
Diamond Hands Investment Fantasy League

Report #1
Software Engineering
14:332:452

[Repository Link](#)

Group 3:

Aarushi Pandey
Aarushi Satish
Apoorva Goel
Christine Mathews
David Lau
Jacques Scheire
Jawad Jamal
Krishna Prajapati
Riya Tayal
Sahitya Gande
Yati Patel
Yatri Patel

Table of Contents

1 Contribution Breakdown	5
2 Work Assignment	5
2.1 Sub-Group 1: User Profiles	5
2.2 Sub-Group 2: Portfolio Management	5
2.3 Sub-Group 3: Graph & News	6
2.4 Sub-Group 4: League Creation w/ Settings	6
2.5 Sub-Group 5: AI Stock Trading Bot	6
3 Customer Problem Statement	7
3.1 Problem Statement	7
3.2 Decomposition into Sub-problems	10
3.2.1 User Profiles	10
3.2.2 Portfolio Management	11
3.2.3 Graphs & News	12
3.2.4 League Creation w/ Customizable Settings	12
3.2.5 Optional AI Players to Leagues	13
3.2.6 Tooltips	13
3.3 Glossary Terms	13
4 Goals, Requirements & Analysis	14
4.1 Business Goals	14
4.2 Enumerated Functional Requirements	15
4.2.1 User Profiles	15
4.2.2 Portfolio Management	15
4.2.3 Graphs & News	15
4.2.4 League Creation w/ Customizable Settings	16
4.2.5 Optional AI Players to Leagues	16
4.3 Enumerated Non-Functional Requirements	17
4.4 User Interface Requirements	17
4.4.1 Registration/Login Page	18
4.4.2 Leagues	18
4.4.3 League Creation + League Manager Settings	19
4.4.4 Join Leagues	19
4.4.5 Portfolio Overview	20
4.4.6 Order Placement	20
4.4.7 Individual Stock	21

4.4.8 News	21
4.4.9 Leaderboard	22
5 Use Cases	22
5.1 Stakeholders	22
5.1.1 Inexperienced Investors	22
5.1.2 Experienced Investors	23
5.1.3 Educational Facilities	23
5.1.4 Sponsors	23
5.2 Actors & Goals	23
5.2.1 Initiating Actors:	23
5.2.2 Participating Actors:	24
5.3 Use Cases	24
5.3.1 Casual Description	24
5.3.1.1 UC-1: Log In	24
5.3.1.2 UC-2: Create League	24
5.3.1.3 UC-3: Join League	24
5.3.1.4 UC-4: Add AI Players	25
5.3.1.5 UC-5: View Portfolio Status & Information	25
5.3.1.6 UC-6: Inspect Transaction History	25
5.3.1.7 UC-7: View League and Rankings	25
5.3.1.8 UC-8: Read News	25
5.3.1.9 UC-9: Research Equity Information	25
5.3.1.10 UC-10: Trade	26
5.3.1.11 UC-11: Inspect Tooltips	26
5.3.1 Use Case Diagram	27
5.3.2 Traceability Matrix	27
5.3.3 Fully-Dressed Description	29
5.3.3.1 UC-4: Add AI Players	29
5.3.3.2 UC-5: View Portfolio Status & Information	30
5.3.3.3 UC-7: View League & Rankings	31
5.3.3.4 UC-9: View Equity Information	32
5.3.3.5 UC-10: Make Trades	32
5.4 System Sequence Diagrams	34
5.4.1 Use Case UC - 4: Add AI Players	34
5.4.2 Use Case UC - 5: View Portfolio Status & Information	35
5.4.3 Use Case UC - 7: View League & Rankings	36
5.4.4 Use Case UC - 9: View Equity Information	37
5.4.5 Use Case UC - 10: Make Trades	38

6 User Interface Specifications	39
6.1 Preliminary Design	39
6.1.1 Create AI Bot	39
6.1.2 View Portfolio Status & Information	39
6.1.3 View League & Rankings	40
6.1.4 View Equity Information	42
6.1.5 Make Trades	44
6.2 User Effort Estimation	46
6.2.1 Create AI Bot	46
6.2.2 View Portfolio Status & Information	47
6.2.3 View League & Ranking	47
6.2.4 View Equity Information	47
6.2.5 Sell/Buy Page	48
7 System Architecture	53
7.1 Identifying Subsystems	53
7.2 Architectural Styles	53
7.3 Mapping Subsystems to Hardware	54
7.4 Connectors and Network Protocols	54
7.5 Global Control Flow	55
7.5.1 Order of Execution	55
7.5.2 Time Dependency	55
7.6 Hardware Requirements	55
8 Project Size Estimation based on use case points	56
8.1 Table I	56
9 Project Management	56
9.1 Groups:	56
9.2 Project Timeline	58
10 References	58

1 Contribution Breakdown

All team members contributed equally.

2 Work Assignment

The division of work was decided based on the functionalities of the project

2.1 Sub-Group 1: User Profiles

- Aarushi Pandey
 - Aarushi Pandey is proficient in C++, python, and is familiar with Java. She has also worked with SQL programming.
- Aarushi Satish
 - Aarushi Satish is proficient in Java, C++ and Python. She also has work experience with Scala and SQL. She is also comfortable working with HTML, CSS and Javascript

2.2 Sub-Group 2: Portfolio Management

- David Lau
 - David is proficient in Python, Javascript, and Java. He is comfortable with a variety of programming paradigms and has experience developing full stack applications.
- Jacques Scheire
 - Jacques has a lot of past experience creating full stack projects and is most comfortable developing web based applications. He works most often with Java, Javascript, and Python and has made use of many available Javascript frameworks important for web development and API development.

- Jawad Jamal
 - Jawad is proficient in Java and C++, and is familiar with Javascript, HTML/CSS and online trading and the stock market. He also has some experience working with Python and object oriented programming.

2.3 Sub-Group 3: Graph & News

- Riya Tayal
 - Riya has past experience with creating web based applications and is most comfortable with frontend development programming. She is proficient in Java, Javascript, C++, HTML, CSS/SCSS and has a good understanding of object-oriented programming. Additionally she has experience using Node.js, Handlebars, Angular JS and React.
- Sahitya Gande
 - Sahitya is proficient in C++ and is familiar with Java. She also has some experience with HTML and iOS development.

2.4 Sub-Group 4: League Creation w/ Settings

- Apoorva Goel
 - Apoorva is proficient in Java and C. She also has a little experience with Python and functional programming languages.
- Yatri Patel
 - Yatri is proficient in Java and C, and is familiar with C++. She also has experience with Python and object oriented programming.

2.5 Sub-Group 5: AI Stock Trading Bot

- Christine Mathews

- Christine is proficient in Java and C. She has some experience developing web applications utilizing Python, HTML/CSS and Javascript.
- Yati Patel
 - Yati is proficient in Java, C and Python. She is comfortable with object-oriented programming. She also has some experience with HTML/CSS.
- Krishna Prajapati
 - Krishna is proficient with the Java, C, and C++ languages, and he is well acquainted with object-oriented programming and RESTful APIs. He also has some experience with the Python, HTML/CSS, and Javascript languages.

3 Customer Problem Statement

3.1 Problem Statement

In the modern day, many students seem to have similar aspirations for their future. These young adolescents plan to go to college, get a degree, find a well-paying job, and simply live their lives working. Most of our time, as youths, is spent learning how to make money with hard work and dedication, but not nearly enough time learning how to make money with money. Investing is a key factor in achieving long-term financial security by creating multiple sources of income for yourself. For many, making sound investments is a critical step towards retirement and financial independence. The notion of investing in the stock market is actually a very well-known and well-tried passive form of income. However, even though millennials acknowledge the fact that investing in the stock market is an intelligent decision, many still have an abundance of valid concerns when it comes to investing, describing the market as complicated, risky, and overwhelming [1].

This fear similarly extends to older generations. Investing is a timeless decision and it is never too late to start. When one makes the conscious decision to set aside money to invest, they are developing discipline by displaying a clear concern for their future and financial wellbeing. Unfortunately, an April 2020 poll indicated that a mere 55% of

Americans own stocks [2], which means that a significant portion of the population have absolutely no stake in the market. While investing is important for the young, who can tolerate more risk and have time on their hands to let their investments grow exponentially, it is equally as important for individuals that are closer to retirement. They tend to have much more buying power than younger generations, which can contribute to larger returns. Even small percent gains for this demographic can result in a significant monetary gain. It may even provide income well after retirement.

This lack of financial literacy is one of the main reasons why many Americans face problems with saving and investing. Approximately half of the American population does not have enough money saved to retire comfortably. Financial literacy is crucial because it equips people with the knowledge and skills necessary to manage money and grow it effectively [3]. Additionally, it aids in making smart, well-calculated financial decisions that help them achieve financial freedom. To make matters worse, a large portion of Americans will not be able to rely on Social Security as a passive income in retirement, since the reserves are projected to be depleted by 2035 [4]. This makes it more imperative than ever for people of all age groups to have multiple sources of income, specifically passive incomes like investments.

While getting involved in the stock market does not require a vast knowledge of technology or years of study, getting involved without the proper background knowledge and experience can be a highly risky endeavour. Especially in light of recent news relating to GameStop and AMC, the market has proven itself to be quite volatile and unpredictable for rookie investors. With new, attractive and accessible online brokers like Robinhood, getting started with investing is the easiest it has ever been. Unfortunately this also means that there will be inexperienced investors that are tempted to throw their money at the market and make uninformed purchases. As we have seen in recent years, this can be a costly mistake and many young, eager investors have suffered as a result.

To combat this national problem, our goal is to offer a platform where both new and experienced investors can trade stocks with imaginary money to learn how to intelligently invest their capital.

New Investor Perspective:

As an inexperienced investor, I have very little awareness of how stocks operate. I want to start investing in the stock market, but I'm terrified at the idea of losing money. I want to learn more about the market before I start investing, but I feel that the only effective way to learn would be through hands-on experience. An app like DHIFL would fix this problem for me entirely by letting me trade with fake virtual money. I can watch my stocks rise and fall as if I had invested in the real market, yet still sleep soundly knowing that my money is safe.

Additionally, I tend to find myself overwhelmed by the many resources that are at my disposal. There is so much material on the stock market and I have no understanding of its meaning or value at this stage in my stock market career. DHIFL would help me tremendously through not only their tooltips for important financial terms, but also through their specialized news feed that shows me exactly what information is relevant to my portfolio.

I also have friends who want to learn more about the market, so this would be a great opportunity to see how they would invest. We would be able to help each other learn more about the market by tracking each other's performance in comparison to ours. Through the app, I am also able to compete against my friends, which motivates me to learn more about the market in order to give myself an edge when climbing the league rankings. I also have the ability to see other users' portfolios, which I can use to size up other players and compare my performance to theirs.

Experienced Investor Perspective:

As an experienced investor, I have already bought and sold stocks through a real broker. I've learned enough to make a profit in the long run, but there are some risky strategies I would like to test out in order to refine my trading tactics. I want to experiment with these new ideas to find what generates the most profit, but I do not want to risk losing actual money in the process. DHIFL provides a stress-free environment where I can invest virtual money and see how my strategies might play out in the real stock market. The application has an extensive portfolio management system that allows me to place market and limit orders on equities. With a variety of options to play around with, there are an endless amount of scenarios that I could implement with no risk.

I would also appreciate having access to a community that will bring a fresh perspective and help me adapt to newer strategies. With the option to join/create leagues and engage in healthy competition, there are so many valuable insights that I can gather from my competitors. I can also compare my portfolio to others through the league rankings, and use other players as a metric to assess the effectiveness of my own investments.

I would want an application that resembles the online broker websites that I currently use to invest because I enjoy their design layout. I would also prefer to have access to metrics to assess my portfolio's growth. I feel that comparing my performance to index funds, such as the S&P 500, as well as assessing various graphs and historical data of all my equities and my portfolio gives me a more complete view of how much of my money's potential is being achieved.

Additionally, the app offers an optional AI trading bot that can be added to my leagues. The bot's portfolio can be viewed as a real player that effectively follows a trading strategy, so it can be used as a baseline metric for my portfolio to be compared to.

While our application would help both new and experienced investors alike, the idea of a stock market fantasy league is not a novel idea. Trading stocks, competing with friends in leagues, and analyzing one's stock data are all features that have been implemented in many past projects, one such project being the Paramount Investments League from Spring 2014. However, our platform will improve upon previous implementations through a new feature and a more modern implementation. Notably, we offer users the option to include AI players in their leagues. This gives new users the opportunity to experience the game alone, which would help those who may be apprehensive about joining leagues with real players. As for more experienced players, the AI component offers another benchmark to evaluate players' performances. The AI will exercise various algorithmic trading techniques, which will utilize data gathered from our stock price endpoint to determine future decisions. The feature that most resembles our AI system is in Group #2's Spring 2014 Project, where they analyzed Twitter posts to determine sentiment value of a stock. However, an AI bot is clearly a distinct approach to this project. It is also worth noting that the last published project to implement a stock market fantasy league was made seven years ago, so we also plan to use more current and cutting-edge technologies to implement our idea. We will also utilize the IEX Cloud API, a modern low-latency API, for gathering stock data and news. The combination of these new technologies will allow our application to have a modern feel with added user responsiveness.

Overall, investing in stocks may seem like a daunting and intimidating task. There are many financial terms, numbers, and graphs that may put off and even overwhelm new investors. For experienced users, potential losses may prevent them from trying new approaches. Our platform will strive to break these barriers by presenting all information in an intuitive, user-friendly, and visually appealing manner that will accelerate the learning process for users of all levels.

3.2 Decomposition into Sub-problems

3.2.1 User Profiles

Users will be able to access a page that contains 3 key elements: current league rankings of users, popular buys and three most recent trades for each particular league. These elements will not necessarily be a literal list or table, but they will be displayed in a way that is easy for the user to see.

On the leagues pages the user can see the name of the league, the number of participants, the user's current ranking, their current portfolio value (with percent gain/loss on a daily basis), and the end date of the league (if applicable) for every league in the table. Therefore, at a glance, the user will be able to see a quick overview of the information they need to play the game. Additionally, each entry name will be clickable and will lead to the actual league page, where the user can actually play the game by managing their portfolio.

3.2.2 Portfolio Management

Every user will have a league-specific portfolio that they can manage. If a user is in multiple leagues, then they will have multiple portfolios to manage. There are three key components to this management: viewing the portfolio, viewing the transaction history, and making a transaction.

When the user views their portfolio, they will be able to see the amount of money they have on-hand, the amount of money in the market, the total value of their portfolio, a graph of their portfolio value, news, and a table of stocks. The graph will display the user's portfolio value over the course of the whole league, as well as the S&P 500 over the same duration. Finally, the user can also view a table of owned stocks, which will display the stock name, stock abbreviation, the current equity price, the average cost of the owned equity, the number of shares, the current value of the equity, and the overall percent gain/loss.

Users can click on any stock to view the stock's price history. They will be brought to the stock screen where they can see general information about the stock, such as market open, market close, P/E ratio, etc. Additionally, the user can see a graph of the stock price over time, which will be explained in "Graphs and News."

The second piece of portfolio management is the history. The user will be able to see their own transaction history during the course of the current league. In a tabular format, the user can see the date and time of the transaction, the stock involved, the type of transaction, and the cost/revenue associated with the transaction.

The last piece to portfolio management is the actual managing itself. Each player will be allowed to perform four main actions as they could do in a real market: buy, sell, limit, and cancel. The user will be able to type the stock name or ticker symbol in a search box to find the desired stock. The information will be retrieved through a back-end API call. From there, they will be given the option to buy shares of the stock if they have sufficient funds or sell their shares if they have sufficient shares. Users are also able to place and cancel limit orders.

3.2.3 Graphs & News

Users will be presented with graphs related to stock and portfolio performance in multiple parts of the application. Upon viewing an individual stock, users will be able to view a standard line time series graph of the stock value over a user-defined interval which will be easy to interpret. Candlestick charts will not be available to provide a simpler new-user experience. The intervals given will be: one day, five days, one month, three months, six months, year-to-date, one year, two years, five years, and all available data up to 15 years. The vertical “stock price” axis will adjust dynamically to capture the maximum variance. Additionally, metrics of percent change over set intervals, e.g. percent change over the past month, will be available below the graph for quick evaluation.

A list of recent news will be available to the user on a separate page. These articles will be aggregated using the same API that was used for stock prices. The articles from API have been pulled from news sites that mention the company name. Additionally, users will have access to a catered news feed, which compiles news of all stocks in a user’s portfolio and presents articles in a list based on date and the user’s positions, i.e. the user’s top holdings will be prioritized. Users will then be able to click on any article and have it open up in a new browser tab.

Within each league, users will be able to view a similarly styled time series graph that depicts the portfolio values of each player. This allows users to easily compare their performance against other players within their league. As with their individual portfolio value graph, S&P 500 will be available to serve as another point of comparison. By comparing the user’s portfolio to the index and observing the degree of this disparity, the user can judge how well they are investing in the current market.

3.2.4 League Creation w/ Customizable Settings

After logging in, a user has an option to create or join a league. When creating a league, the user becomes the league manager. They will have the ability to name the league and send invitations to friends. Upon creation, they are responsible for defining the rules and restrictions for the league, such as starting amount for stock buying power, fees for trading stocks, or even limiting the number of trades within a trading day. These settings cannot be changed after the league is created. League managers also have control over privacy settings, such as determining who is able to join their league.

When users receive an invitation to join a league, they will receive a code that they can input on their screen after they log in. After joining, users will be able to

view the league's dashboard, where they can view the league leaderboard. The league leaderboard will display the user's ranking within the league. Essentially, the user will be able to conveniently see all relevant league information at a glance.

3.2.5 Optional AI Players to Leagues

League managers have the ability to choose if they want to play with a stock trading bot. Our team plans to build a trading bot by utilizing different stock trading algorithms, such as momentum investing and mean reversion, to make calculated assumptions on stock performance to ultimately generate the best return. The data required by the algorithms can easily be pulled from our stock API. In order to make gameplay more fair, the bot will be limited in the amount of transactions that it can perform in a day. A stock trading robot will promote healthy competition within leagues and learning opportunities within the game. These AIs are said to be strategic and resourceful, as they operate with no fear, greed or ego. Investors can benefit from the stock trading bot by using it as a benchmark to compare to their own portfolios.

3.2.6 Tooltips

Users will be able to hover over investment terms and be offered a tooltip. The tooltip will be a brief definition of the term, which will assist in learning and help familiarize the user with the stock market. These terms will be pulled from Investopedia so that we do not have to manually define words. We would also simply be able to automatically search and match words to definitions by matching the words to a dictionary list.

3.3 Glossary Terms

League - A market simulation that has a set of rules decided by a league manager. It allows for several users to compete.

League Manager - League manager has control over the settings of the league. He/She may choose the player starting balance, fees for trading stocks, and limit the number of trades within a trading day. League managers also have control over privacy settings, such as determining who is able to join their league.

Portfolio - Detailed list of all equities owned by a player in a specific league. Will display profits, losses, total return as well as other key information associated with the equity

Market Order - An order placed for immediate execution in the market.

Buy Order - An order placed to purchase a specific number of shares of a stock.

Sell Order - An order placed to sell a specific number of shares of a stock.

Limit Order - An order to either buy/sell a stock at a user-defined price. Investors set a maximum amount they will pay for a stock when buying shares and a minimum amount when selling shares.

Transaction - An agreement between seller and buyer to exchange cash for an asset.

Ticker Symbol - A unique rearrangement of letters assigned to a security for the purpose of trading.

Leaderboard - Ranking system within the league based on total portfolio value of player.

Dashboard - Where users can view current league settings as well as league leaderboard.

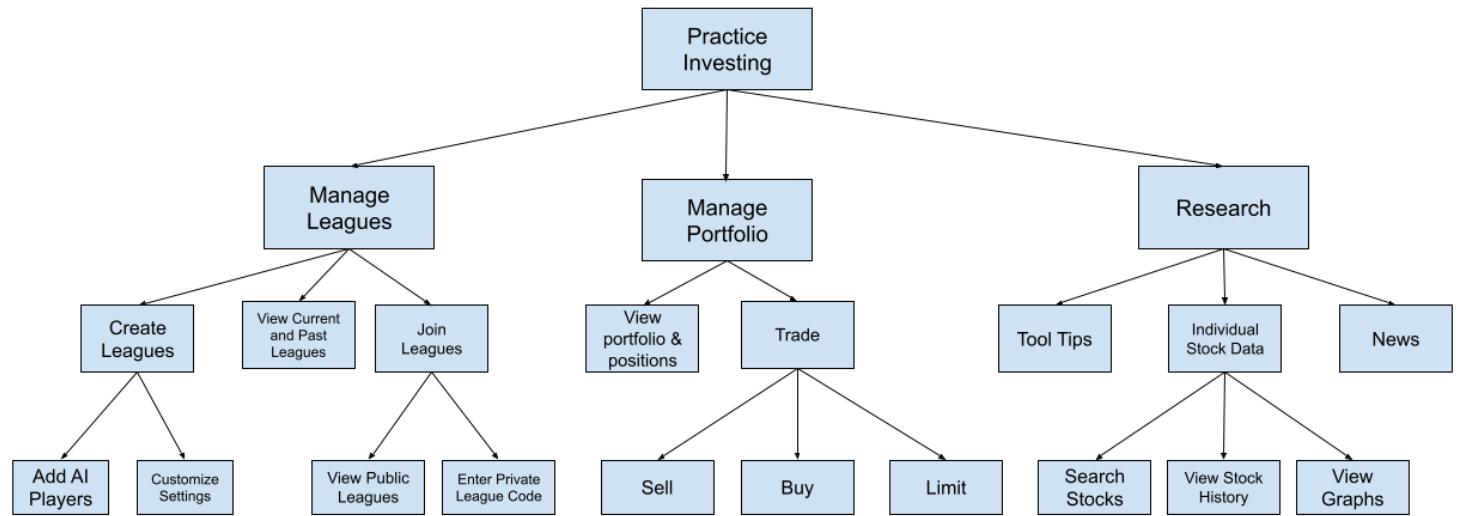
Transaction History- Users can view a transactions from all league members from current leagues

Symbol Lookup - A search bar that allows users to access information about particular stocks.

AI Bot - An artificial intelligence system that competes with other players in leagues to enhance their skill level

4 Goals, Requirements & Analysis

4.1 Business Goals



4.2 Enumerated Functional Requirements

PW = Priority Weight (from 1 to 5)

4.2.1 User Profiles

ID	PW	Requirement Description
REQ-1	2	The system will allow users to register an account and log in with their credentials.

4.2.2 Portfolio Management

ID	PW	Requirement Description
REQ-2	5	The system will allow users to place market and limit orders on equities.
REQ-3	5	The system will allow users to view their positions, transactions, and portfolio value graph in each of their leagues.

4.2.3 Graphs & News

ID	PW	Requirement Description
REQ-4	5	The system will allow users to view prices, graphs, and metrics for individual equities.
REQ-5	2	The system will provide users with a curated news feed for equities in their portfolio.
REQ-6	1	The system will provide users with definitions on financial terms and instruction on how to analyze the graph.

4.2.4 League Creation w/ Customizable Settings

ID	PW	Requirement Description
REQ-7	4	The system will allow users to create new leagues with settings to define limits on gameplay.
REQ-8	2	The system will allow users to join games with an invite from the league manager or public games with unknown users.
REQ-9	1	The system will allow users to view league settings through the league dashboard.
REQ-10	3	The system will allow users to view the portfolio and positions of other users in their league.
REQ-11	2	The system will allow users to view league rankings through a league leaderboard.

4.2.5 Optional AI Players to Leagues

ID	PW	Requirement Description
REQ-12	3	The system will allow league managers to add AI players to their league.
REQ-13	1	League managers will be allowed to choose between a few different trading algorithms for each AI player that they add.
REQ-14	1	The system will limit the AI player from exceeding a certain number of trades per day, determined by the league manager.
REQ-15	5	The AI Player should be able to decide what market trades to

		make throughout the day, regardless of user activity.
REQ-16	3	Multiple AI players in the same league should not have the exact same portfolio.

4.3 Enumerated Non-Functional Requirements

ID	PW	Requirement Description
REQ-17	4	Users are able to access the application via any web browser compatible with ES6 Javascript, e.g., Chrome 51.
REQ-18	2	The application will be consistent across all web browsers.
REQ-19	2	User accounts will be secure and unable to be accessed by other users.
REQ-20	3	The user interface will be responsive with at least eighty percent of actions taking less than 1000 milliseconds to respond.
REQ-21	4	The system should be able to service at least 100 users simultaneously without performance degradation.
REQ-22	3	Equity price data from endpoints will be accessed in under 1000 milliseconds.
REQ-23	3	The system should be able to handle at least 500 transactions per user per day.

4.4 User Interface Requirements

PW = Priority Weight (from 1 to 5)

ID	PW	Requirements
REQ-24	1	Registration/Login Page: The user will be able to create an account to play the game.
REQ-25	3	Leagues: The user will be able to view active and past leagues.
REQ-26	2	League Creation + League Manager Settings: The user will be able to create leagues with settings of their choice.
REQ-27	2	Join League: The user will be able to join a specific league or a random league that is active.

REQ-28	5	Portfolio Overview: The user will be able to view their current holdings and net worth.
REQ-29	4	Order Placement: The user will be able to place orders.
REQ-30	4	Individual Stock: The user will be able to search up a specific stock and view data, graphs and history on it.
REQ-31	2	News: The user will have the option to view news articles based on their holdings.
REQ-32	3	Leaderboard: The user will be able to view their standing in the league.

4.4.1 Registration/Login Page

Diamond Hands Investment League

Username

Password

Not A User? [Register Here](#)

This is the registration/login page. Here, users can put in their identification information to gain access to their fantasy league account. Once they login, they will be directed to the leagues page below. If they are a new member, they can click register here and will be rerouted to an account creation page.

4.4.2 Leagues

The screenshot shows a user interface titled "Your Leagues". On the left, there is a "Profile Pic" placeholder and a sidebar with buttons for "LEAGUES" (highlighted in green), "CREATE LEAGUE", and "JOIN LEAGUE". The main area is divided into two sections: "Current Leagues" and "Past Leagues".

Current Leagues	
(League Name)	\$9432.23 +123.43 (+1.45%)
(League Name)	\$12432.23 -2344.23 (-6.34%)

Past Leagues	
(League Name)	\$9001.00 3rd place
(League Name)	\$145.99 7th place

The leagues page shows what leagues the user is participating in and past leagues. There are options to create a league or join another league. If a league name is pressed, it will direct the user to the portfolio overview page for that specific league. If the create leagues button is pressed, the user will be prompted to the league creation and league manager settings slide below. If the join league button is pressed, the user is redirected to the join leagues page.

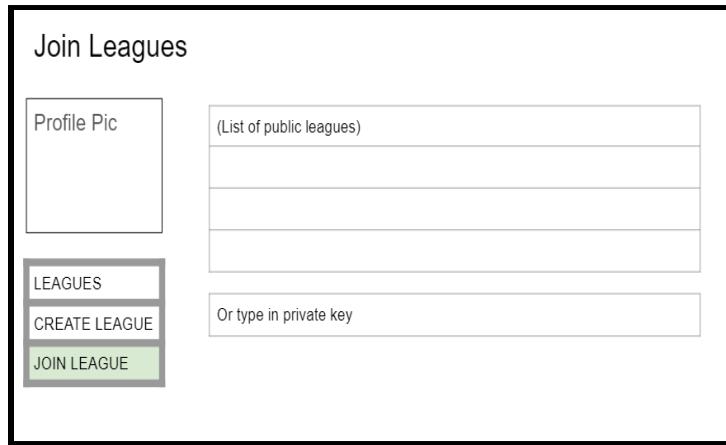
4.4.3 League Creation + League Manager Settings

The screenshot shows a user interface titled "Create League". On the left, there is a "Profile Pic" placeholder and a sidebar with buttons for "LEAGUES" (highlighted in green), "CREATE LEAGUE" (highlighted in green), and "JOIN LEAGUE". The main area has a message "Please fill out the following information:" followed by several input fields:

- League Name
- Starting Balance
- Commission Percentage
- Trade/Day Limit
- Private/Public?
- # of AI players

The league creation page allows the league manager to create their league. The league manager has the ability to customize the league according to his/her preference and must fill out the fields above. League managers can set custom starting costs for all the players in their league. The league manager will also have the ability to limit the amount of stocks each player can trade per day. League managers can also limit the numbers of players that can participate in their league, along with having the option to add AI players to their league.

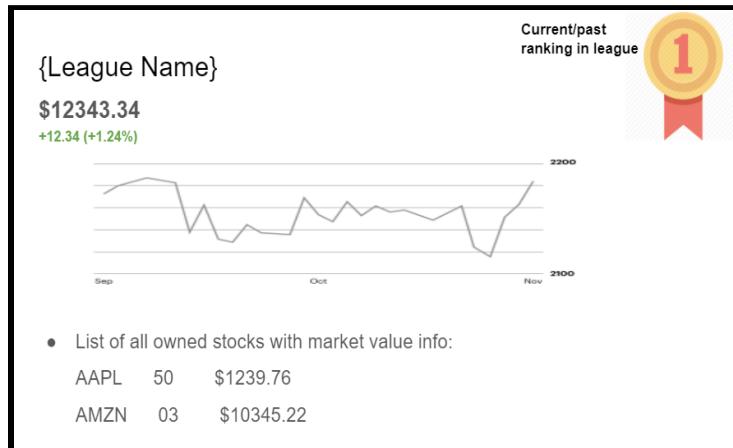
4.4.4 Join Leagues



The interface for joining leagues consists of several input fields and buttons. On the left, there is a placeholder for a profile picture. To its right is a list of public leagues. Below these are three buttons: 'LEAGUES' (disabled), 'CREATE LEAGUE' (disabled), and 'JOIN LEAGUE' (highlighted in green).

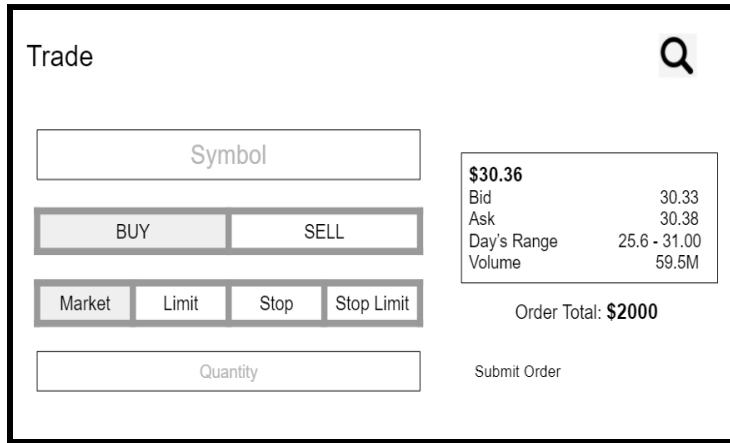
On this page, the user will be able to join public or private leagues. To join a private league, the league manager will receive a code that they will be able to share with the other members that want to join the league. There will also be a list of public leagues that users can find and join if interested.

4.4.5 Portfolio Overview



This page allows the user to view their progress. They will be able to see their portfolio value in that specific league, as well as their daily percent returns. Their portfolio value history will also be displayed in the form of a time series line graph. The user's current ranking will also be shown as well. If the badge is selected, the user will be directed to the leaderboard.

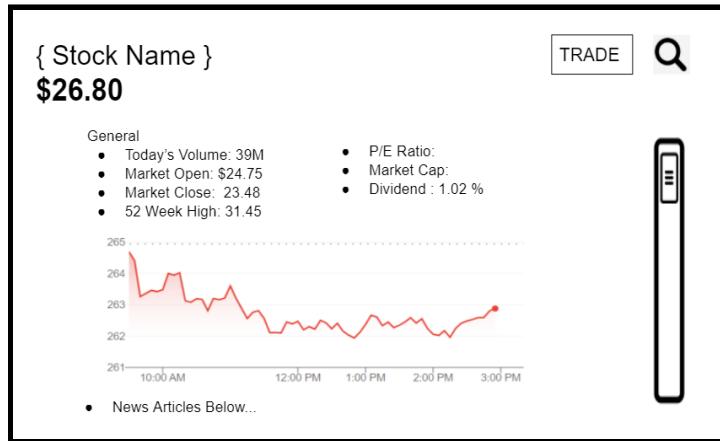
4.4.6 Order Placement



The form is titled "Trade". It includes a search icon and a "Symbol" input field. Below it are "BUY" and "SELL" buttons. Underneath are "Market", "Limit", "Stop", and "Stop Limit" buttons. A "Quantity" input field is present. To the right, there is a box showing market data: Bid (\$30.36), Ask (\$30.38), Day's Range (25.6 - 31.00), and Volume (59.5M). Below this is an "Order Total: \$2000" label and a "Submit Order" button.

On this page, the user can place a trade. They can decide whether they want to buy or sell, the type of order and the number of shares they want to trade. They are also given the option to search another company, which if pursued, will take them to the individual stocks page. Once the submit order button is clicked, the order is executed and a confirmation message will appear.

4.4.7 Individual Stock



On the individual stock page, the user will be able to see the progress of the stock. They can see how the stock is performing in the market. A graph will display the inclines and declines of all stocks, which are kept up to date with real world time. Users have the option to select the trade button to place an order on that stock, which will reroute them to the order placement page, or search up a different stock, which will reroute them to the individual stock page. After viewing information about the stock, the user will be able to carefully make a decision and decide if they want to trade the stock or not.

4.4.8 News

Top Curated News

- Bitcoin Reaches All Time High!
- Tesla invest \$1.5B in Bitcoin!
- Dogecoin: The Next Best Thing?



The illustration features a blue newspaper with the word "NEWS" at the top. To its right is a gold-colored Dogecoin with a dog's face and the word "WOW" on it.

The general news page will allow users to keep up with the latest information related to stocks based on their account holdings. Once the article name is pressed, the article will be opened in another tab. This will allow the user to have a library of articles where they can obtain specific information about how certain stocks are performing. This will assist the user in making decisions concerning their stocks and give them information when is the best time to trade.

4.4.9 Leaderboard



Leaderboard

Rank	Name	Net Worth
1	User2	\$110,000
2	User1	\$100,050
3	User3	\$99,010

The leaderboard will show the ranking of the user in their league with the most profitable holdings. This ranking is based on how they are performing in the stock market compared to the other members of the league. As users join multiple leagues at once, they will be able to see their ranking in each of the leagues. The leader board will include information of the user's current rank, their account value, and the overall percentage, which shows whether the player lost money or gained money in that given day.

5 Use Cases

5.1 Stakeholders

5.1.1 Inexperienced Investors

One of our primary target demographics are inexperienced investors. Inexperienced investors will be able to use our platform to learn about trading and get hands-on experience in a risk-free environment.

5.1.2 Experienced Investors

Experienced investors will also have a keen interest in the performance of our service. Diamond Hands Investment Fantasy League can serve as a way for experienced investors to experiment with new trading strategies. Additionally, the competitive social aspect of the fantasy league appeals to friend groups and coworkers who are interested in putting their investment skills to the test.

5.1.3 Educational Facilities

Educational facilities will want to utilize our platform as a tool to teach new students about investing. With more schools instituting financial literacy requirements, the demand for fun and educational investment tools is on the rise. Our fantasy league promotes engagement in the market, allowing students to actively exercise skills they have learned with no risk.

5.1.4 Sponsors

Our platform provides an excellent space for sponsors, such as popular online brokers like Robinhood or Fidelity, and advertisers to promote their products. With educational facilities and inexperienced traders in mind, our service allows sponsors to directly interact with a population of users who are just getting into the stock market and are interested in checking out new tools and products.

5.2 Actors & Goals

5.2.1 Initiating Actors:

Actor	Role	Goal
Investor	The user is one that is able to use the platform to trade	The goal of the user is to use the platform to improve their

	equities, view their portfolio, and interact with other users in any league they create or join.	financial literacy and understanding of the market. There will also be the added bonus of healthy competition, which would entice any competitive spirits..
League Manager	The league manager is a user with special privileges that allow them to dictate the rules and settings in a league that they create. A user can become a league manager by creating a league and inviting other players to it.	The goal of the league manager extends the goal of the user; the league manager is responsible for making sure that competition amongst users in their league stays fair throughout the active duration of the league.

5.2.2 Participating Actors:

Actor	Role
Stock Service	The stock service is a system that is responsible for providing stock data, including prices and news, to users.
Database	The database maintains information about user accounts, leagues, and stock data.
AI Service	The AI service is a system that is responsible for utilizing equity data and implementing algorithms in order to decide which equity to trade and when.

5.3 Use Cases

5.3.1 Casual Description

5.3.1.1 UC-1: Log In

A registered user can log in using their credentials to access their secured account. New users can create a new account with specified credentials.

- REQ-1, REQ-19, REQ-24

5.3.1.2 UC-2: Create League

A user can choose to create a new league to play in. They will be able to determine league settings to define the restrictions on gameplay. This user will

become the league manager, a role with special privileges within the league, and be able to invite players.

- REQ-7, REQ-8, REQ-12, REQ-26

5.3.1.3 UC-3: Join League

A user can choose to join a league from a list of public lobbies that are available. If they received a private code from a league manager, they may also enter that code to join a private league.

- REQ-7, REQ-8, REQ-27

5.3.1.4 UC-4: Add AI Players

A league manager can choose to add AI players to their league. AI players will utilize user-selected algorithmic trading techniques, but will only be granted a limited number of actions per day.

- REQ-12, REQ-13, REQ-14, REQ-15, REQ-16, REQ-26

5.3.1.5 UC-5: View Portfolio Status & Information

A user can view their portfolio in every league that they are participating in. Their portfolio will display their rankings, a graph of their portfolio value over time, and a curated news feed based on their positions in that league.

- REQ-2, REQ-3, REQ-5, REQ-10, REQ-22, REQ-28, REQ-31

5.3.1.6 UC-6: Inspect Transaction History

A user can view their order history in their active leagues, as well as recently executed orders of fellow league members. They can also see transactions made by all users in their past, inactive leagues.

- REQ-3

5.3.1.7 UC-7: View League and Rankings

A user can view information about leagues they are in, such as the gameplay settings defined by the league manager. Additionally, they will be able to view a leaderboard displaying player rankings within the league.

- REQ-9, REQ-10, REQ-11, REQ-22, REQ-25, REQ-32

5.3.1.8 UC-8: Read News

A user can view news related to stocks in multiple contexts. Users can view news related to a specific stock on the individual security page. Additionally, a curated news feed will be available on each portfolio. A general news page will also be available.

- REQ-4, REQ-5, REQ-31

5.3.1.9 UC-9: Research Equity Information

A user can search and view a page of financial data pertaining to a specific equity. The user will have access to stock quotes and can see its performance over varying time ranges in the form of a graph. Users can also view news related to a specific stock on the individual security page.

- REQ-4, REQ-22, REQ-30

5.3.1.10 UC-10: Trade

A user can make trades on any available stock of their choice during market hours (9:30AM - 4:00PM ET). They can place market or limit orders, whether they choose to buy or sell and can also cancel a pending order before it is executed.

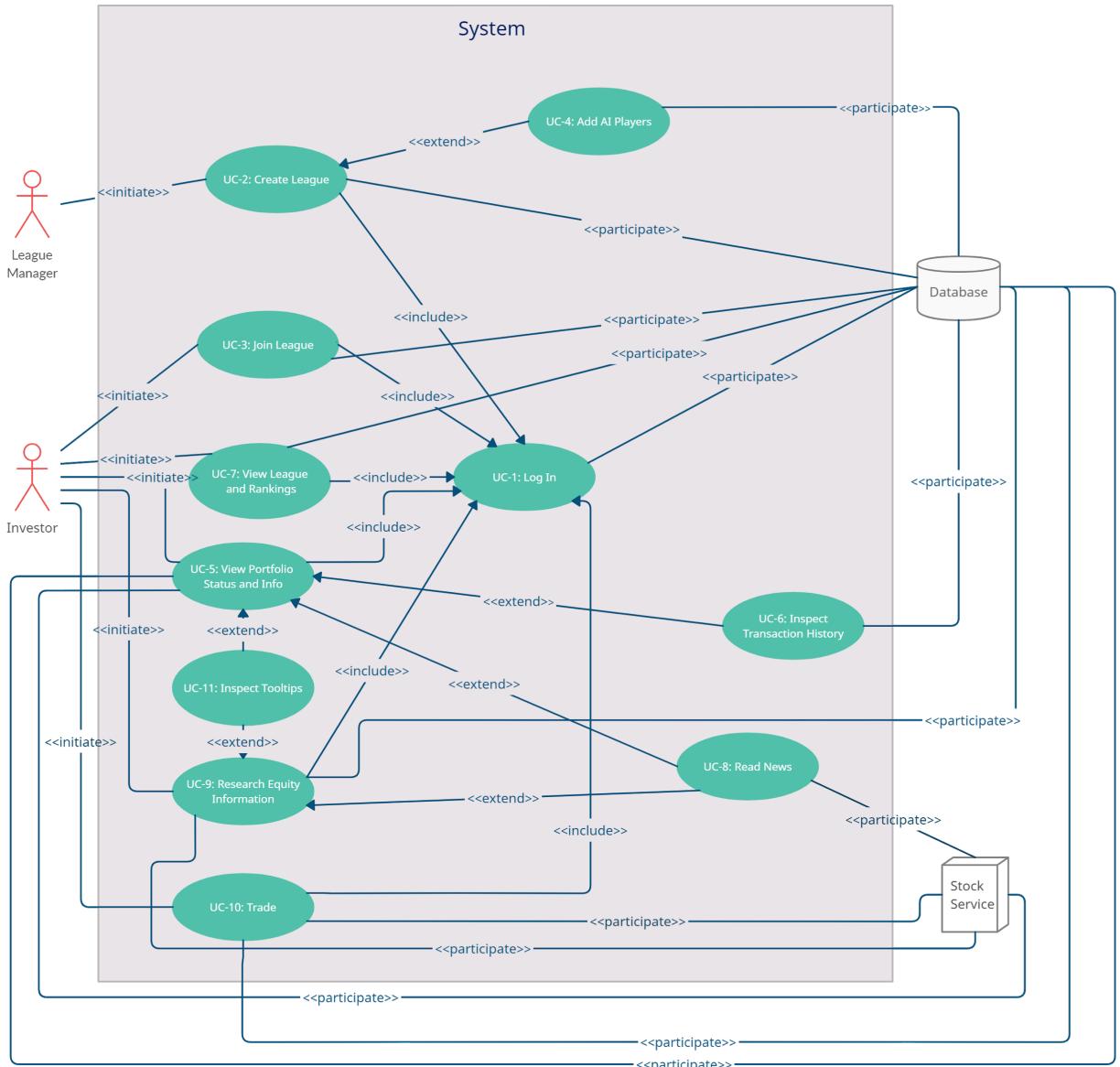
- REQ-2, REQ-22, REQ-23, REQ-29

5.3.1.11 UC-11: Inspect Tooltips

A user can see the definition of different financial terms when they hover over it. Users will also have access to an explanation of the time series graph.

- REQ-4, REQ-6, REQ-30

5.3.1 Use Case Diagram



5.3.2 Traceability Matrix

Priority		UC-1	UC-2	UC-3	UC-4	UC-5	UC-6	UC-7	UC-8	UC-9	UC-10	UC-11
REQ-1	2	X										
REQ-2	5					X						X
REQ-3	5					X	X					

REQ-4	5							X	X		X
REQ-5	2					X		X			
REQ-6	1										X
REQ-7	4		X	X							
REQ-8	2		X	X							
REQ-9	1						X				
REQ-10	3					X		X			
REQ-11	2							X			
REQ-12	3		X		X						
REQ-13	1				X						
REQ-14	1				X						
REQ-15	5				X						
REQ-16	3				X						
REQ-17	4										
REQ-18	2										
REQ-19	2	X									
REQ-20	3										
REQ-21	4										
REQ-22	3					X		X		X	X
REQ-23	3										X
REQ-24	1	X									
REQ-25	3							X			
REQ-26	2		X		X						
REQ-27	2			X							
REQ-28	5					X					

REQ-29	4									X		
REQ-30	4								X		X	
REQ-31	2					X		X				
REQ-32	3						X					
Sum:		5	11	8	15	25	5	15	9	12	15	10

5.3.3 Fully-Dressed Description

5.3.3.1 UC-4: Add AI Players

Related Requirements: REQ-12, REQ-13, REQ-14, REQ-15, REQ-16, REQ-26

Initiating Actor: League Manager

Actor's Goal: To include AI Players in a league they create

Participating Actors: Database, Stock Service, AI Service

Preconditions: The user is the league manager of the league in which they are attempting to add AI players.

Postconditions: One or more AI players are added to participate in the league, each with restrictions on gameplay.

Flow of Events for Main Success Scenario:

- 1) **League Manager** chooses to add AI players to a league that they are managing.
- 2) **System** creates AI players and updates league information in the database.
- 3) **System** tells the AI service to continuously monitor and update this AI player.
- 4) **System** sends a confirmation message to the **league manager**.

Flow of Events for Extensions (Alternate Scenarios or Contingencies):

- 1) **AI Service** is unable to communicate with **Stock Service**
 - a) **AI Service** connection attempt to **Stock Service** times out
 - b) **Stock Service** alerts the **AI Service** of failure to access equity information
 - c) **AI Service** alerts the **System** of failure to access **Stock Service**
 - d) **System** alerts the **League Manager** of AI Players failed performance
- 2) **AI Service** is unable to communicate with the **Database**
 - a) **AI Service** connection attempt to **Database** times out
 - b) **Database** alerts **AI Service** of failure to access and update equity information

5.3.3.2 UC-5: View Portfolio Status & Information

Related Requirements:	REQ-2, REQ-3, REQ-5, REQ-10, REQ-28, REQ-31
Initiating Actor:	Investor
Actor's Goal:	To view one's current positions in an active league and keep track of their total return
Participating Actors:	Database, Stock Service
Preconditions:	The user is logged in, authenticated and a member of an active league.
Postconditions:	The user is now viewing a page that shows their total return on all their holdings in that particular league. A graph of portfolio history is also displayed as well as ranking and recent league activity.

Flow of Events for Main Success Scenario:

- 1) **User** selects a league from the page listing all active leagues.
- 2) **System** requests the user's current positions and league information from the **Database**
- 3) **System** requests all information relevant to the user's portfolio from the **Stock Service** e.g. historical price data, curated news, etc.
- 4) **System** displays **User's** portfolio and ranking in the league selected

Flow of Events for Extensions (Alternate Scenarios or Contingencies):

1. **System** is unable to communicate with the **Stock Service**.
 - a. **System** connection attempt to **Stock Service** times out.
 - b. **System** alerts **User** of failure to access resources.
2. **System** is unable to communicate with the **Database**
 - a. **System** connection attempt to **Database** times out.
 - b. **System** alerts **User** of failure to access resources.

5.3.3.3 UC-7: View League & Rankings

Related Requirements:	REQ-9, REQ-10, REQ-11, REQ-25, REQ-32
Initiating Actor:	Investor
Actor's Goal:	To view league settings and standing in league
Participating Actors:	Stock Service, Database
Preconditions:	The user is logged in, authenticated and a member of an active league.
Postconditions:	The user is now viewing a page where the leaderboard is displayed and the current settings of the league.

Flow of Events for Main Success Scenario:

- 1) **User** selects a league from their current or past leagues list.
- 2) **System** requests league information and user portfolios from the **Database**, which returns the information.
- 3) **System** requests updated stock prices from the **Stock Service** to calculate player portfolio values for current leagues.
- 4) **System** displays league information and rankings.

Flow of Events for Extensions (Alternate Scenarios or Contingencies):

1. **System** is unable to communicate with the **Stock Service**.
 - a. **System** connection attempt to **Stock Service** times out.
 - b. **System** alerts **User** of failure to access resources.

2. **System** is unable to communicate with the **Database**
 - a. **System** connection attempt to **Database** times out.
 - b. **System** alerts **User** of failure to access resources.

5.3.3.4 UC-9: View Equity Information

Related Requirements:	REQ-4, REQ-22, REQ-30
Initiating Actor:	Investor
Actor's Goal:	To view stock quotes, history, and other financial data on a desired company
Participating Actors:	Stock Service, Database
Preconditions:	The user is logged in, authenticated and our stock service is functioning properly.
Postconditions:	The user is now viewing a page with financial data on an equity of their choice.

Flow of Events for Main Success Scenario:

- 1) **User** enters ticker symbol of their desired equity.
- 2) **System** requests **Database** to retrieve all cache data on the selected equity if available.
- 3) **System** validates cache data and requests current equity data from the **Stock service**, which returns updated information.
- 4) **System** transfers all updated information to cache in the **Database**.
- 5) **User** can view updated equity information.

Flow of Events for Extensions (Alternate Scenarios or Contingencies):

1. **User** enters invalid ticker symbol.
 - a. **System** receives invalid notice from **Stock Service**
 - b. **System** alerts **User** of the invalid ticker.

5.3.3.5 UC-10: Make Trades

Related Requirements:	REQ-2, REQ-23, REQ-29
Initiating Actor:	Investor
Actor's Goal:	To place a market or limit order for a specific league's portfolio
Participating Actors:	Stock Service, Database
Preconditions:	The user is logged in, authenticated, and a member of an active league.
Postconditions:	The user's portfolio is updated reflecting the changes in shares of the traded position.

Flow of Events for Main Success Scenario:

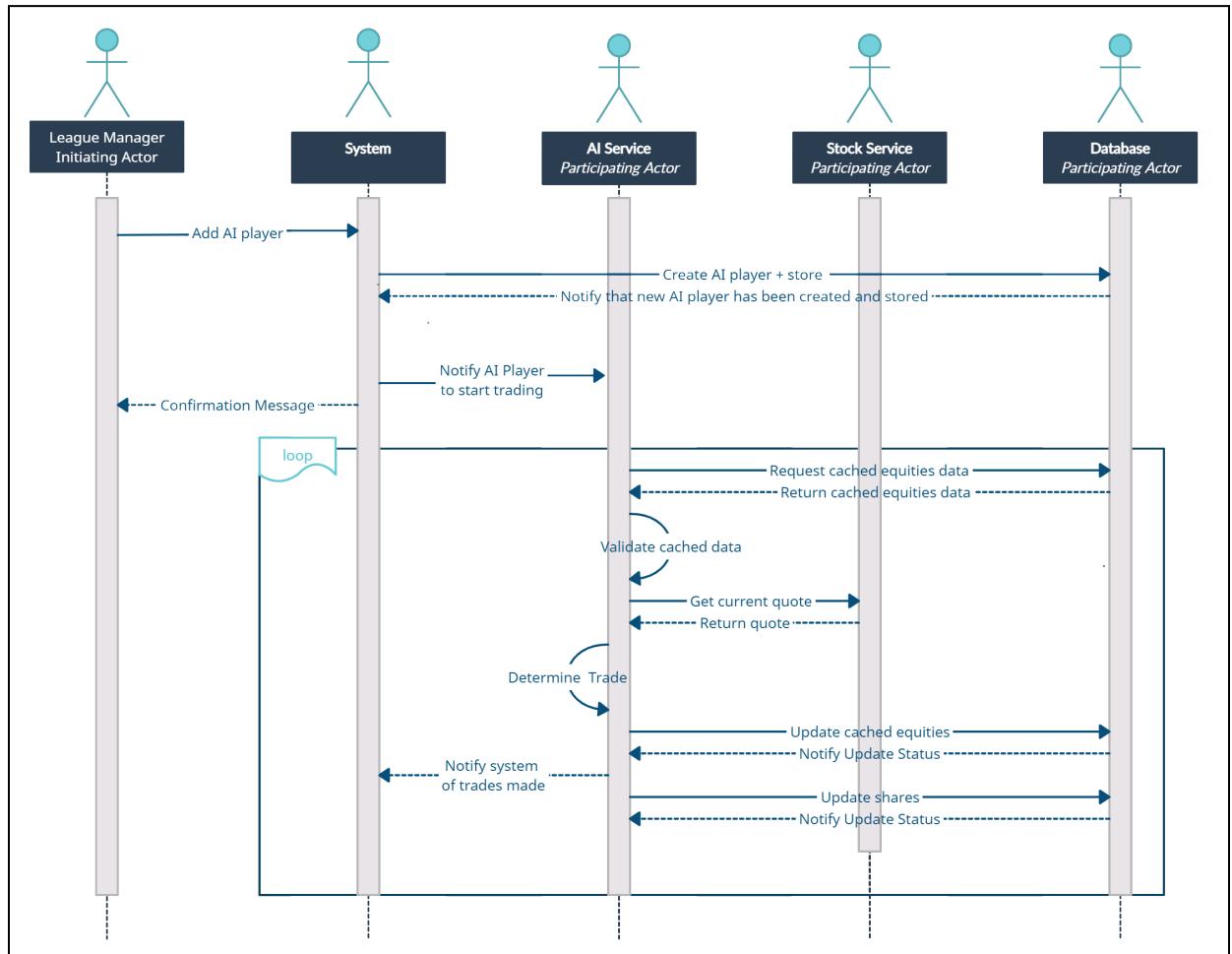
- 1) **User** enters the ticker symbol for their desired equity.
- 2) **System** requests **Stock Service** to validate the ticker symbol query, as well as display relevant information regarding the desired equity.
- 3) **User** selects whether they would like to buy or sell, their desired type of transaction, and the duration they would like the order to stay in effect.
- 4) **User** places their order.
- 5) **System** fulfills the order based on the most recent quote available from the stock service at any given time while the order is active.
- 6) **System** displays confirmation message to **User**.

Flow of Events for extensions (Alternate Scenarios or Contingencies):

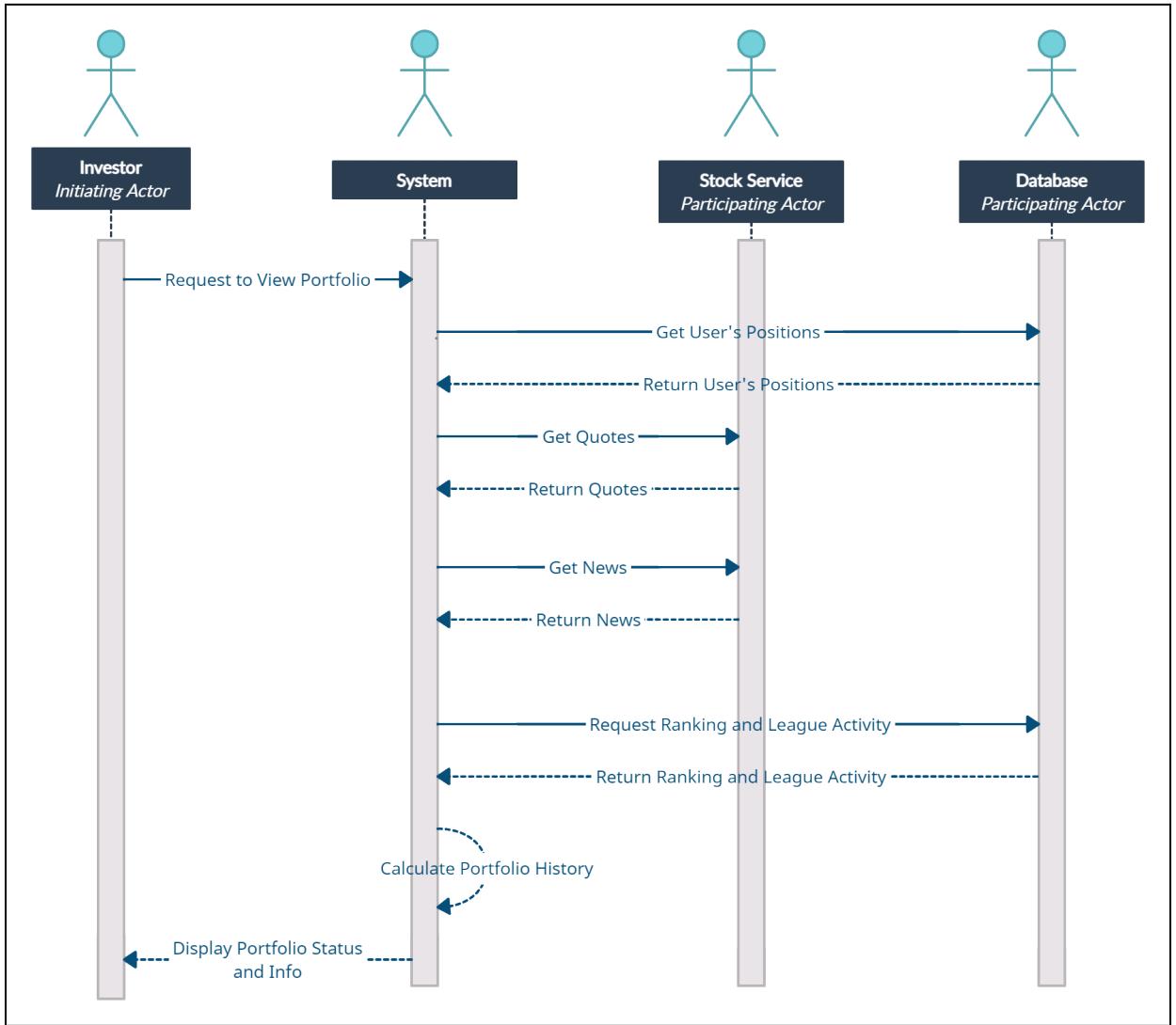
1. **User** does not have enough capital to purchase shares of desired equity or sufficient shares to sell shares of desired equity.
 - a. **System** validates request against **Database** data and determines the request is invalid.
 - b. **System** alerts **User** of trade request rejection.
2. **User** enters an invalid ticker symbol.
 - a. **System** receives invalid notice from **Stock Service**
 - b. **System** alerts **User** of the invalid ticker.

5.4 System Sequence Diagrams

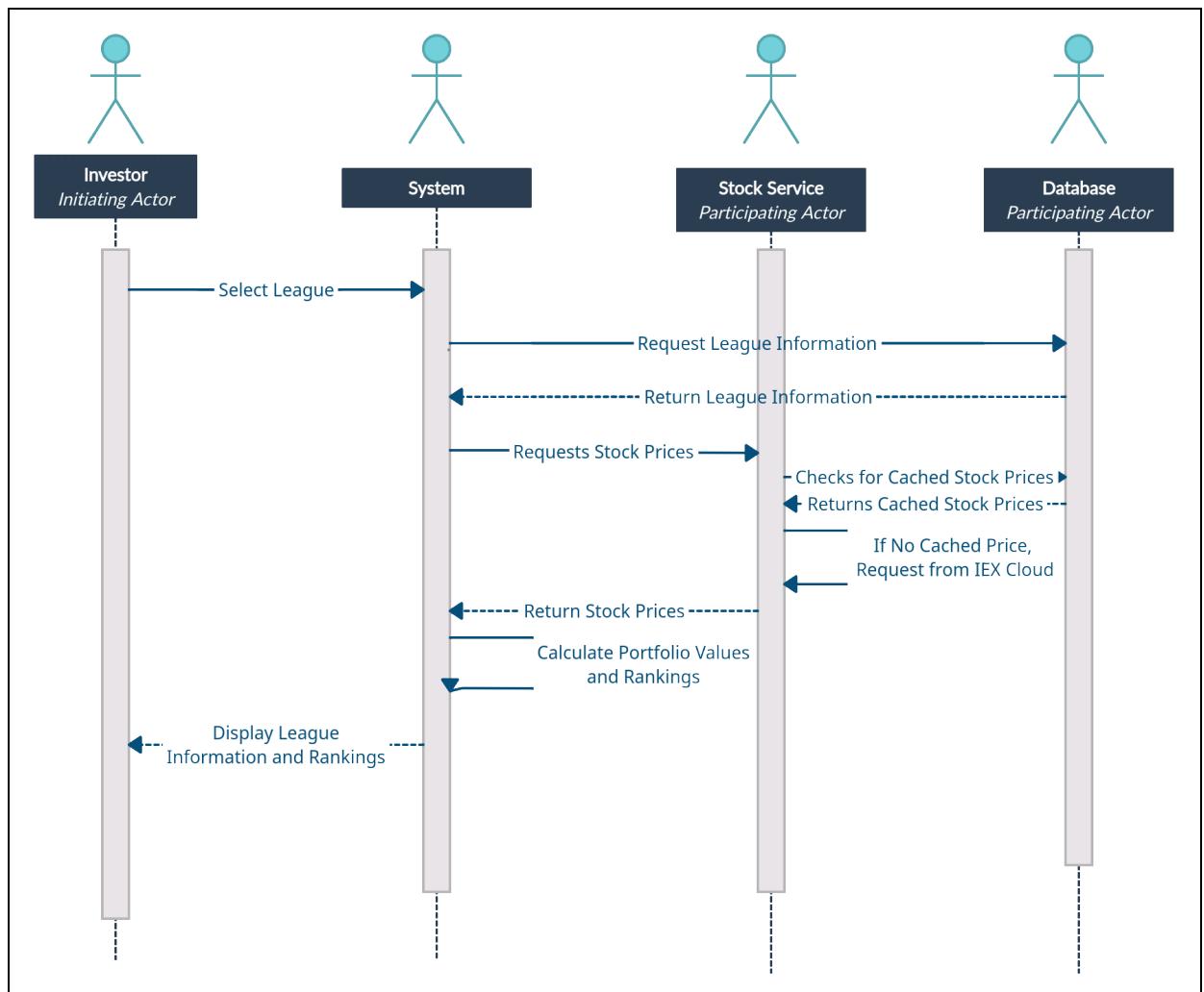
5.4.1 Use Case UC - 4: Add AI Players



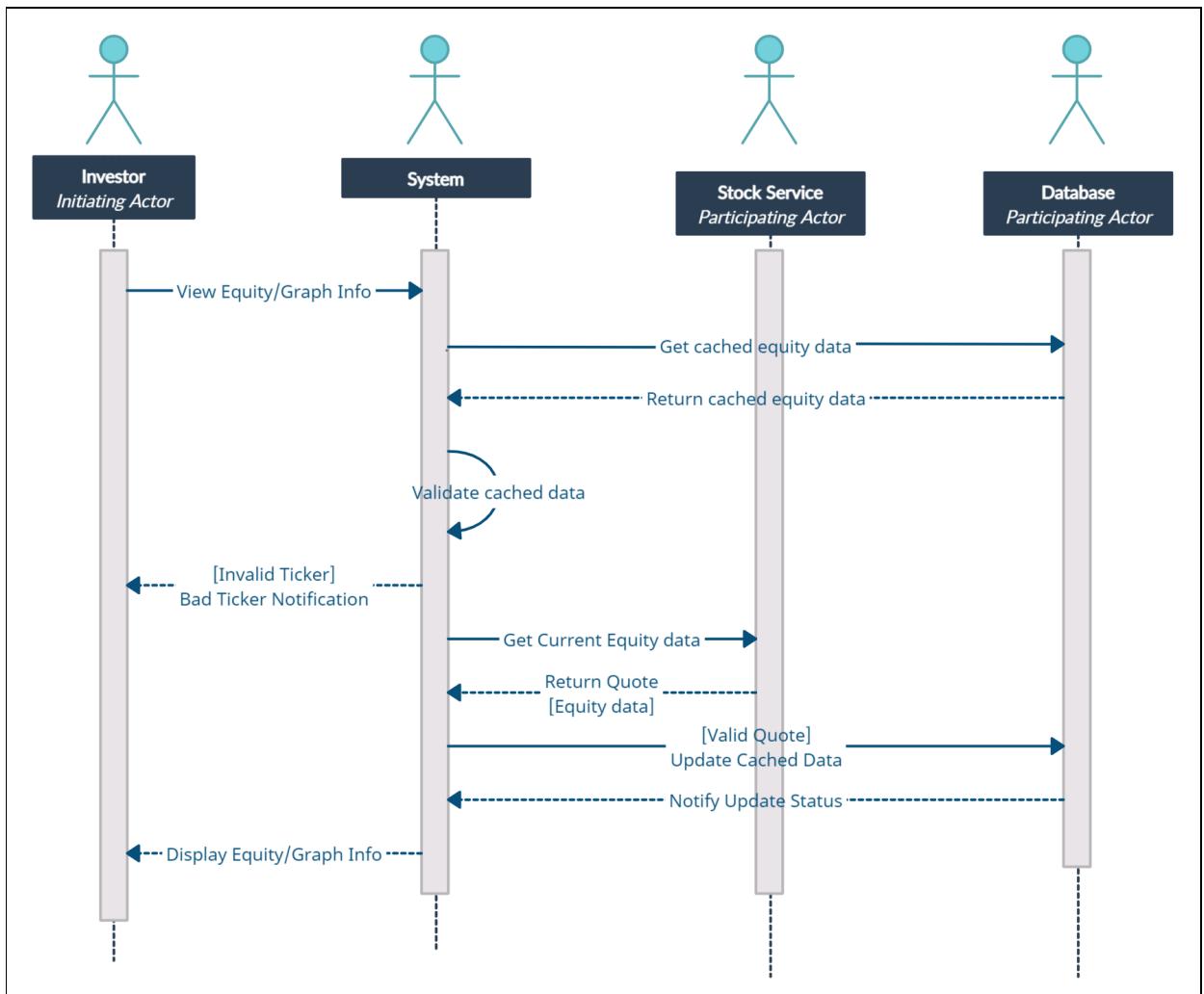
5.4.2 Use Case UC - 5: View Portfolio Status & Information



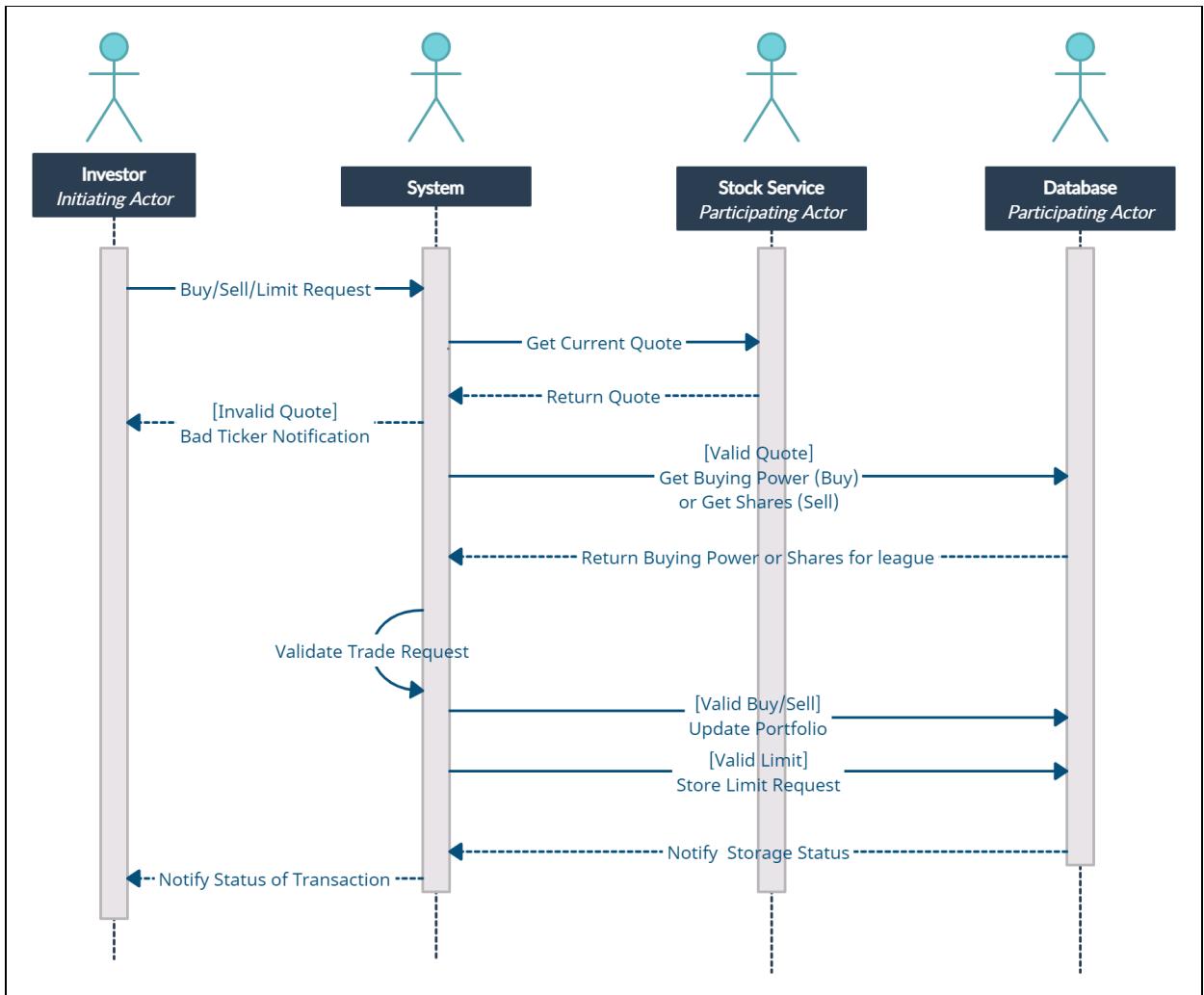
5.4.3 Use Case UC - 7: View League & Rankings



5.4.4 Use Case UC - 9: View Equity Information



5.4.5 Use Case UC - 10: Make Trades



6 User Interface Specifications

6.1 Preliminary Design

6.1.1 Create AI Bot

To create an AI bot, the league manager must go to the create league page using the navigation system located on the left of the website and click on create a new league. While creating a new league the manager will be able to add the desired number of AI bots. In order to do this, they must place a number in the corresponding textbox next to # of AI Player bots before clicking Create.

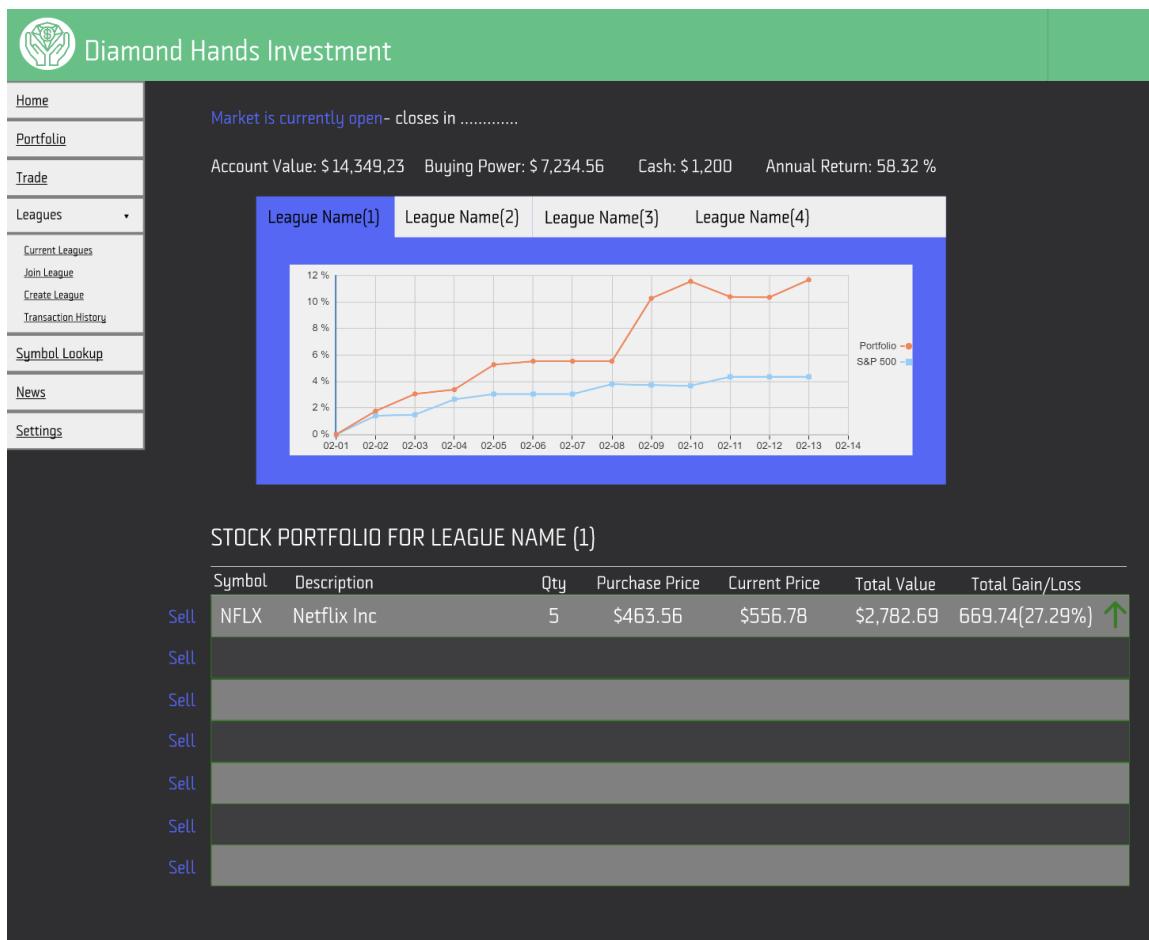
The screenshot shows a dark-themed web application interface. At the top, there is a green header bar with the logo 'Diamond Hands Investment'. On the left side, a vertical navigation menu is visible, featuring links like Home, Portfolio, Trade, Leagues (with sub-links for Current Leagues, Join League, Create League, and Transaction History), Symbol Lookup, News, and Settings. The main content area has a dark background. It starts with a heading 'Please provide the following information:' followed by several input fields. Each field consists of a label on the left and a white input box on the right. The labels are: League Name:, Starting Balance:, Commission Percentage:, Trade/Day Limit:, Private/Public:, # of AI Player Bots:, Start Date:, and End Date:. Below these input fields is a large, rectangular 'Create' button with a thin white border.

6.1.2 View Portfolio Status & Information

On the portfolio page, the user will be able to view important information about their portfolio. Users can locate the portfolio page directly from the navigation bar on the left side of the page. On the portfolio page, the user will be able to see their account value, buying power, cash, and daily return. This information will be displayed for each league depending on which league the user has clicked on. Also, the user can see their performance compared to the S&P 500 for each league that they are part of. The

individual league data can be shown by clicking on the tabs above the graph. The tabs will have all the league names that the user is participating in and the user can switch between tabs to see all the information pertaining to that individual league.

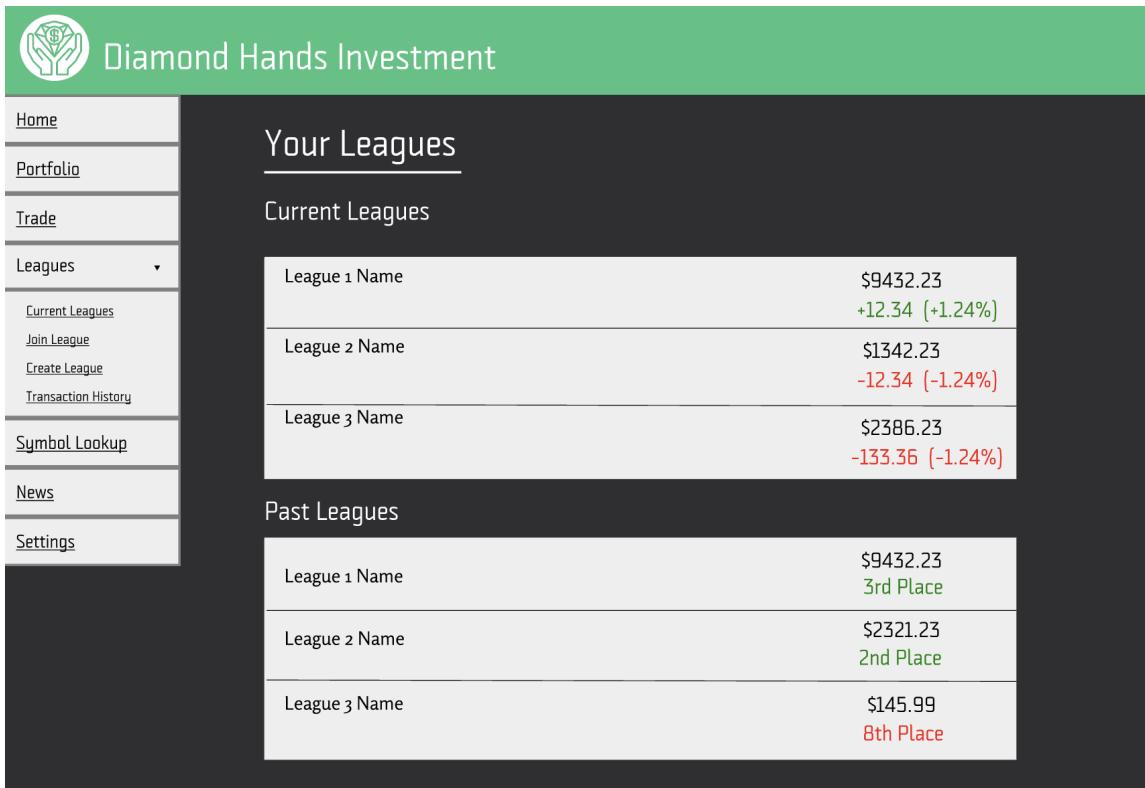
Underneath the graphs, there is also a stock portfolio for each league the user is in. The stock portfolio shows what stocks the user bought in that specific league. The table will show the symbol, a description, the amount of stocks the user bought, the purchase price, current price, total value, and the total gain/loss. The user will also be able to sell their stocks immediately with the “Sell” button right next to the stock. After the user clicks the “Sell” button, the user will be prompted to go to the Trade page as explained in 6.1.5 of the report. The user will also be able to go to the Individual Stock Page by clicking on a particular stock for more detailed information.



6.1.3 View League & Rankings

To view the Individual League activity, users must first go to the navigation bar, located on the left-hand side of the website, click the drop down menu “Leagues” and click on “Current Leagues”. This will bring them to a page where they can see all their current

leagues and past leagues. On the right-hand side of each league name you can see the users progress. In order for the user to see their league's progress, they must click on the name of the league. Once they click on that league they will be redirected to a page where all information pertaining to the league will be displayed. Users will be able to see their current ranking in a ribbon in the top right corner. Under the league name will be a graph that compares every member's progress to the S&P 500 index. This graph will be followed by the league's ranking table stating the current account value and growth of each member of the league.

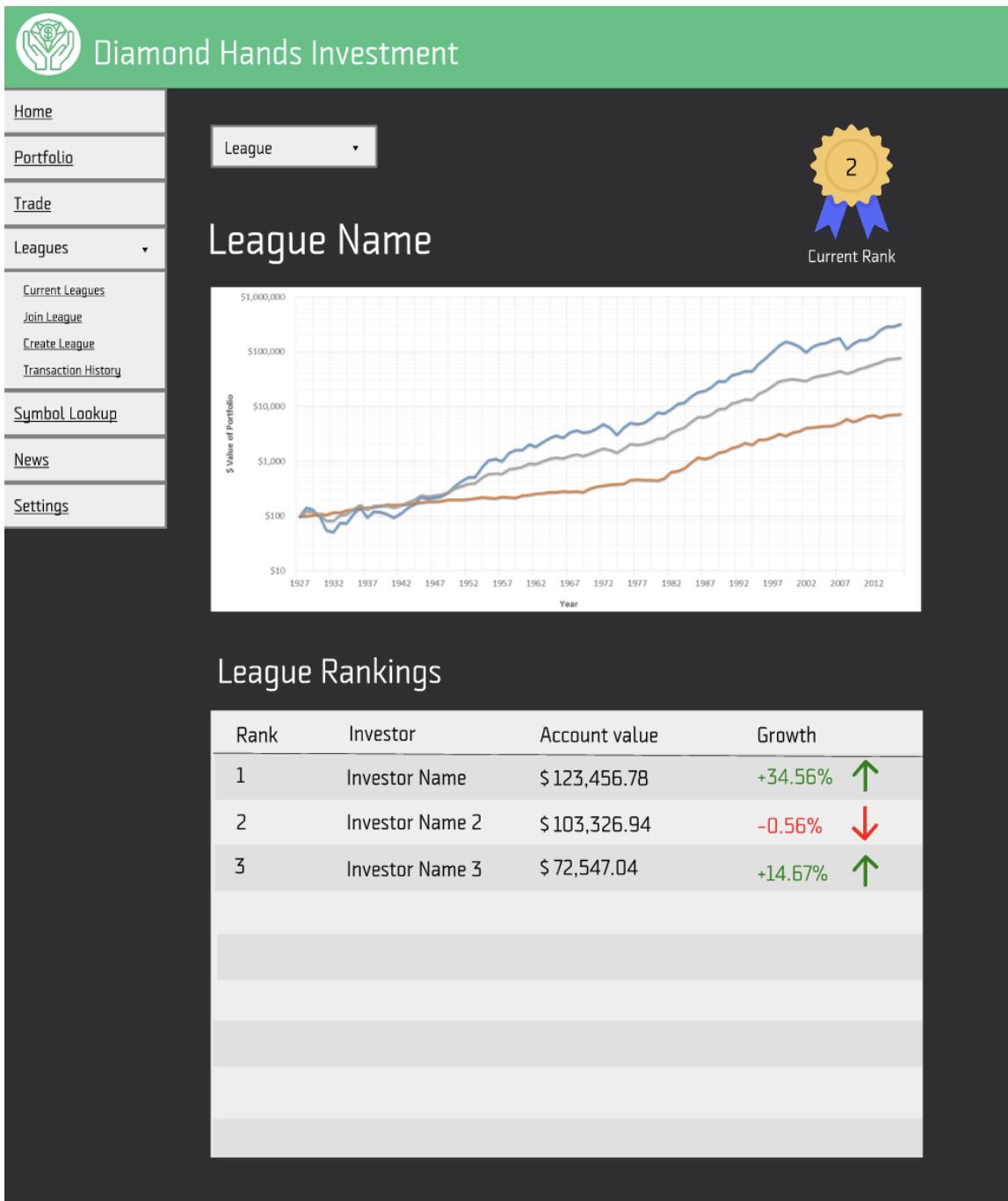


The screenshot shows the 'Your Leagues' section of the Diamond Hands Investment website. The left sidebar has a green header with a logo and navigation links: Home, Portfolio, Trade, Leagues (with dropdown options: Current Leagues, Join League, Create League, Transaction History), Symbol Lookup, News, and Settings. The main content area has a dark background with white text. It starts with a heading 'Your Leagues' and a sub-section 'Current Leagues'. Below this is a table with three rows:

League 1 Name	\$9432.23
	+12.34 [+1.24%]
League 2 Name	\$1342.23
	-12.34 [-1.24%]
League 3 Name	\$2386.23
	-133.36 [-1.24%]

Below this is a sub-section 'Past Leagues' with a table:

League 1 Name	\$9432.23
	3rd Place
League 2 Name	\$2321.23
	2nd Place
League 3 Name	\$145.99
	8th Place

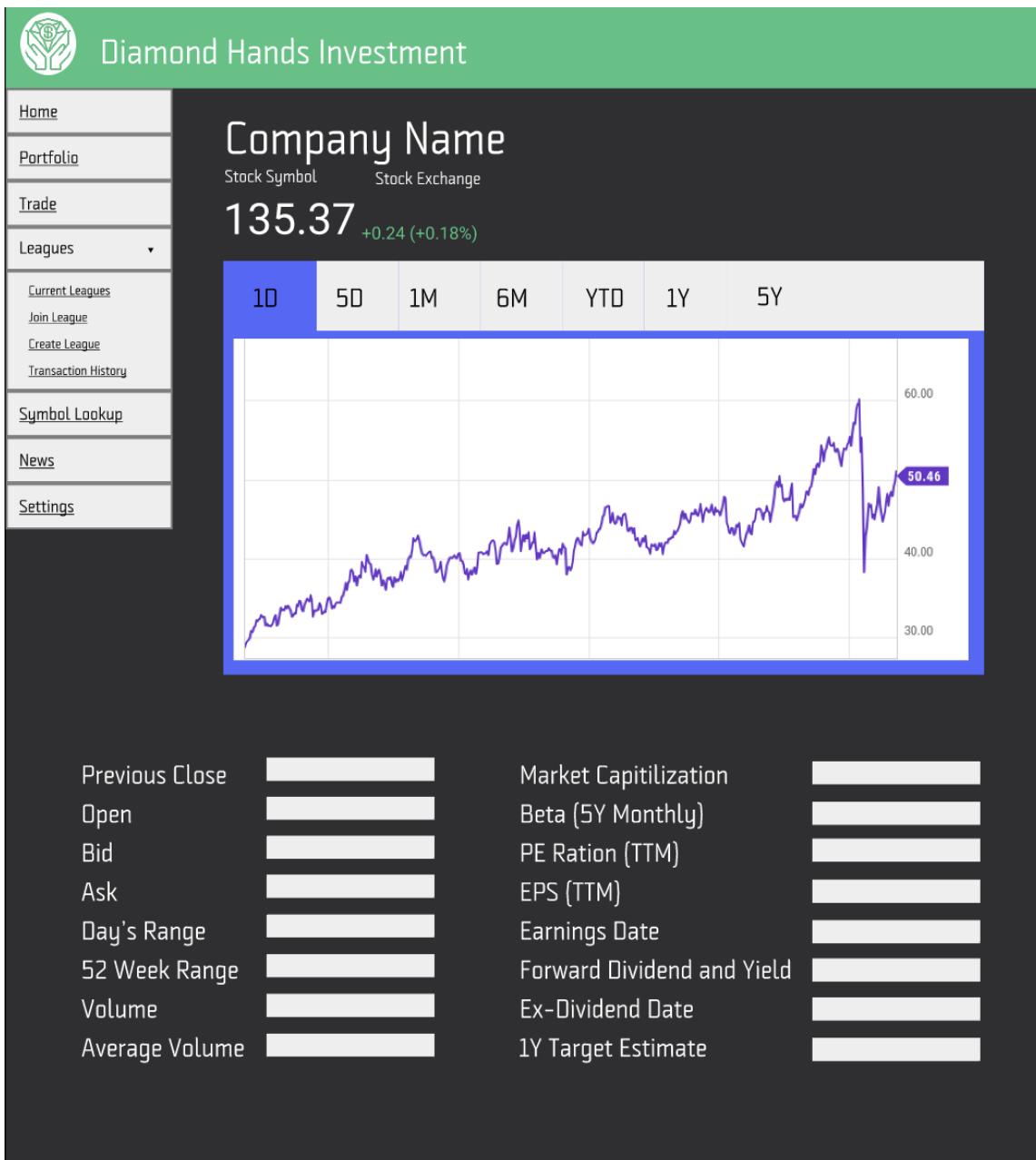


6.1.4 View Equity Information

Symbol lookup provides investors with financial information for prospective trades. This page will give users all the necessary fundamental and technical analysis for the valuation of that particular stock. It can be found on the navigation bar on the left of the page. The previous close shows the prior day's final price of the stock when the market is officially

closed for the day. Open price is the price at which a stock was first traded at the opening of the market. Bid price is the amount of money a buyer is willing to pay for a particular stock. Ask price is the lowest price a seller is willing to accept for a share of that given stock. Day's Range returns the difference between the high and low price of that market day. The 52 week range shows the lowest and highest at which that stock was traded during the previous 52 weeks. Volume shows the product of a market's volume and price over a given interval(usually one day). Average Volume is a trading benchmark used by traders to see the average volume and price of a stock in a given day. Market Capitalization refers to the total dollar market value of a company's outstanding shares. Beta is a measure of volatility and risk for a particular company. PE ratio measures the share price relative to the earnings per share. EPS is the monetary value of a company's earnings per outstanding share. Earnings date is the date of the next release of a company's profitability report. Forward Dividend and Yield is the percentage of a company's current stock price that is expected to pay out as dividends over a period of time. Ex-Dividend Date is the date dividends are paid for most public and private corporations. One year Target Estimate is an estimate of stock price after a period of one year.

The screenshot shows the Diamond Hands Investment website. The header is green with the logo and the text "Diamond Hands Investment". The left sidebar has a dark background with white text links: Home, Portfolio, Trade, Leagues (with a dropdown menu for Current Leagues, Join League, Create League, and Transaction History), Symbol Lookup, News, and Settings. The main content area has a dark background with a light blue header "SYMBOL LOOKUP". It features a search bar with the word "SEARCH" and a magnifying glass icon.



6.1.5 Make Trades

To access the “Sell/Buy” page use the navigation bar on the left hand side and click on “Trade”. This will bring the user to the “Sell/ Buy page” which allows investors to trade equities during market hours. The League icon will allow investors to specify for which league the trade is being made. There are two types of transactions that can be made, buy or sell. Quantity allows users to specify the amount of shares they want to purchase from each company. Price allows investors to buy stocks at market price.

Investors also have the option to perform limit and stop stock orders. Limit order allows investors to buy or sell shares at a specific price. A buy limit order can only be executed at the limit price or lower. A sell limit order can only be executed at the limit price or higher. A stop order allows investors to buy or sell stock at market prices once the stock has traded at a specific stop price. In other words, if the stock reaches the stop price, the order is filled at the next available stop price.

Stocks can be bought and sold in different durations. Our fantasy league allows its investors to choose the duration of their trade. Good till cancelled is an order that works regardless of a time frame until the order is explicitly cancelled. Day order is an order that is only good for the remainder of that trading day in which they are placed.

Account Details are added in this page to show users their account value, buying power and cash of the particular league they are placing the order for. Account value is the total dollar value of all the holdings of the account in a specific league. Buying power is the total amount available to the investor to purchase their equities. Cash values refer to the total amount of liquid funds in the account. This is money that is potentially available to withdraw.

After the user clicks on “Preview Order”, the user will be redirected to the Preview Order page where the user will be able to see a summary of their order. On that page, they will be able to cancel, submit, or change the order.

The screenshot shows the Diamond Hands Investment website interface. The top navigation bar is green with the logo and the text "Diamond Hands Investment". The left sidebar has a dark background with white text, listing "Home", "Portfolio", "Trade", "Leagues" (with "Current Leagues", "Join League", "Create League", and "Transaction History" sub-options), "Symbol Lookup", "News", and "Settings". The main content area has a dark background with white text. It displays a message "Market is currently open - Closes ...". Below this is the "ORDER STOCK" section with fields for "League" (dropdown menu), "Stock Symbol" (dropdown menu), "Transaction" (dropdown menu set to "Buy"), "Quantity" (text input field), "Price" (radio buttons for "Market", "Limit \$", and "Stop \$", with a dropdown menu for "Duration" set to "Good Till Cancelled"), and "Day order" (checkbox). To the right of these fields is the "ACCOUNT DETAILS" section with fields for "Value (USD)", "Buying Power", and "Cash". A modal window titled "Stock Name" is open, listing "Last", "Change", "%Change", "Volume", "Day's High", and "Day's Low". At the bottom of the form is a "Preview Order" button.

The screenshot shows the Diamond Hands Investment application's user interface. On the left is a navigation sidebar with links: Home, Portfolio, Trade, Leagues (with sub-links: Current Leagues, Join League, Create League, Transaction History), Symbol Lookup, News, and Settings. The main content area has a green header bar with the text "Market is Currently Open - Closes in". Below this is a "PREVIEW ORDER" section. It contains fields for Description: Stock Name, Transaction: Stock: Buy at Limit, Stop/Limit, Price:, Quantity:, Comission:, Est. Total:, Duration(Term): Good Till Cancelled, and three buttons: Cancel, Submit Order, and Change order. To the right of the preview order is an "ACCOUNT DETAILS" section with fields for Value (USD):, Buying Power:, and Cash:. Below these fields is a dropdown menu with options: Stock Name, Last, Change, %Change, Volume, Day's High, and Day's Low. At the bottom of the main content area is a "Go Back" link.

6.2 User Effort Estimation

6.2.1 Create AI Bot

Consider the following example for creating a league:

- 1) **Navigation** (Starting at Home Page) - 3 clicks total:
 - a) Click “Leagues” to access drop down menu on navigation bar
 - b) Click “Create League”

--after completing data entry as shown below--

 - c) Click “Create” to finish

- 2) **Data Entry** (10 mouse clicks, 23 keystrokes):
 - a) Click textbox next to “League Name: ”
 - b) Press keys “W”, “i”, “n”, “n”, “e”, “r”, “s” for the league name
 - c) Click textbox next to “Starting Balance:”
 - d) Press keys “1”, “5”, “0”, “0”, “0” for starting balance
 - e) Click textbox next to “Commission Percentage:”
 - f) Press keys “5” for commission percentage
 - g) Click textbox next to “Trade/Day Limit:”

- h) Press “1”, “5” for trade limit
- i) Click textbox next to “Private/Public:”
- j) Press keys “P”, “r”, “i”, “v”, “a”, “t”, “e” to choose private league
- k) Click textbox next to “# of AI Player Bots:”
- l) Press keys “2” to choose 2 AI Player Bots
- m) Click textbox next to “Start Date:”
- n) Click to choose a date to start
- o) Click textbox next to “End Date:”
- p) Click to choose a date to end

6.2.2 View Portfolio Status & Information

- 1) **Navigation** (Starting at Home Page) - 2 mouse clicks:
 - 2) Click “Portfolio” in navigation bar
 - 3) Click on tab with corresponding “League Name” to view information on that particular league

6.2.3 View League & Ranking

- 1) **Navigation** (Starting at Home Page) - 3 mouse clicks
 - a) Click on “Leagues” in the navigation bar
 - b) Click on “Current Leagues” in the dropdown menu
 - c) Click on the desired league name

6.2.4 View Equity Information

Consider the following example: User wants to search for information about Netflix’s stock

- 1) **Navigation** (Starting at Home Page) - 2 mouse clicks
 - a) Click on “Symbol Lookup” in the navigation bar
---after completing data entry as shown below---
 - b) Click the search button (magnifying glass icon)
- 2) **Data Entry** (1 mouse click + 8 keystrokes)
 - a) Click search bar

- b) Press keys “N”, “e”, “t”, “f”, “l”, “i”, “x”

6.2.5 Sell/Buy Page

Example 1: Consider the following example: User wants to buy 5 Tesla stocks at market price and is good till cancelled and they submit the order.

1) **Navigation** (Starting at Home Page) - 9 mouse clicks:

- a) Click “Trade” in the navigation bar
- b) Click specific “League” in dropdown menu
- c) Click “Stock symbol” of that equity in the drop down menu
 - after completing data entry as shown below---**
- d) Click “Buy” in the transaction drop down menu
- e) Click “Quantity”
 - after completing data entry as shown below---**
- f) Click option for market
- g) Click “Good Till Cancelled” in the duration drop down menu
- h) Click “Preview order”
- i) Click “Submit order”

2) **Data Entry** (5 keystrokes):

- a) Press keys “T”, “S”, “L”, “A”
- b) Press keys “5”

Example 2: Consider the following example: User wants to buy 5 Tesla stocks as a limit order for \$800 and is good till cancelled and they submit the order.

1) **Navigation** (9 mouse clicks) :

- a) Click “Trade” in the navigation bar
- b) Click specific “League” in dropdown menu
- c) Click “Stock symbol” of that equity in the drop down menu
 - after completing data entry element a as shown below---**
- d) Click “Buy” in the transaction drop down menu
- e) Click “Quantity”

---after completing data entry element b as shown below---

- f) Click option for “Limit” order

---after completing data entry element c as shown below---

- g) Click “Good Till Cancelled” in the duration drop down menu
- h) Click “Preview order”
- i) Click “Submit order”

2) **Data Entry (8 keystrokes):**

- a) Press keys “T”, “S”, “L”, “A”
- b) Press keys “5”
- c) Press keys “8”, “0”, “0” for the limit order price

Example 3: Consider the following example: User wants to buy 5 Tesla stocks as a stop order for \$800 and is good till cancelled and they submit the order.

1) **Navigation (9 mouse clicks)**

- a) Click “Trade” in the navigation bar
- b) Click specific “League” in dropdown menu
- c) Click “Stock symbol” of that equity in the drop down menu

---after completing data entry element a as shown below---

- d) Click “Buy” in the transaction drop down menu
- e) Click “Quantity”

---after completing data entry element b as shown below---

- f) Click option for “Stop” order

---after completing data entry element c as shown below---

- g) Click “Good Till Cancelled” in the duration drop down menu
- h) Click “Preview order”
- i) Click “Submit order”

2) **Data Entry (8 keystrokes):**

- a) Press keys “T”, “S”, “L”, “A”
- b) Press keys “5”
- c) Press keys “8”, “0”, “0” for the stop order price

Example 4: Consider the following example: User wants to buy 5 Tesla stocks as a market and is a day order and they submit the order.

1) **Navigation** (9 mouse clicks)

- a) Click “Trade” in the navigation bar
- b) Click specific “League” in dropdown menu
- c) Click “Stock symbol” of that equity in the drop down menu

---after completing data entry element a as shown below---

- d) Click “Buy” in the transaction drop down menu
- e) Click “Quantity”

---after completing data entry element b as shown below---

- f) Click option for “market” to order
- g) Click “Day Order” in the duration drop down menu
- h) Click “Preview order”
- i) Click “Submit order”

2) **Data Entry** (5 keystrokes):

- a) Press keys “T”, “S”, “L”, “A”
- b) Press keys “5”

Example 5: Consider the following example: User wants to buy 5 Tesla stocks as a market and is a day order and they cancel the order.

1) **Navigation** (9 mouse clicks):

- a) Click “Trade” in the navigation bar
- b) Click specific “League” in dropdown menu
- c) Click “Stock symbol” of that equity in the drop down menu

---after completing data entry element a as shown below---

- d) Click “Buy” in the transaction drop down menu
- e) Click “Quantity”

---after completing data entry element b as shown below---

- f) Click option for “market” to order
- g) Click “Day Order” in the duration drop down menu

- h) Click “Preview order”
- i) Click cancel order”

2) Data Entry (5 keystrokes):

- a) Press keys “T”, “S”, “L”, “A”
- b) Press keys “5”

Example 6: Consider the following example: User wants to buy 5 Tesla stocks at market price and is a day order and they change the order to duration as good till cancelled.

1) Navigation (12 mouse clicks)

- a) Click “Trade” in the navigation bar
 - b) Click specific “League” in dropdown menu
 - c) Click “Stock symbol” of that equity in the drop down menu
- after completing data entry element a as shown below---**
- d) Click “Buy” in the transaction drop down menu
 - e) Click “Quantity”

---after completing data entry element b as shown below---

- f) Click option for “market” to order
- g) Click “Day Order” in the duration drop down menu
- h) Click “Preview order”
- i) Click “Change order”
- j) Click “Good Till Cancelled”
- k) Click “Preview order”
- l) Click “Submit order”

2) Data Entry (5 keystrokes):

- a) Press keys “T”, “S”, “L”, “A”
- b) Press keys “5”

Example 7: Consider the following example: User wants to sell 5 Tesla stocks

1) Navigation (7 mouse clicks)

- a) Click “Trade” in the navigation bar
- b) Click specific “League” in dropdown menu
- c) Click “Stock symbol” of that equity in the drop down menu

---after completing data entry element a as shown below---

- d) Click “Sell” in the transaction drop down menu
- e) Click “Quantity”

---after completing data entry element b as shown below---

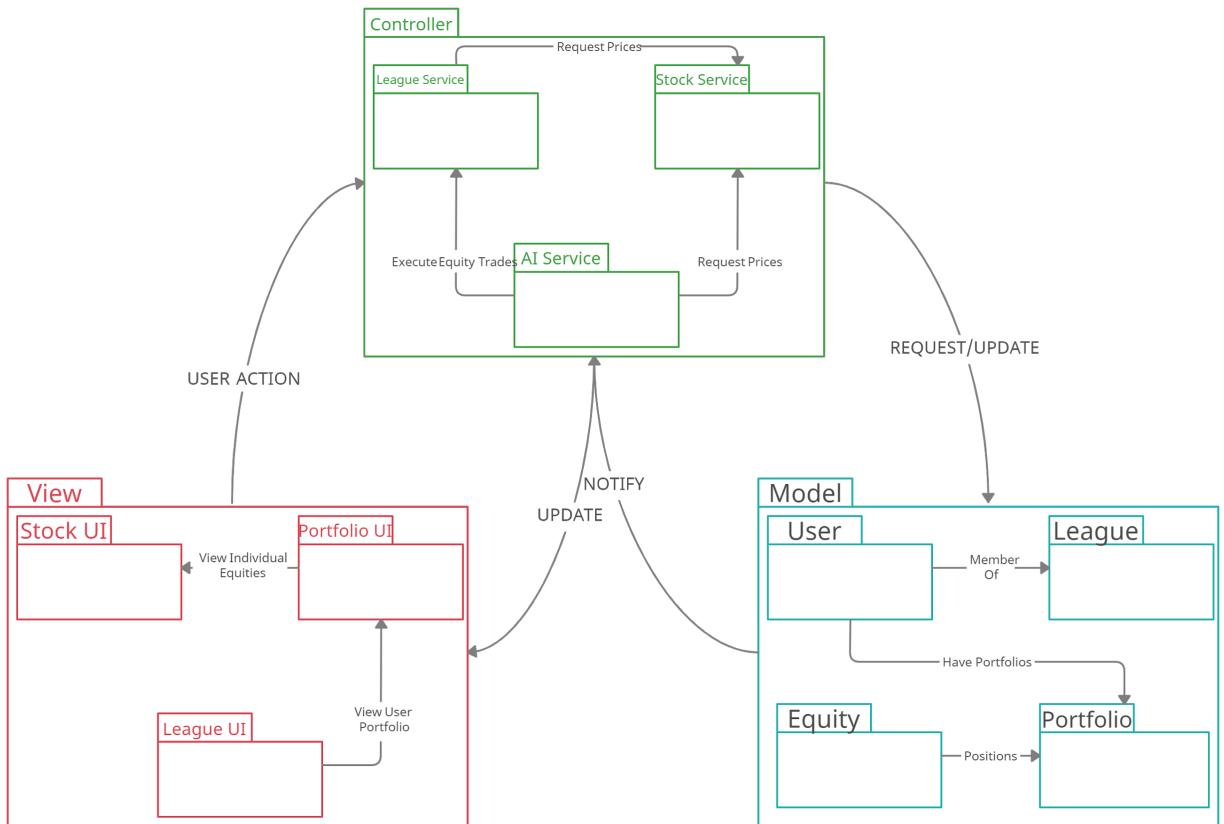
- f) Click “Preview order”
- g) Click “Submit order”

2) Data Entry (5 keystrokes):

- a) Press keys “T”, “S”, “L”, “A”
- b) Press key “5”

7 System Architecture

7.1 Identifying Subsystems



Our system is composed of three subsystems that follow the MVC model and is defined in greater detail in *7.2 Architecture Styles*:

The controller subsystem is comprised of the league service, the stock service, and the AI service. These three services handle all of the business logic within the application. The league service requests for prices and news from the stock service in order to handle trading. The AI service requests for prices and technical indicators from the stock service and utilizes the league service to execute orders.

The view subsystem consists of the stock UI, the portfolio UI, and the league UI. This subsystem provides users with methods that allow them to interact with the system. The league UI is the root, from where users can access their individual league portfolio through the portfolio UI. Users are able to view individual equities through the stock UI, which can be accessed from the portfolio UI.

The final subsystem is the model subsystem, which consists of users, leagues, and equities. This subsystem defines the types of data that we are storing. League models contain information on users participating in the league and their desired settings. Equity models contain historical price data and other relevant financial statistics such as technical indicators. User models keep track of account information and holdings in each league.

Utilizing this MVC architecture along with microservices results in a modular design that allows us to develop components independently of one another.

7.2 Architectural Styles

For our application, we are making use of a Model-View-Controller (MVC) architecture model as outlined above, which has become a popular design pattern for developing web applications. The view component represents the front end of the application and handles all of the application's UI logic. It presents to the user all the data received from the controller. The view component is responsible for displaying all the graphs, statistics, and metrics to the user. The model directly interacts with data from the database and handles all the business logic of the application and responds to requests from the controller. The controller component processes business logic and receives incoming requests from the user. The controller handles user interaction and input and acts as an interface between the model and the view.

Additionally, we will be incorporating a Client-Server model, since many users will want to access our application at once. The user/player will be the main client who interacts with our service, and the server will provide and store all the financial data that the users need upon request. This promotes seamless interaction between the client and infrastructure.

Lastly, we will be using a microservice architecture. Our plan is to create separate services for league logic and financial logic i.e. there will be a league service and a stock service. This allows us to develop these components of the application separately and independently of each other.

7.3 Mapping Subsystems to Hardware

The “View” subsystem gets mapped directly to the user’s device. Our client will be intended for users on either smartphones or PC. Users will be able to use any device with a web browser to access our web application, but only resolutions that are commonly

found on mobile or desktop computers will be supported, as mentioned later in Hardware Requirements (7.6). The user interface will be presented through web pages that are managed with React.

The “Controller” and “Model” subsystems are run by a server. This server will be run on a remote cloud-based machine and will consist of multiple microservices that will communicate with each other through REST APIs. Each microservice (Controllers) will be built in Node.js and will utilize Express for creating routes and building endpoints. The Controllers can also utilize said routes to communicate with MongoDB (Model) for persistent data storage.

7.4 Connectors and Network Protocols

Since the application we are building is a web application, we are using HTTP/HTTPS. This is the standard protocol for anything that is accessed through a web browser, which is why we chose to use it. Additionally, it allows us to use REST APIs to update and retrieve data. This is also another industry standard when it comes to web applications.

7.5 Global Control Flow

7.5.1 Order of Execution

Our system is almost entirely event-driven, as it mainly acts on requests made by the user to the server. Many of the features our system accommodates for are prompted by either the user or another part of the system such as the trading bot. When a user wants to place an order, view their portfolio, or search equity information, they must request the server which will take the necessary steps to facilitate the request. The only features that would be considered procedure-driven are registering and trading. Before any user can access other parts of our application, they must make an account and before they can make a trade, they must be part of a league. However, after making an account, they can trade any stock available to them at any time they want, which means that the system has to wait for the event to occur before it responds to it.

7.5.2 Time Dependency

Our system is also time dependent. The stock market itself only operates during a certain part of the day, so our system depends on this period (9:30AM – 4:00PM) to execute orders. Users can place orders before market open or even after market close, however, orders will only be fulfilled during regular trading hours. Additionally, our system

supports real time transactions, the latest stock quotes and portfolio value, limit orders and our trading bot, all of which are heavily reliant on real-time timers. The limit orders and the AI trading bot will access the stock data at a period that will be specified later. Optimally, they will access the stock data as frequently as possible. This period is constrained by the number of API calls that the stock service can support.

7.6 Hardware Requirements

Requirements for web application client:

- 5 Mbps network connection
- Minimum resolution: 1024 x 768
- Minimum requirements for a web browser compatible with ES6 Javascript

8 Project Size Estimation based on use case points

8.1 Table I

	Group 1	Group 2	Group 3	Group 4	Group 5
UUCP	20	30	21	29	30
TCF	1	1	1	0.7	1
UCP = UUCP*TCF	20	30	21	20	30
Normalized UCP	0.66	1	0.70	0.66	1

9 Project Management

Our team plans to use agile software development practices to create solutions and collaborate on our project. We decided to meet once a week for a formal meeting for about 2 hours to discuss our goals and review our project for the upcoming week. Furthermore, we also scheduled to have 20 min standup meetings every other day. The goal of these meetings is to assist any team members that have immediate concerns on their functionality. All of these meetings will be held on Discord for easy communication. To collaborate on our user interface design, our UI/UX team will be using Figma. This software will give us an opportunity to work together to work on our visual components of the project. For our architecture needs, we plan to use GitHub to track our progress and share code efficiently. The architecture team will host preliminary meetings to decide on the shared infrastructure that will be used in the project, such as any cloud services to host the project. Regarding report integration, we plan to divide the work evenly between all members of the group. This is outlined clearly in our timeline chart below. We will also review the report as a unit and ensure uniformity amongst the team. Our plan of action in ensuring logical integrity throughout the timeline of the project is to divide the work evenly between members of the group.

9.1 Groups:

1 Aarushi S, Aarushi P

2 David, Jacques, Jawad

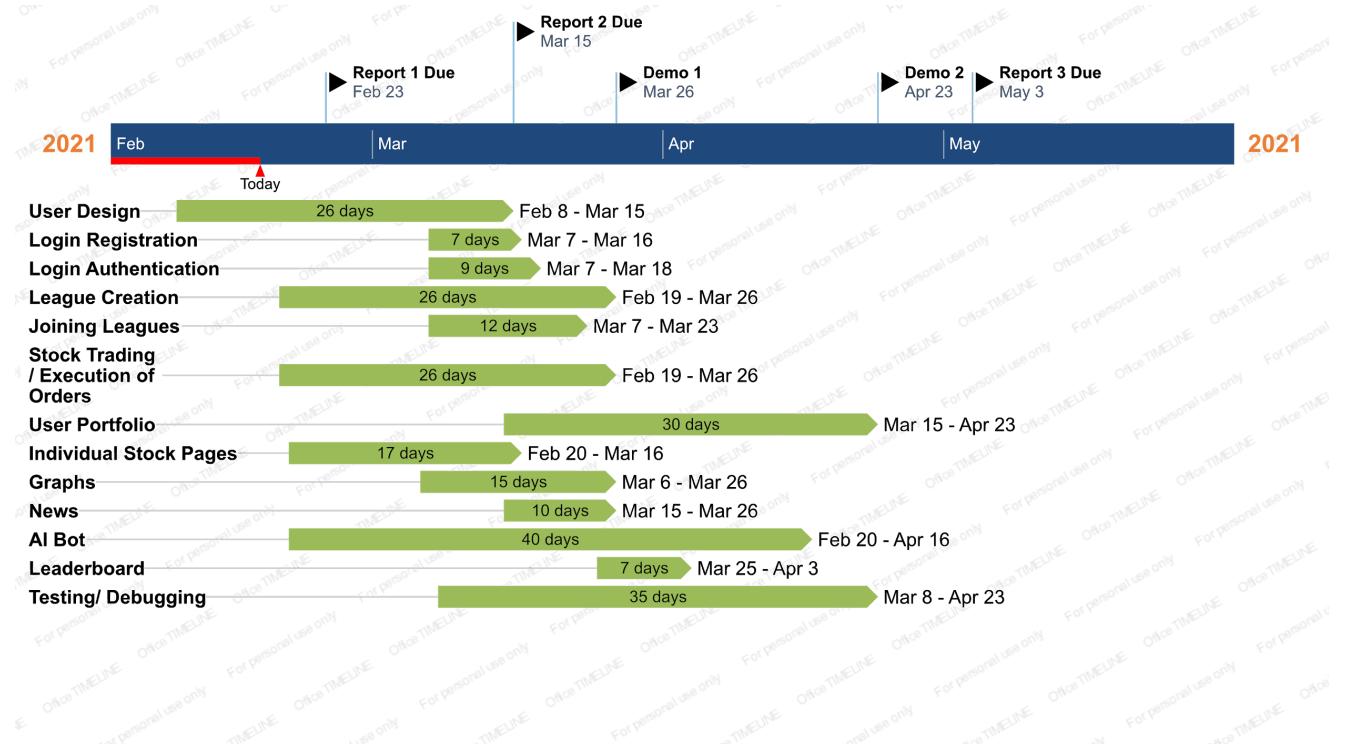
3 Riya, Sahitya

4 Apoorva, Yatri

5 Christine, Yati, Krishna

	Group 1	Group 2	Group 3	Group 4	Group 5
UC-1 (5)	✓				
UC-2 (11)				✓	
UC-3 (8)				✓	
UC-4 (15)					✓
UC-5 (25)		✓			
UC-6 (5)		✓			
UC-7 (15)	✓				
UC-8 (9)			✓		
UC-9 (12)			✓		
UC-10 (15)					✓
UC-11 (10)				✓	

9.2 Project Timeline



10 References

- [1] J. Gobell, “The Investopedia Affluent Millennial Investing Survey,” *Investopedia*, 20-Nov-2019. [Online]. Available: <https://www.investopedia.com/the-investopedia-affluent-millennials-study-4769751>. [Accessed: 09-Feb-2021]
- [2] T. Ghilarducci, “Most Americans Don't Have A Real Stake In The Stock Market,” *Forbes*, 16-Dec-2020. [Online]. Available: <https://www.forbes.com/sites/teresaghilarducci/2020/08/31/most-americans-dont-have-a-real-stake-in-the-stock-market/?sh=68fc3bcd1154>. [Accessed: 09-Feb-2021].
- [3] S. Rose, “7 Expert Perspectives on Why Financial Literacy Is Important,” *OppLoans*, 01-Sep-2020. [Online]. Available: <https://www.opploans.com/oppu/articles/why-is-financial-literacy-important/#:~:text=Financial literacy is important because, a solid foundation for success.&text=Nearly half of Americans don't enough money to retire comfortably>. [Accessed: 09-Feb-2021].
- [4] “How Much Longer Will Social Security Last?,” *AARP*, 22-Sep-2020. [Online]. Available: <https://www.aarp.org/retirement/social-security/questions-answers/how-much-longer-will-social-security-be-around.html#:~:text=En español | According to the,will be depleted by 2035>. [Accessed: 09-Feb-2021].
- [5] *Investopedia*. [Online]. Available: <https://www.investopedia.com/simulator>. [Accessed: 09-Feb-2021].