Rags to Riches

Software Engineering

Team 5:

Alejandro Aguilar
Arjun Ohri
Deep Patel
Kartik Patel
Elisa-Michelle Rodriguez
William He
Bryan Benalcazar

February 19, 2017
Github Page

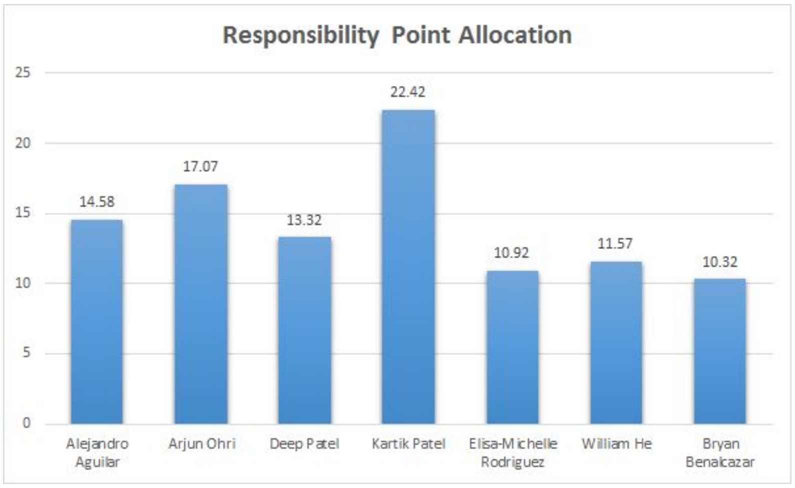
TABLE OF CONTENTS

0 INDIVIDUAL CONTRIBUTIONS BREAKDOWN

0.1 Responsibility Matrix

	Team Members						
	Alejandro Aguilar	Arjun Ohri	Deep Patel	Kartik Patel	Elisa-Michelle Rodriguez	William He	Bryan Benalcazar
Project Management (10 points)	50%			50%			
Sec.1: CSR (9 points)	5 %	30 %	5 %	40 %	5 %	5 %	10 %
Sec.2: System Reqs (6 points)	10 %	15 %	50 %			25 %	
Sec.3: Functional Reqs Specification (30 points)	7%	20%	13%	17%	16%	13%	14%
Sec.4: UI Specs (15 points)	14.2%	14.3%	14.3%	14.3%	14.3%	14.3%	14.3 %
Sec.5: Domain Analysis (25 points)	13.2%	17.3%	14.3%	18.3%	13.3%	13.3%	11.3 %
Sec.6: Plan of Work (5 points)	20 %	20 %	5 %	40 %	5 %	5 %	5 %

0.2 Responsibility Allocation Chart



1 CUSTOMER STATEMENT OF REQUIREMENTS

1.1 Problem Statement

Tabletop games have been a staple in entertainment even before the rise in technology. Nowadays, many favorites have been ported over to virtual systems and are still being enjoyed in their new form. The game should be modelled after these tabletop games, one such in mind is, *Monopoly*. Games like *Monopoly* are competitive by nature and allow for a user to be hooked into a game with other users. The turn-based system employed by these games naturally provides a moment of downtime for each user allowing the user to take in all the information they can and make the best decisions based on this information. The user should be able to join a queue and wait to be matched into a game against other users or be able to join a game with a group of friends through an invitation.

At the start of a game of *Rags to Riches*, each user should start with the same amount of capital so as to not provide any unfair advantages and be given background information on each of the companies relevant to the game. The game should be set in a detailed fantasy world with fantasy companies and products rather than our real world. This will build on the entertainment factor of the game as users will be dropped in an unknown world to explore.

During the game, the users should be able to perform market orders on the various companies available. The user should be able to make informed decisions based on their opponent's decisions as well as the market trend information provided to the user. Once a user has enough influence over a certain company, they should be able to influence the company into making decisions that have a chance of benefitting the user or hurting other users. There should also be a chat system within the game to allow for users to communicate and interact with one another.

If a user goes bankrupt and is completely out of money, they automatically lose and are removed from the game. If a user reaches the amount of money set as the threshold for winning, they are awarded the win for the game. Each user should be shown a quick overview of the game and any achievements they may have unlocked before the game exits.

Achievements should be stored on user accounts and can be utilized as a way of tracking the user's progress in learning the fundamentals. Once a user hits a certain milestone of progress, they should receive a reward that will unlock new abilities or help them in future games. The user should be able to see their progress towards their next milestone/reward and the rewards should be valuable enough that the user would want to reach for those milestones.

Along with being able to see their own achievements, the user should be able to see where they stack up against other players of the game as well as see their lifetime stats. Additionally, the user should be provided with basic account management options relevant to most applications. Administrative tools, such as banning users, should be available for those user's with the correct privileges. The user should have the option of integrating their social media accounts with the game so that they can quickly make posts showing off achievements, bragging about a win, etc. By being able to share to social media the user will be able to get validation of their successes from their peers and utilize that to further their progress through future games.

The user should be able to access *Rags to Riches* quickly and easily. With the prevalence of smartphones these days, it makes sense to design an app for the game so that the user is able to play the game with only the minimal number of strokes. As a result of the game being on an app, the user will be able to play the game on-the-go rather than being tied down to a computer. There is an estimated total of 107.7 million android users, by including the ability of sharing achievements via social media the app can become well known and easier to discover. With this increased ease-of-accessibility, *Rags to Riches* hopes to be an entertaining multiplayer turn-based game for the general public, those who do not have extensive knowledge of stock exchanges and investment management.

1.2 Glossary of Terms

Rags – A colloquial term for not having a lot of money. The user starts with “rags” and the objective is to make more money than their opponents.

Riches – A colloquial term for having a lot of money. A user turning their starting capital into “riches” is the objective of the game.

User Groups:

- **Player** – A basic user who participates in a game and has control over their account settings.
- **Opponent –** The opposing user who participates in a game. An “opponent” is also a “player”.
- **Administrator** – A user with additional privileges allowing for the management of basic users, including banning users from the game.

Game – A multiplayer turn-based game set in a fantasy world where the players manage their investments with the end goal of turning their “rags” into “riches”.

Turn – The time where a player can perform actions. Players are able to manage their investments during this time. Opponents can not perform actions at this time.

Bankrupt – A player has no more money remaining. When bankruptcy occurs, the player loses and is removed from the game.

Win – A player successfully turned their “rags” into “riches”. The player reached the threshold set for winning the game.

Loss – A player went into bankruptcy or another player won the game.

Capital – The amount of money each player starts with at the beginning of a game.

Stock – A type of security that signifies ownership in a corporation and represents a claim on part of the corporation's earnings and assets.

- **Ask Price** – Price at which trader will sell a stock
- **Bid Price –** Price at which a trader will buy a stock

Portfolio – An account of all the assets associated with a player in the game. Each player will have their own portfolio.

Investment – The purchase of stocks that are not to be consumed immediately but rather to be used in the future to create wealth.

Order – An investor must place an order to buy or sell an asset.

- **Buy** – An order to purchase an amount of stock.
- **Sell** – An order to sell an amount of stock.
- **Short** – A sell order performed using borrowed stocks.
- **Cover** – A buy order performed to return previously loaned stocks.
- **Limit** – An order that sets the maximum or minimum at which you can buy or sell stocks.
- **Stop** – An order that will only happen at a defined price called the stop price.

Influence – The greater the amount of stocks and percentage a user has in a company, the greater influence they have. Having a large influence in a company provides the user with a greater variety of decisions.

Decision – When it is the user's turn, they can make a decision. Decisions include but are not limited to the buying of stocks, selling of stocks, etc.

Milestone – A set goal that shows the player is making progress

Achievement – A milestone reached by the player resulting in a reward.

Reward – An item unlocked through an achievement.

2 SYSTEM REQUIREMENTS

2.1 Enumerated Functional Requirements

Identifier	Priority	Requirement
REQ-1	5	The system will allow new users to register accounts.
REQ-2	5	The system will allow returning users to login to their accounts.
REQ-3	5	The system will keep track of user account information.
REQ-4	3	The system will allow users to manage their account: ● Change password ● Change contact info ● Notification settings
REQ-5	1	The system will allow users to check their lifetime stats and achievements.
REQ-6	3	The system will allow administrators to manage user accounts.
REQ-7	4	The system will provide an initial tutorial interface for newly registered users.
REQ-8	4	The system will create a game based on a variety of settings.
REQ-9	4	The system will allow players to host a private game with custom settings: ● Time allowed per turn ● Starting capital ● Threshold for winning
REQ-10	4	The system will allow players to invite friends to their private games using a 4-character passcode.
REQ-11	4	The system will allow players to join a private game with their friends.
REQ-12	2	The system will allow players to join a queue for a public game with other players using default settings.
REQ-13	4	The system will allow players to view various game-related information, such as standings, player portfolios, and ownership percentage.

REQ-14	5	The system will automatically determine market data for the game.
REQ-15	5	The system will allow users to initiate market orders: ● Buy ● Sell ● Short ● Cover ● Limit ● Stop
REQ-16	3	The system will allow players to influence companies once they own a certain percentage of the company.
REQ-17	1	The system will allow players to communicate with each other inside of a game.
REQ-18	4	The system will recognize win and loss scenarios for the game.
REQ-19	1	The system will allow users to integrate their social media accounts to post messages to.
REQ-20	1	The system will provide a leaderboard for users to check how they stack up against other players.

2.2 Enumerated Nonfunctional Requirements

Identifier	**Priority**	**Requirement**
REQ-21	5	The system will be able to run on Android devices.
REQ-22	4	The system will be lightweight to provide fast performance even on low end devices.
REQ-23	3	The system will have a similar theme across the stock information page, the settings page, and the game page.
REQ-24	4	The system will securely store personal user information.
REQ-25	5	The system will store all data and information in a database with no storage being done on the user's device.
REQ-26	3	The system will allow the user to navigate the app in the fewest number of strokes possible.
REQ-27	3	The system will be active 24/7.

2.3 On Screen Appearance Requirements

Identifier	Priority	Requirement
REQ-28	3	**Initial Landing page** – This is shown on first boot. The game and its terms are explained here.
REQ-29	5	**Registration page** – Users will be able to create a new account.
REQ-30	5	**Login page** – Users will be able to login to an existing account
REQ-31	4	**Home Page** – Users will be able to create a game, join a game through an invite, or find an online game. They will also be able to manage their settings and achievements.
REQ-32	4	**Settings page** – Users will be able to change their password, their contact info, their notification settings, and volume.
REQ-33	4	**Create A Private Game page** – User can create a private game with custom settings: time allowed per turn, starting capital, threshold for winning, etc.
REQ-34	3	**Join A Private Game page** – User can enter a code given to them by a friend to join an existing private game.
REQ-35	5	**Game Page** – Users will be able to see their current assets, company information, current game standings, current market trends, each player's wealth, and a button to move on to the next player's turn.

3 FUNCTIONAL REQUIREMENTS SPECIFICATION

3.1 Stakeholders

A stakeholder is defined as a party that has an interest in a company, and can either affect or be affected by the business. For the majority of the time, the primary stakeholders in a typical corporation are its investors, employees, and customers. Stakeholders can be internal or external. Internal stakeholders are those parties whose interest in a company is tied to a direct relationship, such as employment, ownership, or investment. On the other hand, an external stakeholder are those who are not directly tied to the company yet are affected by the company's actions and business outcomes.

The target audience for Rags to Riches is students, novice investors, and anyone who wishes to gain a working knowledge of economics and the stock market. Even those with a greater understanding of the stock market who wish to sharpen and hone their skills and review some of the conceptual background can benefit from using this app. Users who wish to eventually dive in the real stock market will have benefitted from the utilization of the app and be armed with an experiential advantage over those who did not. Rags to Riches can be expanded to educate larger swaths of aspiring investors in the form of being used as an educational resource in high school and college level economic courses. The various features than the app offers will likely garner word of mouth awareness from those who use the app, who can encourage like minded individuals to also experiment around with it.

Rags to Riches is a non-profit, free application with the sole purpose of education. At no point are there plans to incorporate advertisements to generate revenue. Rags to Riches believes that advertisements are distracting and cumbersome, which detract from the user's experience. Because the app's sole purpose is that of education, cluttering the user's interface with ads will distract them from the best experience that can be provided, and perhaps hamper their overall education. Another strategy Rags to Riches strongly opposes is the inclusion of in app purchases or microtransactions. These "pay to win" strategies are potentially game breaking, unfair, and lessen the overall experience. Rags to Riches strives to provide a equal, balanced, enjoyable, and educational experience to all users.

3.2 Actors and Goals

- **Player** – Initiating Actor
  Goals:
  1. To create an account
  2. To login into their account
  3. To create a private game
  4. To find and join a public game
  5. To check their achievements
  6. To perform market transactions in their game
  7. To view the portfolios and balances of players in their game
  8. To view the standings of their game
  9. To communicate with other players in their game
  10. To post achievements or progress updates to their social media account

- **Game Administrator** – Initiating Actor
  Goals:
  1. To set the initial rules for the private game they create
  2. To manage the private game they create

- **Game** – Participating Actor

- **Stock Simulator** – Participating Actor

- **Account Database** – Participating Actor

3.3 Use Cases

3.3.1 Casual Description

UC-1: Register

Allows a new user to register an account for the application. The user will be required to login using this information in the future.
Requirements: REQ-1, REQ-3, REQ-7

UC-2: Login

Allows a registered user to login to their account for the application and link to social media. The system will retrieve all of the user's information and personal settings. The user will be automatically logged in to their account unless they have manually logged out.
Requirements: REQ-2, REQ-3

UC-3: ManageAccount

Allows a registered user to edit their account information, such as account password and email address, as well as manage their application settings and view their achievements.
Requirements: REQ-4

UC-4: FindPublicGame

Allows a registered user to join a queue for a public game. Once four players are found in the queue, they players are removed from the queue and placed in a game.
Requirements: REQ-8, REQ-12

UC-5: CreatePrivateGame

Allows a registered user to create a private game to be played with friends. A 4 character passcode is generated to invite other users to the game.
Requirements: REQ-8, REQ-9, REQ-10

UC-6: JoinPrivateGame

Allows a registered user to join a private game hosted by a friend by entering the 4-character code sent to them.
Requirements: REQ-10, REQ-11

UC-7: ViewMarketData

Allows a player to view information regarding the various companies and their market history to aid them in their decision-making process.
Requirements: REQ-13, REQ-14

UC-8: PlaceMarketOrder

Allows a player to place a market order to buy or sell stocks in the game. The order is placed during the user's turn and changes are reflected immediately.
Requirements: REQ-14, REQ-15

UC-9: ViewPortfolio

Allows a player to view the current assets of any player in the game. The user is able to see information regarding other user's influence in companies to aid them in their decision-making process. User is able to chat with other players if they wish to communicate with those players.
Requirements: REQ-13, REQ-14

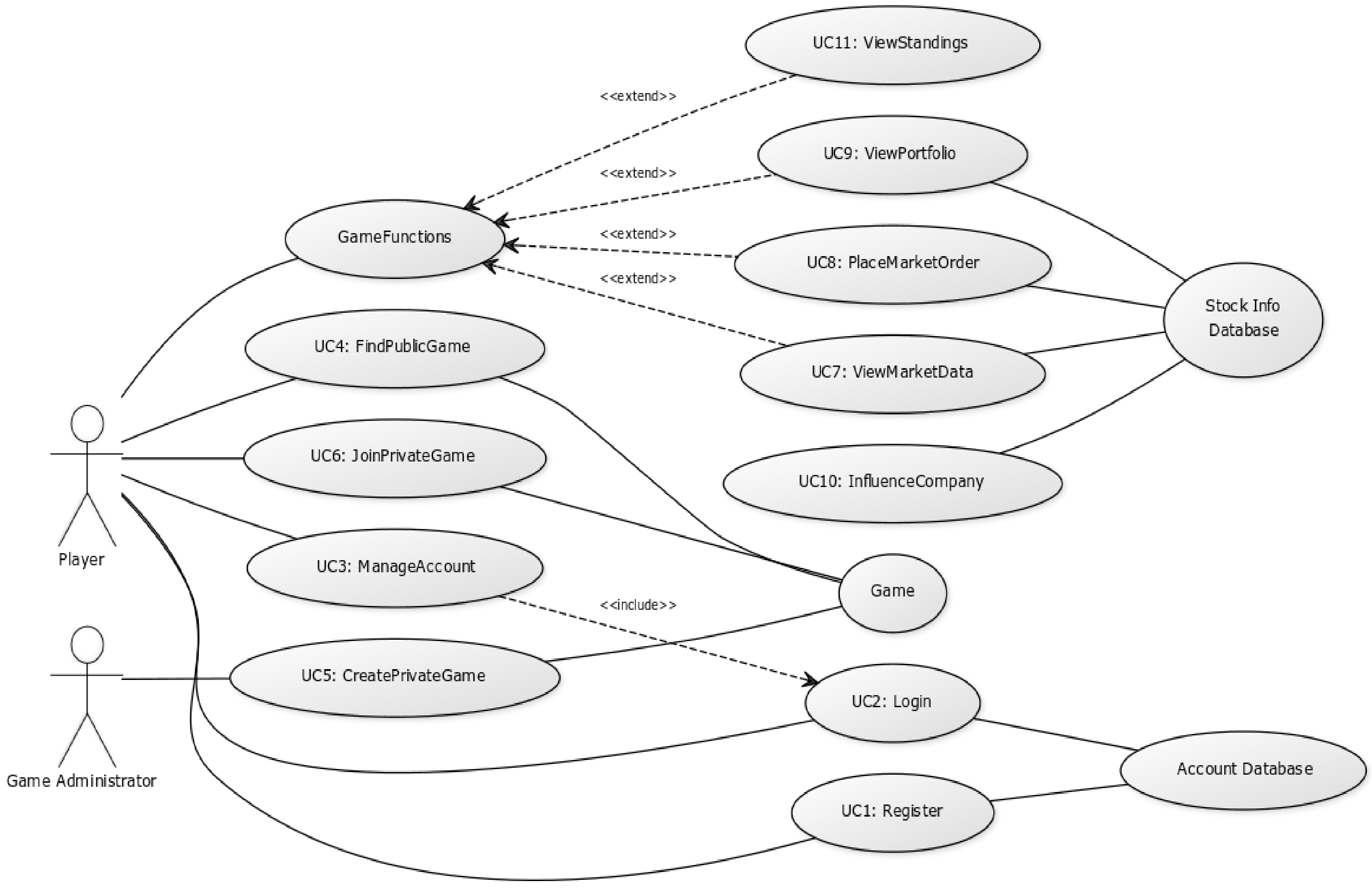
UC-10: InfluenceCompany

Allows a player to influence a company once they own a certain percentage of the company. The user is able to perform additional actions based on this.
Requirements: REQ-13, REQ-14, REQ-16

UC-11: ViewStandings

Allows a player to view the current standings of the players in the game based on net worth. The user is also able to see a brief overview of another player's assets.
Requirements: REQ-13

3.3.2 Use Case Diagram



3.3.3 Traceability Matrix

REQ	**PW**	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8	UC9	UC10	UC11
REQ-1	**5**	**X**										
REQ-2	**5**		**X**									
REQ-3	**5**	**X**	**X**									
REQ-4	**3**			**X**								
REQ-5	**1**											
REQ-6	**3**											
REQ-7	**4**	**X**										
REQ-8	**4**				**X**	**X**						
REQ-9	**4**					**X**						
REQ-10	**4**					**X**	**X**					
REQ-11	**4**						**X**					
REQ-12	**2**				**X**							
REQ-13	**5**							**X**		**X**	**X**	**X**
REQ-14	**4**							**X**	**X**	**X**		**X**
REQ-15	**5**								**X**			
REQ-16	**3**											**X**
REQ-17	**1**											
REQ-18	**4**											
REQ-19	**1**											
REQ-20	**1**											
Max PW		**5**	**5**	**3**	**4**	**4**	**4**	**5**	**5**	**5**	**4**	**5**
Total PW		**14**	**10**	**3**	**6**	**12**	**8**	**9**	**10**	**9**	**4**	**12**

3.3.2 Fully-Dressed Description

Use Case UC-1: Register	
Related Requirements:	REQ-1, REQ-3, REQ-7
Initiating Actor:	Player
Actor's Goal:	To create an account.
Participating Actors:	Account Database
Preconditions:	The user is not logged into an account.
Postconditions:	The user successfully registers an account and can use it to log in to the app from any compatible device. The user account database is updated with the new player's account information.
Flow of Events for Main Success Scenario:	
1.	-> The user is on the initial landing page.
2.	-> On the initial landing page, the player clicks the register option
3.	-> The player enters their desired account name, their password, and their password again to confirm it.
4.	<- If the user confirmed their password correctly, the user account database is updated with the new information and the system welcomes the new user.
Flow of Events for Account Name is taken or Passwords do not match:	
1a.	-> The user enters an account name that is already taken.
1b.	-> The user's password does not match the confirmation password.
2.	<- The user is provided with an error message explaining the situation and is encouraged to try again.

Use Case UC-5: FindPublicGame	
Related Requirements:	REQ-8, REQ-12
Initiating Actor:	Player
Actor’s Goal:	To find and join a public game
Participating Actors:	Player, additional players, Queue system
Preconditions:	Players have valid accounts. Queue system functioning.
Postconditions:	Player is successfully processed through the queue and is placed into a public game against other online players.
Flow of Events for Main Success Scenario:	
1.	-> Player clicks the find public game option on the main screen.
2.	<- Player enters the queue that sorts players into online games on “first-come, first-served” basis.
3.	<- After being successfully processed through the queue, the player is placed into a public game where they play against other online players.
Flow of Events for Extensions (Alternate Scenarios): Queue System Down	
1.	-> Player clicks the find public game option on the main screen.
2.	<- The queue system is down or overloaded at the moment and is unable to place the player into a public game. The player receives an error message apologizing and explaining the situation.
The arrows on the left indicate the direction of interaction: -> Actor’s action; <- System’s reaction	

Use Case UC-6: CreatePrivateGame	
Related Requirements:	REQ-8, REQ-9, REQ-10
Initiating Actor:	Game Administrator
Actor’s Goal:	To create a private game
Participating Actors:	Game Administrator and players
Preconditions:	The game administrator has a valid account.
Postconditions:	The game presents a 4 digit code that the game administrator can share with other players who then use the code to join the private game.
Flow of Events for Main Success Scenario:	
1.	-> Game administrator creates a game from the home screen.
2.	-> Game administrator changes the settings of the game, such as number of players and starting currency amount and chooses the private game option.
3.	<- Game presents the 4 digit code that the game administrator can share.
Flow of Events for Different Sharing Method:	
1.	-> Game administrator creates a game from the home screen.
2.	-> Game administrator changes the settings of the game, such as number of players and starting currency amount and chooses the private game option.
3a.	<- Game administrator chooses the share the 4 digit code by using a social media such as Twitter or facebook and system shares it through this medium.

Use Case UC-8: PlaceMarketOrder	
Related Requirements:	REQ-14, REQ-15
Initiating Actor:	Player
Actor's Goal:	To perform market transactions in their game
Participating Actors:	Players, Stock Information Database
Preconditions:	Each player that has a valid account has its own capital to buy stocks.
Postconditions:	Each player is presented with the option of buying more stocks if capital is available. Also, each player is able to sell their current stocks in order to buy more. Also, given the other market orders, each player can perform a short, cover, limit, and stop during each turn in the game.
Flow of Events for Main Success Scenario:	
1.	-> Player with valid account enters new game, given a capital, and time limit for each turn.
2.	-> Each player in the game at their respective turn with their given capital chooses to place a market order within their given time limit for their turn.
3.	<- The system performs the market transaction and updates the information on the player's screen.
Flow of Events for Market Transaction Error:	
1.	-> Player tries to place a market order that exceeds the amount of money they have.
2.	<- The system can not perform this transaction and returns an error.

Use Case UC-9: ViewPortfolio	
Related Requirements:	REQ-13, REQ-14
Initiating Actor:	Player
Actor’s Goal:	To view the portfolios of players in their game.
Participating Actors:	Player, Desired Player in current game, Account Database
Preconditions:	The player and desired players both have valid accounts and are both present in the same current game session
Postconditions:	The player is presented with the portfolio of the player they desired to view in their current game, which displays their information about their current stocks, achievements, and statistics.
Flow of Events for Main Success Scenario:	
1.	-> Player enters a valid game session
2.	-> Player clicks on their desired player in a session, and clicks on “View Portfolio”.
3.	<- The player is presented with the portfolio of their chosen player. When they are finished viewing it, they press the x icon in the top right.
Flow of Events for Account Database Malfunction:	
1.	-> Player enters a valid game session
2.	-> User clicks on desired player and then upon “View Portfolio”
3.	<- The Account Database is down and cannot bring up that player’s portfolio information. The player receives a message of the error

3.4 System Sequence Diagrams

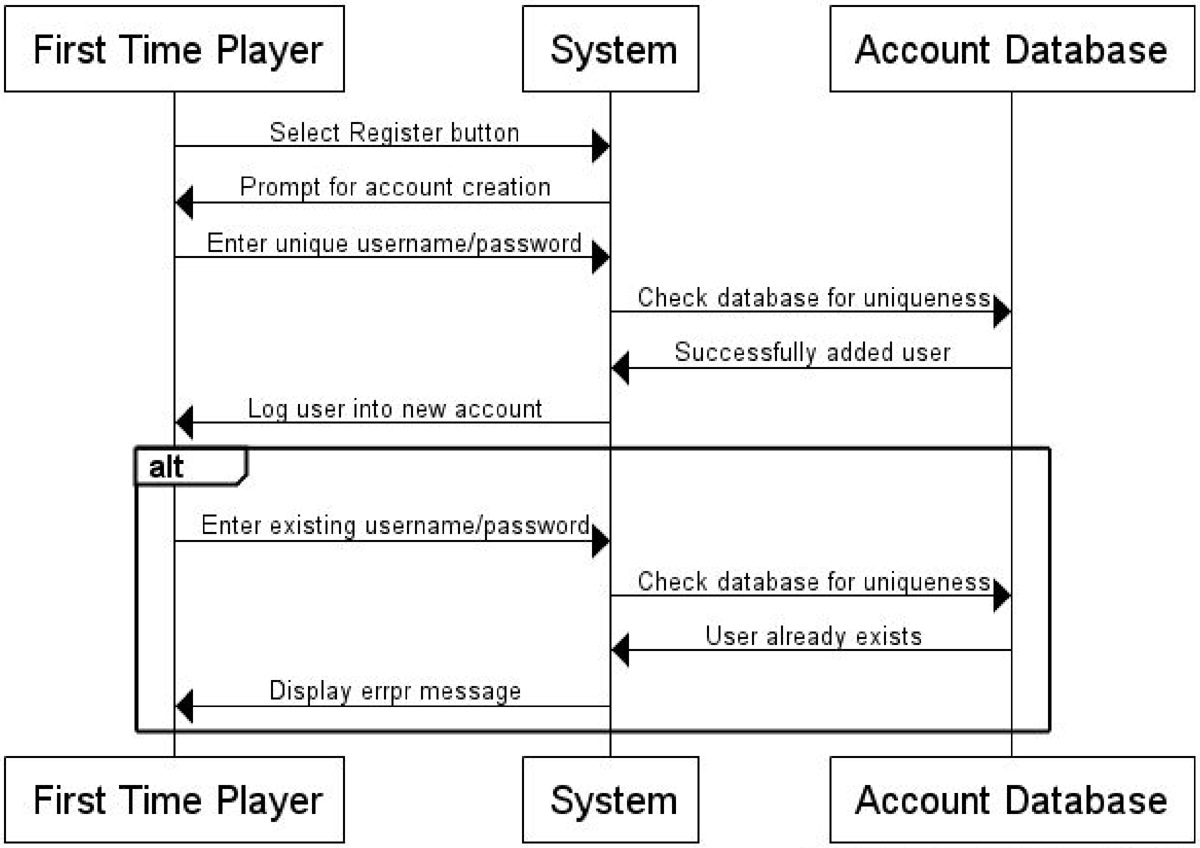

First Time Player
System
Account Database
Select Register button
Prompt for account creation
Enter unique username/password
Check database for uniqueness
Successfully added user
Log user into new account
alt
Enter existing username/password
Check database for uniqueness
User already exists
Display errpr message
First Time Player
System
Account Database

Figure 3.1: UC-1 Register

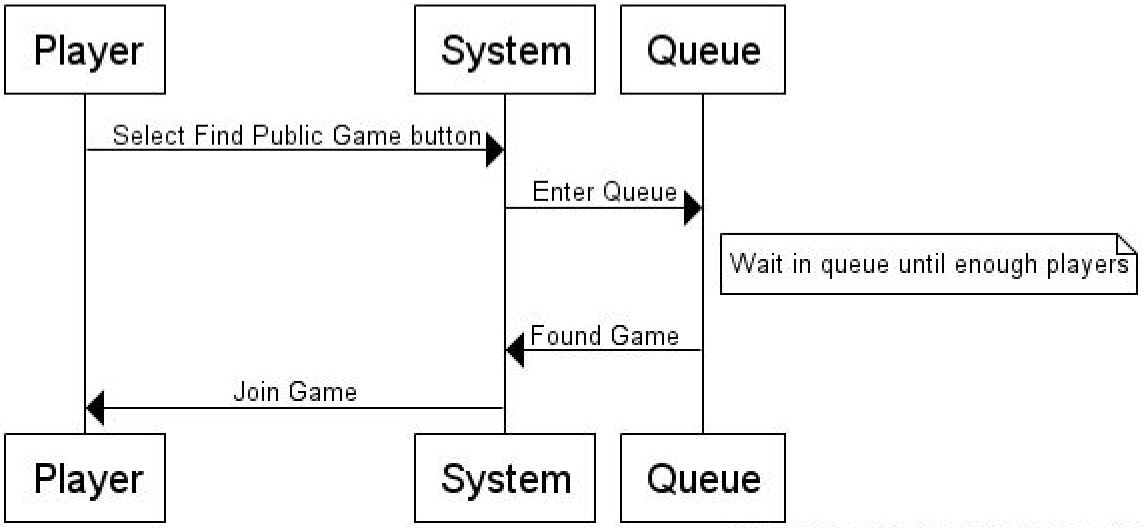

Player
System
Queue
Select Find Public Game button
Enter Queue
Wait in queue until enough players
Found Game
Join Game
Player
System
Queue

Figure 3.2: UC-5 FindPublicGame

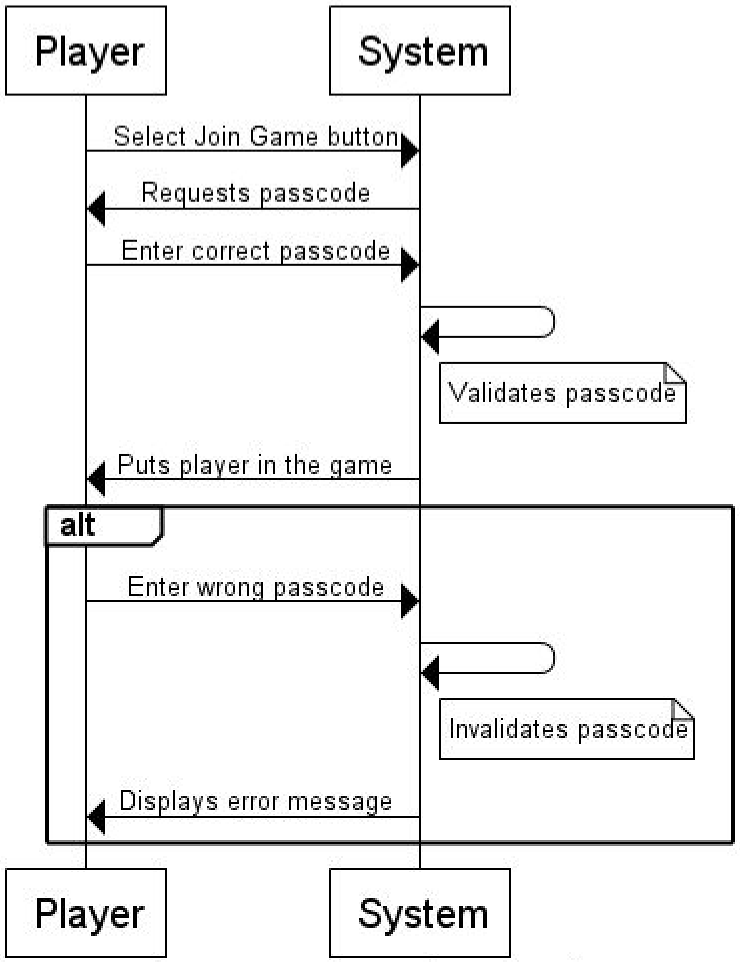


Figure 3.3: UC-6 CreatePrivateGame

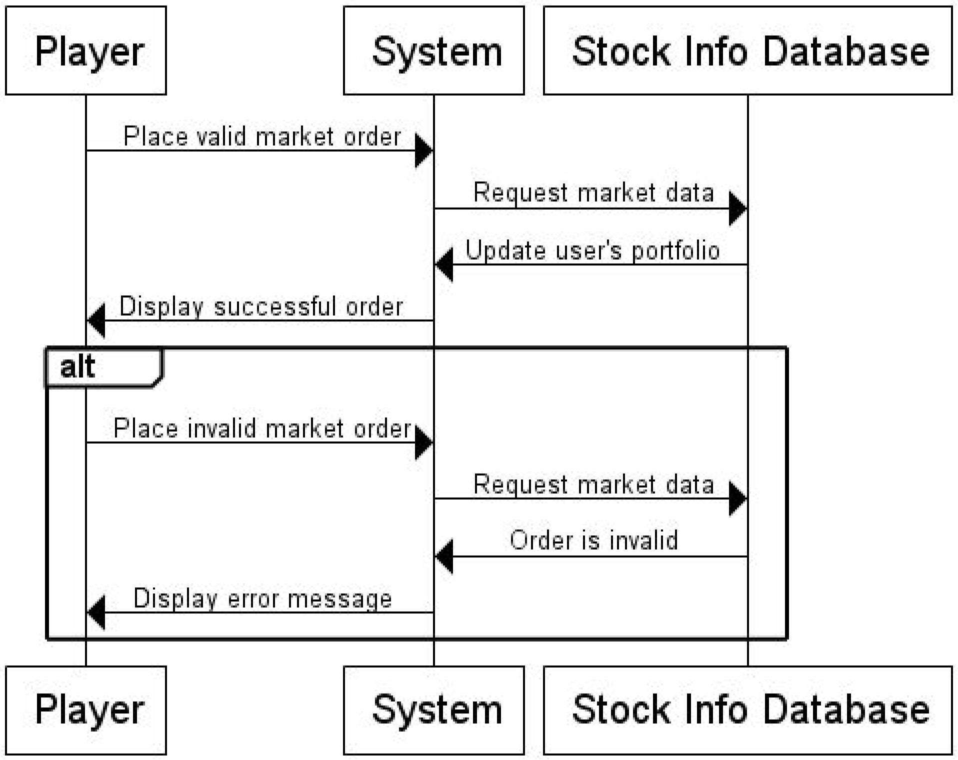


Figure 3.4: UC-8 PlaceMarketOrder

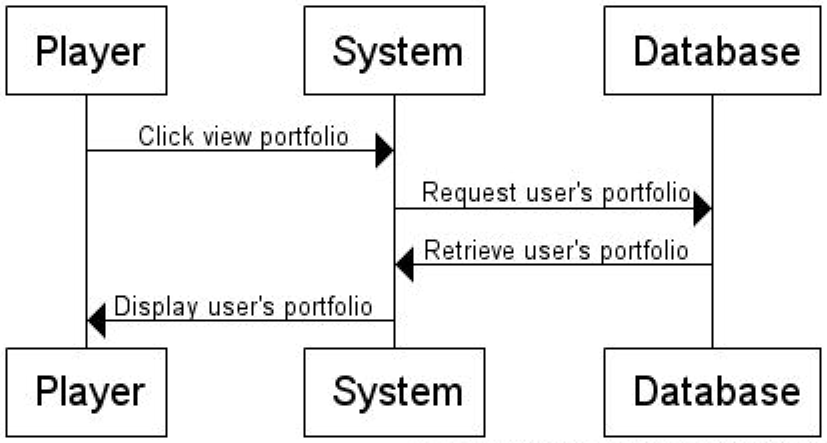


Figure 3.5: UC-9 ViewPortfolio

4 USER INTERFACE SPECIFICATION

4.1 Preliminary Design

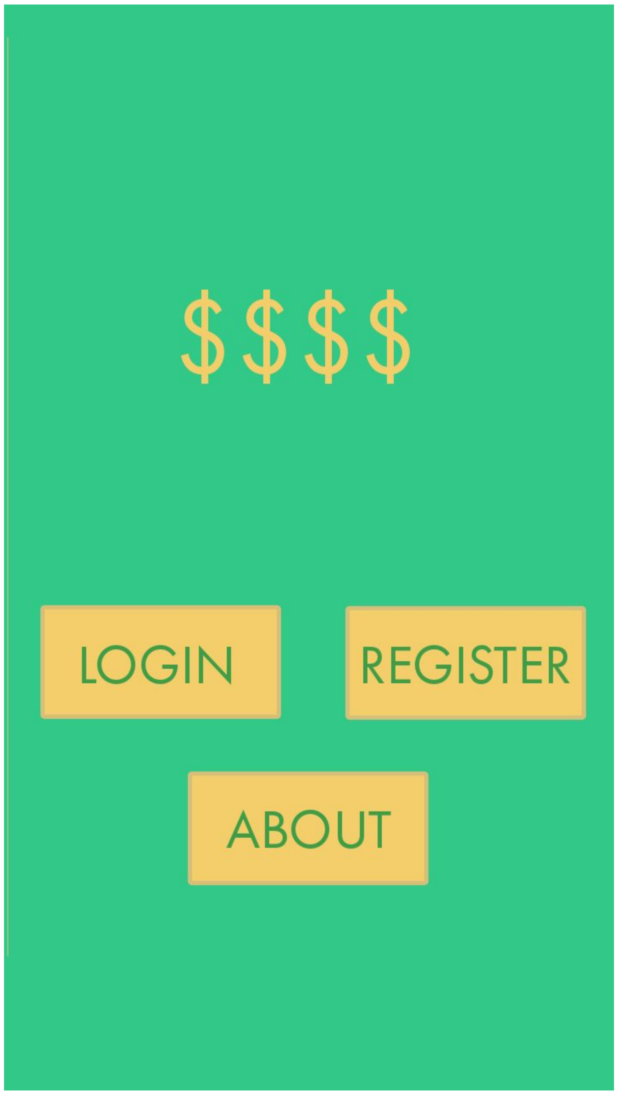


Figure 4.1: Initial Landing Page

Pressing login brings you to login page.
Register brings you to the register page.
About shows you terms and conditions and privacy policy.

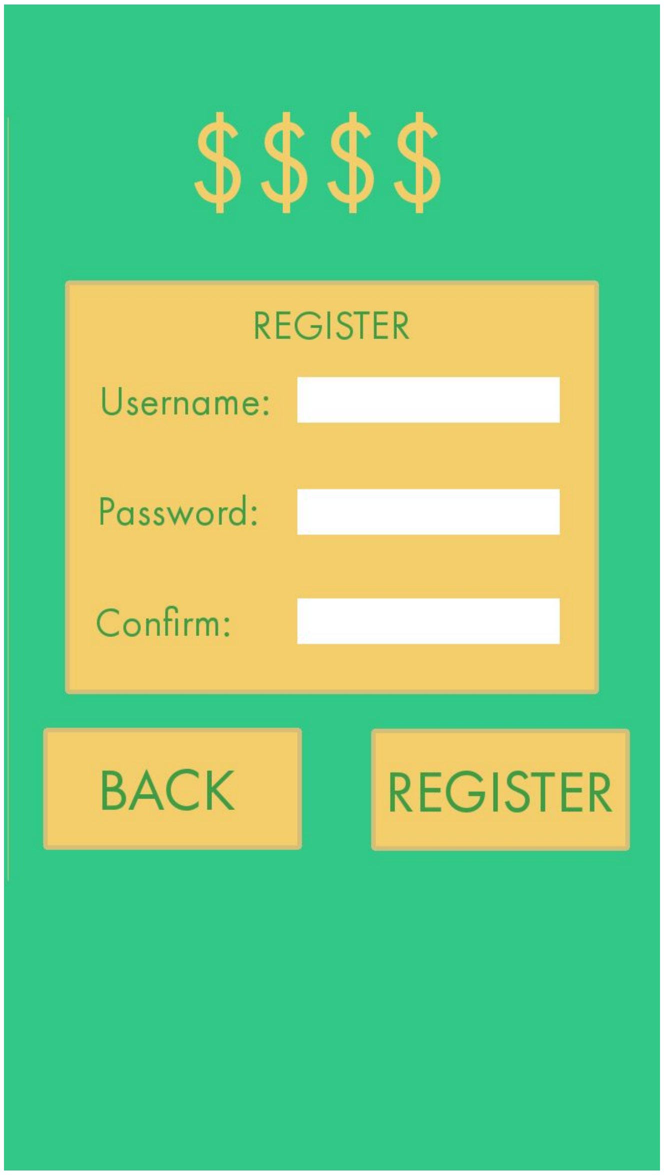


Figure 4.2: Register Page

Enter a username and password in the fields.
If password and confirm are the same, the “Register” button registers the user.
Back returns to start.

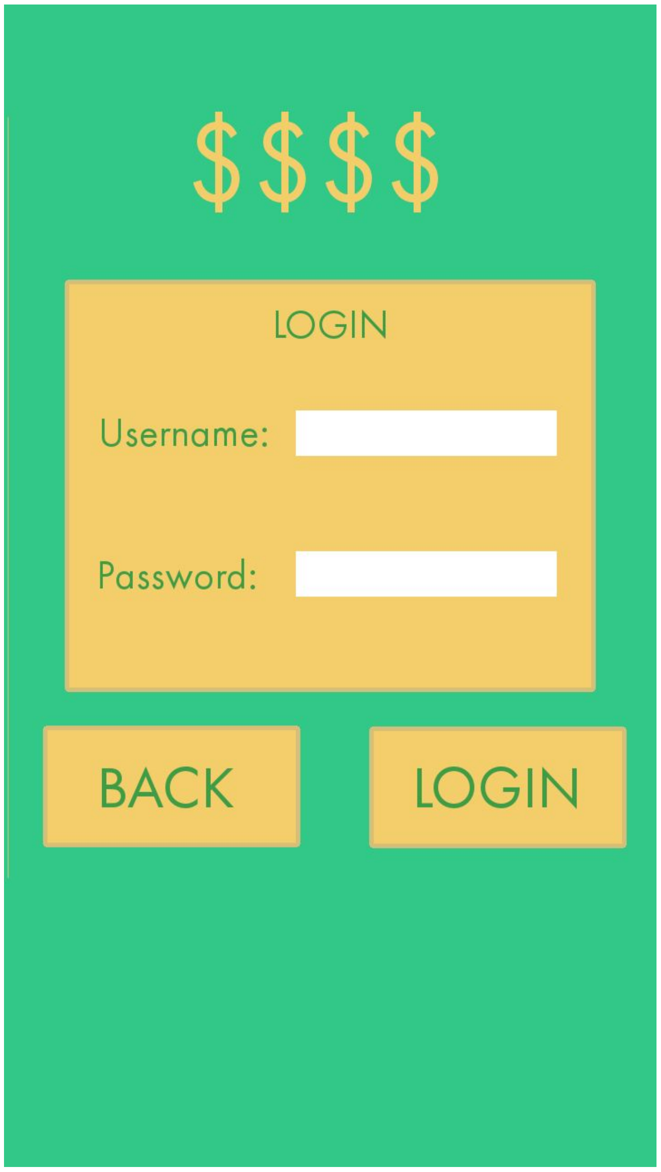


Figure 4.3: Login Page

Enter a username and password in the fields.
Pressing login either logs you in if valid or displays error if not.
Back returns you to start screen.

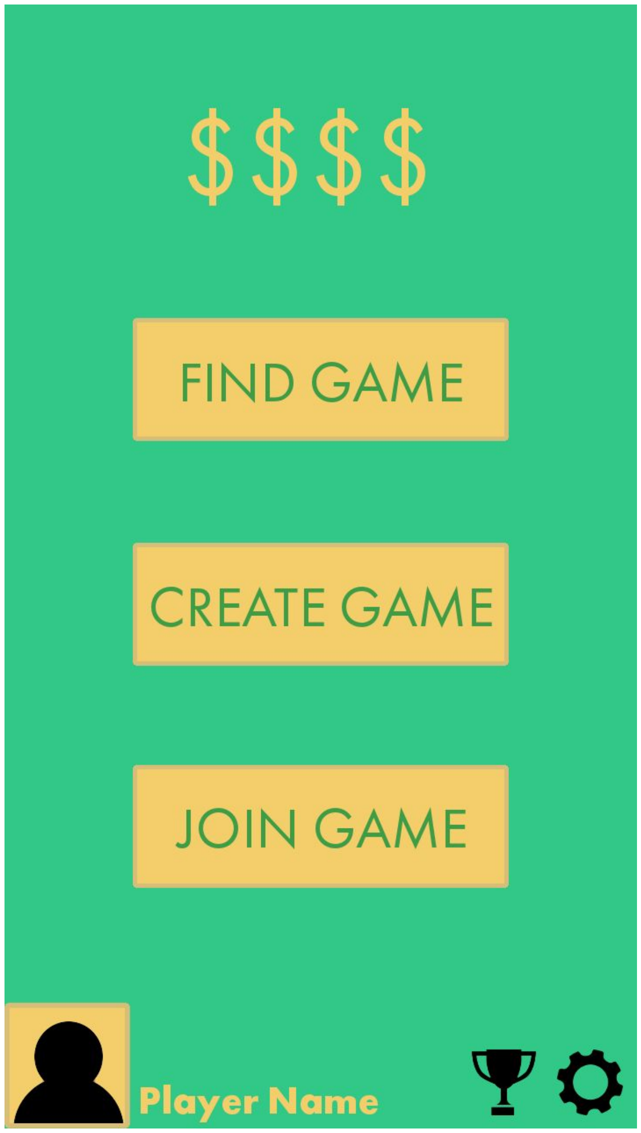


Figure 4.4: Home Page

Gives users the option to Find game, Create game, and Join game

Or click on the icons to go to achievements or settings.

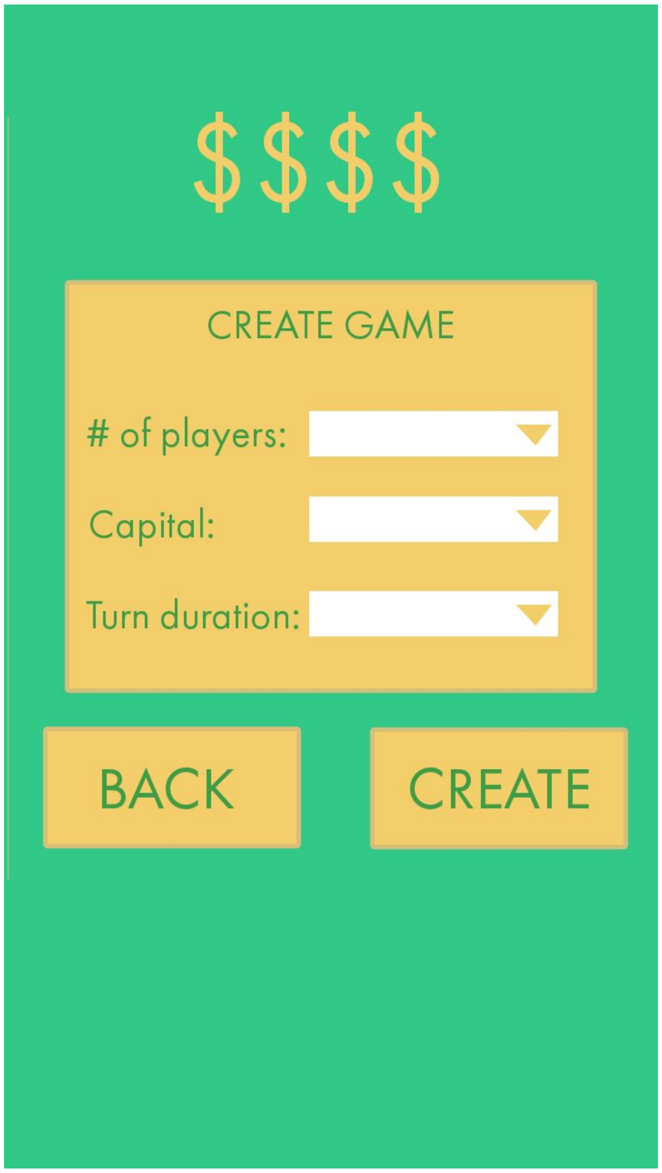


Figure 4.5: Create Game Page

Allows users to select 1-4 players to join
Capital and game duration can also be selected to creator's choice
Hitting "Create" button will prompted creator with a four digit code to share

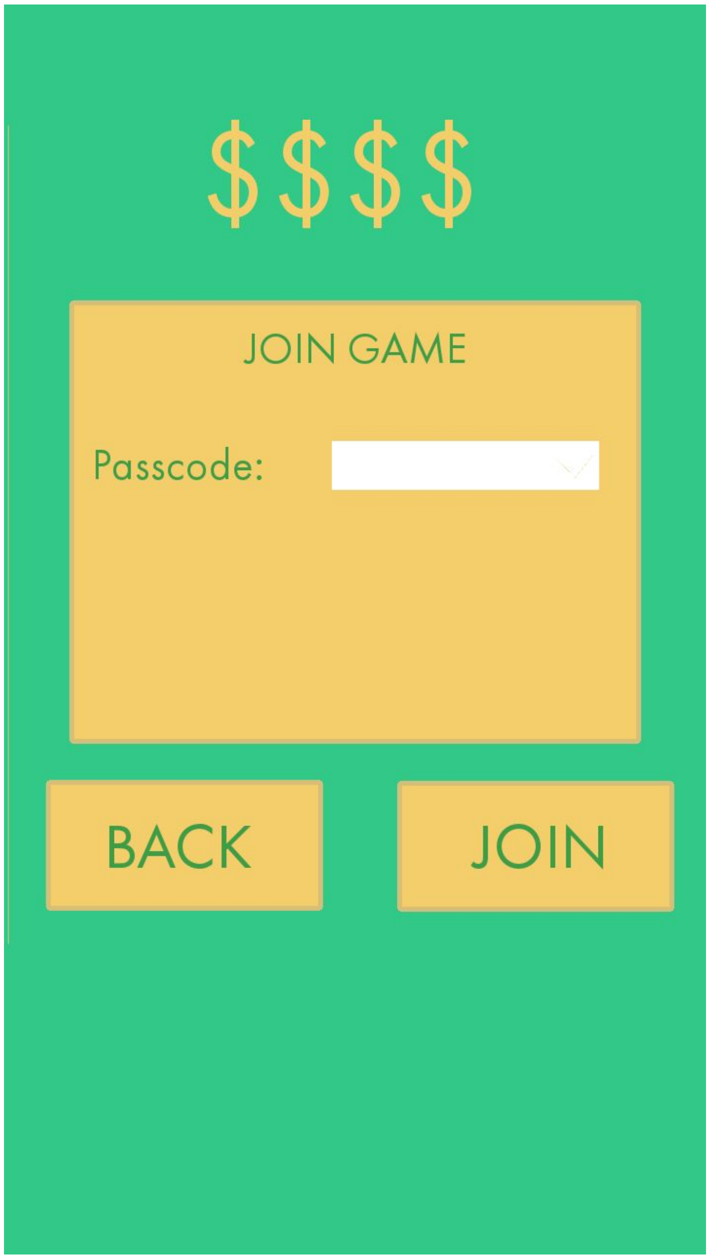


Figure 4.6: Initial Landing Page

The user is asked to enter a passcode in order to join a game that was created by another user. Once passcode is entered the user is prompted to the specific game given the passcode.

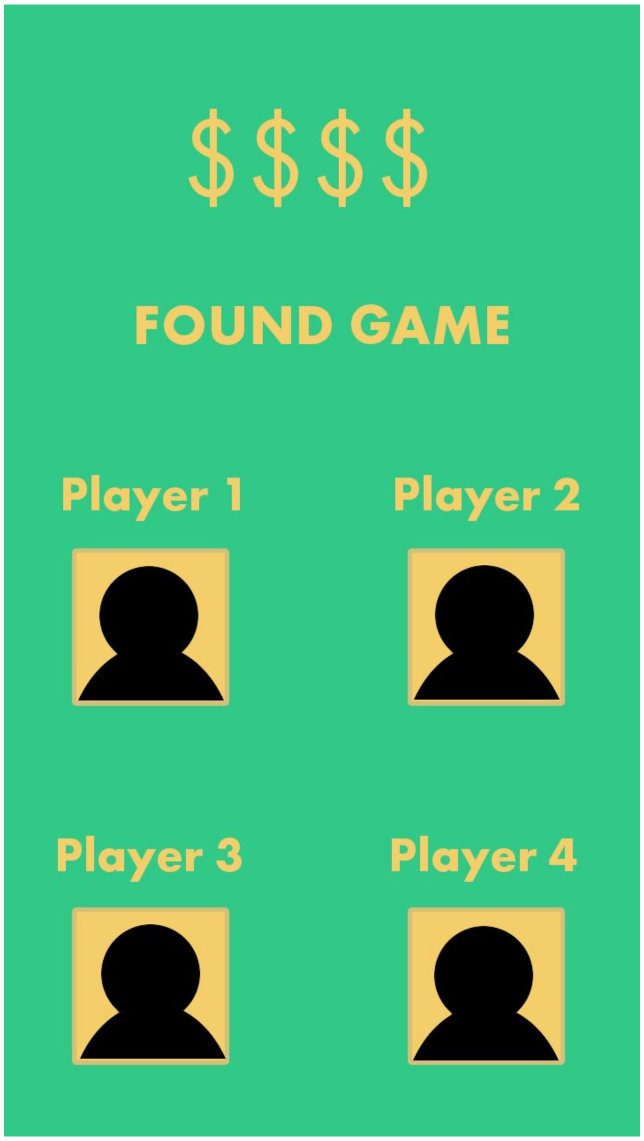


Figure 4.7: Found Game Page

After selecting the option, to join a public game, the user will be processed through a queue. After a short time, this screen will appear, indicating that the player has successfully been processed, other online players have been found, and the public will commence shortly. The player's names and pictures are displayed here as well.

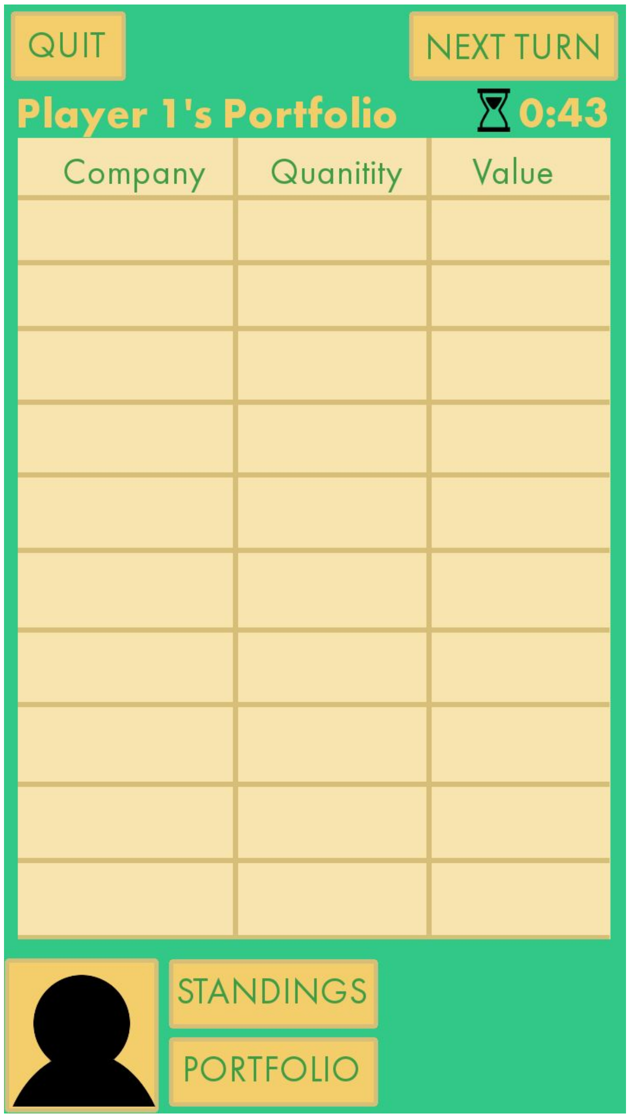


Figure 4.8: View Portfolio Page

While in a public or private game, the player can view their own portfolio or an opponent's portfolio. The above screen is shown, displaying all of the player's stocks, quantities, and values that they own. They can also view the player standings. They can return at any point by clicking the top left, a timer is present allowing them to know when the turn is finished.

4.2 User Effort Estimation

A) Register

Login Page

a) Press “Register” button
b) Fill out text boxes with correct information
c) Press “Register” button to submit (will bring you to main menu)

Data Entry

a) Press on “Username” text field
b) Press any keys for username (6-8 characters)
c) Press the enter key to move to the next text field
d) Press any keys for password (6-8 characters)
e) Press the enter key to move to the next text field
f) Press the same keys as previous text field to confirm password (6-8 characters)

B) Log In

Login Page

a) Press on “Login” button
b) Enter Username and Password
c) Click “Login” button to enter main menu

Data Entry

a) Press on “Username” text field
b) Press any keys for username (6-8 characters)
c) Press the enter key to move to the next text field
d) Press any keys for password (6-8 characters)

C) Create a Private Game

Main Menu

a) Press “Create Game” in the middle of the main menu
b) Enter specifications, number of players, starting currency
c) A screen with a four digit number will be prompted
d) Select sharing method (i.e Facebook. Contacts, private message)
e) Select Start game

Data Entry

a) Press on “# of players” text field
b) Press one key (numbers 1-4)
c) Press the enter key to move to the next text field
d) Enter up to 6 numbers(6 digits)
e) Enter turn duration

D) Join Public Game

Main Menu

a) Click “Join Public Game”, at the bottom of the page in main menu
b) Player is placed into queue that will process them and place them into an online game

E) Place Market Orders

In Game

a) Press desired stock
b) A page will be prompted showing stock trends
c) At the bottom left select buy
d) Select amount of shares, and press the confirm button

F) Manage Account

Main Menu

a) Click on Settings "gear" at bottom right of home screen
b) Two prompts will appear-:
   1) Change username - enter old username, new desired username
   2) Change password - enter old password, enter new password twice

Data Entry

a) Press on "Username" text field
b) Press specific keys for current username (6-8 characters)
c) Press the enter key to move to the next text field
d) Press any keys for new user name (6-8 characters)
e) Press the enter key to move to the next text field
f) Press specific keys for password (6-8 characters)
g) Press the enter key to move to the next text field
h) Press the any keys for new password (6-8 characters)
i) Press the enter key to move to the next text field
j) Press the same keys as previous text field to confirm password (6-8 characters)

G) View Portfolio

In Game

a) Press on avatar, or an opponent's' avatar
b) New screen will be prompted, select View Portfolio". This brings up information about the clicked upon user. (Current stocks, achievements, statistics)

5 Domain Analysis

5.1 Domain Model

Responsibilities Description	Concept Name
Allows a new user to register for an account. The user will be required to login using this information in the future.	Login Controller
Allows a registered user to login to their account. The system will retrieve all of the user's information and personal settings. The user will be automatically be logged in with their account unless they have manually logged out.	Login Controller
Displays login screen to the user and allow them to register or access their account.	Login View
Container for user's account data, such as their login information, match settings, and account settings.	Account
Allows a registered user to edit their account information, such as account password and email address, as well as manage their application settings.	Account Controller
Allows a user to check their achievements and rewards.	Account Controller
Displays option to the user when logged-in to their account but not in a game yet.	Account View
Container to hold a queue of users searching for games	Game Queue
Allows a registered user to join a queue for a public game. Once four players are found in the queue, they players are removed from the queue and placed in a game.	Public Game Controller
Allows a registered user to create a private game to be played with friends. A 4 character passcode is generated to invite other users to the game.	Private Game Controller
Allows a registered user to join a private game hosted by a friend by entering the 4-character passcode sent to them via other users.	Private Game Controller

Container to hold game-related information, such as game settings, game players, and game stats.	Game
Display relevant game information to the user in an organized and easy manner.	Game View
Allows a player to view information regarding the various companies and their market history to aid them in their decision-making process to place a market order..	Game Controller
Prepare a database query to perform the given stock transaction and update the game values as the game moves on.	Database Connection
Allows a player to view the current assets of another player in the game. The user is able to see information regarding other user's influence in companies to aid them in their decision-making process to place a market order.	Game Controller
Allows a player to view the current standings of the players in the game based on net worth. The user is also able to see a brief overview of another player's assets.	Game Controller
Allows a player to influence a percentage of the company. The user is able to perform additional actions based on the influence.	Game Controller
Container to hold information related to a particular company in the game, such as the company description, history, current market value, market trends, etc.	Company
Container to hold information related to the user's portfolio, such as their assets.	Portfolio
Allows players to communicate with each other within the game.	Chat Controller
Container to hold information related to a user's social media account.	Social Media Account
Allows user to post messages onto their social media account(s).	Account Controller
Container to hold information related to a particular achievement.	Achievement
Allow a player to obtain achievements based on milestones in the game.	Achievement Checker

5.1.1 Concept Definitions

Login Controller
For anyone to access the application, they must have a registered account. When a user is prompted to input their email address, the login controller will check the database to see if there is an account that matches the email address. If so, the user will be asked if they forgot their password and direct them to a link to reset their password through email and another link to proceed to the login page. On the contrary, the user will proceed to create an account and the information will be stored in the database.

Login View
The Login View is the first page that will be seen by users of the application. The user will enter their information and a request will be sent to the Login controller for verification. If the data is verified then the home page will load. If not, the Login view will display an error message. Users who have not manually logged out from account settings will be directed to the home page when they restart or open the application.

Account
The Account is a container that has information regarding to the particular user's account.

Account View
The Account View will display information to the user while they are logged-in to their account, but not currently in a game. They will be able to find games, join or create games, see their achievements, view their settings, etc.

Account Controller
A logged-in user can manage their account settings or see any achievements they may have gotten. The user can also post messages on their social media.

Game Queue
The Game Queue is a container that has information regarding all players currently searching for a game.

Public Game Controller
Once a player has successfully logged in, one of the game options that will be displayed on the home page is Find Game. When the user chooses the Find Game option then they will be put into a queue. Once there are four players in the queue, the system creates a game.

Private Game Controller
Once a player has successfully logged in, one game option that will be displayed on the home page is Create Game. When the user chooses the Create Game options then another page loads prompting the game specifications such as number of players, captial and turn duration. The user

will input the requested information and the game will be created. A four digit passcode entry is prompted when Join Game is chosen. Upon successful entry, the player will be added into the game.

Game
The Game is a container that has information regarding a particular game being played.

Game View
The game view will be seen by any player in a game. It is where information regarding the game will be displayed and will allow for a multitude of functions that will be relevant to the game, such as placing market orders.

Game Controller
Handles all logic behind the game functions and will handle any queries made by the user on the Game View.

Database Connection
Necessary to access data on user accounts as well as stock simulation data used in the game. When data needs to be verified a request will be sent to the database to look for it. The database will also store information such as a company's market history and other primary information.

Chat Controller
Handles the chat system between players in a game.

Company Stock Controller
Information regarding a particular company in the game will be stored in this container, such as the company description, history, current market value, market trends, etc. Other information such as their stock value, or how much stock can be bought will also be here.

Portfolio
The Portfolios is a container that has information regarding a particular player in the game, such as assets.

Social Media Account
The Social Media Account is a container that has Information regarding a user's social media account.

Achievement
The Achievement is a container that has Information regarding a particular achievement.

Achievement Checker
Periodically checks if a user has hit a certain milestone and is to be awarded an achievement.

5.1.2 Association Definitions

Concept Pair	Association Description	Association name
Login View ↔ Login Controller	Login View passes on login information that needs to be verified by the Login Controller.	Send login data
Account View ↔ Account	The accounts view page requests information about a user which it retrieves from the Account containers.	Send User account information
Public Game Controller ↔ Game Queue	The game queue handles multiple players trying to enter a public game. This information is passed onto the public game controller.	Find Match
Game View ↔ Game	Information regarding a game is passed onto the game view.	Send game information
Game View ↔ Chat Controller	The chat controller sends chat data to the game view.	Send Chat Information
Game View ↔ Game Controller	Information about a game is sent to the game view.	View Game
Game ↔ Database controller	The database controller passes on primary information about a company to the game.	Send Company Info
Game View ↔ Portfolio	Information about players and the stocks they own is passed onto the game view.	Send Player Info
Achievement ↔ Achievement Checker	Information that an achievement has been accomplished is passed onto the achievement container.	Send achievement information
Account ↔ Social Media Account	Social media account passes on social media information about a user's to the accounts container.	Send account information
Game ↔ Game Controller	Information about stocks that are bought or sold is passed onto the game.	Send stock information. Update stock information. Send other players game information.

5.1.3 Attribution Definitions

Concept	Attribution	Attribution Definitions
login controller	login	If username and password input match, logs in, else return wrong password.
account controller	changePass	After confirming current password and making sure new password and confirm new password are the same, change password to new password.
public game controller	pblcontrolGame	Handles entering users into a public game
private game controller	createPvtGame, createPvtCode joinPvtGame	Controls the local game information and creates a 4 digit code that will be shared to other players that will be joining the private game. If entered code is correct, allows a user to join the private game,
chat controller	sendChatInformation	Users, chat statements. Sends chat information to the game
achievement checker	checkAchievement	Checks if an achievement is accomplished.
game controller	sellStock buyStock borrowStock updatepricedata viewRankings viewPlayerInfo viewCompanyInfo influenceCompany	Decreases the stock in user assets and user gains in game currency equal to the value of the stock. Decreases user currency and gain stock amount equal to the price. Gain currency equal the the value of the stock borrowed, but lose currency equal the the value of the stock borrowed after a period of time. Request updated stock price data from database. Show the players ranked by how much currency they would have if they liquidated all their assets. Show how much money and what stocks a player has. View the current information, profile, price, ownership, background, history and more of a company. User Influences a company causing repercussions in future prices.

5.1.4 Traceability Matrix

		Domain Concepts																	
Use Case	**PW**	Login Controller	login view	Account	Account View	Account Controller	Game Queue	Public Game Controlle	Private Game Controller	Game	Game View	Database Connection	Game Controller	Chat Controller	Company	Portfolio	Social Media Account	Achievement	Achievement Checker
UC-1	14	✓		✓		✓											✓		
UC-2	10	✓	✓	✓													✓		
UC-3	3			✓	✓	✓											✓	✓	✓
UC-4	6						✓	✓											
UC-5	12								✓										
UC-6	8								✓										
UC-7	9									✓	✓	✓	✓		✓				
UC-8	10									✓	✓		✓		✓	✓			
UC-9	9									✓	✓		✓	✓		✓			
UC-10	4									✓	✓		✓	✓		✓			
UC-11	12									✓	✓	✓	✓		✓	✓			

5.2 System Operational contacts

UC-1 Register/Create an Account

• **Pre-conditions**
– If a new user is opening the application for the first time, in order to jump right into the game, the user must create a login and register within the system
• **Post-conditions**
– After registration, the database is updated with the basic information of the new player.

UC-5 FindPublicGame
• **Pre-conditions**
– Investor must be logged in to their account
– User must not been in a current game
• **Post-conditions**
– player has joined a game
– Queue holding games is updated

UC-2 Create/Join League
• **Pre-conditions**
– User must be logged in
– User must fill in all required text boxes
• **Post-conditions**
– A four digit pin is generated and given to creator
– Other user must input pin to join game

UC-8 Place a Market Order
• **Pre-conditions**
– User is logged into their account.
– User must be in a public or private game
– User must have enough funds in their account to place a market order
• **Post-conditions**
– User profile is reflected with any change to funds and stocks.
– In Game data has been updated with these changes.

UC-9 ViewPortfolio
• **Pre-conditions**
– User is logged into their account.
– User is in a public or private game
• **Post-conditions**
– Any adjustments made to the player's portfolio has been updated in the database.

5.3 Mathematical Model

Rather than use a statistical model for stock price prediction, our application will utilize data of stocks from previous years.

Our application involves players can either buy stocks, sell stocks, request a short, buy a cover, limit a specific stock, and create a stop price of a specific stock from various companies. However, the companies that exist within the app are fictional. Each of these corporations will possess its own characteristics, traits, as well as a backstory. We will develop these qualities for each of the companies, providing users with the information and background that they can apply when making the decision of investment. The aim of this background knowledge about the companies is to provide the user with both incentive and deterrents about the individual and unique companies. This will allow the user to proceed and make well informed and educated decisions.

While the backstories and information about each company will be unique, each company's data will be mapped to a company that exists in the real world. The term mapped is used in order to illustrate how the company performs economically according to the stock market. Since the data will be from previous years, the year 2013 applies well to Rags to Riches because it was a strong year for the stock market in which there were no market crashes like the one in 2010. Since 2013 is considered a positive year for the stock market, it allows users to have a more pleasant experience, which simplifies the game for all users regardless of age or skill level. This year is also selected to ensure that users do not possess any recent economic knowledge about major corporations, and thus make the experience more balanced and fair for all users. Selecting the year 2015 or 2016 is less appropriate, because while most users might not house any recollection of the stock market performance of that year, a small percentage of users might hold that knowledge, which could result in player disadvantage and less overall game balance.

In essence, the “model” that our app will utilize will be a set of data from several real world company’s stock performance in the year 2013. Each of the fictional companies that exist within the app will be mapped to one of the real world companies, and thus possess that company’s stock performance of 2013. The user will be provided with the background information about the individual companies, but not what real world company’s stock data it will be mapped to nor the fact that the data is from 2013. The remaining aspects of the fictional companies will be entirely unique and specialized to heighten the experience of the app and the user’s enjoyment. This method of utilizing old stock data rather than develop a statistical model for stock price prediction simplifies this portion and allows for further effort and creative software development to be poured into more novel and interesting parts of the Rags to Riches app.

6 PROJECT MANAGEMENT

6.1 Plan of Work

Our plan of work will be dividing into three sub teams, each responsible for a “mini project”. This stack organization is much more effective than a chain organization in that it allows less dependency on other teams and less communication overhead. In turn, this allows for more creative software development.

Our team size is appropriate to develop such an app with this many functionalities. Every member of our team has their own particular strengths, therefore each subgroup will focus on development on a particular functionality of the project. A smaller team would not be able to design and implement a project of this breadth and one that includes this number of features.

The main objective for the next few weeks is to develop the core components of our project. We want to make sure that we have a base build of our project before branching off and developing the additional features. The additional features are to make the game more enjoyable for the user and even though that is important, we need to make sure the barebones version of our app will function first. Each group will be working on the most important aspects of their sections of the project. The specific short term goals of each group are outlined below.

6.2 Group 1: Alejandro and Arjun

The first group in our team will focus on various portions of the project having to do with the user account creation and management, achievements, and social media integration.

User Interfaces: Landing page, Login page, Create Account page, Settings page

Short Term Goals: The plan is to make a skeleton of the framework that will be the account page, settings page, and login page. As well as setup a database that will hold account information securely.

6.3 Group 2: Elisa-Michelle and Bryan

The second group in our team will focus on various portions of the project having to do with creating and joining a private game, joining a public game, leaderboards, and the chat system.

User Interfaces: Create Room page, Join Room page, Leaderboards page, Waiting Room page, and chat UI.

Short Term Goals: The plan is to have basic game creation implemented along with an invitation system for other players.

6.4 Group 3: Kartik, Deep, and William

The third group in our team will focus on various portions of the project having to do with virtual stock simulation, market transactions, and the end-game scenarios .

User Interfaces: User Portfolio page, User Performance pages, Transaction page, and Achievements page, Game page

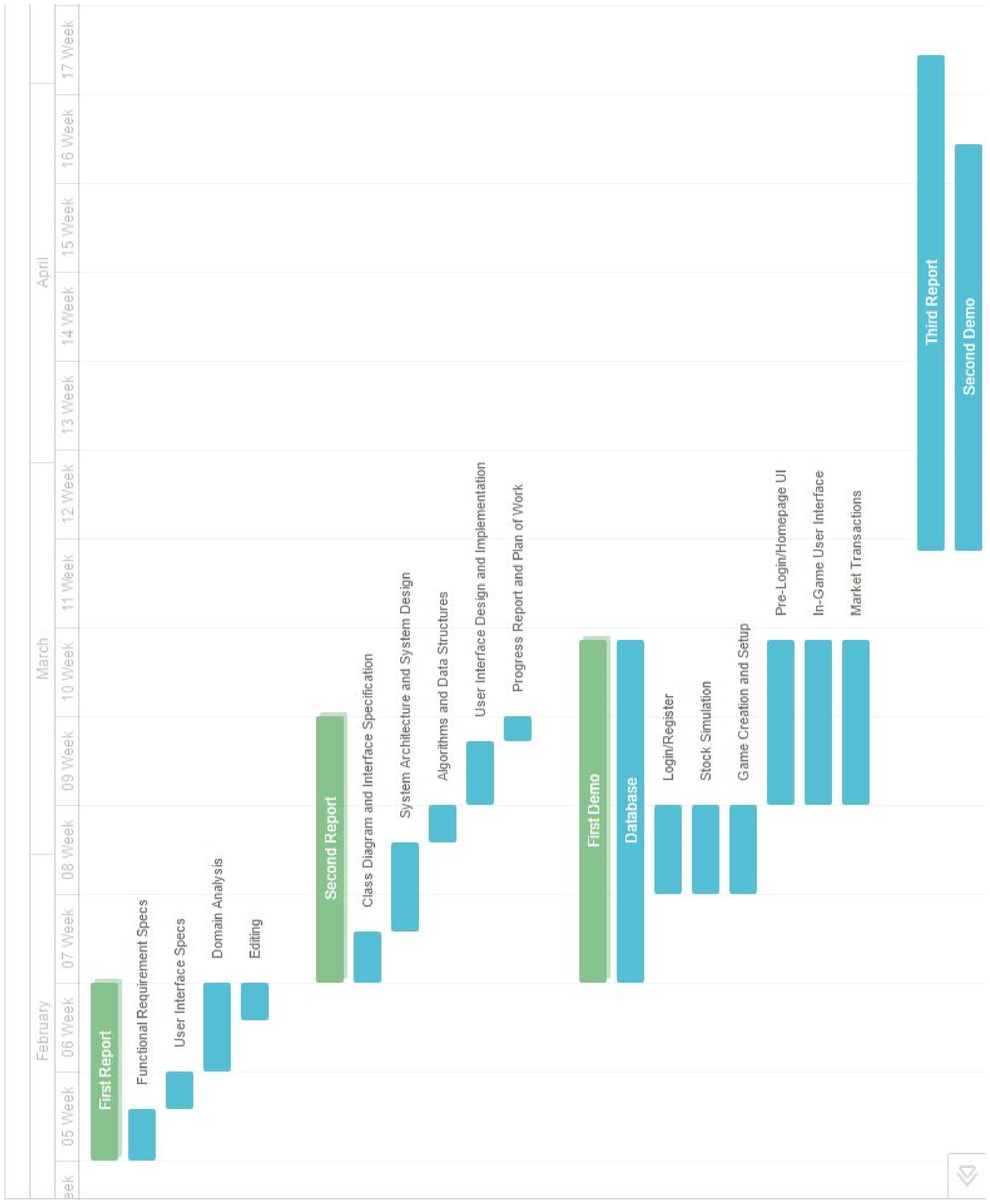
Short Term Goals: The plan is to focus on the storing previous stock information for stock simulations as well as storage and retrieval of user portfolios.

6.5 Measure of Success

The success of the application in educating the user will be determined quantitatively by the user's wealth at the end of the game. If the user was able to increase their wealth from their starting capital, then they demonstrated that they were able to understand and apply good investment strategies.

Furthermore, the amount of achievements a user has gotten also demonstrates the degree to which they understand the investment concepts. If a user is able to reach one of the achievements we have laid out for them, then they will have reached that "checkpoint" of understanding the material, and hopefully continue to reach higher and higher achievements until they have a complete grasp of investment strategies.

6.6 Gantt Chart



7 REFERENCES

"Create A Project Plan With Free Online Gantt Chart Software | Ganttpro". *GanttPro*. N.p., 2017. Web. 5 Feb. 2017.

"Investopedia - Sharper Insight. Smarter Investing.". *Investopedia*. N.p., 2017. Web. 5 Feb. 2017.

Grant, Peter. "How To Write A Software Proposal | Ehow". *eHow*. N.p., 2017. Web. 5 Feb. 2017.