User and Functional Requirements

Report 1
Software Engineering
14:332:452

Group 18

Joseph Coleman
Murali Gunti
Matthew Jackson
Himateja Madala
Andrew Marfitsin
Akshaykumar Patil
Paul Stanik
David Wang

February 10, 2019

Contributions Breakdown

Joseph Coleman (GUI Management):

-On-Screen Appearance Requirements - 70%
-User Interface Specification - 33%
-Project Size - 100%

Murali Gunti (Product Functionality)

-User Interface Specification - 33%

Matthew Jackson (Website Operator and Advertiser Interaction)

-Problem Statement - 33%
-Glossary of Terms - 100%
-User Stories - 100%
-Nonfunctional Requirements - 50%
-On-Screen Appearance Requirements - 30%
-Functional Requirements Specification[Stakeholders, Actors and Goals, Use Cases(Use Case Diagram, Traceability Matrix, Fully-Dressed Description), System Sequence Diagrams] - 70%
-Domain Analysis - 100%
-References - 100%

Himateja Madala (Portfolio and Market Insight Tools, Achievements)

-Plan of Work - 33%

Andrew Marfitsin (Portfolio and Market Insight Tools, Achievements)

-Problem Statement - 33%
-Nonfunctional Requirements - 50%
-Functional Requirements Specification[Use Cases(Casual Description)] - 30%

-Plan of Work - 33%

Akshaykumar Patil (GUI Management)

Paul Stanik (Website Operator and Advertiser Interaction/Product Functionality)

-Problem Statement - 33%

-Plan of Work - 33%

-User Interface Specification - 33%

David Wang (Product Functionality)

Summary of Changes

- Changed the contribution breakdown to better reflect the updated team oriented setup.
- Added references.
- Added early gantt chart.

Contents

1. Customer Statement of Requirements

1.1 Problem Statement

Investing can be very intimidating for anyone starting out, especially when real money is involved. In order to prevent investment failures from discouraging a newcomer from continuing in the field, we will provide a more comfortable and fun investment platform that gets rid of any investing anxieties and bridges the gap between higher and lower level investing. A good way to introduce a novice to the world of investing is through a simulation that features true-to-life trading mechanics, but involves fantasy money. This way, the investor-to-be has the opportunity to learn from mistakes and explore market trends without putting real money on the line.

To further simplify and de-stress the experience, the user will be offered a tutorial and walkthrough of the features offered by the application, and how to use them. This will include pointers on navigating the application interface and where to find functions such as the user's own stock portfolio, the user's group or league membership, and other views. Depending on the user's comfort level and knowledge, additional, more complex functions will be (optionally) offered to make the experience as engaging to a seasoned investor as it would be to a newcomer. These features, such as the mutual fund vs. portfolio graphs, "shorting" stock, and otherwise would be offered incrementally, with the passage of time and the user's advancement in the game, and not included in the basic first tutorial so as not to confuse a novice player.

To make the experience more engaging and exciting, in-game achievements will be implemented, rewarding the user for successful investing strategies. These achievements could range in difficulty from something like "first week with a net portfolio value gain" to "first 5% profit". The user would also be shown achievements that have not yet been reached, and the requirements for them, giving the user concrete goals to work towards. Achievements could vary for groups, to give leagues of players more challenging goals to work towards together.

Users of the application will have the option to join groups and communicate with each other, to make the experience more social if the user so chooses. Additionally, "Leagues" can be formed among players. The purpose of the league is to introduce competition among players by ranking the players within the league based on a parameter such as percentage of profit. League settings/rules can also be changed by the league manager(the user that creates the league), and the league manager would have the option of adding/removing players from the league as well. Messaging will be implemented between individual users and within leagues, such that users can effectively communicate their ideas on investment strategy and observations of the market at large. Each league will have a public message board that can be seen by all the users within the league, which would allow for league updates to be made and viewed on a regular basis. A "friend" system would be put in place, such that if two users have agreed to become friends within the game, they can see each other's portfolios and coordinate future investments. This is a useful step in making the social aspect of the game a clear option to the user, accommodating users who would want the social functionality and those who would prefer a more "single player" experience.

The user would also be offered insightful, advanced tools to aid in stock buying decisions, including per-stock price charts, portfolio comparison charts, and a historic value chart for the user's portfolio. These functions would expand the user's experience with the application to encompass more of the functions available to investors in the real world, and ease the user into familiarity with them. The user could compare their own portfolio against ETFs or index funds using the mentioned comparison charts, to observe the behavior of the market at large and evaluate how their trading decisions compare. These statistics could show the user whether or not professionally managed investment products (such as mutual funds) are really worth the price of entry.

Experience with data visualizations could help investors decide whether they prefer long-term planning, riskier short term, high payoff investments, or another trading strategy. This can be accomplished through the use of bar graphs, pie charts, box-and-whisker chart, and ext. Knowing this and having a general concept of the profit margins the user can expect that their strategy could later help in the investor's choice of brokerage account or investment product. In general, these are common tools offered by brokerage accounts to their users, and a functionality that the investor should become comfortable using.

1.2 Glossary of Terms

League - Users can join different leagues that each have their own rules such as start dates, end dates, starting cash, commission values, etc. This is where users will invest.

- **Public -** Any investor can join a public league. A public league can be made by a user who will then become the league manager.
- **Private -** A private league can only be entered by a request or through a unique code. A private league can be created a user.

League Manager - A league manager has the ability to make a league and decide whether it will be public or private. League managers set the rules for the league.

Site Administrator - A site administrator monitors the whole web application to make sure it is operating properly and can remove leagues.

Achievement - Goals set that reward users for good investments.

Marketable Securities - Financial instruments that can be converted into cash.

Investment Vehicle - A product used by investors for gaining positive returns.

- **Mutual Fund -** An investment vehicle that involves multiple investors pitching in money to invest into securities.
- **Exchange-Traded Fund(ETF) -** A type of security that tracks a stock index.

Stock - A type of security that represents ownership of a company.

- **Ask Price -** Price at which a trader is willing to sell a stock.
- **Bid Price -** Price at which a trader is willing to buy a stock.
- **Bid-Ask Spread** - The amount by which the ask price exceeds the bid price. The difference between the highest price that a buyer is willing to pay for an asset and the lowest price that a seller is willing to accept to sell it.
- **Commission -** A charge due to a broker handling the purchase or sale of a security.

Order - An investor must place an order for the purchase or sale of a stock.

- **Market Order -** An order to buy or sell a security immediately at the best available price.
- **Limit Order -** An order to buy or sell a security by setting a maximum or minimum price.
- **Stop Order -** An order to buy or sell a security once the price of the stock reaches a specific price. Different from a market order or a limit order because a stop order is not active until the stock reaches that specific price. Once the stock reaches that price, the stop order changes to a market order.
- **All or None(AON) Order -** A type of order that guarantees an investor gets the entire amount of stocks they requested or none at all.
- **Good 'Til Canceled(GTC) -** A time restriction placed on an order that remain active until an investor cancels it.

Stock Symbol (Ticker) - A unique set of letters that represent a security for trading purposes.

Symbol List - A list of the market's stock symbols.

Portfolio - A grouping of financial assets.

Rights - Stockholders can buy new shares issued by a company at a predetermined price based on how many shares the stockholder already owns.

SSO - Single sign-on. A user can authenticate once and have access to all parts of the web application.

2. System Requirements

2.1 User Stories

Identifier	User Story	Weight
ST-1	As a user, I can create an account through the web application using my email.	10 pts
ST-2	As a user, I can join as many leagues as I desire.	6 pts
ST-3	As a user, I may opt to create a league and become a league manager so I may have my own personal league.	10 pts
ST-4	As a user, I can search for a company by symbol or company name.	6 pts
ST-5	As a user, I can report another user who demonstrates negative behavior.	2 pts
ST-6	As a user, I will have access to a tutorial that introduces me to the web application by explaining different mechanics and tabs of the web app.	4 pts
ST-7	As a user, I will have access to a forum where I can talk to other users about different topics.	4 pts
ST-8	As a user, I can manage my portfolio to track investments.	8 pts
ST-9	As a user, I can buy or sell stocks so that I may build a portfolio.	10 pts

ST-10	As a user, I can browse a companies profile and view the performance data over a configurable span of time so that I may determine whether or not I want to invest in them.	6 pts
ST-11	As a user, I can see trades being made by all other users in real-time so I can have a quick overview of current trends	3 pts
ST-12	As a user, I can visually track my finances via graphs and charts so I may more easily manage my portfolio	4 pts
ST-13	As a user, I can view a portfolio leaderboard so I may have a summary of my performance in comparison to other users.	2 pt
ST-14	As a user, I can add other users and see their online status.	2 pts
ST-15	As a user, I can message other users individually or in a league.	2 pts
ST-16	As a user, I can recover or change my password so I may always have access to my own account.	5 pts
ST-17	As a user, I can access my profile and settings on a dashboard on the top of every page within the site.	8 pts
ST-18	As a league manager, I can add league rules, a league name, and a league logo to personalize my league.	8 pts
ST-19	As a league manager, I may manage players within the league so I may invite players I want to join, ban players that are being abusive, and assign other league managers.	8 pts
ST-20	As a league manager, I can moderate and delete comments in the league page.	4 pts
ST-21	As a league manager, I can create league announcements.	3 pts

ST-22	As a site administrator, I can view reports of users.	2 pts
ST-23	As a site administrator, I can delete abusive/offensive comments and ban users or IP addresses.	6 pts
ST-24	As a site administrator, I may post front page news or announcements.	3 pts
ST-25	As a site administrator, I may have access to a user count, number of active leagues, total leagues, quantity of daily transactions, the most/least popular stocks, and newly created or banned users so I may have reliable site statistics.	9 pts

2.2 Nonfunctional Requirements

Functionality

Additional features for security will be enabled through the use of a SQLite database API. A SQLite database is not very resource-intensive and provides more space than Web storage. SQLite allows for select and sort actions along with transactions that affect the state of the database, and is easily implemented with the webapp infrastructure. SQLite will also be used to store hashed passwords, provide recovery options for users that have forgotten their password, and store investment transactions. Locally hashed passwords improve speed of use and simplicity of implementing an SSO (single sign-on) system for the whole application and its sub-services. Alpha Vantage's developer toolset will be implemented to obtain stock data, both historic and current. The syntax is similar to that of SQL, and scripted queries can be used to serve data per the user's input, or navigation of the UI. This way, the application will have up-to-date and real security prices to make the experience authentic. Using an HTML and CSS security checklist, the web application will follow best coding practices to ensure sound security.

Usability

The web application should be designed in a way that is welcoming to new users but can offer more complex functions for veteran users. Using the CSS framework Bootstrap, we will be able to main a high quality web application with a clean and sleek UI. Through CSS, every page will have a header and navigation bar that offers access to all other pages of the web application. Portfolio and stock price views can be implemented graphically using graph options built into Bootstrap, and ready to implement with a connection to the SQLite database and Alpha Vantage services for current and historic stock price visualizations. These views should be able to adapt per user query, relaying the request to Alpha Vantage, fetching the requested data, and displaying it. Javascript will be used to provide a higher level of responsiveness and it will be the main framework of the welcoming tutorial.

Reliability

In order to avoid inconsistency with a user's account, all transactions must be confirmed before being registered on the server. This is done so that in the case of internet or server failure, no changes to the account are made until after this confirmation, and inconsistent data is not retained by the server. As a result, the user's portfolio will remain in a consistent state and will be restored when the user is able to log back in. A user that leaves the application without logging out and returns later will remain logged in. The effects of server failure should also be mitigated by keeping backups of user data. In case of an incomplete set of database transactions or an error state, the last valid backup should be reverted to, in avoidance of completely stopping the user's experience with the application. It should also be noted to a user when a particular stock source is not available.

Performance

Using CSS and Javascript, the aim is to keep the web application running smoothly while keeping hardware demands to a minimum on both the client and server sides. The web application should be able to complete any task initiated by the user in a timely manner without many errors or stalls. The web server should be able to serve concurrent requests to handle when a large number of users are logged in. The use of SQLite reduces the use of computing resources and will offer a faster operating experience. SQLite only offers the core functions of SQL, thereby reducing its footprint and improving its responsiveness. Additionally, locally hashing passwords will remove the need to contact external SSO services, and thus reduce latency and increase the speed of authentication. To most simply and efficiently implement communication within the application between its users, ad-hoc server based messaging will be leveraged. This choice is made due to the small resource impact of the method and the small data size it will use, which will reduce network resource use and improve the speed of communication.

Supportability

The web application will be made with desktop users as the focus. Despite this, it will still be possible for users to login to the app and use it on their phones using a mobile web browser. This will be achieved by using dynamic webpage practices, which allow the on-screen interface to adapt to any screen size it is viewed on and retain usability. This would offer the greatest possible user reach and convenience in use of the application. Locally hashed passwords allow greater flexibility in implementing new security standards, such as a required password change or update, requirements towards the password length or complexity, and so on. Without the need to contact an external SSO provider, authentication can be easily maintained and updated. The web application should be coded in a way that makes it easy to extend or update any server components and push improved versions of modules which can be installed by administrators. Preparation will be made to include an additional number of servers to achieve load balancing. The system should be backed up to a remote server for version experimentation and a rollback in the case that a new version creates a glitch.

2.3 On-Screen Appearance Requirements

Identifier	Requirement
OSR-1	The login page will display the web application's logo and present the user with a login/registration page.
OSR-2	Every page except for the login page will have a scrolling ticker at the bottom to update the users on stock prices.
OSR-3	The webapp will have text in the top right corner that shows the user's username and options to check their account settings, leagues they are a part of, and a search bar for leagues, stocks, companies, and currencies.
OSR-4	Each user will have an option to report them next to comments they leave on certain things.
OSR-5	New users will have a popup asking them if they would like to do the tutorial after their first login.
OSR-6	Leagues will display a leaderboard as well as stocks being bought and sold in that league and whether the league is private/public.

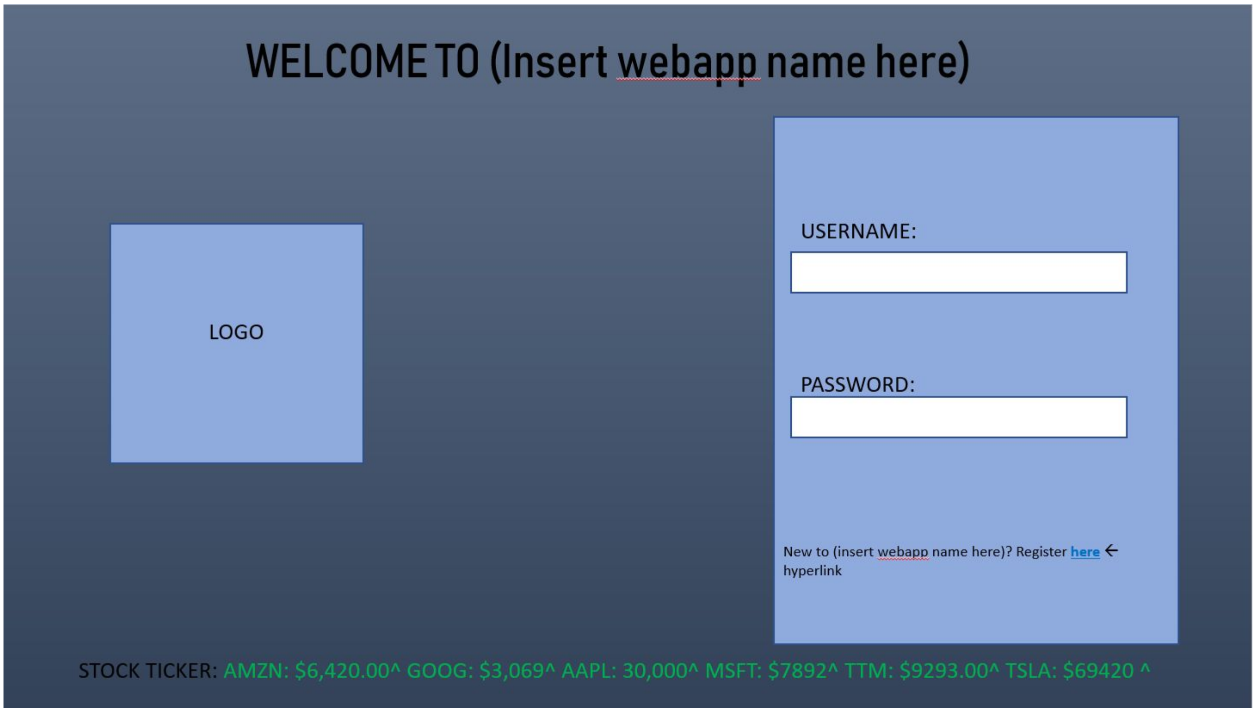

WELCOME TO (Insert webapp name here)
LOGO
USERNAME:
PASSWORD:
New to (insert webapp name here)? Register here ← hyperlink
STOCK TICKER: AMZN: $6,420.00^ GOOG: $3,069^ AAPL: 30,000^ MSFT: $7892^ TTM: $9293.00^ TSLA: $69420 ^

Figure 2.1: A mockup for a login screen

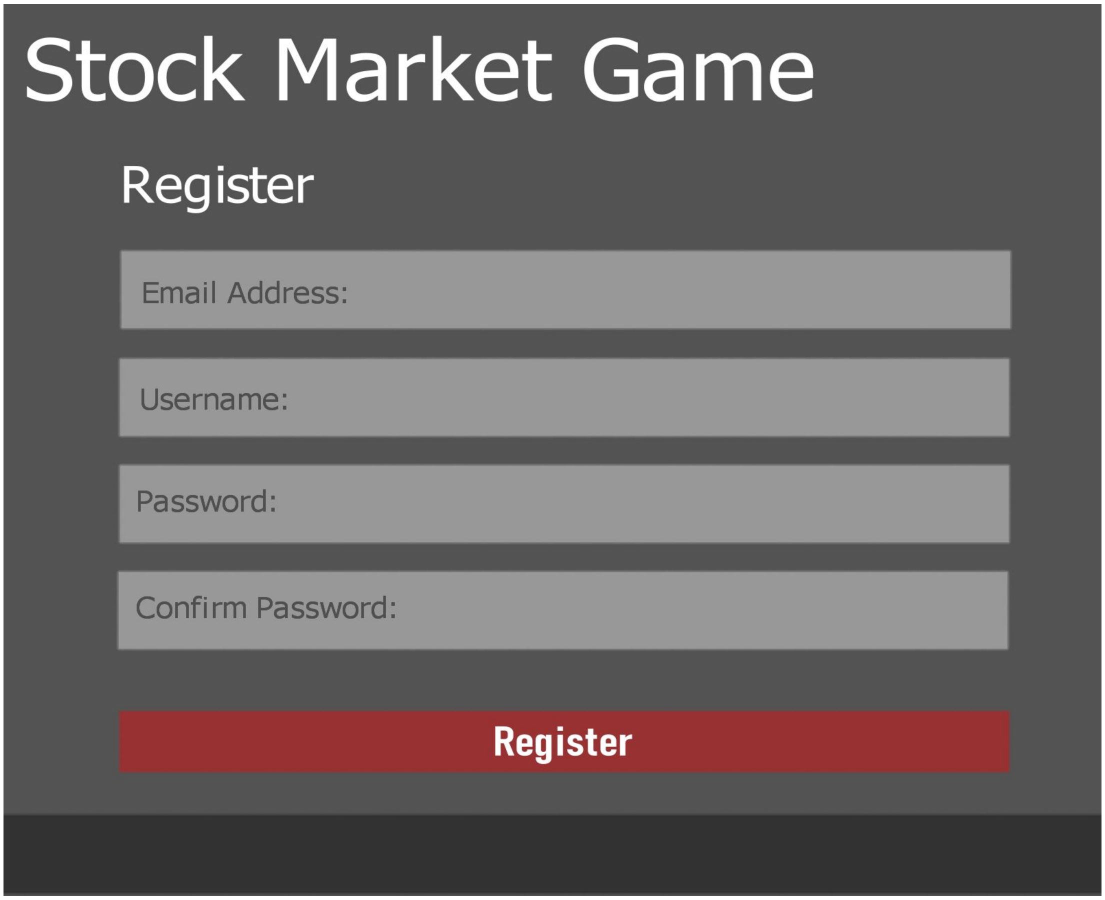
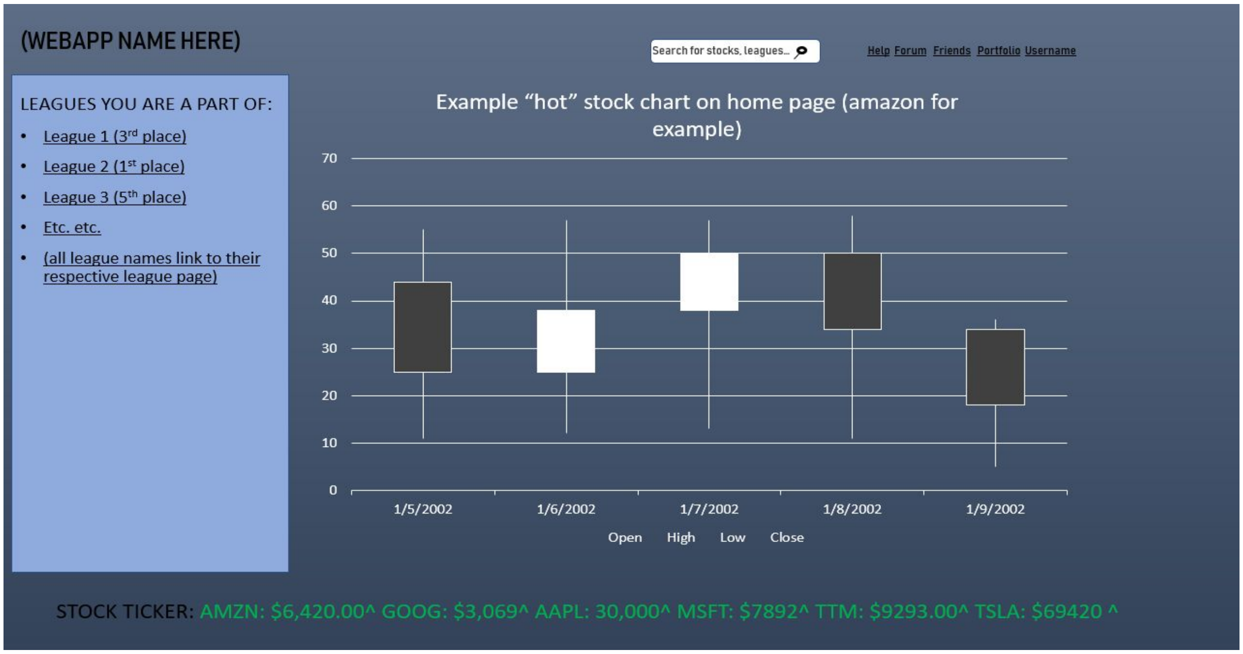


Figure 2.2: A mockup of the registration page



(WEBAPP NAME HERE)

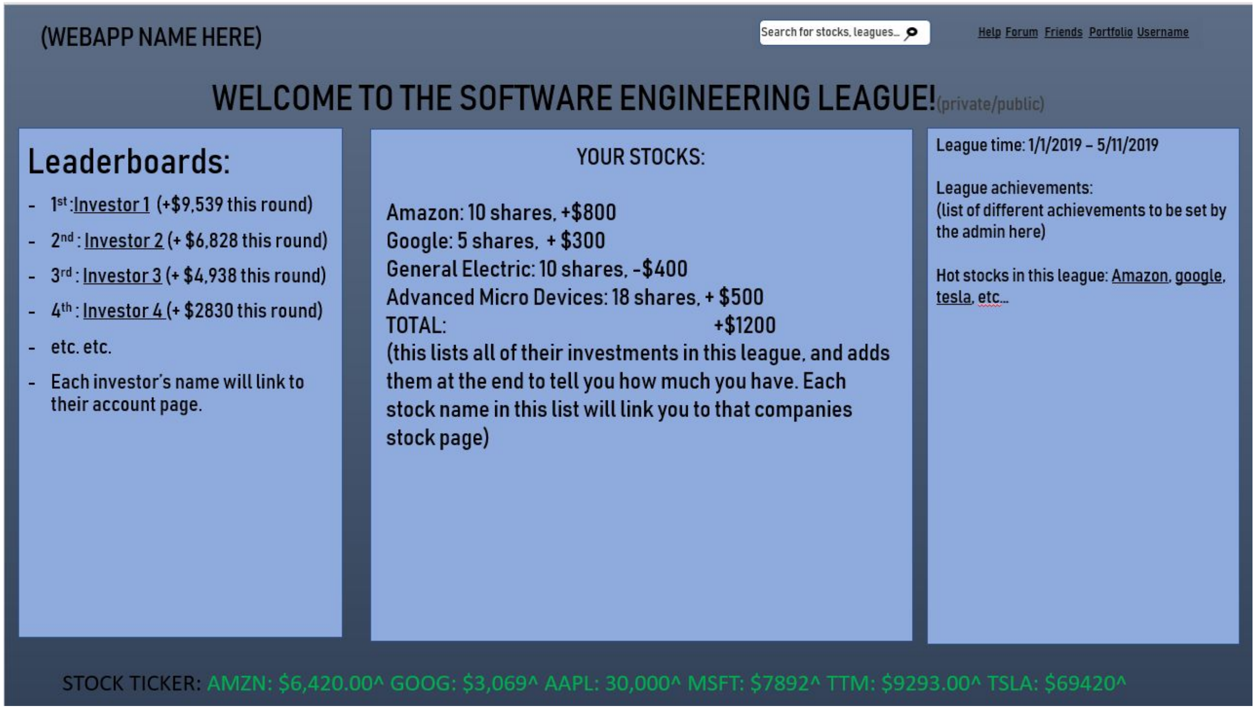
LEAGUES YOU ARE A PART OF:

- League 1 (3rd place)
- League 2 (1st place)
- League 3 (5th place)
- Etc. etc.
- (all league names link to their respective league page)

Example "hot" stock chart on home page (amazon for example)

STOCK TICKER: AMZN: $6,420.00^ GOOG: $3,069^ AAPL: 30,000^ MSFT: $7892^ TTM: $9293.00^ TSLA: $69420 ^

Figure 2.3: A mockup for the "home page"


Search for stocks, leagues...
Help Forum Friends Portfolio Username

(WEBAPP NAME HERE)

WELCOME TO THE SOFTWARE ENGINEERING LEAGUE!(private/public)

Leaderboards:

- 1st:Investor 1 (+$9,539 this round)
- 2nd : Investor 2 (+ $6,828 this round)
- 3rd : Investor 3 (+ $4,938 this round)
- 4th : Investor 4 (+ $2830 this round)
- etc. etc.
- Each investor's name will link to their account page.

YOUR STOCKS:

Amazon: 10 shares, +$800
Google: 5 shares, + $300
General Electric: 10 shares, -$400
Advanced Micro Devices: 18 shares, + $500
TOTAL: +$1200
(this lists all of their investments in this league, and adds them at the end to tell you how much you have. Each stock name in this list will link you to that companies stock page)

League time: 1/1/2019 – 5/11/2019

League achievements:
(list of different achievements to be set by the admin here)

Hot stocks in this league: Amazon, google, tesla, etc...

STOCK TICKER: AMZN: $6,420.00^ GOOG: $3,069^ AAPL: 30,000^ MSFT: $7892^ TTM: $9293.00^ TSLA: $69420^

Figure 2.4: A mockup of a "league page"

3. Functional Requirements Specification

3.1 Stakeholders

Although the web application can be used by anyone interested in investing, one potential target of the web application would be students. High school students can be introduced to investing through the software to get a head start on learning investing tactics. High school students looking to major in economics would benefit from experimenting with the software and develop a heightened interest in the stock market. This same thing applies to college students with an economics major. Just like high school student, university students can use the software to learn financial concepts. Unlike high school students, college students are more likely to partake in real world investing, and so they can more seriously experiment with investing techniques through the web application and bring that knowledge over with them when engaging in real investing. As mentioned earlier, the software can be used casually by anyone looking to dip their toes into investing and is not limited to students.

The web application will be a free service but include advertisements placed throughout the webpage. This means that companies looking to promote their products or services can contact us to discuss marketing. These advertisements would be used to support the web application and allow for continual development of the platform. The use of advertisers would allow for the web application to remain free which will allow it to broadcast to the largest number of customers.

3.2 Actors and Goals

Guest

A user of the web application who has either not logged in or not registered.

- Create an account.
- View leaderboard and stock prices.

Investor

A user who has registered and logged into an account.

- Join and create leagues.
- View portfolio.
- View leaderboard and stock prices.
- Buy or sell securities.

League Manager

A user who has created or is in control of a league.

- Create a league competition by setting the rules.
- In charge of adding/removing players participating in the league
- Can set league to public or private.
- Edit league settings.

Site Administrator

Manages the whole web application.

- Remove users who exhibit inappropriate behavior.
- Make front page announcements and post news.

Database

Holds the information of all users' accounts.

- Retrieve information as soon as a user account is created.
- Store new data about users or events, including user portfolios.

Alpha Vantage API

Holds information about current financial statistics

- Retrieves information about stock prices and sends it to the database or live view in application.

3.3 Use Cases

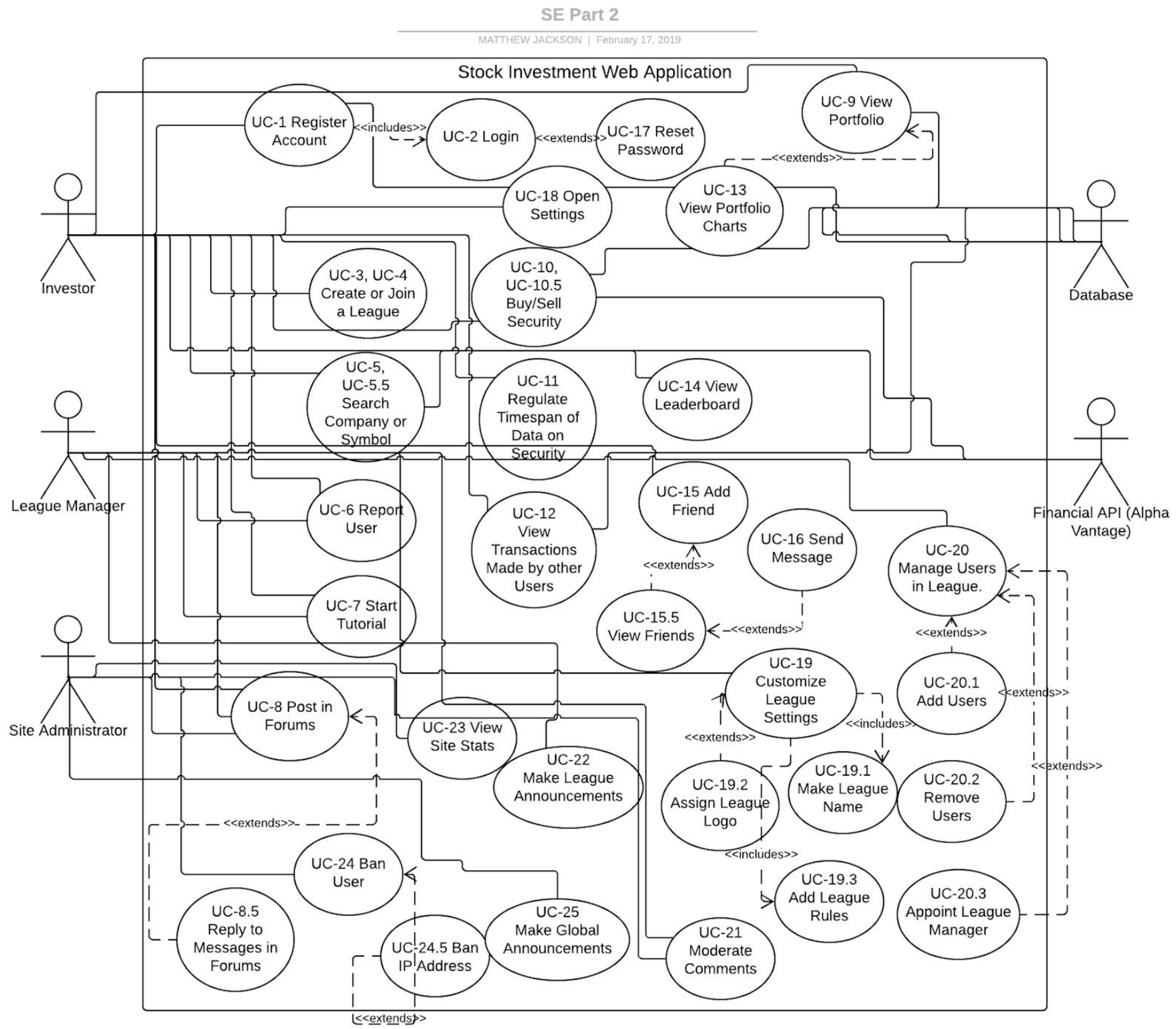
i. Casual Description

Use Case	Use Case Name	User Story	Actor	Actor's Goal
UC-1	CreateAccount	ST-1	Investor	To create an account using own email.
UC-2	Login	ST-1	Investor	To login to user account.
UC-3	JoinLeague	ST-2	Investor	To join any/all leagues desired.
UC-4	CreateLeague	ST-3	Investor/League Mgr	To create a league and become its manager.
UC-5	SearchCompany	ST-4	Investor	To search for company data using symbol or name.
UC-5.5	SearchSymbol	ST-4	Investor	Search non-company security using its symbol.

UC-6	ReportUser	ST-5	Investor	To report other abusive users.
UC-7	StartTutorial	ST-6	Investor	To be led through the basics of the application, and shown its functions.
UC-8	ForumPost	ST-7	Investor	To post a message in the user forum.
UC-8.5	ForumReply	ST-7	Investor	To reply to a posted message in the forum.
UC-9	PortfolioView	ST-8	Investor	To assess the status of all investments using a portfolio.
UC-10	BuySearched	ST-9	Investor	To purchase a found security.
UC-10.5	SellOwned	ST-9	Investor	To sell owned security.
UC-11	SetTimespan	ST-10	Investor	To regulate the timespan of data provided on a chosen security.
UC-12	ViewOtherTransactions	ST-11	Investor	To see transactions made by other users.
UC-13	PortfolioChartView	ST-12	Investor	To assess one's own portfolio's status via data visualizations such as charts.
UC-14	LeaderboardView	ST-13	Investor	To compare one's own profits to those of other users in a leaderboard format.
UC-15	AddFriend	ST-14	Investor	To "add" other users as friends.
UC-15.5	ViewFriend	ST-14	Investor	To view added users' online status.
UC-16	SendMessage	ST-15	Investor	To send messages to friends and/or league members.
UC-17	ResetPwd	ST-16	Investor	To recover or change password if need be.
UC-18	OpenPortfolio	ST-17	Investor	To access own profile/portfolio from any active page.
UC-18.5	OpenSettings	ST-17	Investor	To access settings from any active page.
UC-19	CustomizeLeague	ST-18	League Mgr	To customize league settings.
UC-19.1	LeagueName	ST-18	League Mgr	To customize league name.
UC-19.2	LeagueLogo	ST-18	League Mgr	To customize league logo.
UC-19.3	AddLeagueRule	ST-18	League Mgr	To customize league rules.
UC-20	ManageUser	ST-19	League Mgr	To manage user membership in a league.

UC-20.1	AddLeagueUser	ST-19	League Mgr	To manage user membership: add league users.
UC-20.2	DelLeagueUser	ST-19	League Mgr	To manage user membership: remove league users.
UC-20.3	AppointMgr	ST-19	League Mgr	To appoint other league managers.
UC-21	ModComment	ST-20/23	League Mgr/Site Admin	To moderate/remove league page comments.
UC-22	LeagueAnnounce	ST-21	League Mgr	To issue announcements to owned league.
UC-23	SiteStats	ST-22/ST-25	Site Admin	To view reports on users activity, including user count, active and total league number, daily transactions, favored stocks, created/banned users.
UC-24	BanUser	ST-23	Site Admin	To issue user bans.
UC-24.5	BanIP	ST-23	Site Admin	To issue IP bans.
UC-25	GlobalAnnounce	ST-24	Site Admin	To issue news or announcements directly on the application site.

ii. Use Case Diagram



iii. Traceability Matrix

User Story	PW	UC-1	UC-3	UC-5	UC-7	UC-8	UC-9	UC-10	UC-19	UC-23
ST-1	10	X								
ST-2	6		X							
ST-3	10		X							

ST-4	6			X						
ST-5	2									X
ST-6	4				X					
ST-7	4					X				
ST-8	8						X			
ST-9	10							X		
ST-10	6			X			X			
ST-11	3						X			
ST-12	4						X			
ST-13	2						X			
ST-14	2									
ST-15	2									
ST-16	5	X								
ST-17	8									
ST-18	8								X	
ST-19	8								X	
ST-20	4								X	
ST-21	3								X	
ST-22	2									X
ST-23	6									X
ST-24	3									X
ST-25	9									X
Total Priority		15	16	12	4	4	23	10	23	20

iv. Fully-Dressed Description

Use Case UC-1	Register an Account
Related Requirements:	ST-1
Initiating Actors:	Guest
Actor’s Goal:	Register with our servers to enter the database.
Participating Actors:	Guest, Database
Preconditions:	-The guest must not be a registered user.
Postconditions:	-The Database gets updated with information about the user and the guest becomes an Investor.
Flow of Events for Main Success Scenario	
→	1. Guest visits the web application and attempts to register.
←	2. Database is accessed and checks that an investor is not found.
←	3. Guest is registered as an investor in the database.
→	4. Registration is confirmed to the user and start portfolio is displayed.
Flow of Events for Extensions	
→	1. Guest attempts to make an account.
←	2. Database is checked and the account already exists.
←	3. Message signaling that the account already exists is displayed to the user.

Use Case UC-3,4	Create/Join a League
Related Requirements:	ST-2, ST-3, ST-18, ST-19, ST-20, ST-21
Initiating Actor:	Investor
Actor’s Goal:	Create or participate in a league that will hold competitions.
Participating Actors:	Database, Investors

Preconditions:	-User is logged in. -League is not already created. -Investor is not already in league.
Postconditions:	-The league is created with all settings filled in. -The Investor joins the league. -Database is updated about new league and investors in it.
Flow of Events for Main Success Scenario	
→ ← → ←	1. Investor navigates to leagues page. 2. All public leagues are displayed to the Investor. 3. Investor selects to join a league. 4. Investor is registered into the league and the Database is updated.
Flow of Events for Extensions	
3a. The investor selects to create a league.	
→ ← ←	1. Investor inputs league name and settings. 2. The league is created and updated in the Database. 3. Investor is turned into the League Manager of the league.
4a. Investor attempts to join or create a league without permission.	
←	1. The request is rejected and the reason for the rejection is displayed.

Use Case UC-5	Browse Companies
Related Requirements:	ST-4, ST-10
Initiating Actor:	Investor
Actor’s Goal:	View information on securities and companies.
Participating Actors:	Database, Alpha Vantage
Preconditions:	-Investor is logged in. -Alpha Vantage is up and running appropriately.
Postconditions:	-Correct companies and security prices must be displayed.
Flow of Events for Main Success Scenario	

→	1. Investor begins searching for a company.
←	2. Investor is given suggestions based on the characters inserted.
→	3. Investor enters their own search or clicks on a suggestion.
←	4. Information is retrieved from Alpha Vantage and is displayed to the Investor.
Flow of Events for Extensions	
3a. Investor’s search is invalid.	
←	1. A message is displayed that the entered company does not exist and similarly named companies are displayed.

Use Case UC-10	Place Market Order
Related Requirements:	ST-9
Initiating Actor:	Investor
Actor’s Goal:	To purchase or sell a security through a market, limit, or stop order.
Participating Actors:	Database, Alpha Vantage
Preconditions:	-Investor is logged in. -Alpha Vantage is up and running appropriately. -Investor is a member in a league.
Postconditions:	-Investor’s portfolio is updated based on order.
Flow of Events for Main Success Scenario	
→	1. The investor places an order from inside a league.
←	2. Prompt for order type, stock amount, and company are displayed.
→	3. Investor fills out the prompt and the order is placed.
←	4. Market price is received from Alpha Vantage and places the order into the Database.
Flow of Events for Extensions	
1a. The Investor places an order from a company’s profile and not from inside a league.	
→	1. The Investor selects which league to place the order in.
←	2. The same prompt explained in Step 2 above is displayed to the

	investor.
4a. The Investor does not have enough funds or margin to place an order.	
←	1. A message alerting the Investor that they do not have enough funds or margin is displayed and the order is prevented from completing.

Use Case UC-9	Manage Portfolio
Related Requirements:	ST-8, ST-12, ST-13
Initiating Actor:	Investor
Actor's Goal:	View current standings and securities.
Participating Actors:	Database
Preconditions:	-Investor is logged in. -Investor is a member of a league.
Postconditions:	-Correct portfolio information must be displayed.
Flow of Events for Main Success Scenario	
→	1. Investor opens up portfolio.
←	2. Portfolio information is retrieved from the Database and is displayed to the Investor.

Use Case UC-7	Start Tutorial
Related Requirements:	ST-6
Initiating Actor:	Investor
Actor's Goal:	To learn how to navigate the web application while learning about investing.
Participating Actors:	None
Preconditions:	-Investor is logged in.
Postconditions:	-It will be marked that the user has completed the tutorial.
Flow of Events for Main Success Scenario	

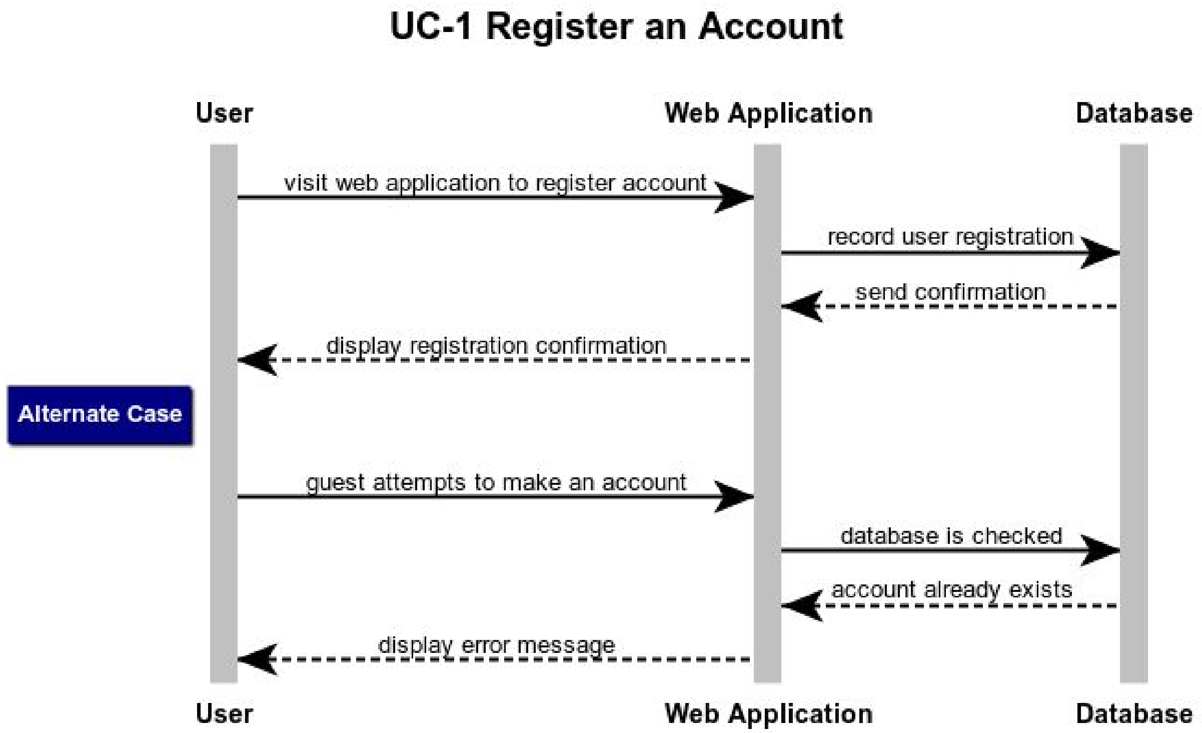
→ ← ←	1. Investor selects tutorial option. 2. What will be taught in the tutorial is displayed. 3. The investor is taken through a step-by-step tutorial on the workings of the web application.

Use Case UC-23	Take Administrative Actions
Related Requirements:	ST-22, ST-23, ST-24, ST-25
Initiating Actor:	Site Administrator
Actor’s Goal:	Manage the website and the Database
Participating Actors:	Database, Investors, League Managers
Preconditions:	-User is the Site Administrator.
Postconditions:	-The Database is updated to reflect any actions taken by the Site Administrator.
Flow of Events for Main Success Scenario	
→ ←	1. The Site Administrator requests logs from the Database. 2. Database sends the data requested by the Site Administrator to the Site Administrator.

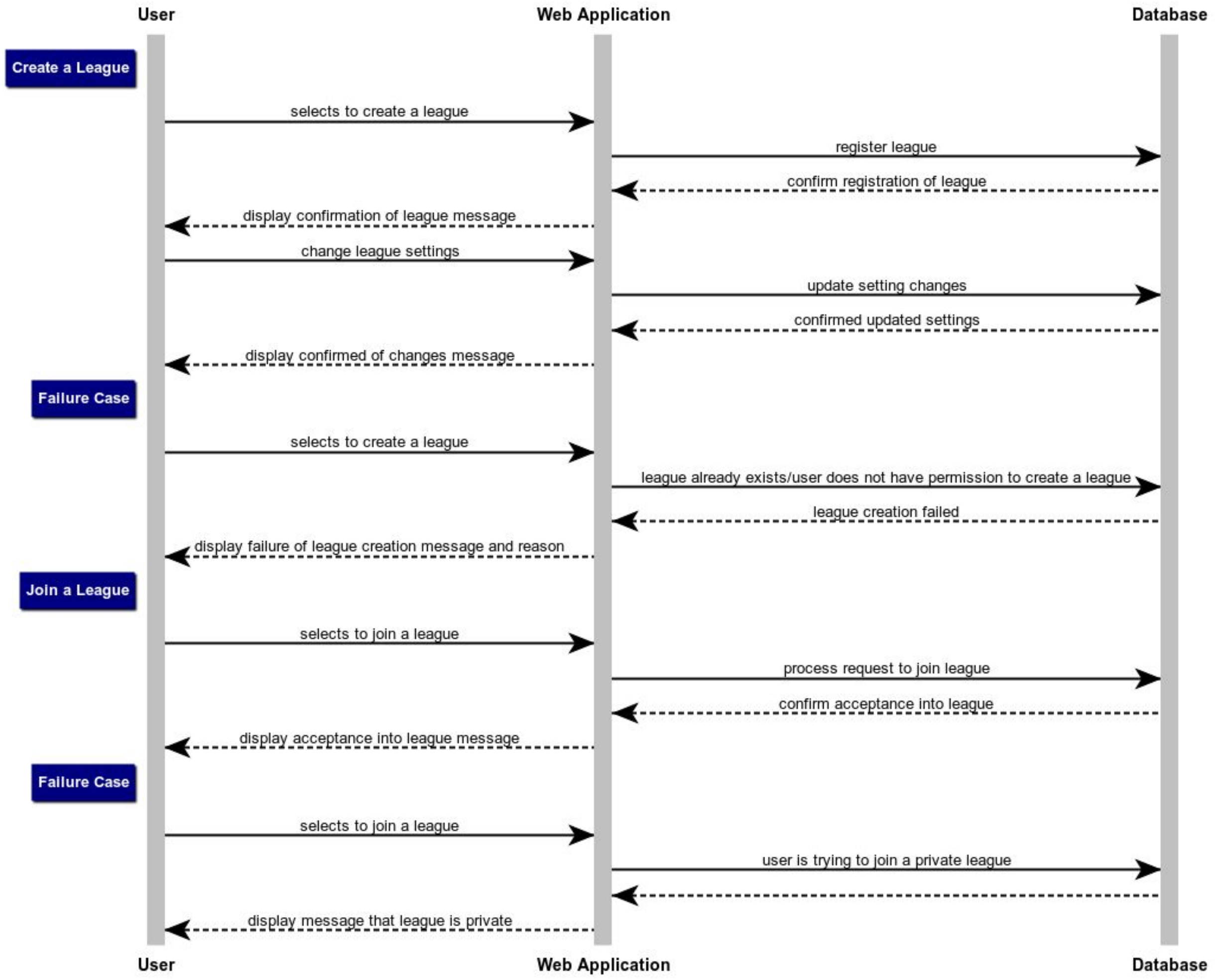
Use Case UC-19	Manage League Settings
Related Requirements:	ST-18, ST-19, ST-20, ST-21
Initiating Actor:	League Manager
Actor’s Goal:	Set/Change league settings
Participating Actors:	Database, Investors
Preconditions:	-User is a League Manager.
Postconditions:	-Database is updated to reflect the changes made by the League Manager.
Flow of Events for Main Success Scenario	

→ ← ←	1. League Manager makes adjustments to league settings. 2. Database is updated with the setting changes. 3. The updated changes are displayed to the League Manager.

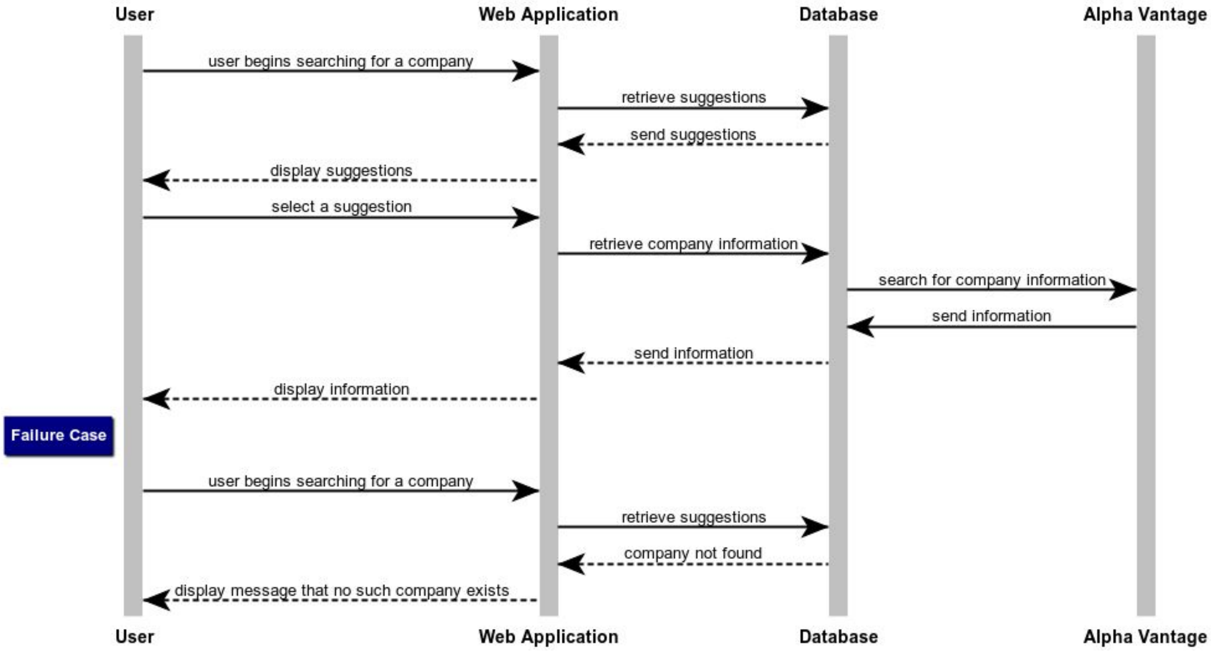
3.4 System Sequence Diagrams



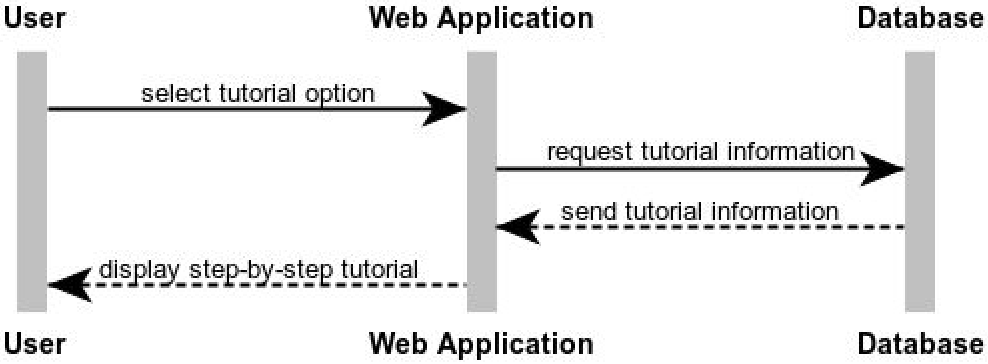
UC-3,4 Create/Join a League


User
Web Application
Database
Create a League
selects to create a league
register league
confirm registration of league
display confirmation of league message
change league settings
update setting changes
confirmed updated settings
display confirmed of changes message
Failure Case
selects to create a league
league already exists/user does not have permission to create a league
league creation failed
display failure of league creation message and reason
Join a League
selects to join a league
process request to join league
confirm acceptance into league
display acceptance into league message
Failure Case
selects to join a league
user is trying to join a private league
display message that league is private
User
Web Application
Database

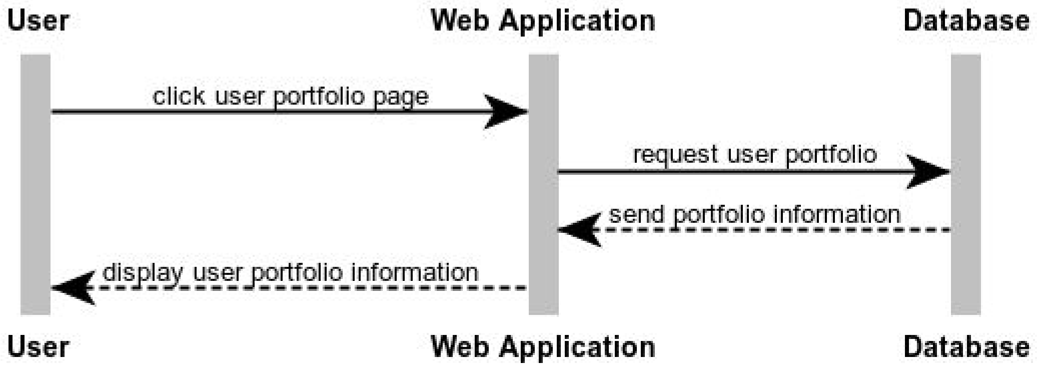
UC-5 Browse Companies



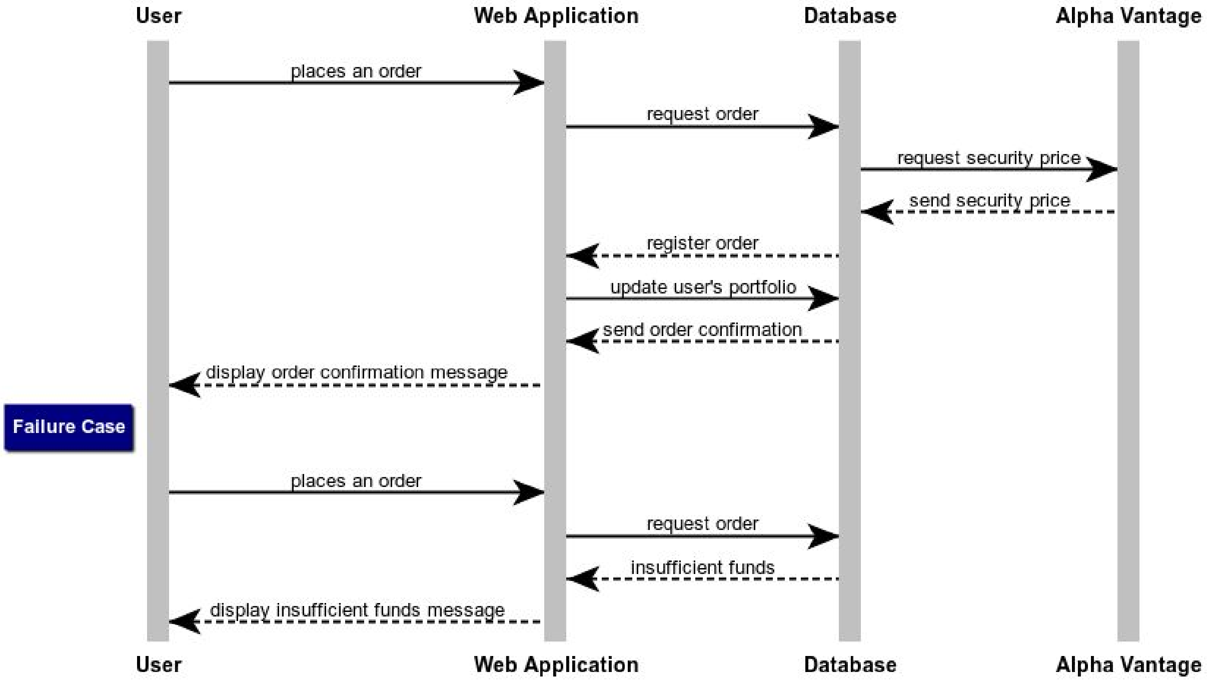
UC-7 Start Tutorial


User
Web Application
Database
select tutorial option
request tutorial information
send tutorial information
display step-by-step tutorial
User
Web Application
Database

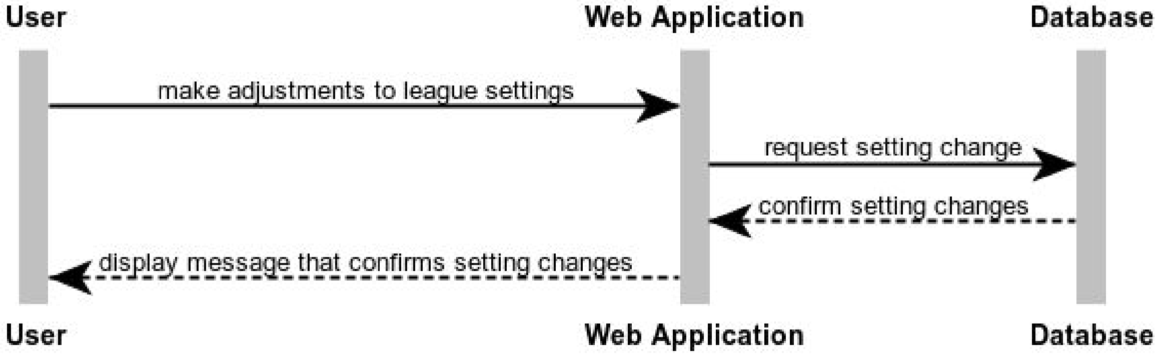
UC-9 Manage Portfolio


User
Web Application
Database
click user portfolio page
request user portfolio
send portfolio information
display user portfolio information
User
Web Application
Database

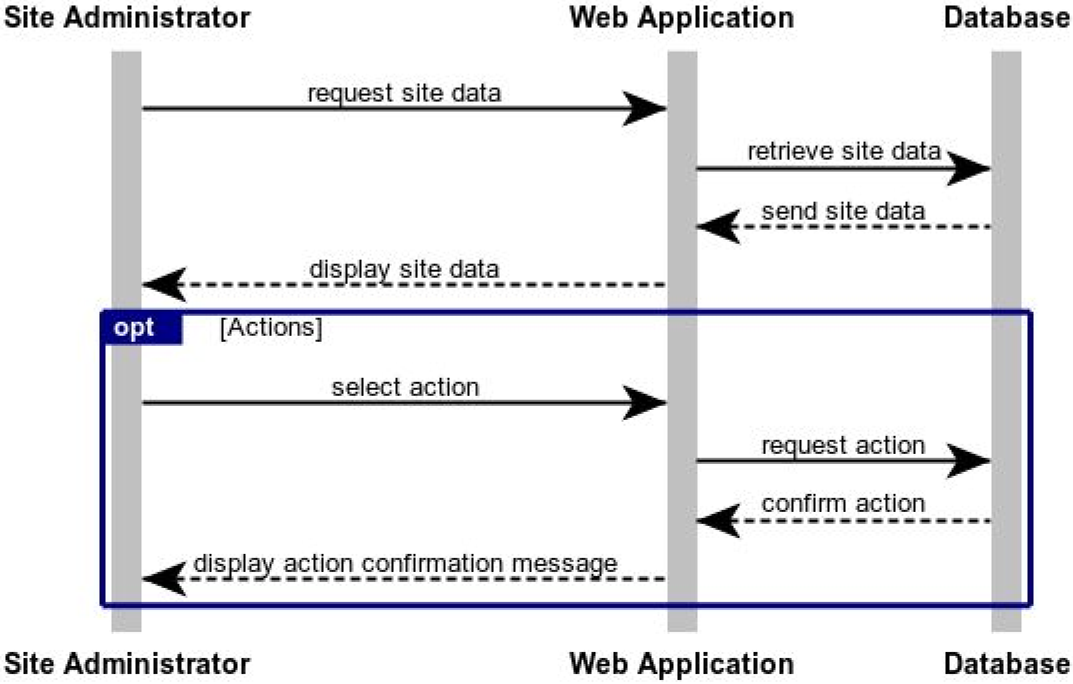
UC-10 Place Market Order



UC-19 Manage League Settings



UC-23 Take Administrative Actions



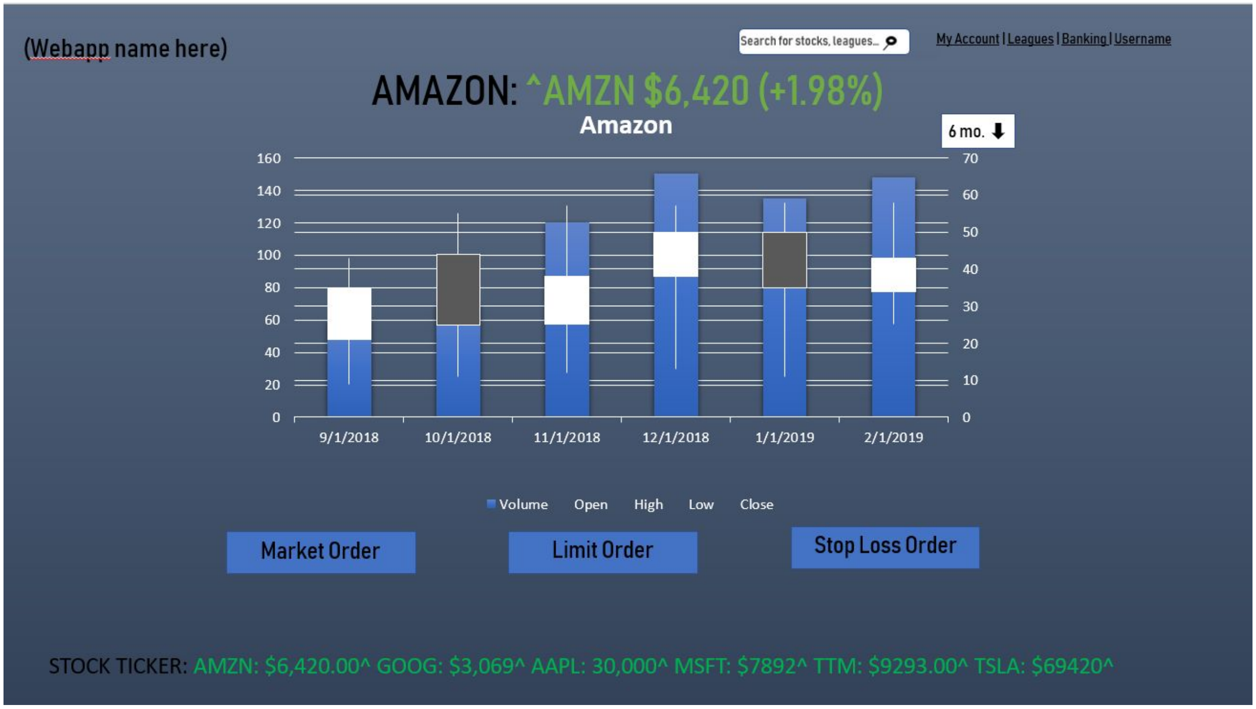
4. User Interface Specification

4.1 Preliminary Design

Our User Interface will behave as the main area our users will see when investing. It would display the stock prices, stocks that are available, current portfolio, its value, global rank, a dash to perform market orders, a news feed, and a graphic to provide a quick view of stock performance. We hope to add different views, making this not the only place a user can see certain information.

Intro Page/Screen will have 4 elements. The first element at the top is the title of the app. The second element is the logo, which will take up the left half of the page. The right half is for the user to sign in or sign up. The fourth element is at the bottom where we have live ticker updates for the use to view. If the use selects to create a new account, the layout of the page does not change, just the right part of the page will display a new box with details regarding sign-up details. After this page the use will be displayed the home page.

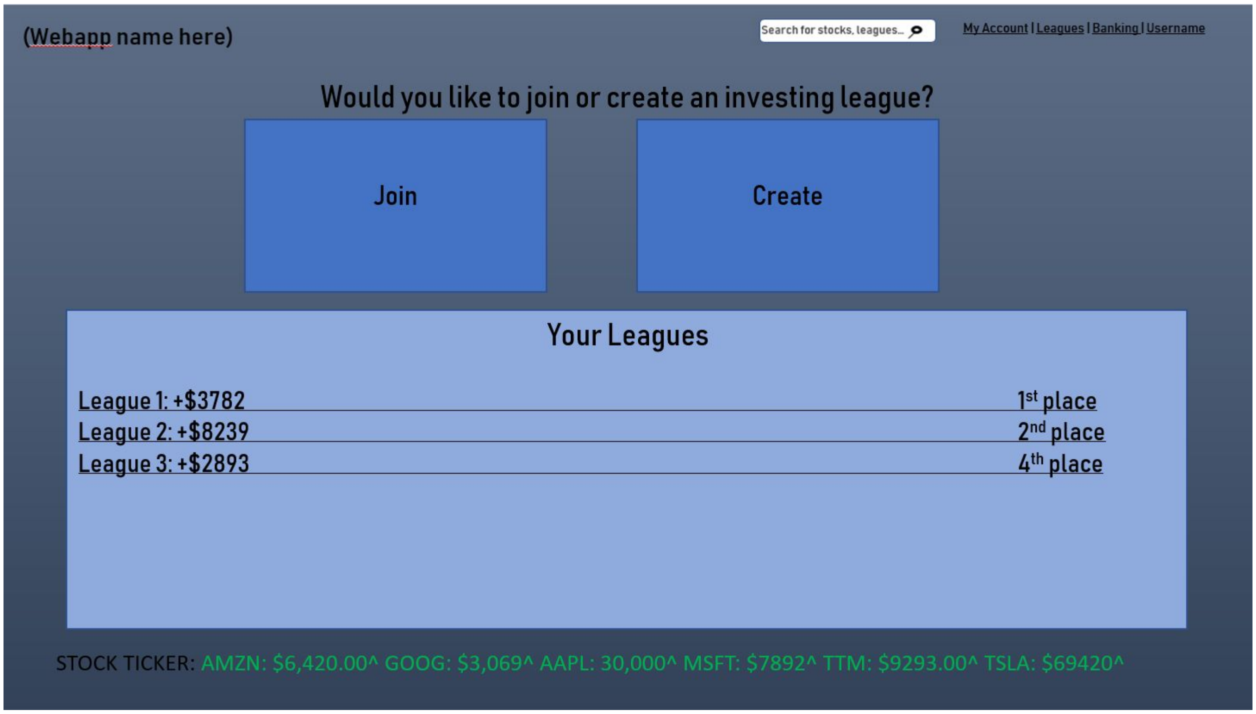
The home page has the top reserved for the logo and menu. The menu has options to direct the use to My Account | My Leagues | Banking | Search Stock. Below the menu, the page is divided into 3 vertical sections,the sections on the left is ¼ size of the page, middle is 1/2 , right is 1/4. The section on the left is used to list all information regarding league updates, league messages, league manager, league news. The middle section displays current portfolio and/or has stocks the user has interest in. The section on the right has news related to the stocks of interest, or any important updates/notification sent my admins/managers. Once a stock is selected from the middle screen, it will navigate to a different page, which is the transaction page. This page has the logo on the top along with the menu described above. Below the page displays various option to buy the stock: Market Order | Limit Order | Stop Loss Order. Once the option is selected, the page displays a message when the order has been executed and immediately the page updates to shows the charts, total gain and total return on the stocks.



AMAZON: ^AMZN $6,420 (+1.98%)

STOCK TICKER: AMZN: $6,420.00^ GOOG: $3,069^ AAPL: 30,000^ MSFT: $7892^ TTM: $9293.00^ TSLA: $69420^

Example stock buy page (note that all numbers are placeholders)


(Webapp name here)
Search for stocks, leagues...
My Account | Leagues | Banking | Username
Join
Create

Would you like to join or create an investing league?

Your Leagues

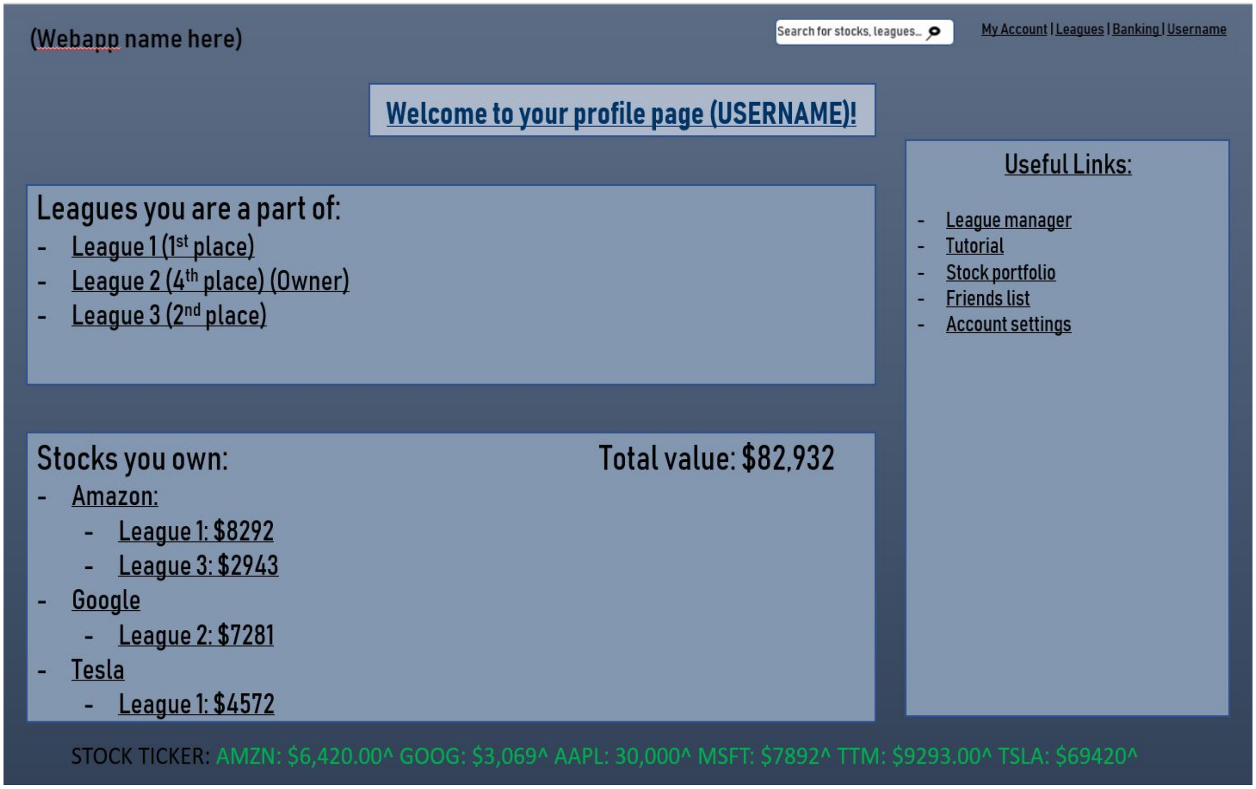
League 1: +$3782 1st place

League 2: +$8239 2nd place

League 3: +$2893 4th place

STOCK TICKER: AMZN: $6,420.00^ GOOG: $3,069^ AAPL: 30,000^ MSFT: $7892^ TTM: $9293.00^ TSLA: $69420^

Example “leagues” tab. Can join/create leagues and see ones user is already a part of.


(Webapp name here)
Search for stocks, leagues...
My Account | Leagues | Banking | Username

Welcome to your profile page (USERNAME)!

Leagues you are a part of:

- League 1 (1st place)
- League 2 (4th place) (Owner)
- League 3 (2nd place)

Stocks you own: Total value: $82,932

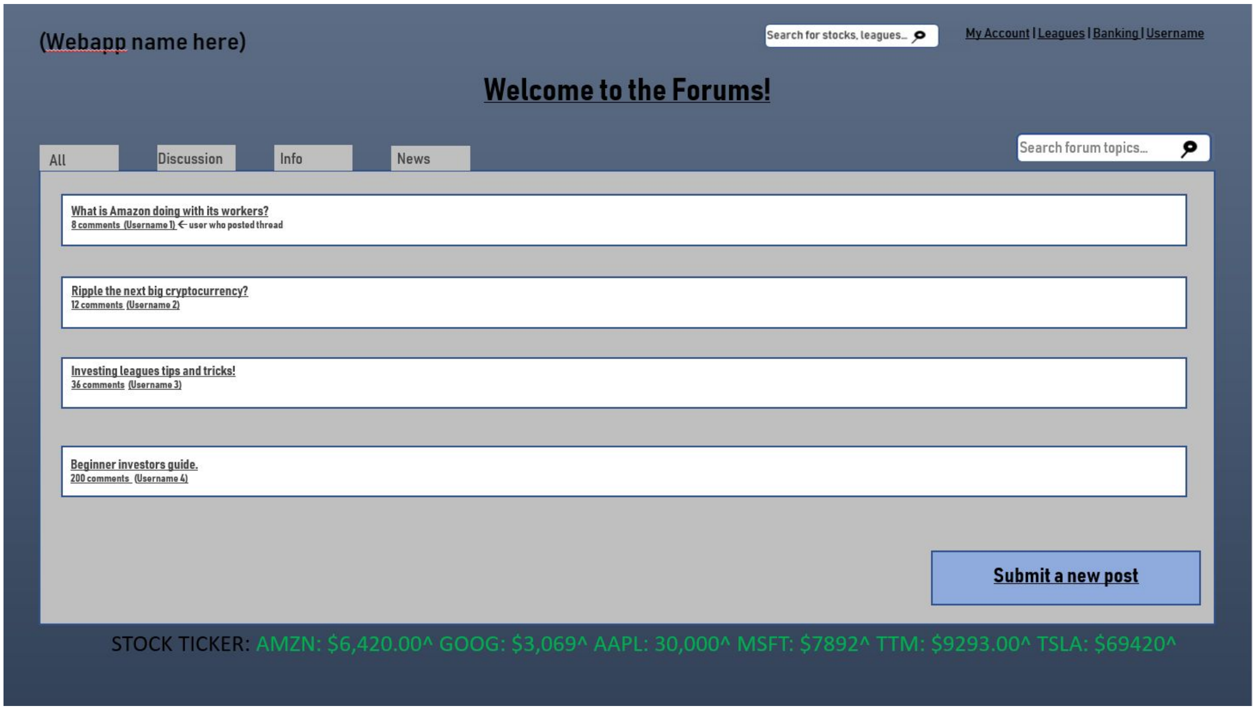
- Amazon:
  - League 1: $8292
  - League 3: $2943
- Google
  - League 2: $7281
- Tesla
  - League 1: $4572

Useful Links:

- League manager
- Tutorial
- Stock portfolio
- Friends list
- Account settings

STOCK TICKER: AMZN: $6,420.00^ GOOG: $3,069^ AAPL: 30,000^ MSFT: $7892^ TTM: $9293.00^ TSLA: $69420^

Example profile page



Welcome to the Forums!

What is Amazon doing with its workers?
8 comments (Username 1) ← user who posted thread

Ripple the next big cryptocurrency?
12 comments (Username 2)

Investing leagues tips and tricks!
36 comments (Username 3)

Beginner investors guide.
200 comments (Username 4)

STOCK TICKER: AMZN: $6,420.00^ GOOG: $3,069^ AAPL: 30,000^ MSFT: $7892^ TTM: $9293.00^ TSLA: $69420^

Example Forums home/all page

4.2 User Effort Estimation

We hope that the user doesn't have to put too much effort into navigating through our app, to facilitate this, we will have clearly labeled buttons that the user can read. To execute an order, we will make sure this button is obvious and clear. In terms of how many clicks it takes to access each specific part of the website are minimal.

- From login page to list of leagues (2 clicks)
    - Login page
    - Leagues hotbar on left side
- From login to any stock or league (3 clicks & some keystrokes)
    - Login page
    - search bar search for whatever is desired by the user (with any number of keystrokes as a search)
    - select desired query
- For a specific forum post (4 clicks & some keystrokes)
    - Login
    - Forum hyperlink in top right corner
    - forum search bar (any # of keystrokes)
    - Desired forum post
- to access a fellow league member's profile (3 clicks or 3 clicks and keystrokes)
    - Login
    - League hotbar
    - Investor username in list of league members
    - **OR**
    - Login
    - Search bar searching for profile name (any # of keystrokes)
    - Select desired profile
- To access your own profile (2 clicks):
    - Login
    - Profile tab in the top right
- Find the tutorial (3 clicks):
    - Login
    - Profile page
    - Tutorial link

This app is being designed so that each page is rich in links and features in order to swiftly navigate to any part of the site desired within 5 clicks or so with an occasional set of query keystrokes in between.

5. Domain Analysis

5a. Domain Model

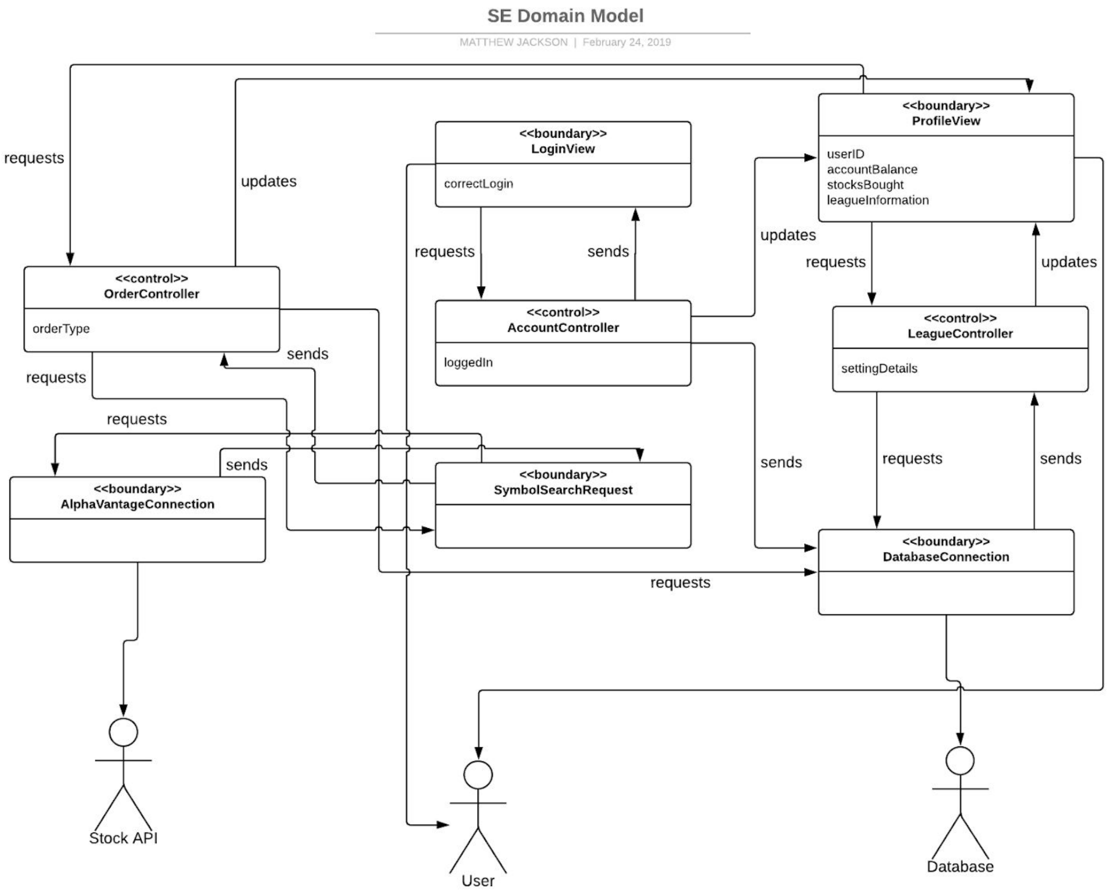


Figure 5.1: Domain Model

i. Concept Definitions

Responsibility Description	Type	Concept Name
Rs1. Controls information regarding to a new user’s account registration and an existing user’s login.	D	Account Controller

Rs2. Controls information related to an Investor purchasing a security through a certain order type.	D	Order Controller
Rs3. Controls information regarding league managers, league members, and league settings.	D	League Controller
Rs4. Database query that sends information to and from the database that can revolve around league, portfolio, or security information.	D	Database Connection
Rs5. Specifies search parameters for retrieving company or stock information from the Alpha Vantage API.	K	Symbol Search Request
Rs6. API that contains information on stocks, trades, and companies.	K	Alpha Vantage Connection
Rs7. HTML document that displays a user's profile, which contains information related to their own securities.	K	Profile View
Rs8. HTML document that displays the login and account registration screen.	K	Login View

Account Controller

The Account Controller works in conjunction with Login View, Profile View, and Database Connection to handle all the information related to a user's account. Once a user logs in and registers an account, the Account Controller takes this data and sends it to Database Connection. When the user clicks to view their profile, Account Controller changes the view from Login View to Profile View.

Order Controller

The Order Controller updates the Profile View when a user makes an order of any security. Order Controller works with Symbol Search Request to get the price of the security to be purchased or sold when a user makes an order.

League Controller

The League Controller updates the Profile View when a change is made to any league. This includes the creation or removal of a league, changes made to a league's settings, and the addition or removal of an investor to or from the league.

Database Connection

Database Connection takes on the big job of sending and retrieving data to the database from the rest of the model. Database Connection mainly works in conjunction with the controllers to handle all data related to a user or a league. It is in charge of saving new data and retrieving existing data pertaining to users' information.

Symbol Search Request

Symbol Search Request is in charge of making suggestions to the user when the user beings entering letters to search for a stock or a company. Symbol Search Request works with Alpha Vantage Connection to get the list of companies and stocks that a user can invest in.

Alpha Vantage Connection

Alpha Vantage Connection is used to send information from the finance API to the rest of the model. This contains data related to stock prices and other securities along with companies.

Profile View

The Profile View is where a user will see all information regarding their account. When a user is curious about their portfolio or account balance, they can navigate to their profile. Profile View is updated through the Account Controller which has access to Database Connection.

Login View

The Login View is displayed to a new user or a user who is not logged in. The user can navigate to this page manually, but upon trying to get access to any features that require registration, they will be forcefully redirected to this page. Login View works with Account Controller to send data to the database through Database Connection that revolves around account login information.

ii. Association Definitions

Concept Pair	Association Description	Association Name
Login View ↔ Account Controller	Login View sends a request to create an account or login to one. Account Controller sends back an error if something goes wrong.	requests, sends
Account Controller ↔ Database Connection	Account Controller sends login or registration information. Database connection confirms account registration.	sends
Account Controller ↔ Profile View	Account Controller changes view to Profile View.	updates
Profile View ↔ League Controller	Profile View sends a request to join, leave, or alter a league. League Controller updates Profile View with this new information.	requests, updates
League Controller ↔ Database Connection	League Controller requests league information from Database Connection. Database Connection sends information.	requests, sends
Profile View ↔ Order	Profile requests security order information from	requests, updates

Controller	Order Controller. Order Controller updates Profile View with this new information.	
Order Controller ↔ Database Connection	Order Controller requests Database Connection to store orders into the database.	requests
Symbol Search Request ↔ Alpha Vantage Connection	Symbol Search Request requests information about securities and companies. Alpha Vantage Connection returns the information.	requests, sends
Order Controller ↔ Symbol Search Request	Order Controller requests information about symbols and companies for when a user goes to make an order.	requests, sends

iii. Attribute Definitions

Concept	Attributes	Attribute Description
Account Controller	logged in	Used to determine if the user is logged in
Profile View	user's identity	The user's account name and password.
	account balance	Holds information related to the user's funds.
	stocks bought	Holds information regarding what stocks the user has invested in.
	league information	Contains information related the leagues a user is in.
Order Controller	order type	Holds information relating to whether the user has requested a market, limit, or stop order.
Login View	correct login	Checks if the user has input the correct account information when logging in.

League Controller	setting details	All the information regarding a league's rules.

iv. Traceability Matrix

		Domain Concepts							
Use Case	PW	Account Controller	League Controller	Order Controller	Login View	Profile View	Symbol Search Request	Database Connection	Alpha Vantage Connection
UC-1	10	X			X			X	
UC-3,4	6		X			X		X	
UC-5	3						X		X
UC-7	2	X							
UC-9	5	X				X		X	X
UC-10	6			X				X	X
UC-19	5		X					X	
UC-23	4	X						X	

5b. System Operation Contracts

UC-1 Register/Create an Account

- *Preconditions*
  - If a new user is visiting the web application, they must first register before joining/creating a League.
- *Postconditions*
  - After registration, the database is updated and the new user is now acknowledged as being an investor.

UC-3,4 Create/Join League

- *Preconditions*
  - User is logged into their account.

    - There cannot already exist a League with the same name.
    - User has not joined a League yet.
- *Postconditions*
    - User has joined a League.
    - Database has been updated.
    - League has been set with selected settings.

UC-5 Browse Companies

- *Preconditions*
    - User is logged into their account.
    - Alpha Vantage is accepting inquiries.
- *Postconditions*
    - None

UC-7 Start Tutorial

- *Preconditions*
    - User is logged into their account.
- *Postconditions*
    - User is marked as having completed the tutorial, which is required for any user to become a league manager.
    - The tutorial is not closed to a user who has completed it. It can be repeated as many times as desired.

UC-9 Manage Portfolio

- *Preconditions*
    - User is logged into their account.
    - Alpha Vantage is accepting inquiries.
- *Postconditions*
    - Any adjustments made to the investors portfolio have been updated in the database.

UC-10 Place a Market Order

- *Preconditions*
    - User is logged into their account.
    - Investor has enough funds in their account to place a market order.
    - Alpha Vantage is accepting inquiries.
- *Postconditions*
    - User profile is reflected with any change to funds or position.
    - Database has been updated with these changes.

UC-19 Manage League Setting

- *Preconditions*
    - Initiating actor is the League Manager.

  - League Manager is logged into their account.
- *Postconditions*
  - Database is updated to reflect any changes made to their account.
  - All users are notified of any changes made in their League.

UC-23 Take Administrative Actions

- *Preconditions*
  - User is the Site Administrator.
  - There exists an issue that needs to be resolved.
  - There are outstanding abuse reports.
- *Postconditions*
  - Conflicts/Issues have been resolved.
  - The reported user has been notified of any actions taken against them.

6. Project Size

Actor classification:

Actor Name	**Relevant characteristics**	**complexity**	**weight**
investor	Interacts with finance system through a graphical user interface in order to look at stocks and manage leagues	complex	3
League manager	Manages leagues through the gui interface on the finance site	complex	3
Site admin	Interacts with the site through text based means in order to administer the site	average	2
database	Is accessed by the users through a dedicated api that we develop	simple	1
Finance api (Alpha Vantage)	Pulls finance data from outside sources using a set api	simple	1
guest	Makes new account on site, only happens if account not already made for user	average	2

Unadjusted actor weight: 2*3 + 2*1 + 2*2 = 12

Use case weight description:

Use Case	**description**	**category**	**weight**
UC-1	Guest registers for an account in our database, 2 actors (investor, database) <5 steps for success with <5 accessory steps	average	10
UC-2	User logs into our database, 2 actors (investor, database) but <5 steps for success	simple	5
UC-3	Investor attempts to join a league that already exists in our database, 2 actors (investor, database)	average	10
UC-4	Investor attempts to create a league to hold competitions (investor, database)	average	10
UC-5	Investor searches for information on securities and/or companies, 3 actors (investor, database, alpha vantage).	average	10

	Average work because 4 events for success and 1 extension event		
UC-7	Investor selects the start tutorial link. 1 actor (investor) <5 events for success	Simple	5
UC-9	Investor wants to manage his stock portfolio, 2 actors (investor, database) <5 events for success	average	10
UC-10	Place market order, 3 actors (investor, database, alpha vantage) <10 total events	Complex	15
UC-19	Manage league settings, 3 total actors (league manager, investor, Database) with <5 events total	average	10
UC-23	Take administrative actions, be able to make changes to site and database, 4 total actors (database, investors, league managers, site administrator) with <5 events for completion but with many different options	complex	15

Unaltered Use Case Weight: 2x5 + 6*10 + 2*15 = 100

Technical Complexity Factors:

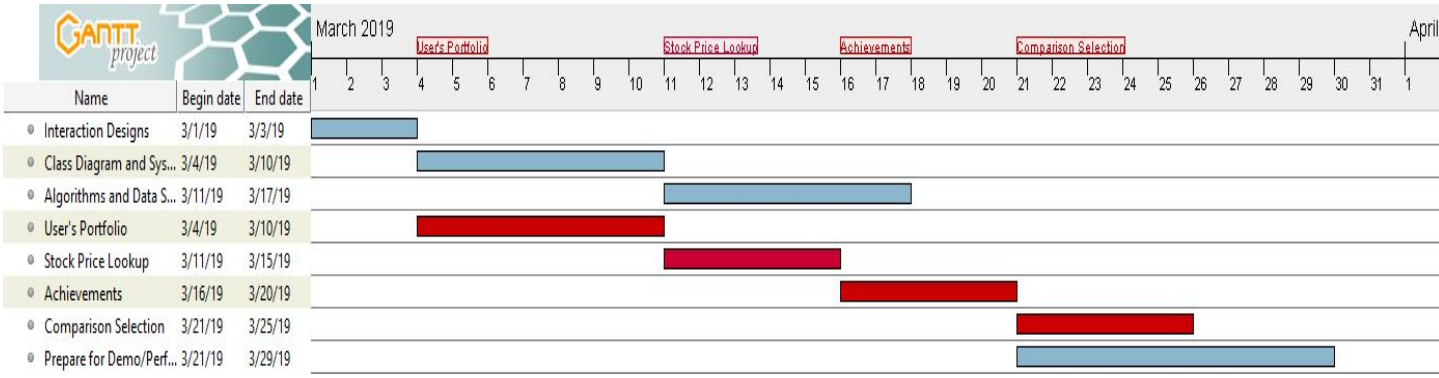
Technical factor	Description	Weight	Percieved complexity	Calculated weight factor
T1	Users expect good performance	1	3	3
T2	Web based system to be used on multuple machines	2	3	6
T3	Easy to modify features of leagues and profiles	1	3	3
T4	Easy enough to change from a site administrator standpoint	1	2	2
T5	Security is very important to protect users finances	1	4	4
T6	Some unique training required through the tutorial but should be easy	.5	1	.5

	to use			
T7	Response times critical to keep up with stock changes	2	2	4
T8	Internal processing should not be too complex	1	2	2
T9	Concurrent use of the system is required as there are multiple users	2	2	4
T10	System needs to remain up to date to follow stock prices every day	3	1	3
T11	User pages need to have multiple areas of information to track different leagues	3	2	6
T12	Easy to “install” and use as the app is web based	.5	1	.5
T13	Has some direct access from third party finance API’s (alpha vantage)	2	1	2
	Technical Factor totals:			**40**

7. Plan of Work

7.1 Development and Report Milestones

- Create user’s portfolio - 3/10
  - Simple stock list view that provides basic information about the user’s portfolio
- Stock price lookup - 3/15
  - Allow to search for a stock by ticker
  - Display the current price of the stock
  - Display historic price chart
- Achievements - 3/20
  - Create achievements page that shows completed and incomplete achievements
  - Show requirements for incomplete achievements
- Comparison section - 3/25
  - Create comparisons page housing dynamic view chart
  - Allow user to compare portfolio to stock or other charts



Name	Begin date	End date
Interaction Designs	3/1/19	3/3/19
Class Diagram and Sys...	3/4/19	3/10/19
Algorithms and Data S...	3/11/19	3/17/19
User's Portfolio	3/4/19	3/10/19
Stock Price Lookup	3/11/19	3/15/19
Achievements	3/16/19	3/20/19
Comparison Selection	3/21/19	3/25/19
Prepare for Demo/Perf...	3/21/19	3/29/19

7.2 Breakdown of Responsibilities

Portfolio and Market Insight Tools, Achievements: Andrew Marfitsin, Himateja Madala

- Stock price lookup with historic price chart
- Own portfolio value chart
- Own portfolio vs. stock/ETF/index/mutual fund chart
- User’s list of achievements, accomplished and incomplete

Website operator and Advertiser Interaction: Matthew Jackson, Paul Stanik

- Manage the webapp

- Monitor advertisers who want to sign up to get their products displayed on the website
- Track advertisers’ balances due

Product Functionality: David Wang, Paul Stanik, Murali Gunti

- Buying and selling stocks
- Creating/deleting a portfolio
- Detecting Low/High price, executing sell/buy based on said prices
- Making sure achievements are triggered

GUI Management: Joseph Coleman, Akshaykumar Patil

- Ensuring the user has a smooth and easy interaction with software
- Putting together a clear and coherent manual describing how to use software

References

- Information about ETFS
https://www.investopedia.com/terms/e/etf.asp
- Information about Mutual Funds
https://www.investopedia.com/terms/m/mutualfund.asp
- Information about Stocks
https://www.investopedia.com/terms/s/stock.asp
- Information about Orders
https://www.investopedia.com/terms/o/order.asp
- Information about Portfolios
https://www.investopedia.com/terms/p/portfolio.asp
- Stock API
https://www.alphavantage.co/