Portfol.io

PREPARED FOR

CS 408 Class Project

PREPARED BY

Team 1: Riley Robertson, Jake Koontz, Walter Jacquette, Kyle Brown, Jeremy Putnam, and Eric Thiem

1. Problem Statement

With the increase in technology, the ability for novice investors to get involved with the stock market has increased dramatically. However, the average person does not have sufficient knowledge of the different types of investments and what their risk tolerance is. This leads many people young and old people to put their hard-earned money into something they do not completely understand. That is why we have set out to create a safe place to learn and practice the fundamentals of trading and investing in the stock market. With our app the users will be allocated a set amount of fake money by the coordinator of the game that they will be able to invest or trade in an attempt to make the most money by the end of the time period. This allows the user to try different investment strategies without having to worry about losing their hard-earned money. Our application will feature the ability to search for many different stock tickers for Blue Chip stocks, Mutual Funds, Penny Stocks, and hopefully Bitcoin. We will then give the user relevant data such as graphs of their investment track record and other useful data such as market cap and 52w high / low, and finally allow them to purchase as many of that stock as their budget allows.

2. Background Information

Our web application will be a safe learning environment for anyone that is wanting to get into stock market investing without risking their hard earned money. The stock market is a great way for people to build wealth, however many people are very scared to get started with it. That is why it is our goal to make this barrier of entry a lot lower in allowing people to practice without their own money.

With our solution, each user will have access to thousands of different stocks, index funds, mutual funds, and possibly cryptocurrencies that will allow users with high risk tolerance to push their earnings high and people that are a bit more conservative can invest in more blue chip stocks. It will also allow these users to leave their comfort zone and test out some investment theories that they have heard about without the true risk behind it.

3. Environment

Our project will be built upon the ReactJS framework for the front-end visuals along side a mixture of MDB Bootstrap, Reactstrap, and Bootstrap for CSS components. For our charting framework, we will be using Highcharts to show the graphs of historical stock data. In the backend, we will be using NodeJS for the server and a MongoDB database, and our server will be utilizing the Alpha Vantage API to obtain our real-time stock data. For authentication users, we will be using Firebase Auth system.

4. Requirements (Backlog)

Milestone	Tasks		
#	As a user	Hours	Planned Status
1.01	As a user, I would like to be able to create an account	7	Sprint 1
1.02	As a user, I would like to be able to login in to my account	7	Sprint 1
	As an Investor		
1.03	As an investor, I would like to have my portfolio performance on my home page	22	Sprint 1
1.04	As an investor, I would like to have a list of my currently owned stocks and current prices	15	Sprint 2
1.05	As an investor, I would like to have a list of my stock watchlist and current prices	15	Sprint 2
1.06	As an investor, I would like to be able to sort and filter stocks on any stock pages (watchlist/owned list/ search)	20	Sprint 2
1.07	As an investor, I would like to be able to search for new stock tickers	20	Sprint 1
1.08	As an investor, I would like to be able to add a stock to my watchlist	12	Sprint 2
1.09	As an investor, I would like to be able to view a stock page that has data on historical performance and market data for the current day	25	Sprint 1
1.10	As an investor, I would like to be able to buy a specified quantity of the stock I am currently on	18	Sprint 2

1.11	As an investor, I would like to be able to sell a specified quantity of the stock that I currently own	18	Sprint 2
1.12	As an investor, I would like to be able to see a leaderboard with the portfolio values of everyone in the game	12	Sprint 2
1.13	As an investor, I would like to be able to click on a user on the leaderboard and see a graph of their portfolio's value	15	Sprint 2
	As a Game Creator		
1.14	As a game creator, I would like to be able to start a game lobby and specify the game rules	10	Sprint 1
1.15	As a game creator, I would like to be able to change the rules of a game which is already created before it starts	12	Sprint 1
1.16	As a game creator, I would like to be able to invite other users to join the game	12	Sprint 1

Total Hoursl: 240 Hours (60 hours a week for 6 weeks)

5. Non-Functional

a. Security

Security will be a concern of our third party authenticator, Firebase, as it is backed by google and a safe, easy way to safely encrypt a user's information. That way user's sensitive information will be securely kept in Google's authenticator. MongoDB is a safe and secure database as it encrypts all of our data as it saves it into the database.

b. Scalability

Scalability will be reliant upon the API's and authentication restraints. Our current financial data API, Alpha Vantage, will have some limitations on requests per day that we would need to start paying a monthly fee if wanted to extend that number but for the sake of this class the limitation will suffice. In terms of authentication, our solution, Firebase has a limitation on the number of users the system can have before you have to pay a monthly subscription fee. Those numbers will not be met in this course however if we needed it, it would be accessible.

c. Usability

Usability is a large concern for a product like this. We are expecting novice investors to fully use this application, so we need to make sure no feature is hard to understand or get to. We will do this by making a minimalist home page for easy navigation for where they would like to go. The website will also be responsive, so whether they are using a phone, tablet, or computer, they will have a pleasant viewing experience.

d. Reliability

In terms of reliability, our website will be using a very robust tech stack, ReactJS, NodeJS, and MongoDB, so we will not be using any crazy new technologies that have not been thoroughly tested. However, the reliability will come more down to the speed of our API calls for financial data and handling when the API takes a long time to respond.

e. Hosting

We will be hosting our frontend website from our group Github repository using Github Pages. This system works extremely well because we can set up a branch and whenever we push to that branch, the site automatically updates with the new code. This is convenient not only because all of our code is already saved onto github but it makes it nice so we can very consistently push out updates to the live version of out site. In terms of backend hosting we plan on using Bluemix.

6. Use Cases

Milestone	Use Cases		
#	Task	Hours	Sprint
1.01	As a user, I would like to be able to create an account	X	Sprint 1
Action		System Response	
1.	Click create account	2.	Create account page is displayed

3.	Input account information		
4.	Click create	5.	If valid information, log user into new account.
1.02	As a user, I would like to be able to login in to my account	X	Sprint 1
Action		**System Response**	
1.	User visits our page with no account	2.	Login page is displayed
3.	Input account information		
4.	Click login	5.	If valid information, log user into account.
1.03	As an investor, I would like to have my portfolio performance on my home page	X	Sprint 2
Action		**System Response**	
1.	User visits their home page	2.	Display a line graph with current and historical data on the users portfolio
3.	User can hover over graph	4.	Display price at that point in time
4.	User can change their graphs time period	6.	Display the corresponding portfolio data with the new date range
1.04	As an investor, I would like to have a list of my currently owned stocks and current prices	X	Sprint 2
Action		**System Response**	
1.	User visits their home page	2.	Display a list of current stocks the user owns
		3.	Display current price, and percentage change
4.	User clicks on their owned stock	5.	Display that specific stock page to user
1.05	As an investor, I would like to have a list of my stock watchlist and current prices	X	Sprint 1
Action		**System Response**	
1.	User visits their home page	2.	Display a list of current stocks the user owns
		3.	Display current price, and percentage change
4.	User clicks on their owned stock	5.	Display that specific stock page to user
1.06	As an investor, I would like to be able to sort and filter stocks on any stock pages (watchlist/owned list/ search)	X	Sprint 2
Action		**System Response**	

1.	User visits any stock page	2.	Display the normal list of stocks
3.	User changes the results to filter out a type of stock	4.	Display the stocks that match the filtered criteria
5.	User changes the results to sort by name/value	6.	Display the stocks that match the expected order
1.07	As an investor, I would like to be able to search for new stock tickers	X	Sprint 1
Action		**System Response**	
1.	Click search bar	2.	Allow user to search stock name or stock ticker
3.	User click on the stock they are searching	4.	Bring user to that stocks data page
1.08	As an investor, I would like to be able to add a stock to my watchlist	X	Sprint 1
Action		**System Response**	
1.	User is on a stock page	2.	Display “Add to watchlist” button
3.	User clicks on add to watchlist	4.	Add a checkmark to the watchlist box
		5.	Add that stock to the current user’s watchlist
1.09	As an investor, I would like to be able to view a stock page that has data on historical performance and market data for the current day	X	Sprint 1
Action		**System Response**	
1.	User clicks on a stock ticker	2.	Display a page with live data on that stock
		3.	Display a historical graph
		4.	Display market cap, 52W high, and other important data
1.10	As an investor, I would like to be able to buy a specified quantity of the stock I am currently on	X	Sprint 2
Action		**System Response**	
1.	User specifies number of stock to buy	2.	Display total price for that many stocks
3.	User confirms the price and quantity of stocks to buy	4.	Display check on buy stock box
		5.	Subtract that money from their wallet
1.11	As an investor, I would like to be able to sell a specified quantity of the stock that I currently own	X	Sprint 2

Action		**System Response**	
1.	User specifies number and type of stocks to sell	2.	Display total price for that many stocks
3.	User confirms the price and quantity of stocks to sell	4.	Add money back into user's wallet
1.12	As an investor, I would like to be able to see a leaderboard with everyone in the game's portfolio value	X	Sprint 2
Action		**System Response**	
1.	User clicks on leaderboard icon	2.	Display a page showing all members and their portfolio's value
1.13	As an investor, I would like to be able to click on a user on the leaderboard and see a graph of their portfolio's value	X	Sprint 2
Action		**System Response**	
1.	User clicks on another user on the leaderboard	2.	Loads the selected user's profile
1.14	As a game creator, I would like to be able to start a game lobby and specify the game rules	X	Sprint 1
Action		**System Response**	
1.	User clicks the create game button	2.	Game is created and a new dialog is presented to the user
3.	User enters desired end time, start time, dollar amount, and trade limits	4.	The game is created with the desired specifications
1.15	As a game creator, I would like to be able to change the rules of a game which is already created before it starts	X	Sprint 1
Action		**System Response**	
1.	User clicks the options button in the game lobby	2.	A dialog box appears with the current game settings

3.	User edits the settings and clicks the save button	4.	The game updates with the new settings
1.16	As a game creator, I would like to be able to invite other users to join the game	X	Sprint 2
Action		**System Response**	
1.	Game creator clicks on the game page	2.	Display a page with a table showing players that were already invited
3.	Game creator enters the username of a player and clicks the Invite button	4.	The invited user receives an invitation to the game through their dashboard
		5.	The game shows that the invitation was sent and updates the pending invites table