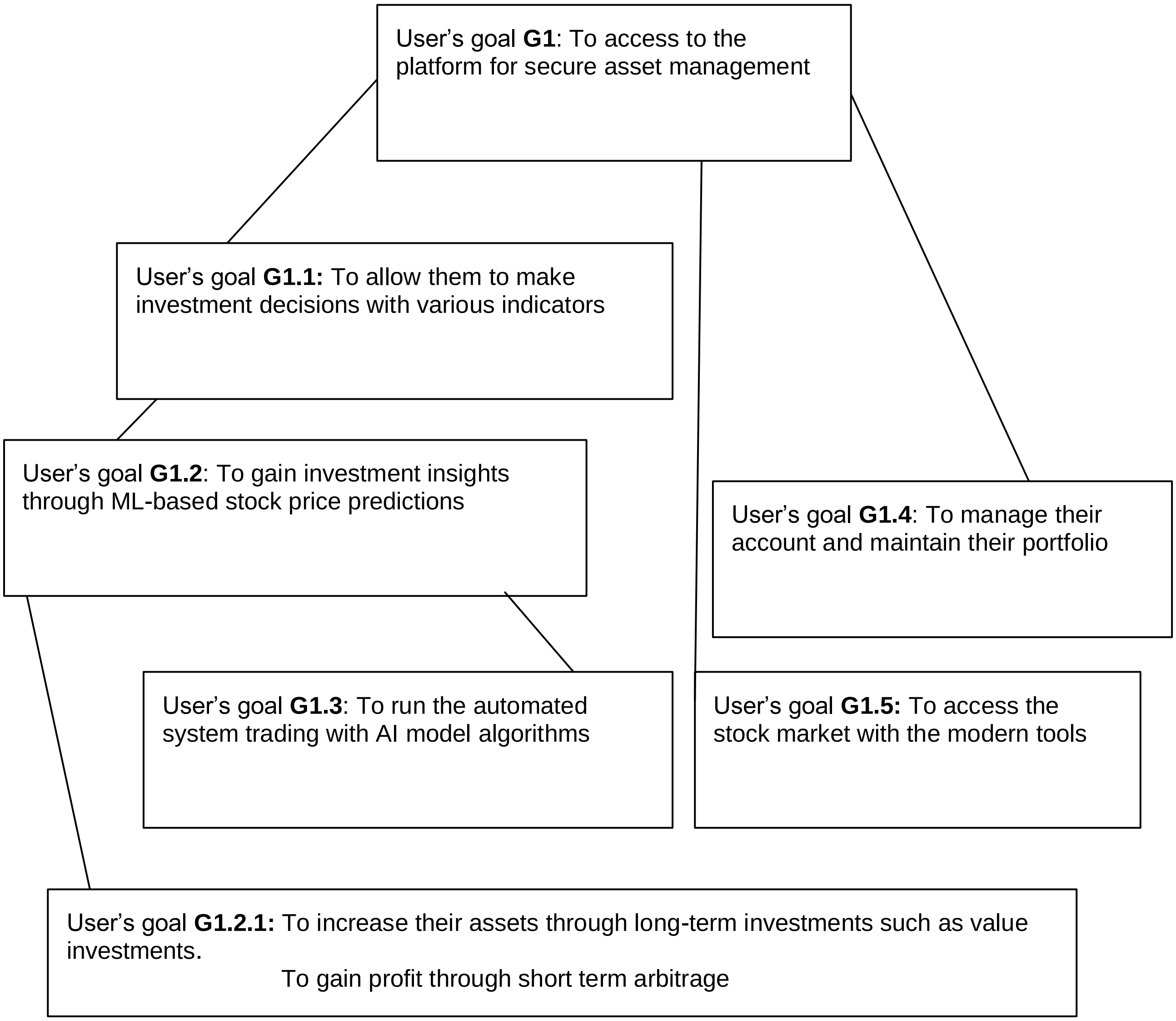
Relative Strength Index (RSI) − RSI is a momentum oscillator that measures the speed and change of price movements. It ranges from 0 to 100 and is used to identify overbought and oversold conditions. Values above 70 suggest overbought, while values below 30 suggest oversold.

Moving Average Convergence Divergence (MACD) − MACD is a trend-following momentum indicator that shows the relationship between two moving averages of an asset's price. It consists of the MACD line and the signal line.

Historical Volatility − Historical Volatility measures the past price fluctuations of an asset over a specified period. It helps traders assess the level of risk associated with an asset and can be used to make informed decisions about position sizing and risk management.

Goals, Requirements, and Analysis

a. Business Goals


User's goal **G1**: To access to the platform for secure asset management
User's goal **G1.1:** To allow them to make investment decisions with various indicators
User's goal **G1.2**: To gain investment insights through ML-based stock price predictions
User's goal **G1.4**: To manage their account and maintain their portfolio
User's goal **G1.3**: To run the automated system trading with AI model algorithms
User's goal **G1.5:** To access the stock market with the modern tools
User's goal **G1.2.1:** To increase their assets through long-term investments such as value investments.
To gain profit through short term arbitrage

b. Enumerated Functional Requirements

The user stories below demonstrate the application functionality in perspective of the user. Each user story is identified by an identifier in the left column and requirement weight in the right column. These requirements are functions the user should expect upon regularly interacting with the ML-Stock Prediction application.

Identifier	User Story	Weight
ST-1	As a user, I can create and register an account to participate in the ML-Stock Prediction application.	10
ST-2	As a user, I can manage and authenticate my account so that I can securely access my information.	8
ST-3	As a user, I can search for and select companies by stock symbol so that I may track potential and current investments.	8
ST-4	As a site administrator, I can capture data and store historical stock data so that the ML-based stock price predictions are reliable.	10
ST-5	As a user, I can view ML-based stock price predictions so that I may determine whether to invest in a stock.	10
ST-6	As a user, I can easily navigate the price prediction interface so that I can quickly find stock information needed for investments.	4
ST-7	As a user, I can view real-time data integrated with financial API so that I can see current market trends.	6
ST-8	As a user, I can use data visualization tools so that I may have a better understanding of the platform.	4
ST-9	As a user, I can access the ML-Stock Prediction application across mobile devices, tablets, and desktop computers so that I can have a seamless and consistent user experience.	10
ST-10	As a user I have access to company news, interactive charts and market activity updates so that I may manage my investment portfolio more easily.	4

c. Enumerated Nonfunctional Requirements

Functionality: To ensure data integrity and security, user registration and authorization will be a priority. Authentication features include user log-ins with encrypted and stored passwords, account management, and session validations. For the security system, a multi-signature system will be implemented. Measures against viruses and hacking, phishing emails, and phishing sites will also be taken.

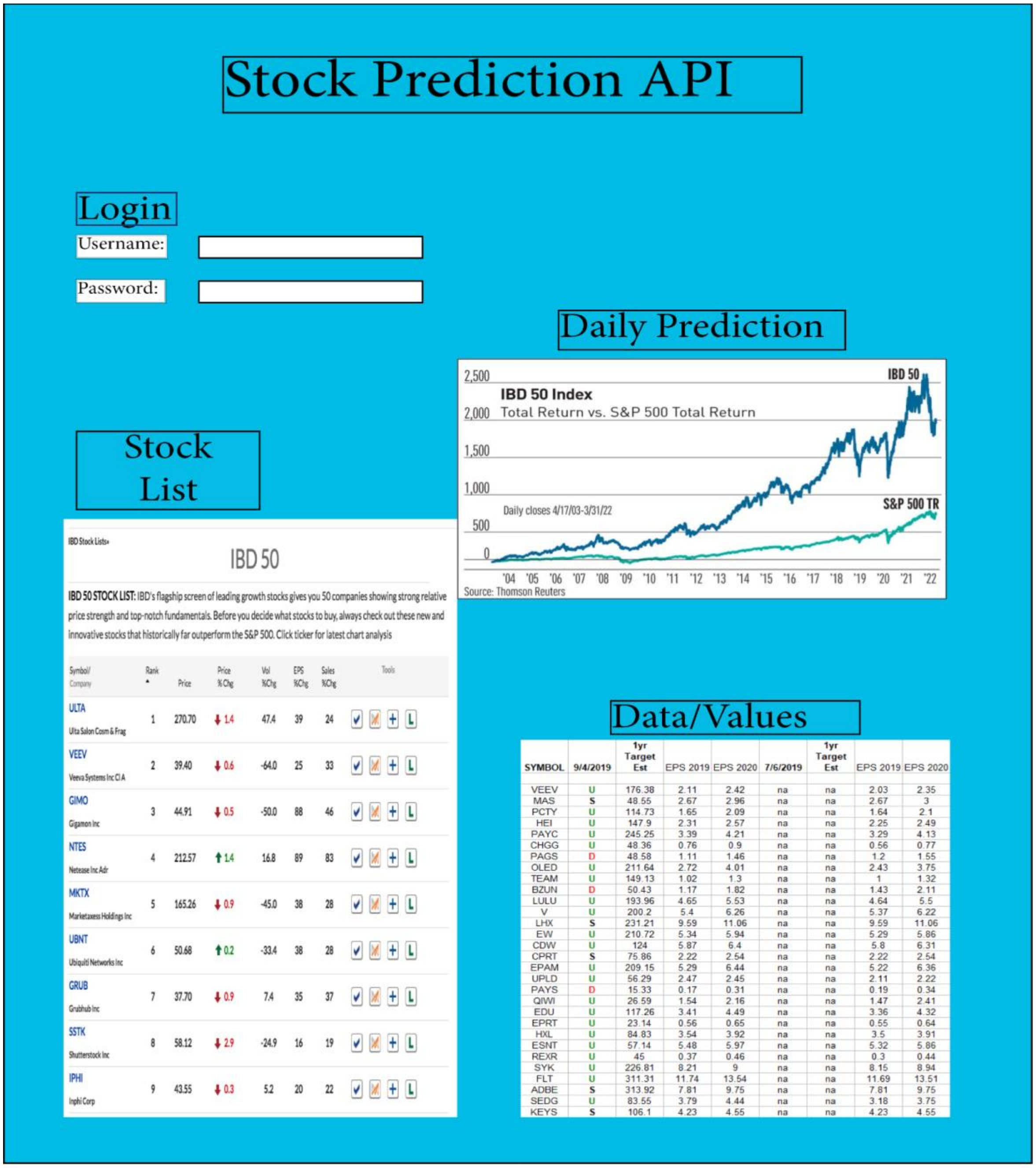
Usability: The application's ease of use will enhance the user experience. The user experience will be consistent and seamless across mobile devices, tablets, and desktop computers. The approach aims to be an engaging experience for users to gain knowledge about the stock market performance and predictions.

Reliability: To ensure the application's reliability under certain operating conditions, user error and server failure will be minimized through transaction confirmation messages and user data backup.

Performance: The performance of the application will be prioritized through efficiency in communication between the client and the server. User tasks and server requests will be performed timely to ensure speed, efficient resource consumption and short response time.

Supportability: The application will be compatible with current and future versions of web servers with platform independence and adaptability. Front end support will provide access scalability for lower resolution devices.

d. User Interface Requirements



Source: Thomson Reuters

IBD Stock Lists»

IBD 50

IBD 50 STOCK LIST: IBD's flagship screen of leading growth stocks gives you 50 companies showing strong relative price strength and top-notch fundamentals. Before you decide what stocks to buy, always check out these new and innovative stocks that historically far outperform the S&P 500. Click ticker for latest chart analysis

Symbol/ Company	Rank ▲	Price	Price % Chg	Vol %Chg	EPS %Chg	Sales %Chg	Tools
ULTA Ulta Salon Cosm & Frag	1	270.70	↓ 1.4	47.4	39	24	☑ ☒ + L
VEEV Veeva Systems Inc Cl A	2	39.40	↓ 0.6	-64.0	25	33	☑ ☒ + L
GIMO Gigamon Inc	3	44.91	↓ 0.5	-50.0	88	46	☑ ☒ + L
NTES Netease Inc Adr	4	212.57	↑ 1.4	16.8	89	83	☑ ☒ + L
MKTX Marketaxess Holdings Inc	5	165.26	↓ 0.9	-45.0	38	28	☑ ☒ + L
UBNT Ubiquiti Networks Inc	6	50.68	↑ 0.2	-33.4	38	28	☑ ☒ + L
GRUB Grubhub Inc	7	37.70	↓ 0.9	7.4	35	37	☑ ☒ + L
SSTK Shutterstock Inc	8	58.12	↓ 2.9	-24.9	16	19	☑ ☒ + L
IPHI Inphi Corp	9	43.55	↓ 0.3	5.2	20	22	☑ ☒ + L

SYMBOL	9/4/2019	1yr Target Est	EPS 2019	EPS 2020	7/6/2019	1yr Target Est	EPS 2019	EPS 2020
VEEV	U	176.38	2.11	2.42	na	na	2.03	2.35
MAS	S	48.55	2.67	2.96	na	na	2.67	3
PCTY	U	114.73	1.65	2.09	na	na	1.64	2.1
HEI	U	147.9	2.31	2.57	na	na	2.25	2.49
PAYC	U	245.25	3.39	4.21	na	na	3.29	4.13
CHGG	U	48.36	0.76	0.9	na	na	0.56	0.77
PAGS	D	48.58	1.11	1.46	na	na	1.2	1.55
OLED	U	211.64	2.72	4.01	na	na	2.43	3.75
TEAM	U	149.13	1.02	1.3	na	na	1	1.32
BZUN	D	50.43	1.17	1.82	na	na	1.43	2.11
LULU	U	193.96	4.65	5.53	na	na	4.64	5.5
V	U	200.2	5.4	6.26	na	na	5.37	6.22
LHX	S	231.21	9.59	11.06	na	na	9.59	11.06
EW	U	210.72	5.34	5.94	na	na	5.29	5.86
CDW	U	124	5.87	6.4	na	na	5.8	6.31
CPRT	S	75.86	2.22	2.54	na	na	2.22	2.54
EPAM	U	209.15	5.29	6.44	na	na	5.22	6.36
UPLD	U	56.29	2.47	2.45	na	na	2.11	2.22
PAYS	D	15.33	0.17	0.31	na	na	0.19	0.34
QIWI	U	26.59	1.54	2.16	na	na	1.47	2.41
EDU	U	117.26	3.41	4.49	na	na	3.36	4.32
EPRT	U	23.14	0.56	0.65	na	na	0.55	0.64
HXL	U	84.83	3.54	3.92	na	na	3.5	3.91
ESNT	U	57.14	5.48	5.97	na	na	5.32	5.86
REXR	U	45	0.37	0.46	na	na	0.3	0.44
SYK	U	226.81	8.21	9	na	na	8.15	8.94
FLT	U	311.31	11.74	13.54	na	na	11.69	13.51
ADBE	S	313.92	7.81	9.75	na	na	7.81	9.75
SEDG	U	83.55	3.79	4.44	na	na	3.18	3.75
KEYS	S	106.1	4.23	4.55	na	na	4.23	4.55

Figure 2.4: A mock-up view of the webpage in early planning stage. This shows the information we wish to show the user on various webpages.

Functional Requirement Specifications

a. Stakeholders

The described software is primarily aimed at two key demographics: first-time investors and existing investors. However, it is designed to evolve into a comprehensive tool for both short and long-term trading decision planning. The intention is for the software to eventually expand its user base beyond comparable platforms by offering enhanced features, such as portfolio importing, which can entice investors to spend more time using the platform.

Initially, the software will be offered as a free service, with the long-term plan of transitioning to a subtle advertising model that does not disrupt the user experience. Advertisements will only be introduced once a significant user base has been established, providing a source of revenue for the company. This approach, as a free service with future ads, is expected to attract many users and, due to its increased functionality, keep them engaged for extended periods.

The software's target audience is not limited to potential investors; it is intended to cater to anyone seeking a deeper understanding of the financial industry and those interested in practicing trading as a learning experience before venturing into the actual market.

b. Actors and Goals:

Guest:

1. Description: A visitor to the website who has either not logged in or is just a simple visitor.
2. Capabilities:
    1. Register and create an account using (Future OpenID or OAuth2)
    2. View a random stock.

Investor:

3. Description: A user who has an account on the server and is logged into their account.
4. Capabilities:
    1. Research the latest updates in the market.
    2. View their portfolio.
    3. Execute orders of any kind.

Database System:

5. Description: Stores information for the accounts of all users.
6. Responsibilities:
    1. Insert information as accounts are created.

    2. Push data back to views about users/events.
    3. Store new data about users and events.

Financial API:

7. Description: Provides the stocks in the database with up-to-date prices.
8. Responsibilities:
    1. Fetch real-world information and update the database accordingly.

Site Administrator:

9. Description: Manages the overall website.
10. Responsibilities:
    1. Ensure Site is running smoothly.

Browser:

11. Description: The middleman between the user and the system.
12. Responsibilities:
    1. Present data to the user.
    2. Retrieve data from the user.

Yahoo! Finance:

13. Description: The unit that knows about current financial statistics.
14. Responsibilities:
    1. Retrieve data about stocks.

c. Use Cases

UC-1 Register/Create Account - A user can register, create, and authenticate their account to fully access all the features of the application by using OpenID or OAuth2. Derived from ST-1 and ST-2.

UC-2 View Market Data - A user can search for stocks and view real time market data from Yahoo! Finance API. Derived from ST-3, ST-6, ST-7, and ST-8.

UC-3 Manage Portfolio - A user can manage their stock portfolio and access up to date information regarding company and market activity that use interactive charts. Derived from ST-9 and ST-10.

UC-4 Pick Stock and ML-Language Prediction - A user can execute market orders and view ML-based stock price predictions to aid in investment decisions. Derived from ST-3 and ST-5.

UC-5 Take Administrative Actions - The site administrator can capture and store historical stock data, as well as manage and maintain the application site to run smoothly. Derived from ST-4.

Use Case Diagram:

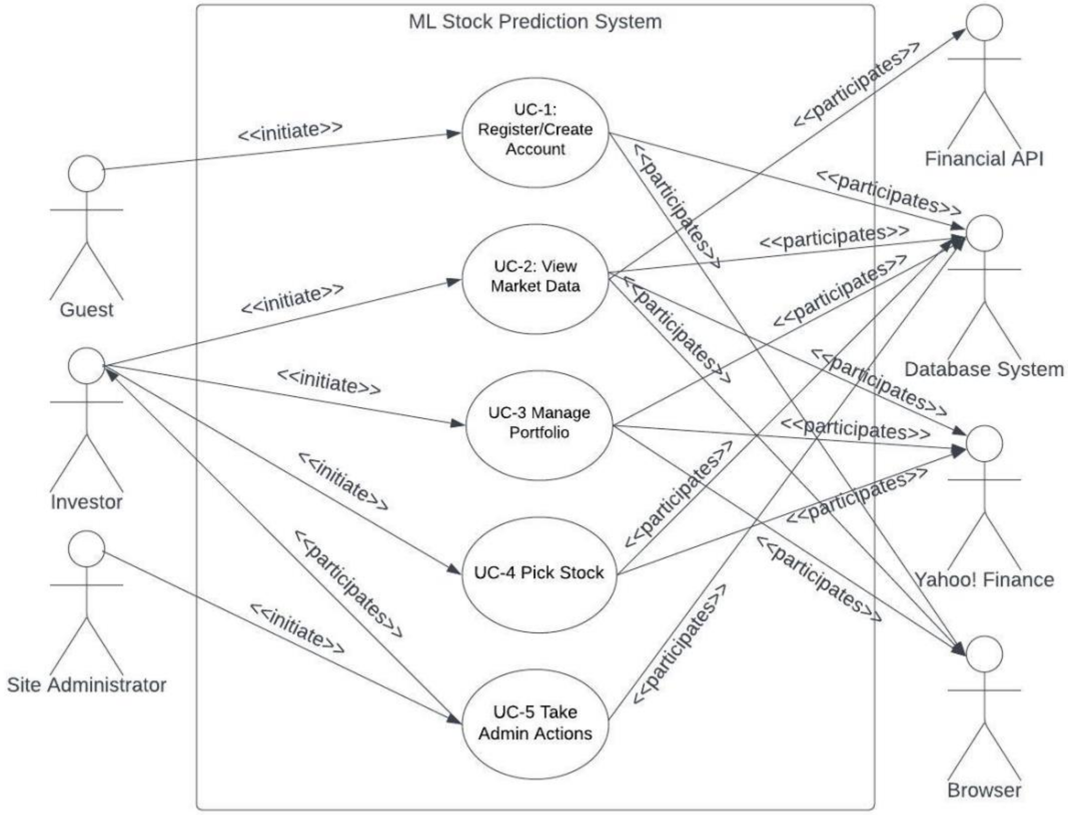


Figure 3.1: This graphic illustrates the relationships between the core actors of our platform.

Use Case UC-1:	**Register/Create Account**
Related Requirements:	ST-1, ST-2
Initiating Actor:	Guest
Actor's Goal:	Register and create account with website servers
Participating Actors:	Guest, Database System, Browser
Preconditions:	Guest is not a registered user
Postconditions:	New user account information and data is stored by the Database System.
	Flow of Events for Main Success Scenario:
→	1. Guest navigates to application website and attempts to log in on Browser
→	2. System checks Database for user information of Investor to see if found or not
←	3. System receives OpenID/OAuth2 data and registers user information in the Database as Investor
←	4. System sends registration confirmation to Investor and displays portfolio in Browser
	Flow of Events for Alternative Scenario:
→	1. Investor navigates to application website and attempts to log in
→	2. System checks Database for user information of Investor to see if found or not
←	3. Systems retrieves Investor user information from Database
←	4. System displays Investor's profile on Browser

Use Case UC-2:	**View Market Data**
Related Requirements:	ST-3, ST-6, ST-7, ST-8
Initiating Actor:	Investor
Actor's Goal:	Search for stocks and view real time market data
Participating Actors:	Database System, Yahoo! Finance API, Browser
Preconditions:	Investor is logged in and Yahoo! Finance stock data requests are working

Postconditions:	Stock and market information is displayed for the Investor
	Flow of Events for Main Success Scenario:
→	1. Investor searches for a stock using its symbol
→	2. System sends search request to Database
←	3. Database returns suggested systems to system
←	4. System displays suggested stock symbols
→	5. Investor picks a symbol from suggested symbols and sends request to System
→	6. System sends the request to Database and Yahoo! Finance
←	7. Database is updated with Yahoo! Finance stock information and ML price prediction
	Flow of Events for Alternative Scenario: Search Fails
←	1. Yahoo! Finance returns no results from stock symbol search
→	2. System alerts investor that the search failed on Browser

Use Case UC-3:	**Manage Portfolio**
Related Requirements:	ST-9, ST-10
Initiating Actor:	Investor
Actor’s Goal:	Manage investments by viewing portfolio
Participating Actors:	Database System, Yahoo! Finance, Browser
Preconditions:	Investor is logged in and Yahoo! Finance stock data requests are working
Postconditions:	Investor can view and make changes to their portfolio
	Flow of Events for Main Success Scenario:
→	1. Investor requests to view their portfolio and navigates to area on Browser
←	2. System send request for Investor portfolio from Database
→	3. System displays investment portfolio to Investor in Browser
→	4. Investor makes applicable changes needed to portfolio
←	5. System updates Database portfolio information of user

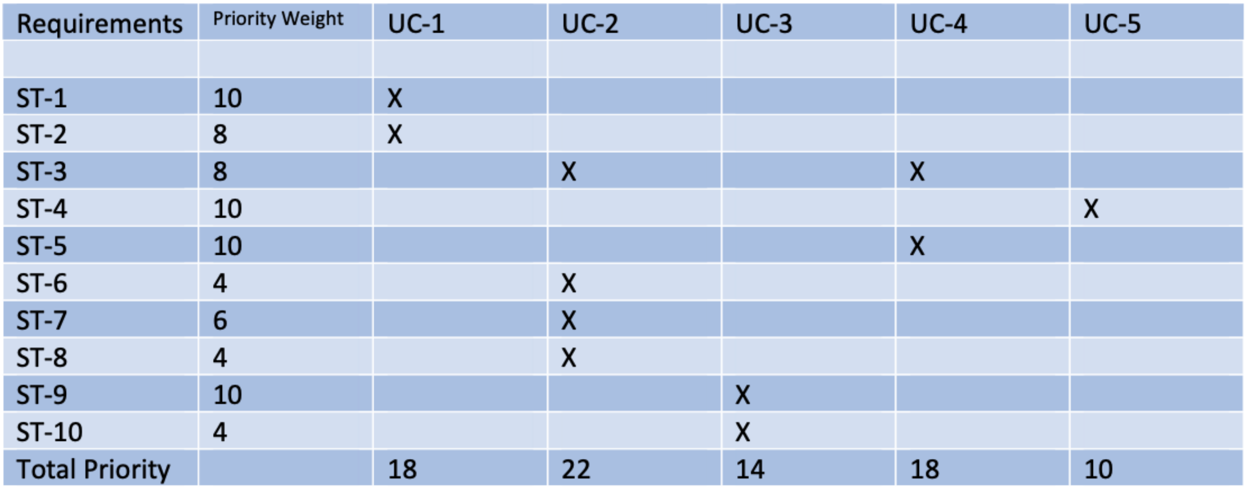
→	6. System shows confirmation to Investor of updated portfolio in Browser

Use Case UC-4:	**Pick Stock and ML-Language Prediction**
Related Requirements:	ST-3, ST-5
Initiating Actor:	Investor
Actor’s Goal:	View ML-based stock price predictions and execute market orders
Participating Actors:	Database System, Yahoo! Finance
Preconditions:	Investor is logged in and Yahoo! Finance stock data requests are working
Postconditions:	Investor positions are updated in Database
	Flow of Events for Main Success Scenario:
→	1. Investor selects stock and places a market order
←	2. System requests update from Yahoo! Finance
←	3. System receives data from Yahoo! Finance
←	4. System records order in Database
→	5. System displays portfolio changes to Investor in Browser
	Flow of Events for Alternative Scenario: No information found in Yahoo! Finance
←	4. System alerts Investor that request failed in Browser

Use Case UC-5:	**Take Administrative Actions**
Related Requirements:	ST-4
Initiating Actor:	Site Administrator
Actor’s Goal:	Manage and maintain the application site to run smoothly. Derived from ST-4.
Participating Actors:	Database System, Investors

Preconditions:	User is logged in and has Site Administrator permissions
Postconditions:	Maintain application site to capture and store historical stock data
	Flow of Events for Main Success Scenario:
→	1. Site Administrator requests System logs
←	2. System saves, closes and returns logs

d. Traceability Matrix



Requirements	Priority Weight	UC-1	UC-2	UC-3	UC-4	UC-5
ST-1	10	X				
ST-2	8	X				
ST-3	8		X		X	
ST-4	10					X
ST-5	10				X	
ST-6	4		X			
ST-7	6		X			
ST-8	4		X			
ST-9	10			X		
ST-10	4			X		
Total Priority		18	22	14	18	10

Figure 3.2: The traceability matrix presented here is based on only the full dressed use cases above.

e. System Sequence Diagrams

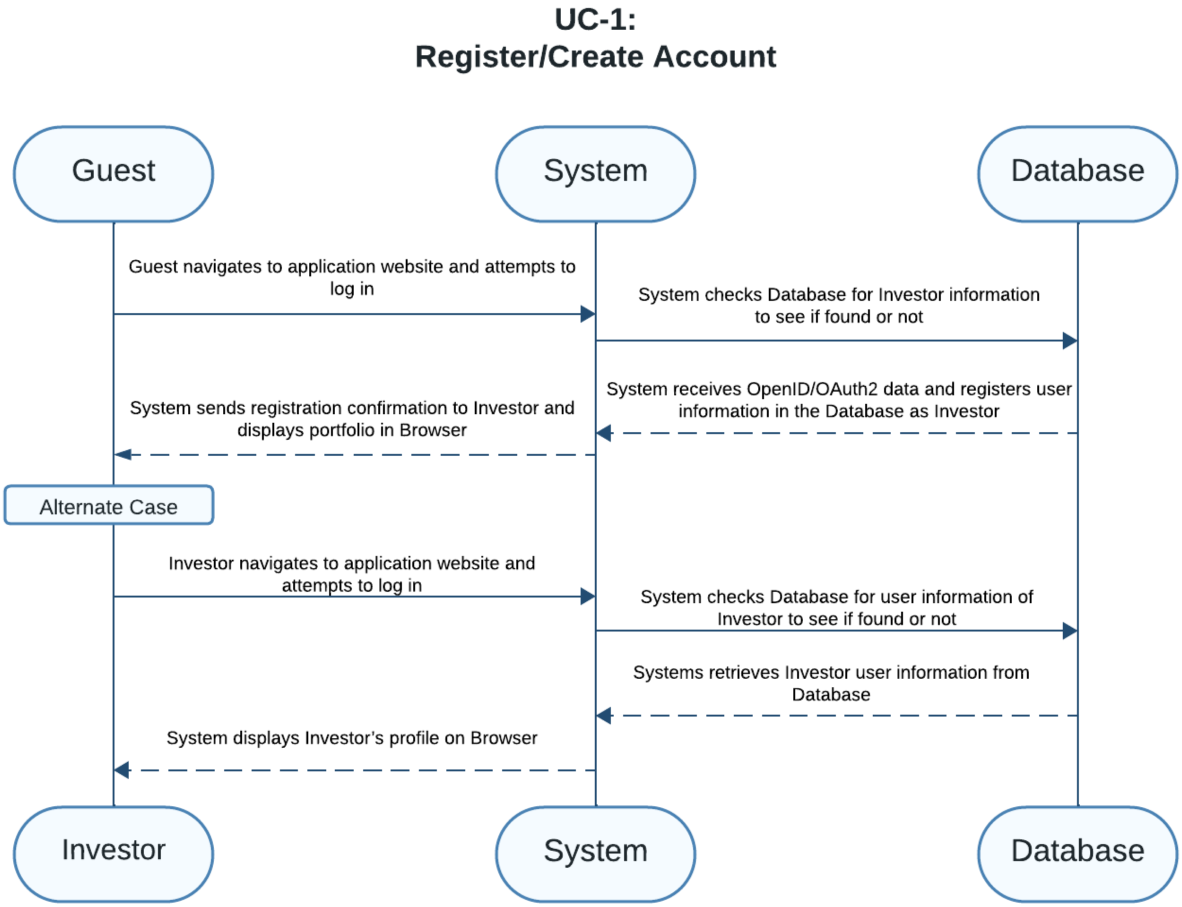

UC-1:
Register/Create Account
Guest
System
Database
Guest navigates to application website and attempts to log in
System checks Database for Investor information to see if found or not
System receives OpenID/OAuth2 data and registers user information in the Database as Investor
System sends registration confirmation to Investor and displays portfolio in Browser
Alternate Case
Investor navigates to application website and attempts to log in
System checks Database for user information of Investor to see if found or not
Systems retrieves Investor user information from Database
System displays Investor's profile on Browser
Investor
System
Database

Figure 3.3: See UC-1 on page 14. When the guest navigates to the application website, this use case is triggered. The system checks to see if there is investor information in the database, and if not, it receives guest OpenID/OAuth2 data. The guest information is then registered as an investor. The system sends the registration confirmation to the investor and displays the portfolio in the browser. If the system confirms that the guest information is already registered in the database, it receives the investor information and displays the profile in the browser.

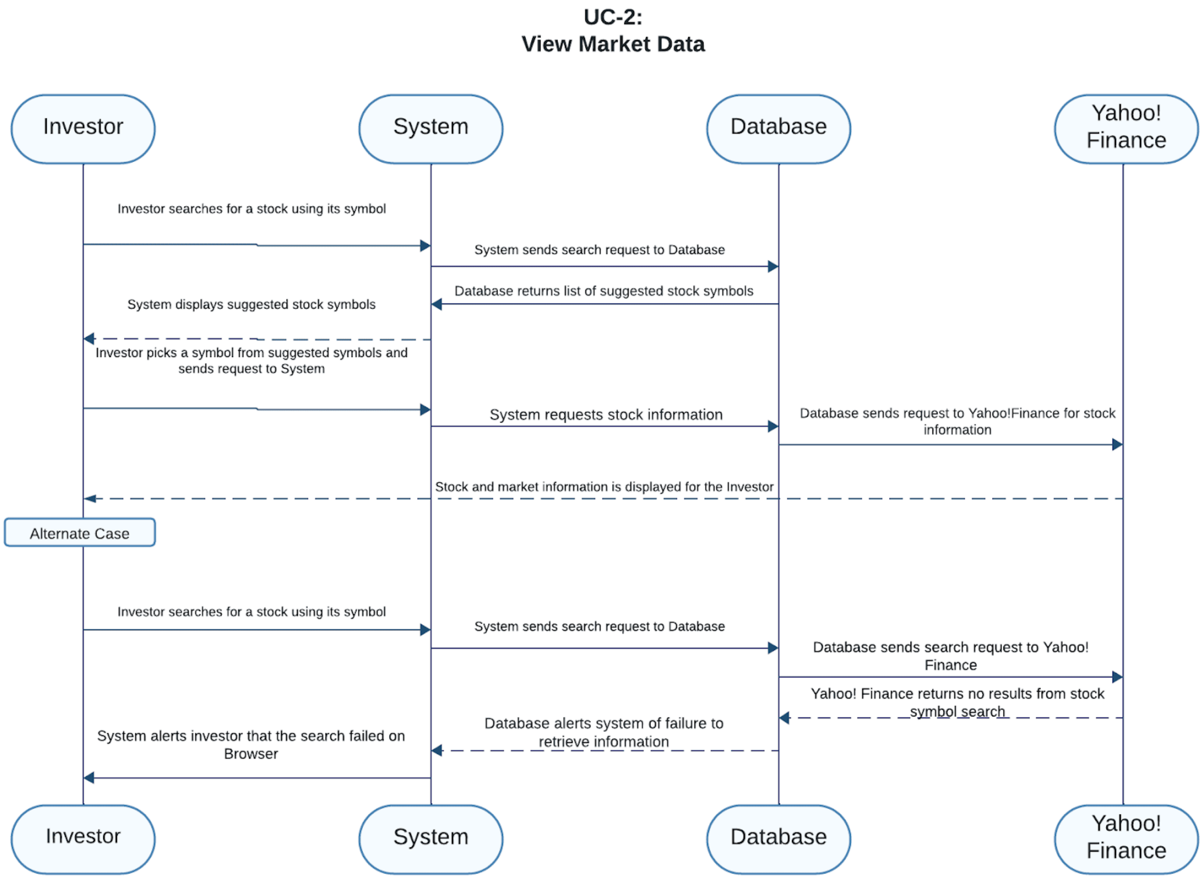

UC-2:
View Market Data
Investor
System
Database
Yahoo! Finance
Investor searches for a stock using its symbol
System sends search request to Database
Database returns list of suggested stock symbols
System displays suggested stock symbols
Investor picks a symbol from suggested symbols and sends request to System
System requests stock information
Database sends request to Yahoo!Finance for stock information
Stock and market information is displayed for the Investor
Alternate Case
Investor searches for a stock using its symbol
System sends search request to Database
Database sends search request to Yahoo! Finance
Yahoo! Finance returns no results from stock symbol search
Database alerts system of failure to retrieve information
System alerts investor that the search failed on Browser
Investor
System
Database
Yahoo! Finance

Figure 3.4: See UC-2 on page 15. This use case is triggered when an investor searches a stock with symbols. The system sends a search request to the database, which returns a list of corresponding stock symbols to the system, which displays the stock symbols. And the investor selects the symbol of the stock he/she wants to see from the list. The system requests the stock information corresponding to the symbol from the database, which requests the stock information from Yahoo! Finance. Finally, Yahoo! Finance provides the relevant stock and market information to the investor. If Yahoo! Finance cannot find the information requested by the database, it returns a no result to the database, which returns a failure to retrieve to the system. If Yahoo! Finance cannot find the information, it returns "no result" to database. Database returns "failure to retrieve" to system and system returns "failed" to investor.

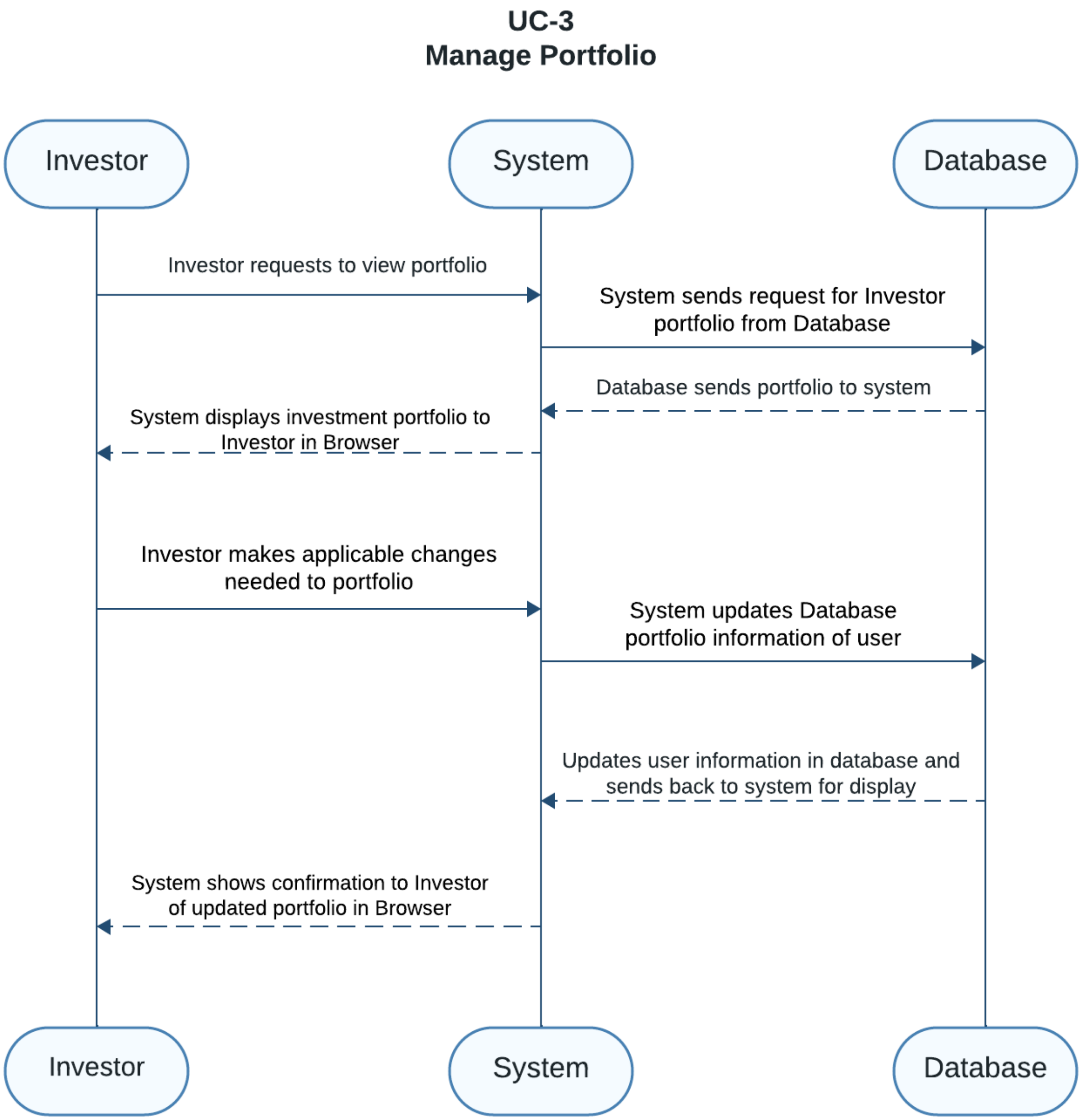

UC-3
Manage Portfolio
Investor
System
Database
Investor requests to view portfolio
System sends request for Investor portfolio from Database
Database sends portfolio to system
System displays investment portfolio to Investor in Browser
Investor makes applicable changes needed to portfolio
System updates Database portfolio information of user
Updates user information in database and sends back to system for display
System shows confirmation to Investor of updated portfolio in Browser
Investor
System
Database

Figure 3.5: See UC-3 on page 15. This shows the investor to system to database back to investor in regards for the user to manage the portfolio.

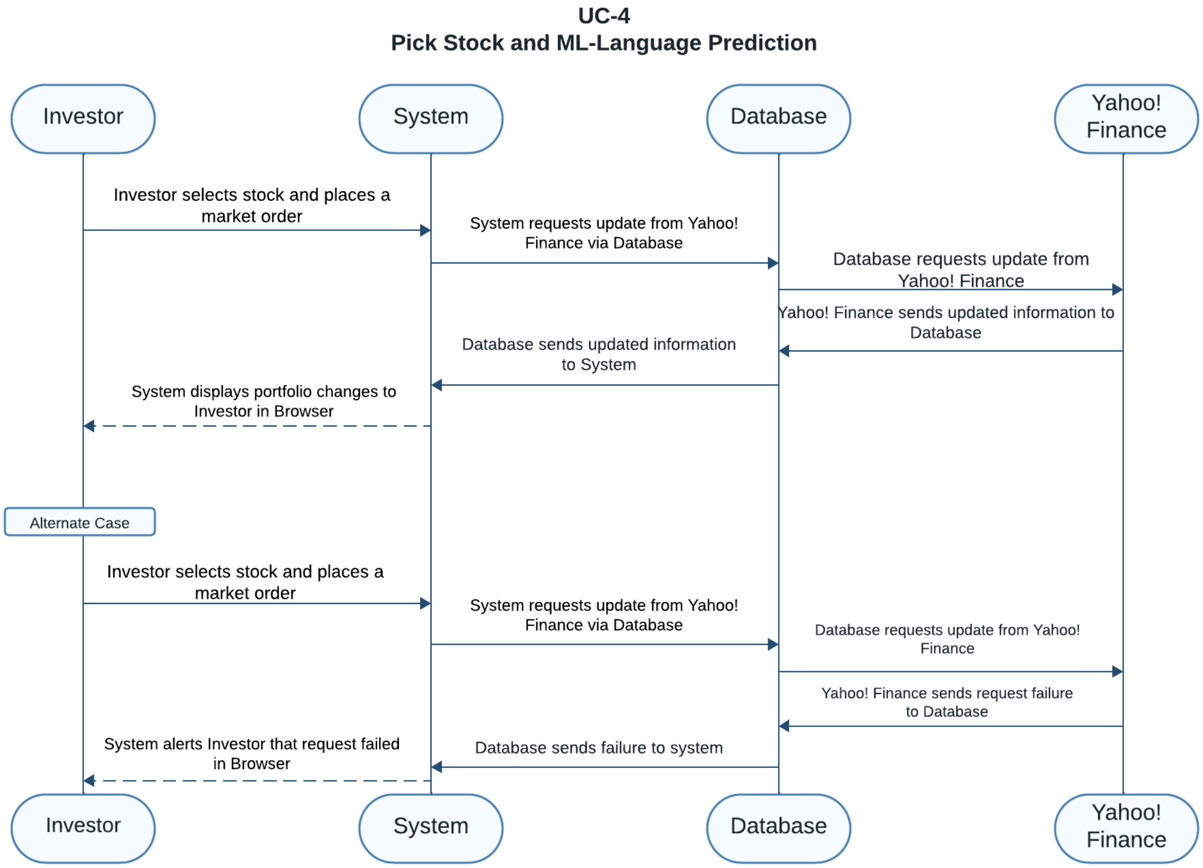

UC-4
Pick Stock and ML-Language Prediction
Investor
System
Database
Yahoo! Finance
Investor selects stock and places a market order
System requests update from Yahoo! Finance via Database
Database requests update from Yahoo! Finance
Yahoo! Finance sends updated information to Database
Database sends updated information to System
System displays portfolio changes to Investor in Browser
Alternate Case
Investor selects stock and places a market order
System requests update from Yahoo! Finance via Database
Database requests update from Yahoo! Finance
Yahoo! Finance sends request failure to Database
Database sends failure to system
System alerts Investor that request failed in Browser
Investor
System
Database
Yahoo! Finance

Figure 3.6: See UC-4 on page 16. This shows the Investor to our system to the database to Yahoo Finance API and back to pick a stock and use it for ML-Language prediction.

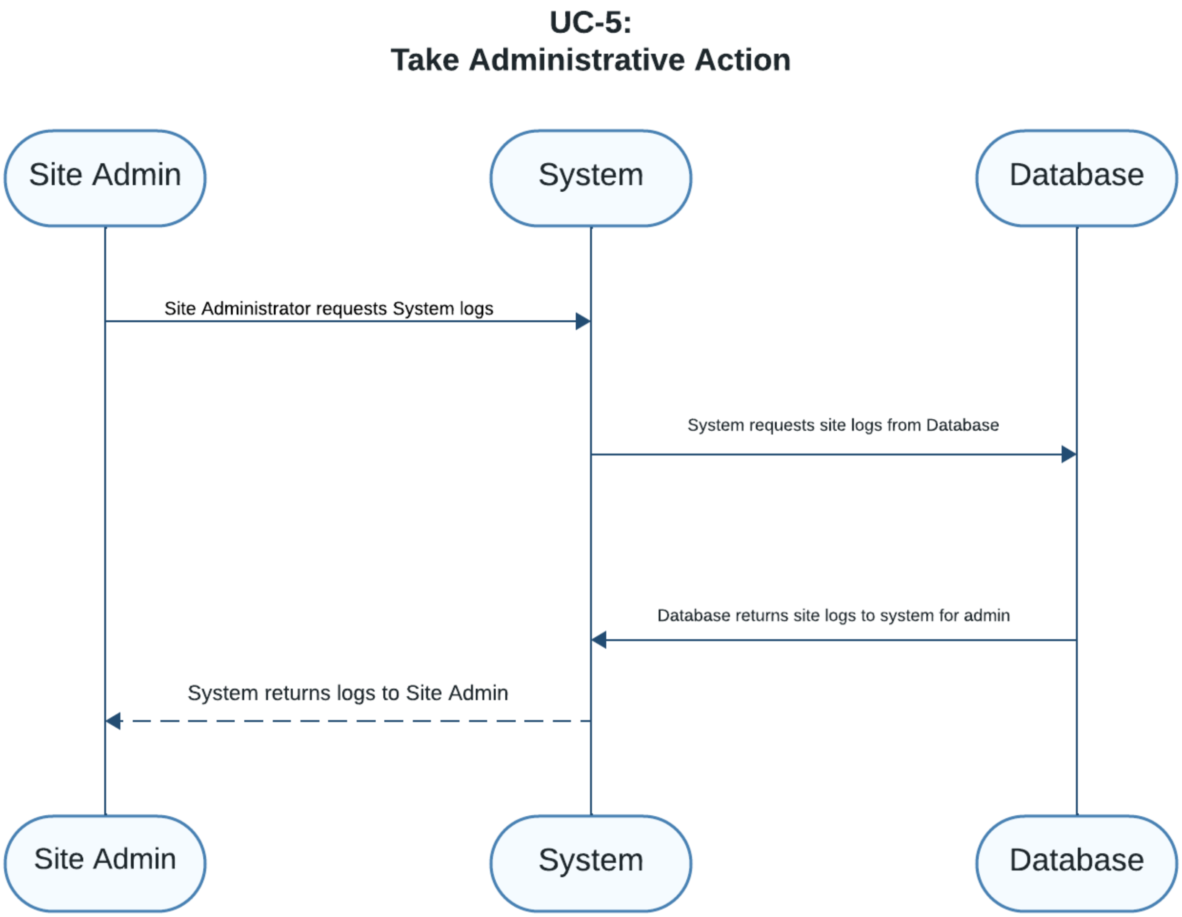

UC-5:
Take Administrative Action
Site Admin
System
Database
Site Administrator requests System logs
System requests site logs from Database
Database returns site logs to system for admin
System returns logs to Site Admin
Site Admin
System
Database

Figure 3.7: See UC-5 on page 16. This figure shows how the site admin will verify and look into various issues within the system and database via logs.

User Interface Specification

a. Preliminary Design

The user interface (UI) for ML-Stock Prediction will act as the a “command center” for users to interact with various stock values for research and possible orders. This “command center” will act as the primary but not only way for users to interact with the system. It will provide a snapshot of the users’ current stock prediction, current prices of Stocks.

The UI should be lightweight to insure it will not burden mobile and tablet users. The color scheme will be chosen to be easy on the viewer, even though this is subjective. The color scheme will be a basic pallet of grey, black, blue, white, possibly red. It will be using more pastel and web supported colors for any additional items.

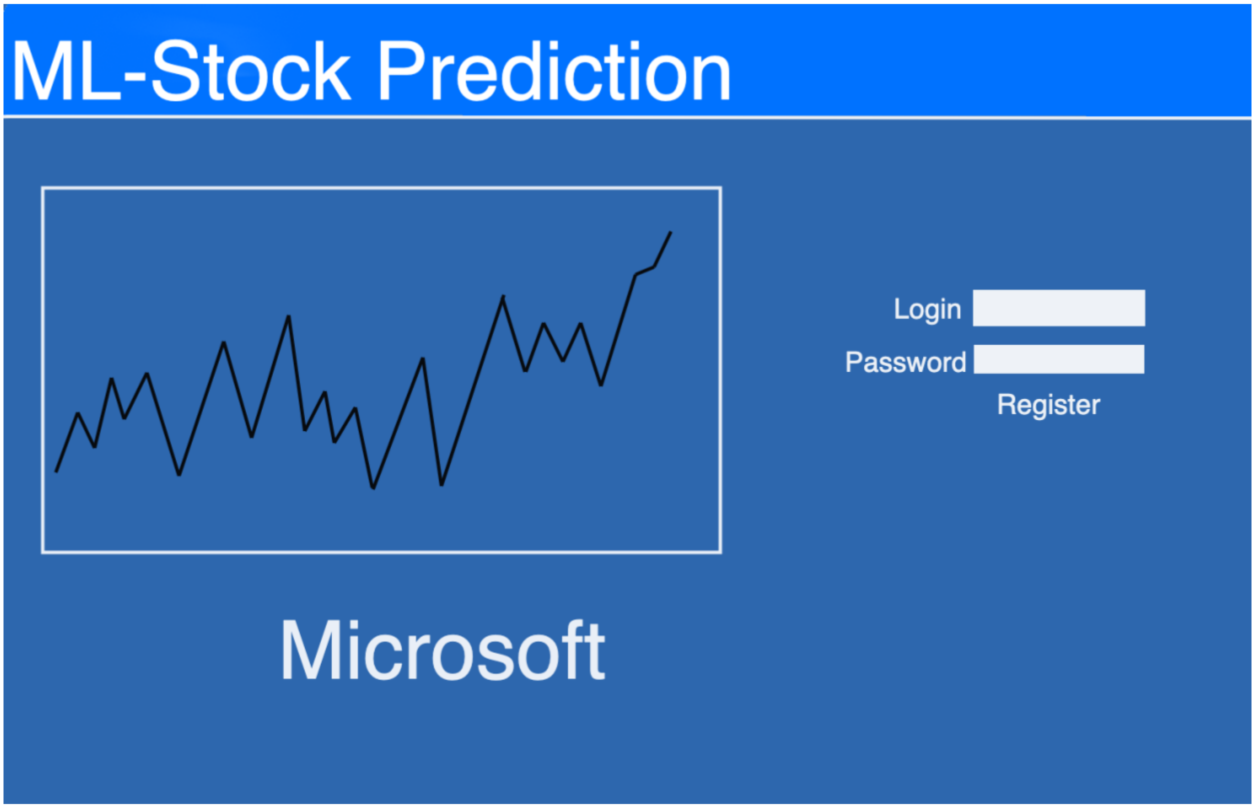
The UI will be built on top of an open-source framework with: ‘Bootstrap CSS’ framework being an option. This is to help deliver content to the three target platforms: desktop, mobile and tablet browsers.

Landing Page and Login

ML-Stock Prediction is designed around users to begin using the service. To accomplish this, the user will need to register with a username/account with our system. We hope in future to allow to transition to a zero-effort registration system allowing users to use their Google accounts to log in. The landing page will have a random graph with a stock showing the chart of price of historical data and then future prediction value.

Global Header

The header across the website will remain persistent across the website once the user is logged into the system. Navigation will essentially 2-3 views in the order: User Information, Stock Selection, Analyze/Predict Stock.

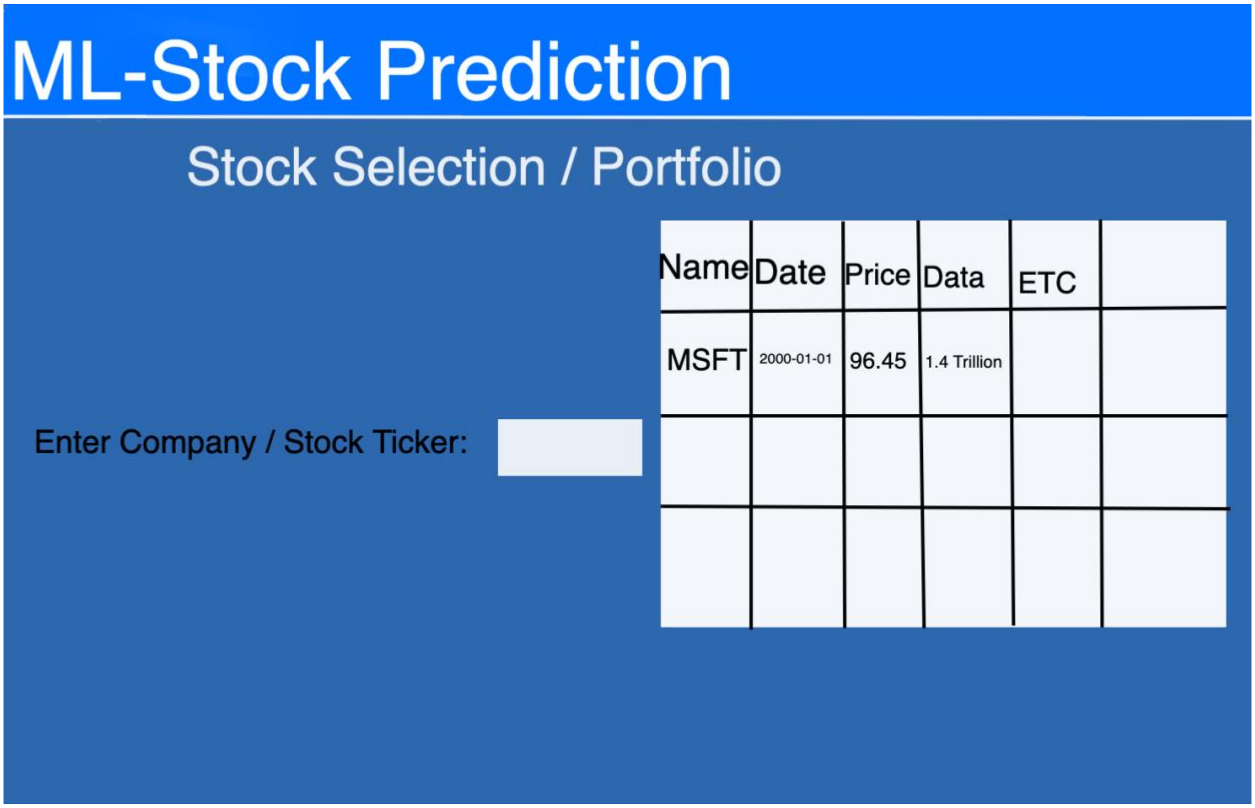


Home page Mock-up (subject to change)

Stock Selection

The ‘Stock Selection’ view of the website will act as the ‘command center’ for a user to get information about companies. This will include information about the stock pricing, and other data available via the Yahoo Finance API.

Most Importantly it provides what stock the user picked over time for stock prediction.



ML-Stock Prediction

Stock Selection / Portfolio

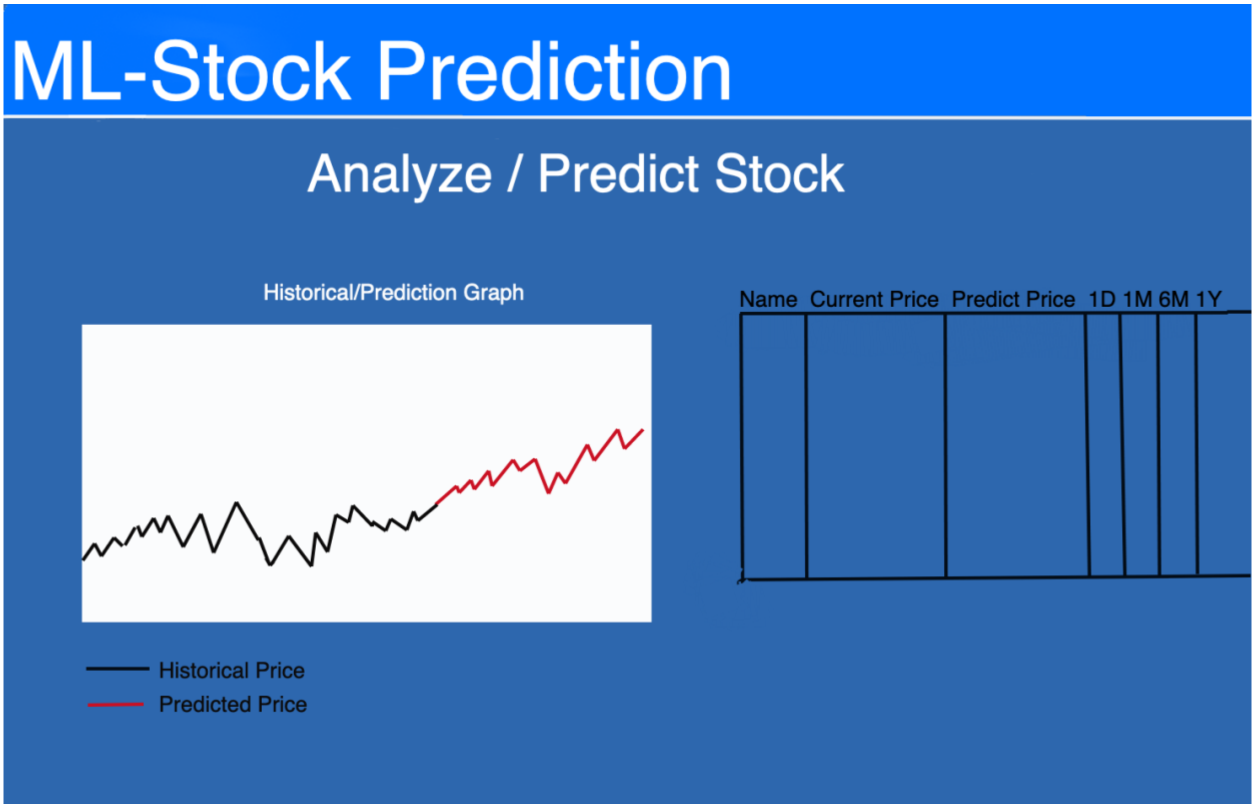
Enter Company / Stock Ticker:

Name	Date	Price	Data	ETC	
MSFT	2000-01-01	96.45	1.4 Trillion		

Stock Selection Page (*subject to change)***

Analyze/Predict Stock

This webpage will show the stocks the user selected and current value and what it predicts it will be in near future and on various timelines, e.g. (next day, 1 week, 1 month, 1 year, 5 year).
The user will receive a graph showing values and a chart that details the values. The chart will include company name, predicted stock price, current stock price, and which timeline selection.



ML-Stock Prediction

Analyze / Predict Stock

Name	Current Price	Predict Price	1D	1M	6M	1Y

***Analyze / Predict Stock Page (**subject to change)*

b. User Effort Estimation

Several of the Most common usage scenarios for ML-Stock Prediction

Usage Scenario	Clicks	Keystrokes
Register/Login	2-3	1-20
Stock Selection	3-6	2-12
Analyze/Predict Stock	2	2-12

Register/Login

Assume the user has come to the website and wishes to Login if already registered, or register if already a user:
Navigation:

1. Click on Login/Register icon
2. Click on your account
3. Click on login or hit enter.

Stock Selection

Assume the user has already logged in and they wish to place a look up a stock ticker:

Navigation:
1. Navigate to ’My Stocks’, 0-1 clicks.

Data Entry:
1. Click textbox. 1 click,
2. Enter assets stock ticker name e.g.: ’M’, ’S’, ’F’, ’T’, 2-12 keystrokes
3. Click start button, 1 click

Analyze/Predict Stock

Assume user is logged in and want to start an in-depth analysis of an asset:

Navigation:
1. Click on Stock, 1 click.

Data Entry:

1. Click on the textbox for entering a stock name, 1 click.
2. Enter Stock Name/Stock Ticker, 2-12 keystrokes.
3. Hit enter, 1 keystroke.

Project Management and References

a. Plan of Work and Product Ownership

Team members from all three groups will support one another based on the complexity of the task, and knowledge sharing is crucial throughout this process.

Team Pair 1: Lisa & Brian

- Functionality: User registration, account management, and authentication.
- Qualitative Property: Ensuring data security and user privacy.
- Ownership: Lisa will lead the registration and authentication development, while Brian will focus on account management.

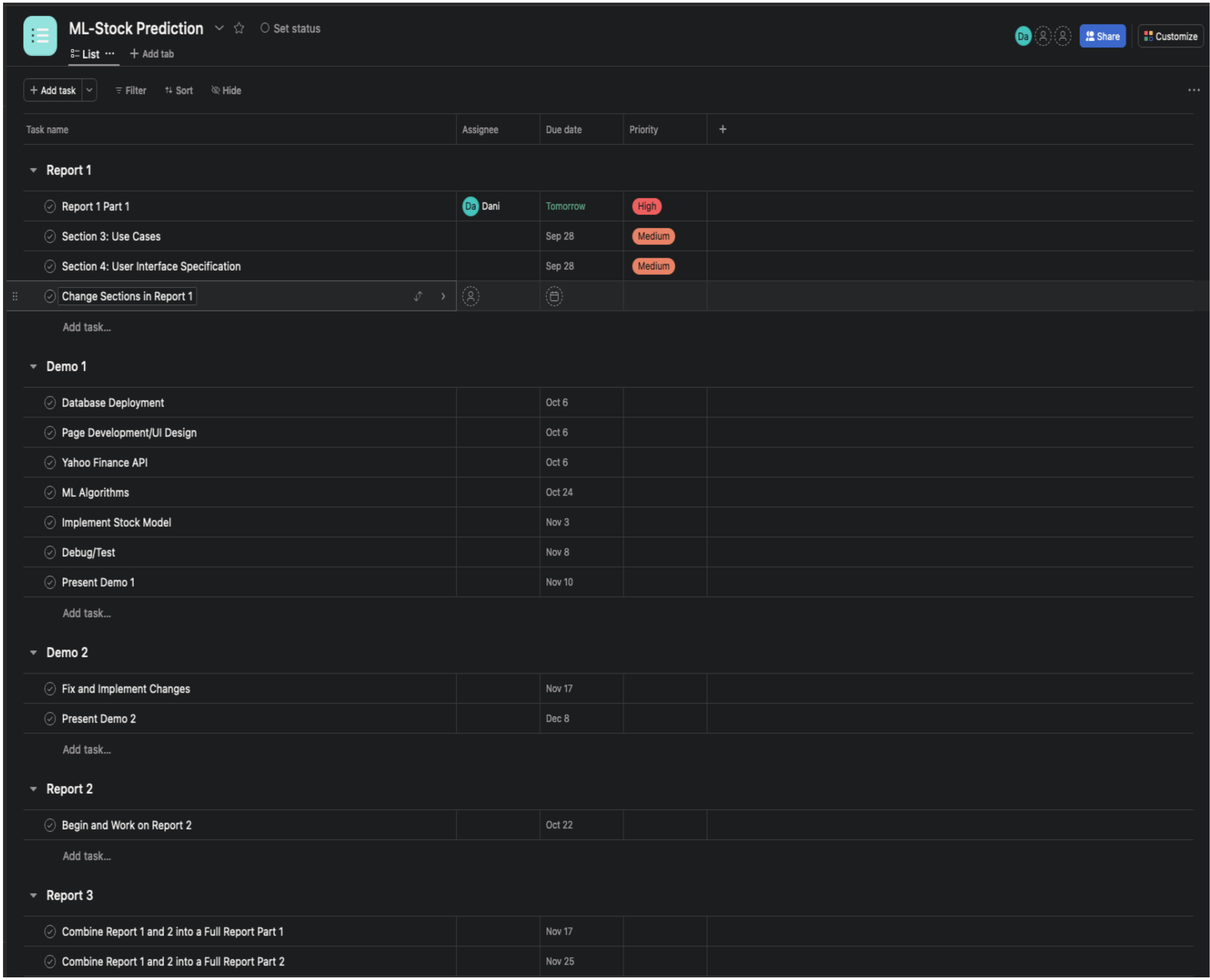
Team Pair 2: Keita & Curtis

- Functionality: Data capture, storage, and integration with financial APIs.
- Qualitative Property: Optimize API response time for improved speed.
- Ownership: Keita will work on data capture and integration, while Curtis will focus on optimizing API responses.

Team Pair 3: Alex & Taylor

- Functionality: ML-based stock price predictions, data visualization, and portfolio management.
- Qualitative Property: Ensure the accuracy of ML predictions.
- Ownership: Alex and Taylor will lead the development of the ML prediction model and data visualization, with the backup from team assisting as needed.

b. Project Roadmap



Task name	Assignee	Due date	Priority
Report 1			
Report 1 Part 1	Da Dani	Tomorrow	High
Section 3: Use Cases		Sep 28	Medium
Section 4: User Interface Specification		Sep 28	Medium
Change Sections in Report 1			
Add task...			
Demo 1			
Database Deployment		Oct 6	
Page Development/UI Design		Oct 6	
Yahoo Finance API		Oct 6	
ML Algorithms		Oct 24	
Implement Stock Model		Nov 3	
Debug/Test		Nov 8	
Present Demo 1		Nov 10	
Add task...			
Demo 2			
Fix and Implement Changes		Nov 17	
Present Demo 2		Dec 8	
Add task...			
Report 2			
Begin and Work on Report 2		Oct 22	
Add task...			
Report 3			
Combine Report 1 and 2 into a Full Report Part 1		Nov 17	
Combine Report 1 and 2 into a Full Report Part 2		Nov 25	

c. References

1. https://bitflyer.com/en-jp/s/regulations/security