TradeHub Investment Simulator Proposal

COMP3900 - Computer Science Project

Group:

H11A-UNDEFINED

Team Members:

Domenic Prestia - z5061488
Jet Young Lim - z5195170
Jiaying Yi - z5135198
You Shi Zhu - z5060508
Yukun Shi - z5141536

Submission Date:

04/10/2020

Table of Contents

Executive Summary

This proposal summarises the plan of action of making an investment simulator web application, TradeHub. The problem statement is outlined and three existing investment simulators are

investigated. The existing systems are compared and their individual drawbacks are identified.

The product backlog of user stories presented in this proposal describe the core functionality required in the project objectives as well additional features not in existing investment simulators. The timeline of three sprints are defined, as well the user stories that will be included in the first sprint and projected user stories for the second and third sprints.

Storyboards in this proposal demonstrate how users interact with the system, and all user stories are covered by the story boards. The system architecture shows the presentation, business and data layers of the system as well as how the different types of external actors interact with the system. The technologies and languages to be used are described, which include Python, Flask, JavaScript, React, HTML, CSS, PostgreSQL, and yfinance API.

1.0 Background

1.1 Problem Statement

The stock market trading game has been a valuable learning tool for investors of all levels of experience. Students and first-time investors can learn how to trade stocks, while experienced investors are able to test out different investment strategies, all without risk. For our system, we looked into several simulators already in the market and identified drawbacks.

1.2 Existing Platforms

Table 1.1: Comparison of Existing Platforms

	ASX sharemarket game (ASX)	**SIFMA Stock market game (SMG)**	**NZX virtual trading (NZX)**
Stock details page	Includes graph, detailed info, announcements	No detailed info or graph	No detailed info or graph
Starting fund	$100,000	$ 100,000	$50,000 (can reset account)
Place orders during closed market	Yes	No	Yes
Educational Popups	Yes	No	Yes
Self-study material for stock market	Yes	Yes	Yes
Glossary page	Yes	Yes	Yes
Support Watchlists	Yes	No	No
Game cycle	3 months	Minimum 4 months	No defined cycle

As seen in the table above, our group looked into several websites including ASX Sharemarket Game (ASX), SIFMA Stock Market Game (SMG), and NZX Virtual Trading (NZX). All three simulators are designed with a user-friendly interface, instructional guidelines and provide extra

study material. This gives first-time users background knowledge to start trading. Amongst the three simulators, the ASX Sharemarket Game is the most mature website with detailed stock information pages and support for a watchlist with notification function. The initial balance is also substantial, though even the minimum amount amongst the three simulators is $50,000, which may be adequate for users to explore different investment strategies. However, there are a few drawbacks in the design of simulators: First, the ASX Sharemarket Game is the only simulator that displays detailed stock information along with graphs of stock performance (see Figure1).

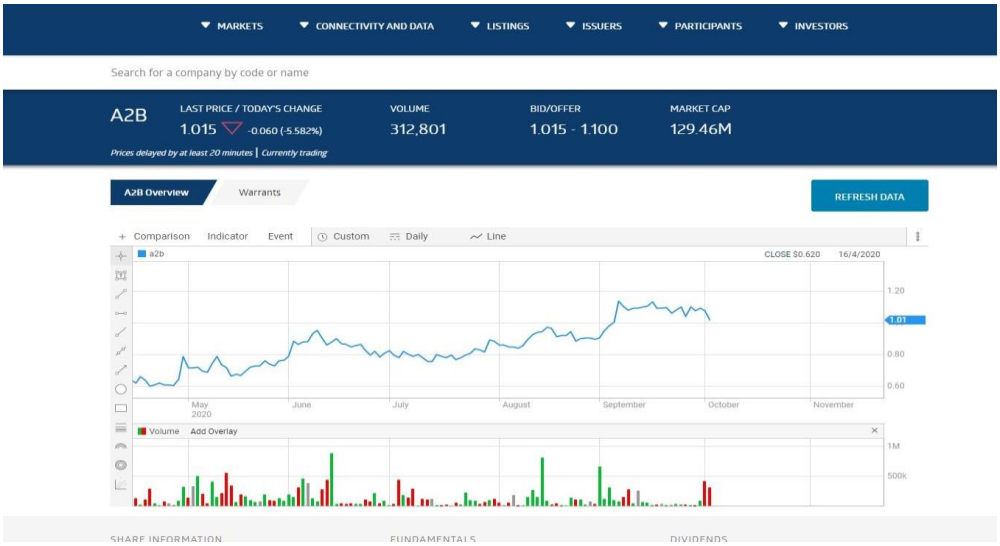


Figure 1.1: Screenshot for detailed page. Retrieved from ASX website

Second, although all three simulators offer educational material for basic concepts and general strategies, none of simulators provide a case-study opportunity based on the user's personalized preference. Users are unable to test their investment skills in a short period of time. Our proposed system will address the problems by inserting a detailed page for individual stock, watchlist and novel feature call investment challenge.

2.0 User Stories and Sprints

2.1 Product Backlog

Table 2.1: Epic Story 1 - Unregistered Users

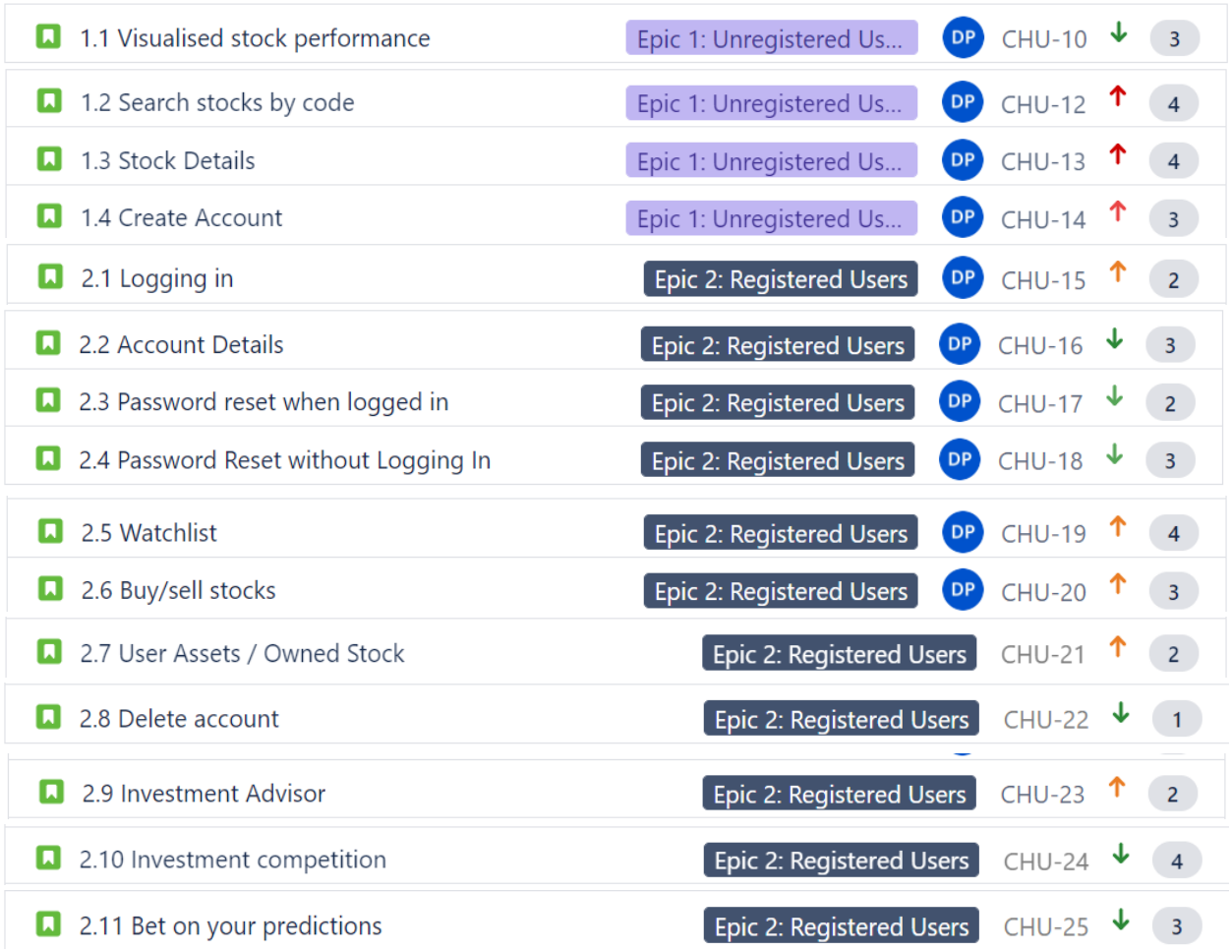
US 1.1	As an unregistered user, I want to be able to see graphs of how stocks are doing, so I have a visual reference of stock performance
US 1.2	As an unregistered user, I want to search stocks by code and/or name, so I can navigate different stocks more efficiently
US 1.3	As an unregistered user, I want to be able to see the details of specific stocks, so I can be informed about specific stock's details
US 1.4	As an unregistered user, I want to be able to make an account, so that I can get access to registered features.

Table 2.2: Epic Story 2 - Registered Users

US 2.1	As a registered user, I want to log in to the website with my credentials, so that I can access my account
US 2.2	As a registered user, I want to be able to see my account details, so I can view things specific to me
US 2.3	As a registered user, I want to be able to change my password, so I can stop using any compromised passwords
US 2.4	As a registered user, I want to be able to change my password if I forgot it, so I can log in
US 2.5	As a registered user, I want to be able to add stocks to my watchlist, so I can easily see how all the things im interested in are doing
US 2.6	As a registered user, I want to be able to buy/sell stocks, so I can profit.
US 2.7	As a registered user, I want to be able to view and interact with my assets so that I can improve my overall portfolio performance.
US 2.8	As a registered user, I want to be able to delete my account, so I can stop playing this game
US 2.9	As a registered user I want to see how a cat randomly invests, so that I can make profit better than most of the investors.
US 2.10	As a registered user I want to challenge the system to have a chance to profit

US 2.11	As a registered user, I want to bet on how the stocks will perform, so that I can test my investment skill.

Figure 2.1-Jira Screenshots



2.2 User stories that satisfy the Objectives

Table 2.3: Objectives and matching user stories

Investors must be able to search for a stock using a "stock code" (also known as stock “symbol”), with results indicating the stock name, and latest available unit price for the stock.	**US 1.2**
Each investor must be able to add stocks from search results to their personal visible watchlist, and remove stocks from this watch list, with each stock on this watch list showing: ● the stock code, ● latest available stock price per unit, and ● the percentage change in the stock unit price when comparing the latest available stock unit price to the previous day's known stock unit price.	**US 2.5**
Investors must be able to view a graph showing the historical daily closing unit price for any stock on their watchlist, where historical data must at least be available from the day when the stock was added to the watchlist to now.	**US 1.1** **US 1.3**
Investors must be able to "simulate" a "buy" order for a given stock at the current market price per unit, for a given number of units (note that a "simulated" buy order means that a "buy" order isn't actually executed, and the investor doesn't actually own the stock - hence we call stock that is bought using a simulated "buy" order "simul-owned").	**US 2.6**
An investor must be able to "simulate" a "sell" order for a given stock at the current market price per unit, for a given number of stock units, only for stock-units that they simul-own.	**US 2.6**
Investors must be able to view the total profit or loss they would make if all the stock units they currently simul-own were sold at their current market price per unit.	**US 2.2**
They must be able to view the total profit or loss they would make for any given stock they simul-own, if all the units they simul-own for that stock were sold at the current market price per unit.	**US 2.7**
Investors must be able to see a page that lists aggregate statistics for each stock type simul-owned, including: ● total units simul-owned, ● total current worth of simul-owned units, ● total paid for currently simul-owned units.	**US 2.7**

These user stories completely cover the required objectives. Other user stories, excluding the novelties, pertain to user account maintenance.

2.3 Sprint Plans

2.3.1 Sprint dates

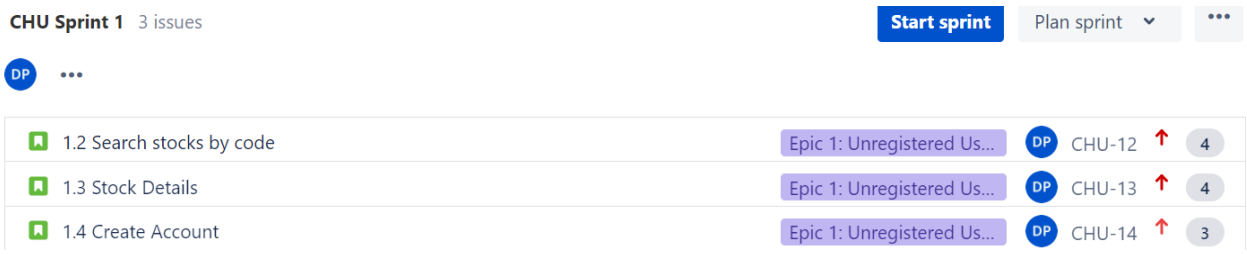
The project will be broken up into 3 variable length sprints. The sprints coincide with the demonstrations and retrospectives.

Table 2.4: Planned Sprints

Sprint Number	**Sprint Dates**	**Sprint Lengths**
Sprint 1	5th Oct - 15 Oct	11 days
Sprint 2	16 Oct - 5 Nov	20 days
Sprint 3	5 Nov - 16 Nov	12 days

2.3.2 First Sprint User Stories

Sprint 1

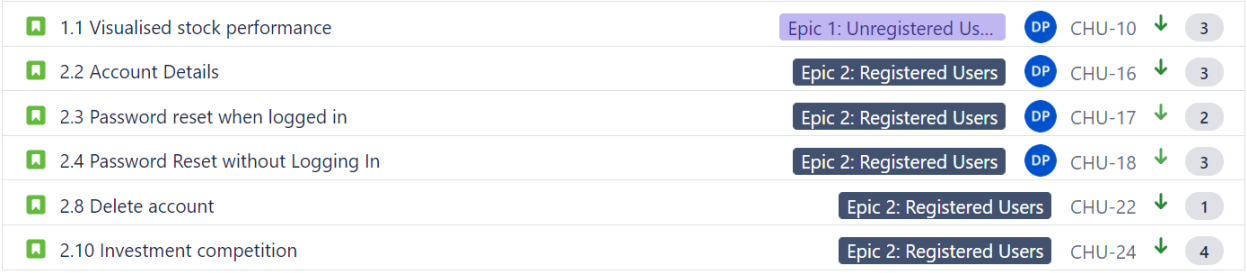


2.3.3 Projected Future Sprints

These are the user stories we expect sprint 2 and 3 to contain, but this could still subject to change

Sprint 2

Sprint 3



2.4 Novel Functionalities

Our investment simulator provides three novel functions.

Table 2.5: Novel User Stories

US 2.9	As a registered user, I want advice on what stocks to buy, so I can make better investment choices.
US 2.10	As a registered user I want to challenge the system to have a chance to increase my profits
US 2.11	As a registered user, I want to bet on how specific stocks will perform, so I can make profit off my bets

US 2.9 introduces the 'Investment Advisor', which is based on the 'random walk hypothesis'. This proposes that stock market prices cannot be inferred from past stock movements, so cannot be predicted. An advisor selects stocks to invest in. User's can look at the advisor's choices and choose to follow it, or not. No other system includes an actual implementation of such a theory. This allows the client to become aware of the unpredictability of actual stock markets, and allow them to interact with it.

US 2.10 introduces an investment competition game which gives users a chance to challenge our system. The competition is week long, in which users choose a starting money amount and a subset of stocks to trade in (minimum of 5). Over this period of time the user and the system will trade in these stocks, and whoever has made the most money at the end of the time period is the winner. We believe narrowing the scope of investment is important. In this way, users can only compete in the investment direction they are interested in, instead of being confused in the face of the huge stock market. At the same time, such competition is more unique and interesting, players will not be able to copy the operation of top investors.

User's also wager some of their money on the challenge, doubling the money if they win, and losing it all if they lose. They're also blind to the system's investment choices during a challenge. This encourages the user to invest more aggressively to ensure a win by the end of the challenge.

These features serve to increase the user's enjoyment by playing this high stakes game in which users try to beat the system.

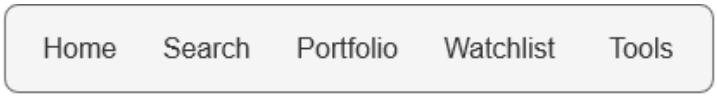
The closest thing that other systems have is a leader board that is shared amongst a group of people. These are not personalised, users are aware of the portfolios of others and there is no

notion of risk involved. Furthermore, users cannot decide most of the core elements in a leaderboard. By comparison, our system's degree of customization is novel and enticing.

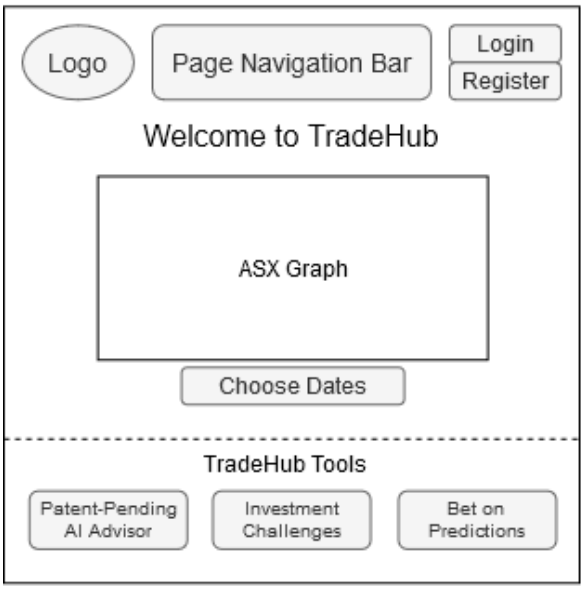
US 2.11 describes a user's ability to bet on the performance of a particular stock. A user will predict the range of percentages that a specific stock will increase/decrease at the end of the day/week/month. A user wins the bet if the increase/decrease falls within this range. The reward is the initial bet multiplied by the reward multiplier, which depends on the period selection (day/week/month). If they do not predict correctly, they lose the money they betted. Predicting when a stock will decrease and being rewarded for it simulates the practice of 'short selling'. This is the practice of borrowing stocks, and then selling them, with the intention of subsequently repurchasing them at a lower price to gain a profit. Therefore, our system will allow users to practice a simplified version of short selling and will allow users to profit even when a stock decreases.

3.0 Interface/Flow Diagrams

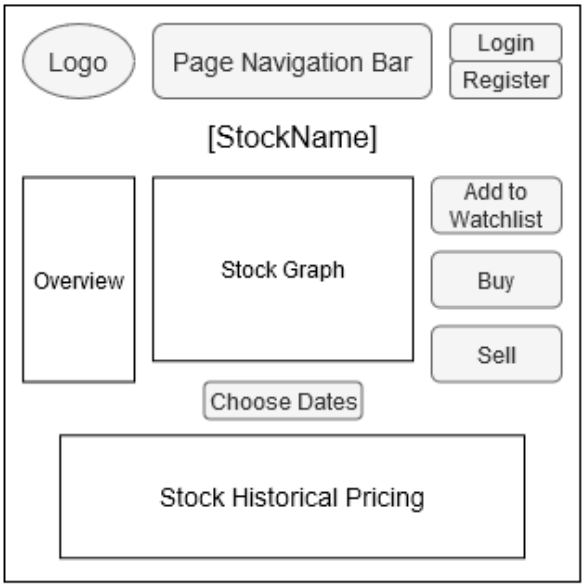
3.1 Unique Pages



Page Navigation Bar

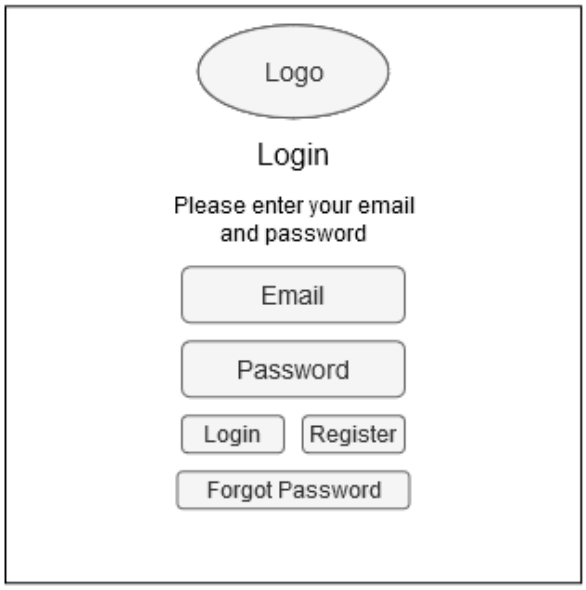


Front Page



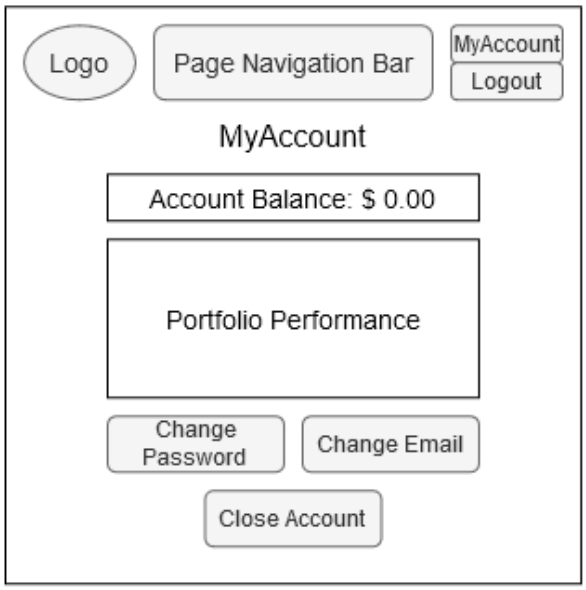
Stock Details Page

Search Page

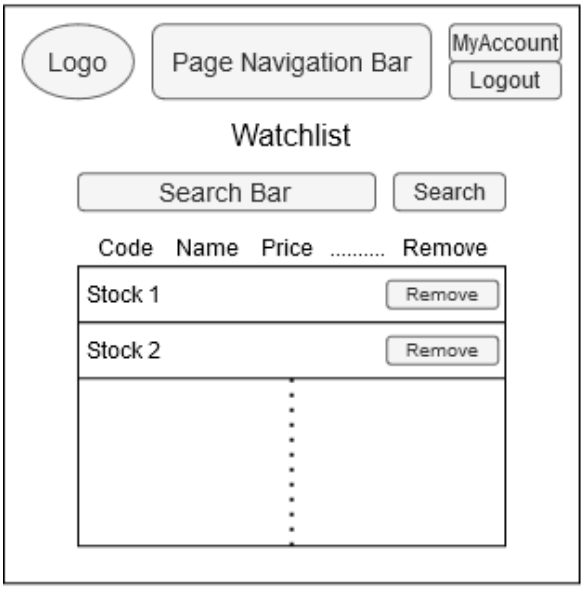


Login Page

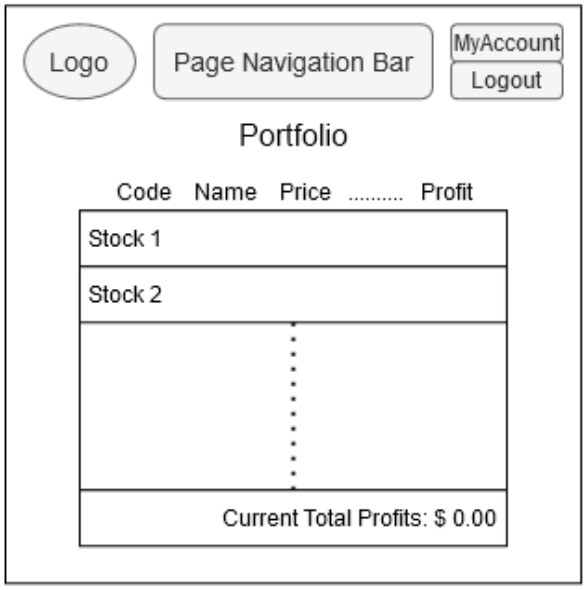
Register Page



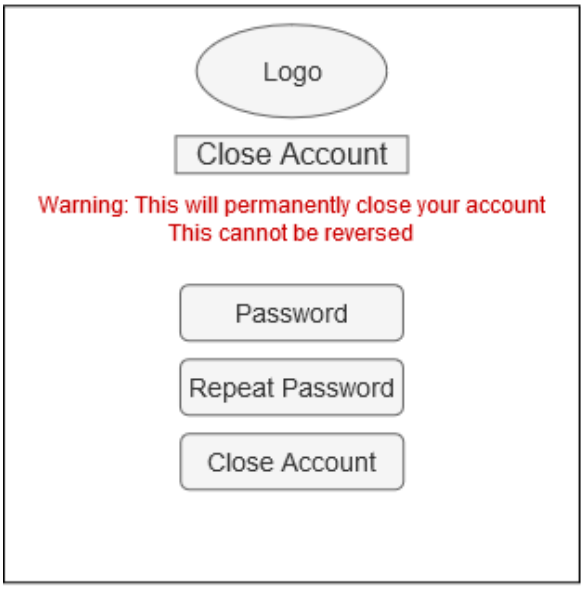
MyAccount Page



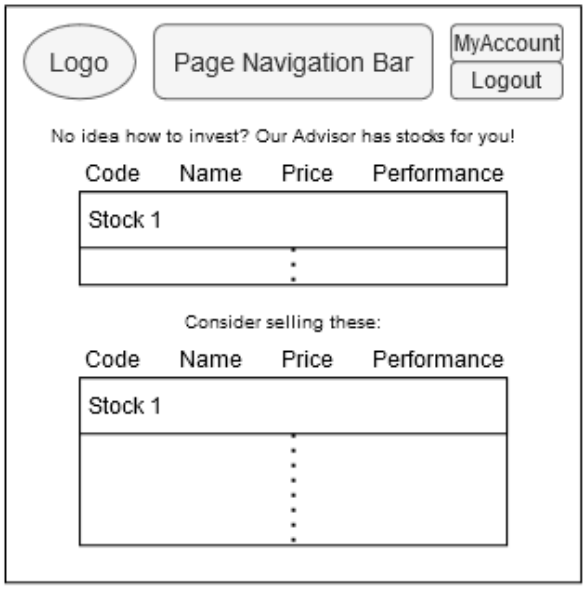
Watchlist Page



Porfolio Page

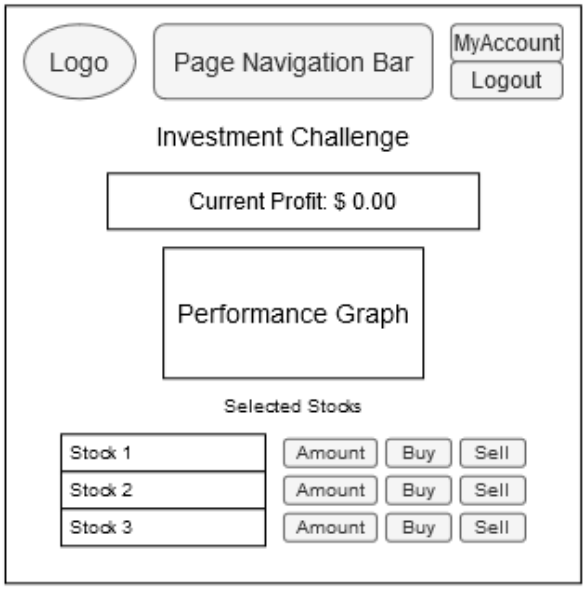


Account Deletion Page

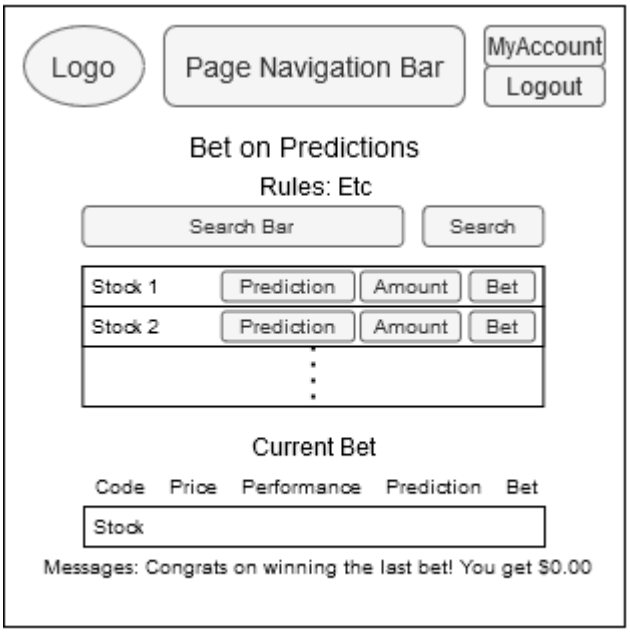


AI Advisor Page

Challenge Setup Page



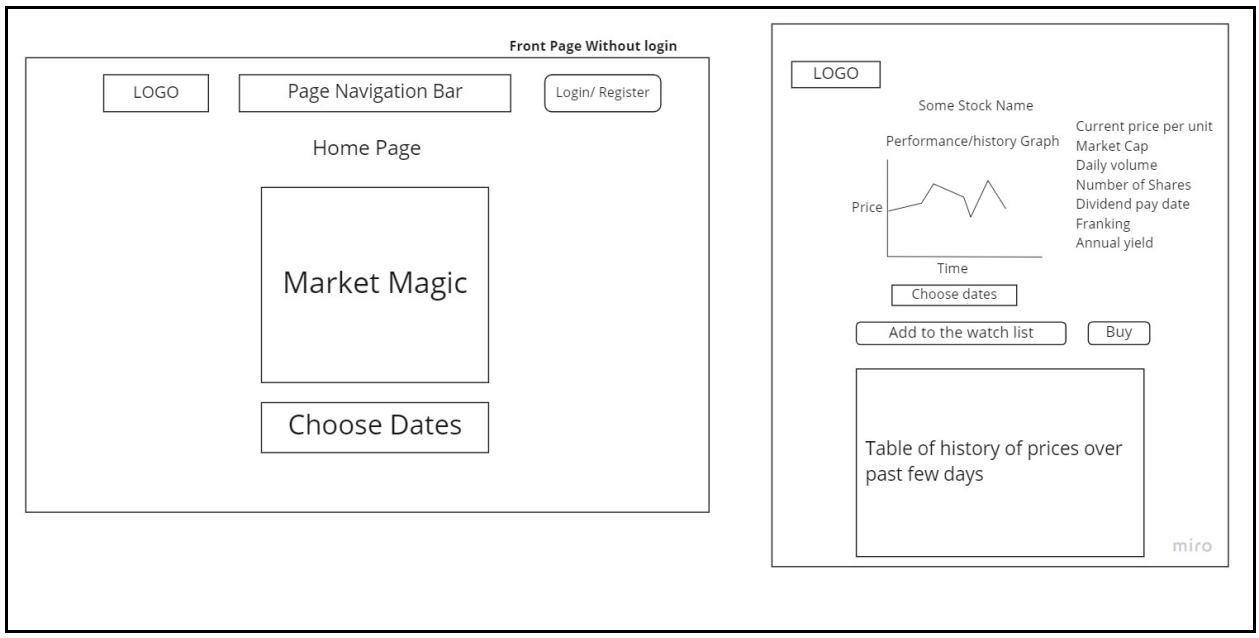
Challenge Play Page



Prediction Betting Page

3.2 Epic Story 1 - Unregistered Users

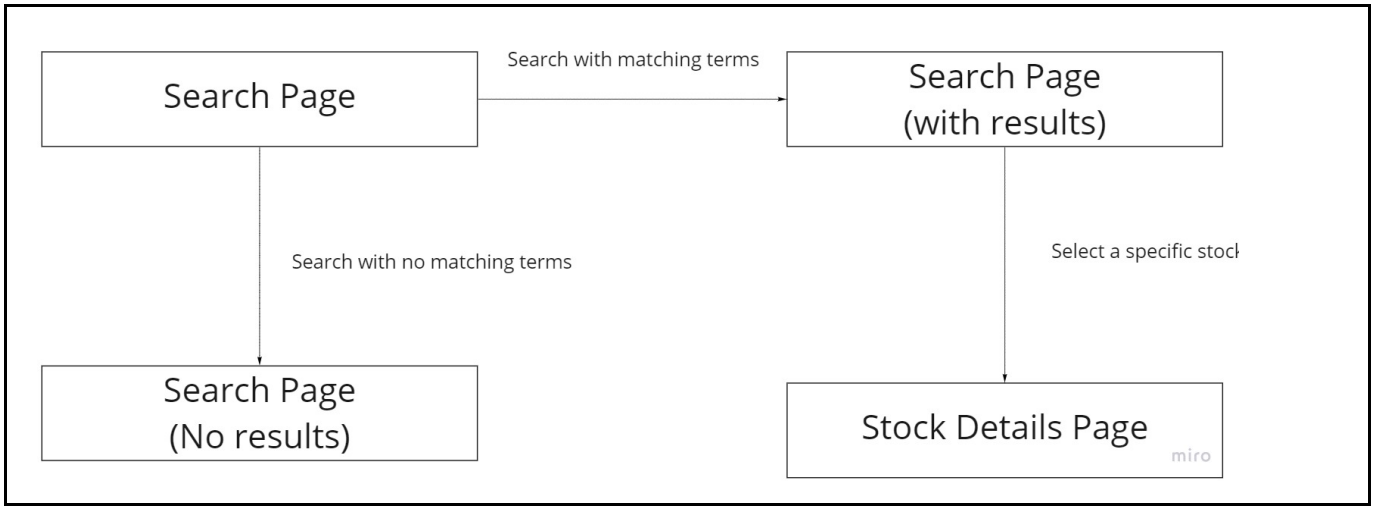
User Story 1.1



The general stock market's performance is shown at the front/home page and the individual stock's performance shown in its own details page.

Name	Visualize Stock Performance
Acceptance Criteria	● Graph Labels: ○ Price ○ Time ● Must display a graph of current stock market performance. ● Each stock detail page will have a graph visualizing the stock's performance.

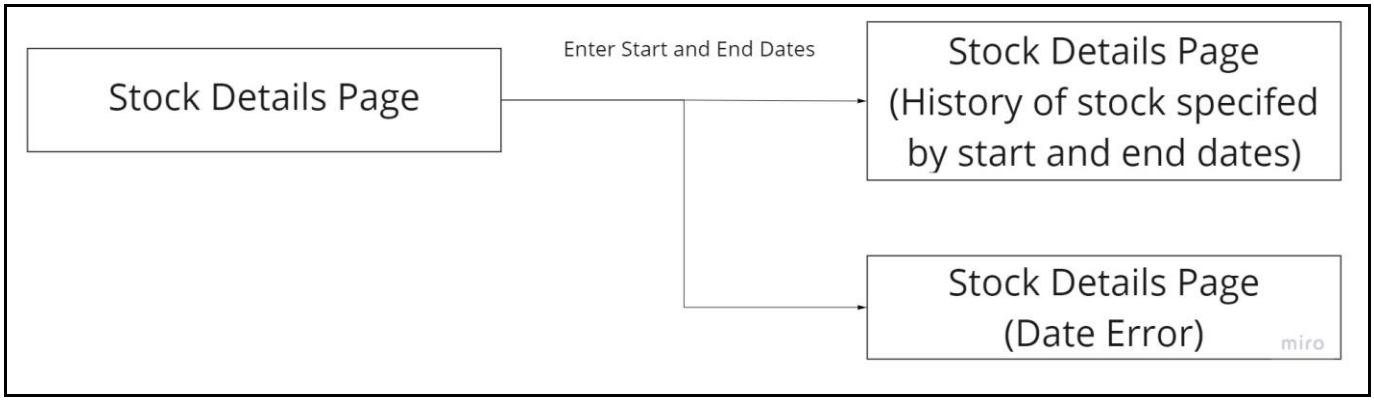
User Story 1.2



Stock search query flow diagram, on success the user will be shown a list of stock with the matching term that allows the user to access a specific stock detail page. On failure a failure message is provided.

Name	Stock search by code
Acceptance Criteria	● Search page with search bar: ○ Searching displays a list of matching items. ○ No results display text indicating search failed. ● Clicking on searching items navigates to the corresponding stock page. ● Before searching, default stocks are displayed. ● Results should display stock name and latest available unit price for the stock.

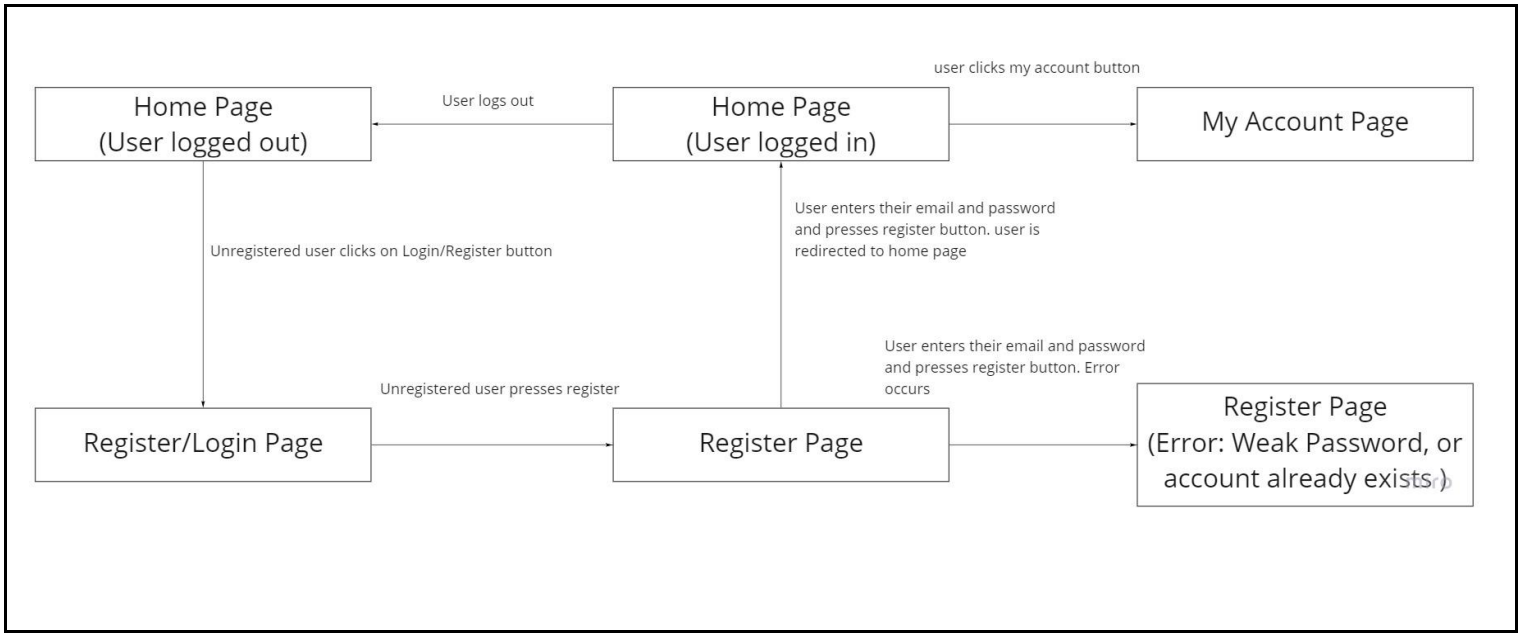
User Story 1.3



Stock price history query, on success returns users with price details of a stock in a specific date range. Failure provides an error message.

Name	Stock details
Acceptance Criteria	● Stock page should display the following details: ○ Stock name ○ Market cap ○ Current price ○ Daily volume ○ Number of shares ○ Dividend pay date ○ Franking ○ Annual yield ○ Price history ● Able to input a start date and end date, table of history price updated to match according to the data range. ○ Invalid date input will result in invalid date popup error.

User Story 1.4

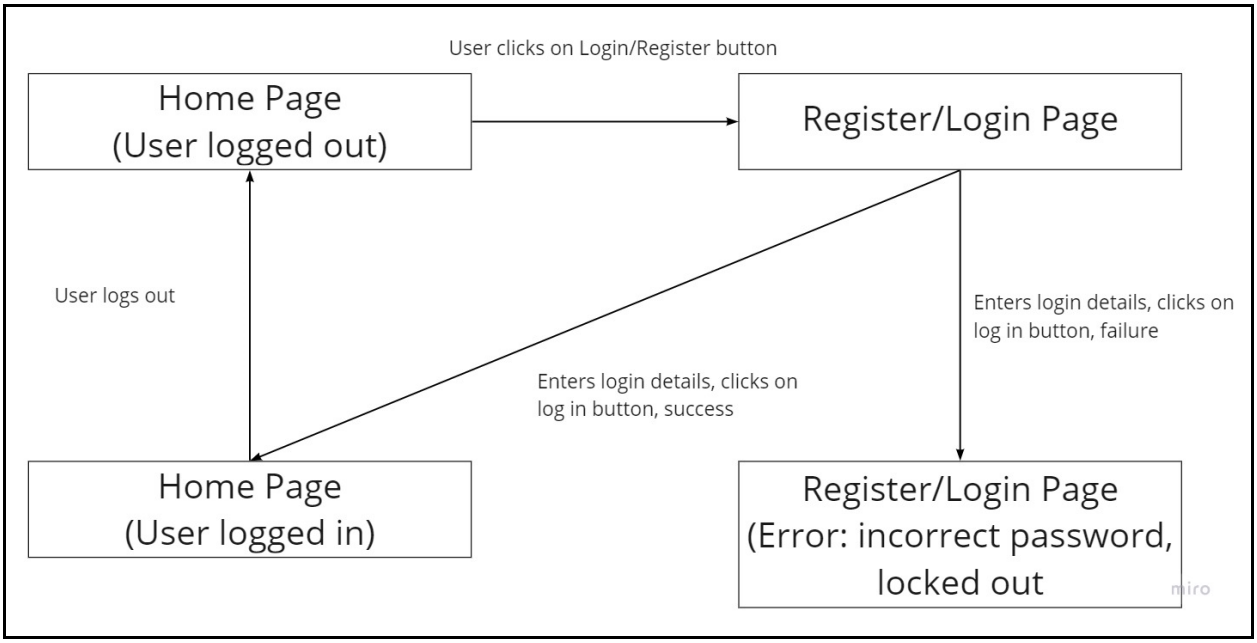


Stock price history query, on success returns users with price details of a stock in a specific date range. Failure provides an error message.

Name	Account Creation
Acceptance Criteria	● Should be able to click on the “sign up” at the home page. ● Unregistered users should be able to type in a password and email address to register ○ “Email already exists” error ○ “Weak password” error ○ Sanitised inputs ● Passwords should generally follow NIST password guidelines ○ https://specopssoft.com/blog/nist-password-standards/ ● Should be logged in if successfully registered ● After creating account, has access to Registered user only functions ● Can log out of account ● Accounts are saved and can be logged back in ● Accounts start of with $1bil

3.3 Epic Story 2 - Registered Users

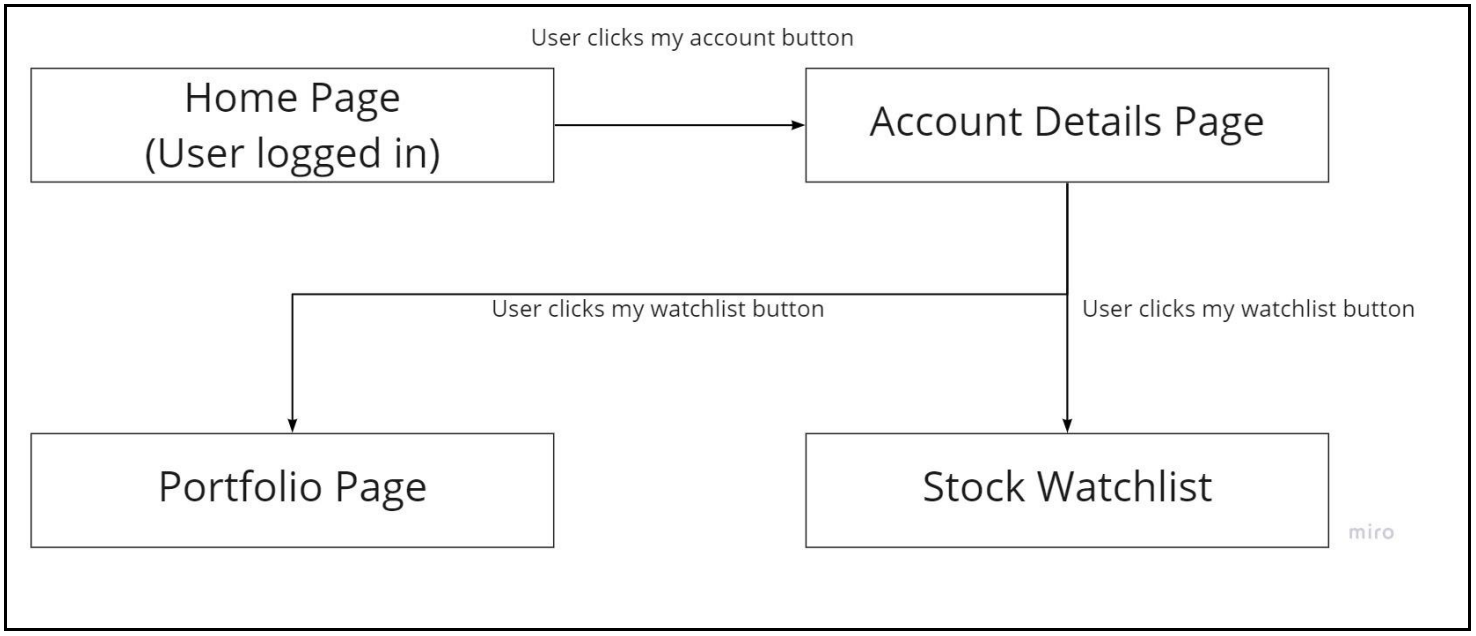
User Story 2.1



Registered users login flow diagram, successful login reroutes user to home page and on failure prompts user to re-enter their credentials and a warning message is provided.

Name	User Login
Acceptance Criteria	● Should be able to click on the login button at home page. ● User can enter credentials ○ Let user login if match ○ Error and prevent login if mismatch ● If fail to login more than 10 times, lock for period of time ● After logging, reroute user to home/front page

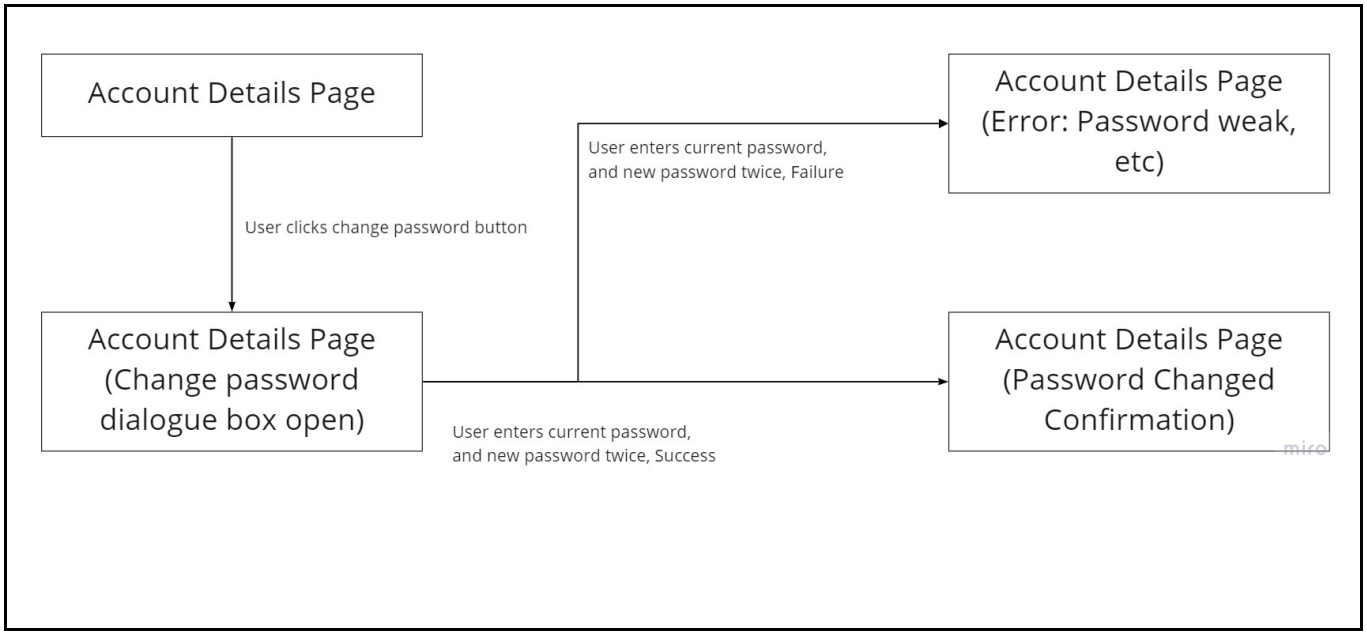
User Story 2.2



Flow diagram for registered user stories accessing pages that are account specific to themselves.

Name	User Account Details
Acceptance Criteria	● Users can click on the “My Account” button at home page to navigate to their main account details page, this button is replaced from “Login” to “My Account” upon successful login by the user. ● Account details page should contain the following ○ Watchlist and Portfolio page reroutes ○ Search bar in both Portfolio and Watchlist page ○ Wallet details in main account details page ● Portfolio page contains: ○ Current profit if all assets are sold immediately ○ List of owned stocks and their basic details ● Have update/change email and password buttons at the main account details page.

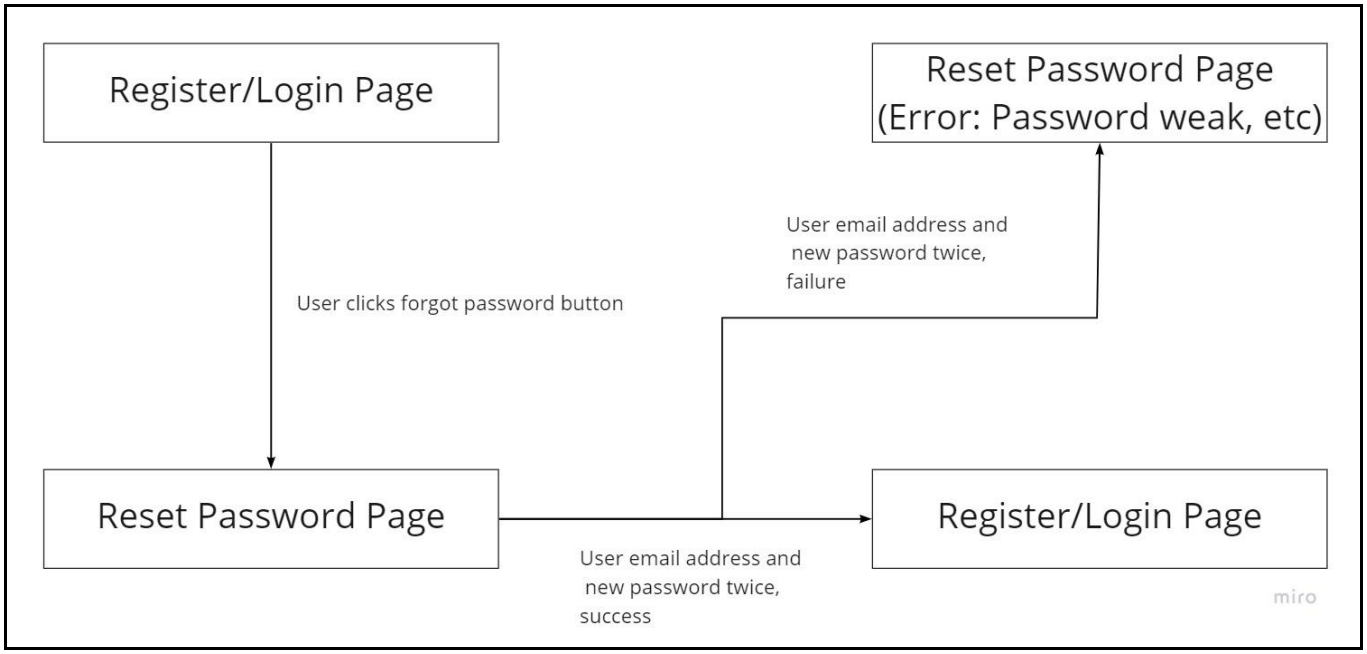
User Story 2.3



Registered users changing their passwords, weak passwords prompt users to enter a stronger password and a confirmation for a password change is presented when the user successfully enters their new password.

Name	Password Reset (Logged in Users)
Acceptance Criteria	● Able to click on “Change Password” button on “My Account” / main account detail page ● Registered users should be able to reset password ○ If current password doesn’t match, prevent changing password ○ Should type a new password twice, reset if matched, error if failed ○ “Weak password” error, new password must adhere to NIST password guideline.

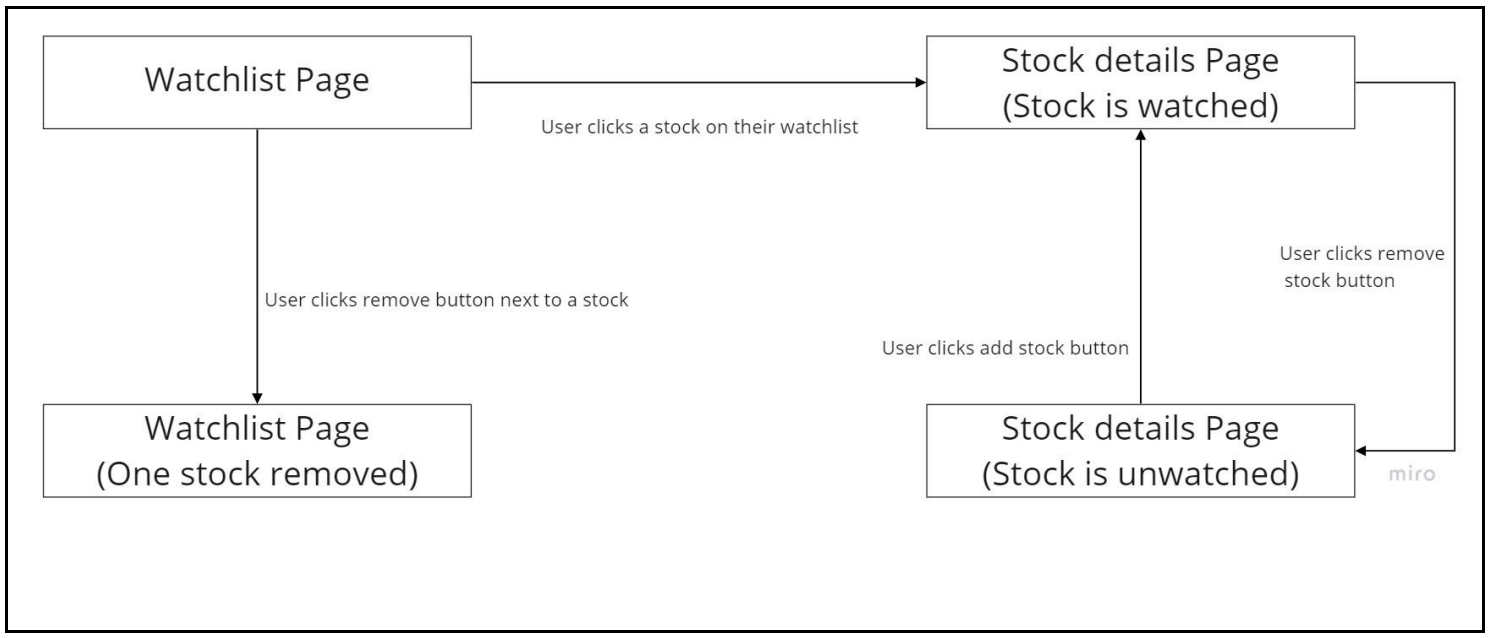
User Story 2.4



Registered users are able to reset/change their password at the login page with a “Forgot Password” button at the login page.

Name	Password Reset Without Logging In
Acceptance Criteria	● User is able to click “Forgot Password” button on login page ● User change password for an email ○ If email doesn’t exist, don’t give any warning

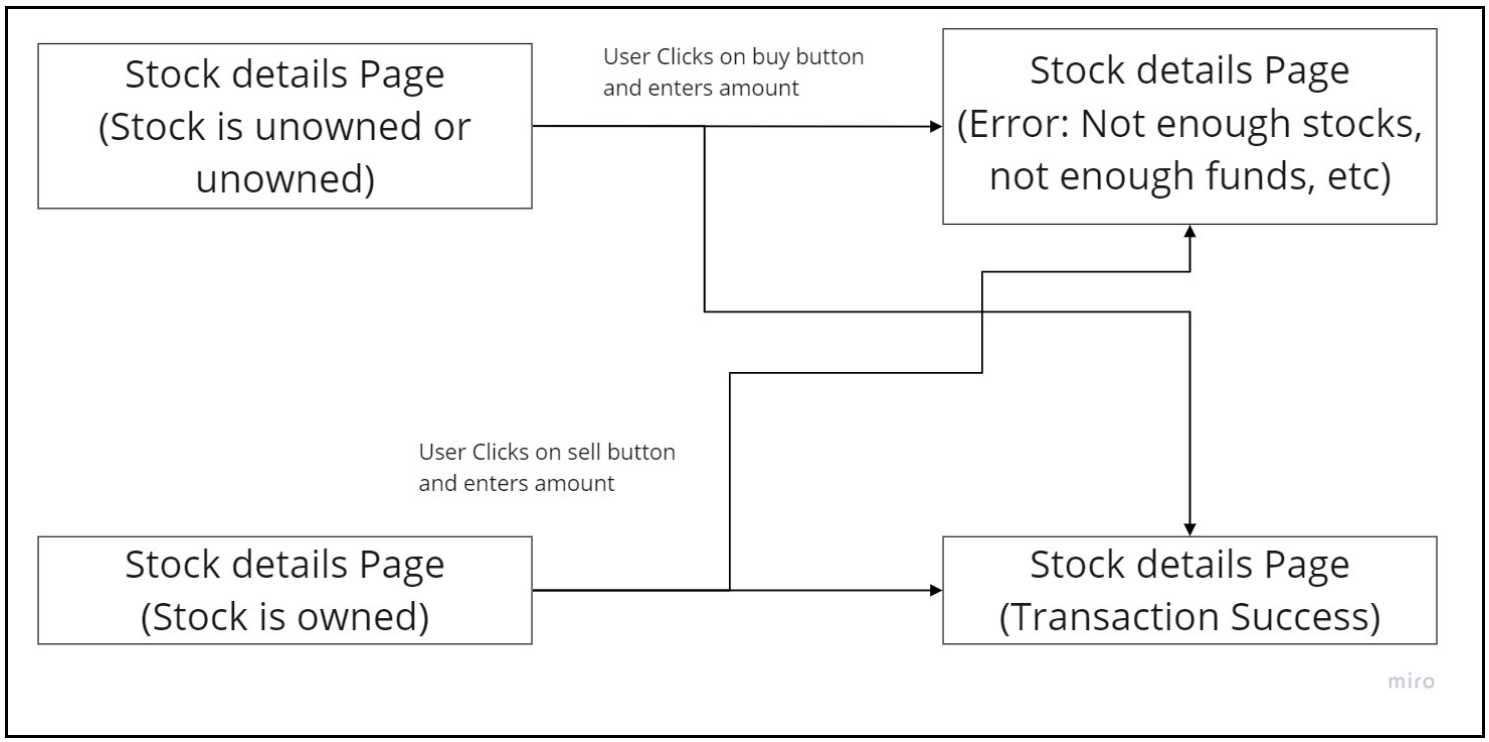
User Story 2.5



Registered users are able to add/remove stocks they've seen into their watchlist that is accessible at their watchlist page. They are also able to navigate stocks in their watchlist page.

Name	Watchlist Page Functionalities
Acceptance Criteria	● User should be able to add to the watchlist through the stock details page ● User should be able to remove a stock from the watchlist on the stock details page or via the watchlist page ● User should be able to click on "watchlist" then display a list of stocks ● User should be able to view the details page of stock in watchlist when clicked ● Watchlist should display some basic statistics for each stock: ○ Stock code ○ Stock price ○ Stock performance

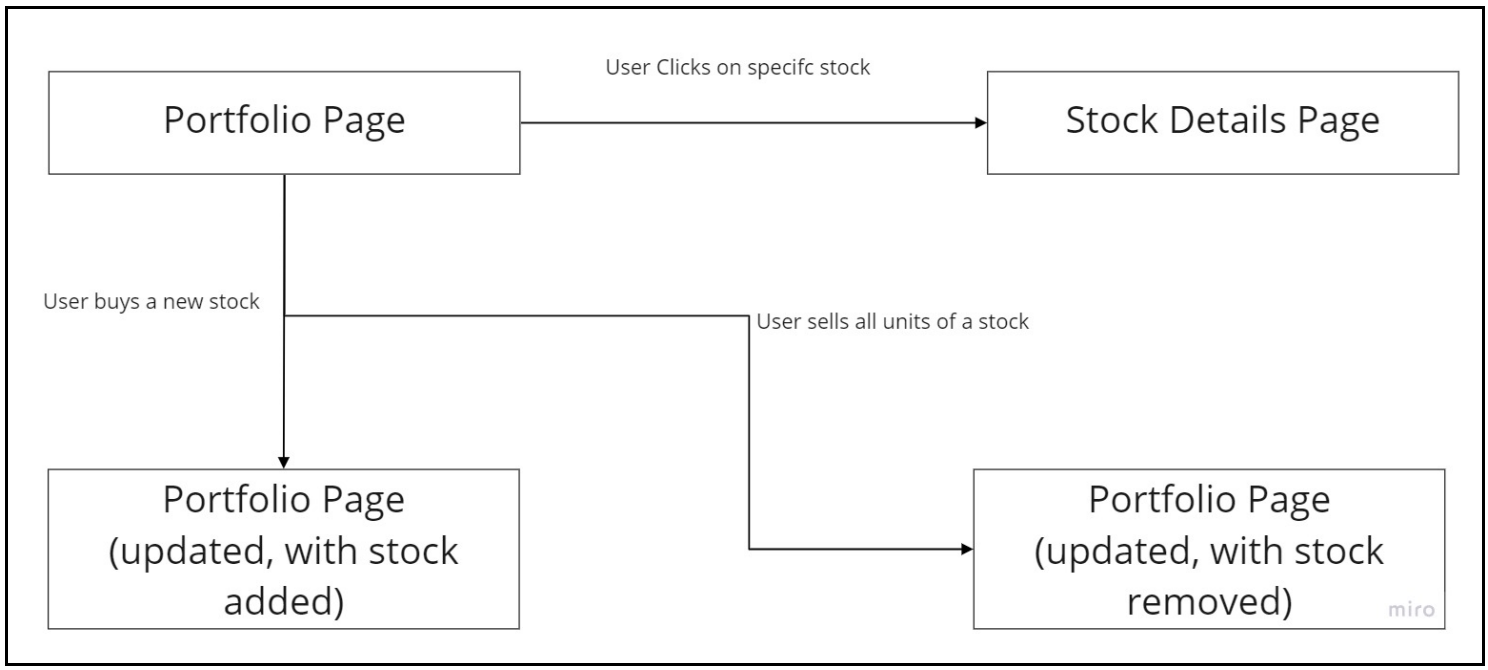
User Story 2.6



Registered users are able buy and sell stocks if they have the appropriate amount and number of stocks respectively.

Name	Buy/Sell Stocks
Acceptance Criteria	● Registered users can buy stocks when they have enough capital. ○ Error message if buying more stocks than they have capital ● Registered users can sell stocks within their holdings. ○ Sell from the stock details page if it is owned ○ Error message if selling more stocks than they own ● Users can see the total value of the stock they are selling ● Users can buy/sell based on the latest price even if the market closes

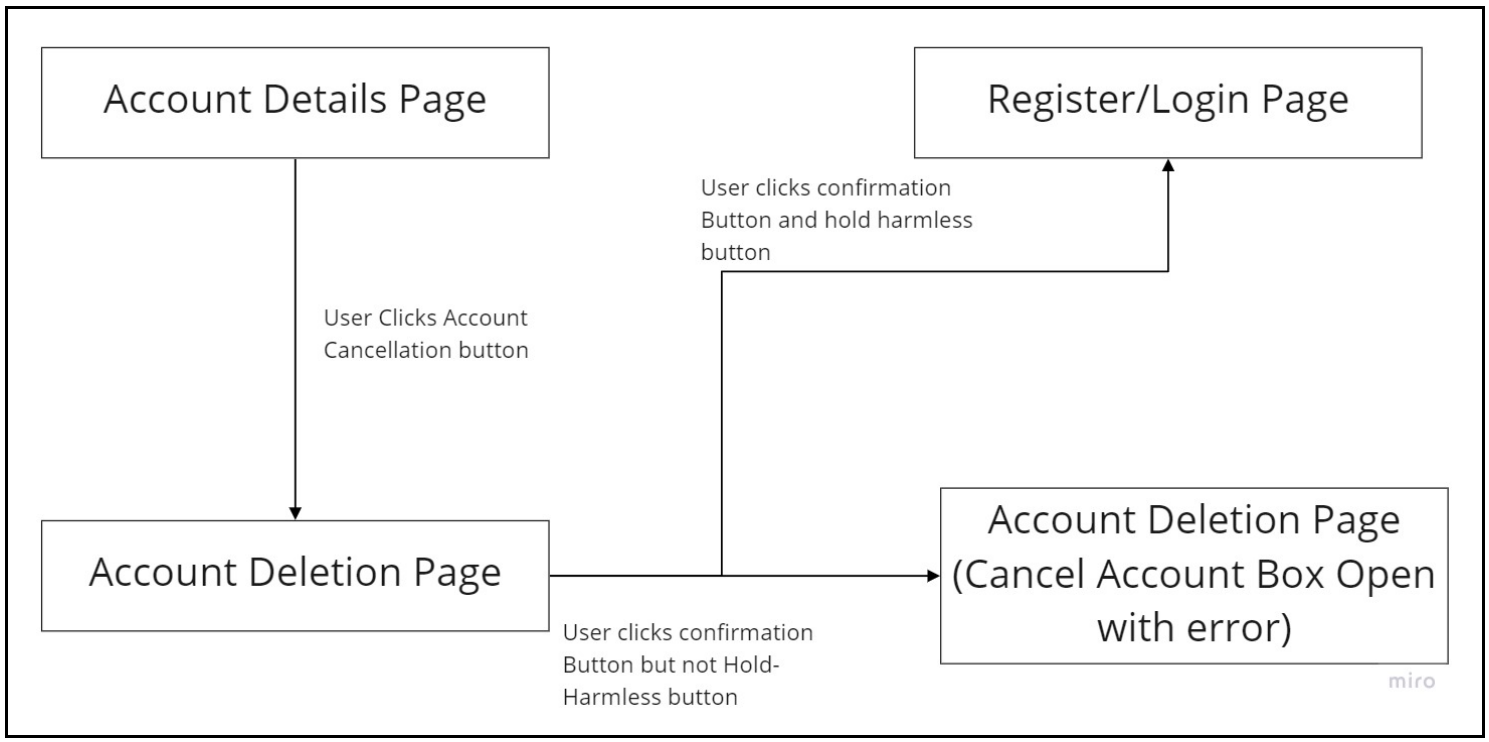
User Story 2.7



Registered users are able to view their assets overall performance and manage it with their portfolio page.

Name	User Assets / Portfolio
Acceptance Criteria	● Users should be able to access individual investments through the user assets page. ● Users should be able to click on “owned stocks” then display a list of user owned assets. ● Individual stock pages should display some basic statistics ○ Stock code ○ Stock price ○ Stock performance ○ Total current worth ○ Total paid for ○ Total profit if all units of an individual stock were sold ● Stocks should automatically update or remove itself from the user assets/owned stocks page at each transaction.

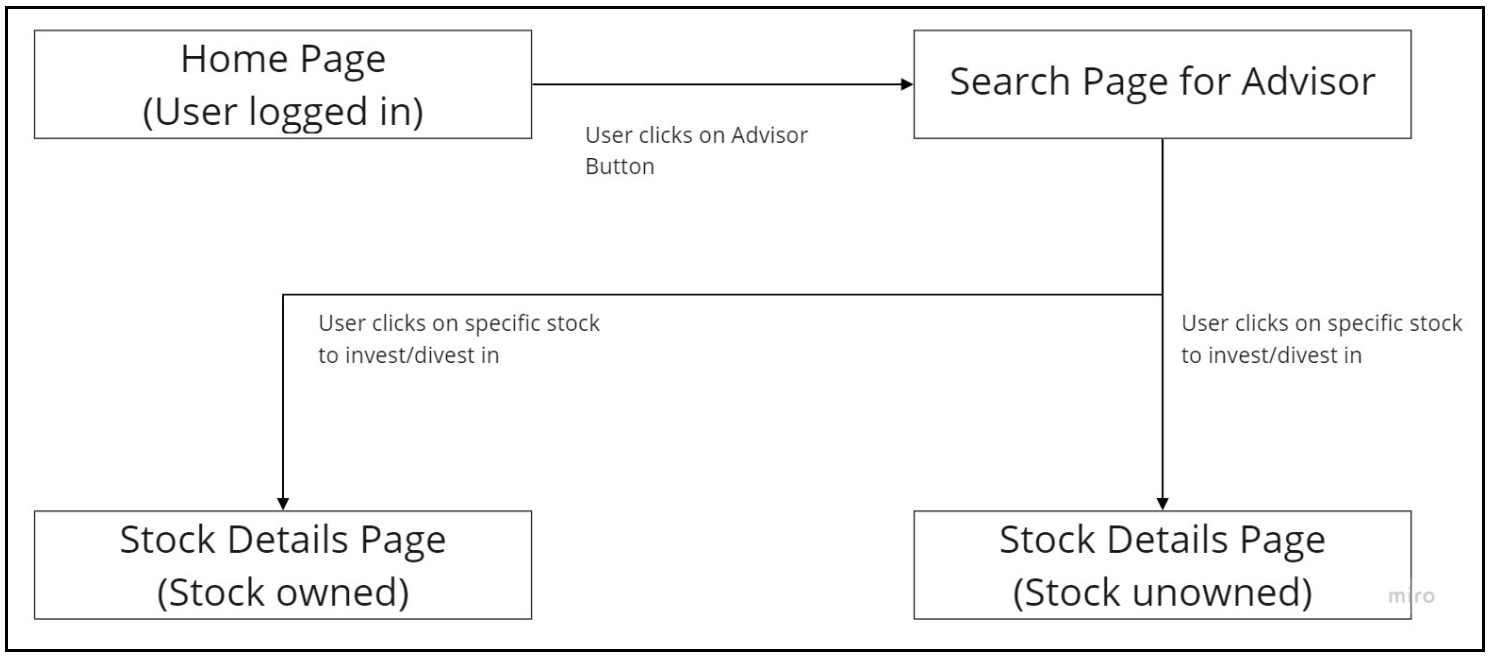
User Story 2.8



Registered users account deletion flow diagram.

Name	Account Deletion
Acceptance Criteria	● User can click delete button on MyAccount page ● Allow a check before actually cancel the account ● Account scrubbed from database ● User can no longer login with those credentials and can create new account with the email address

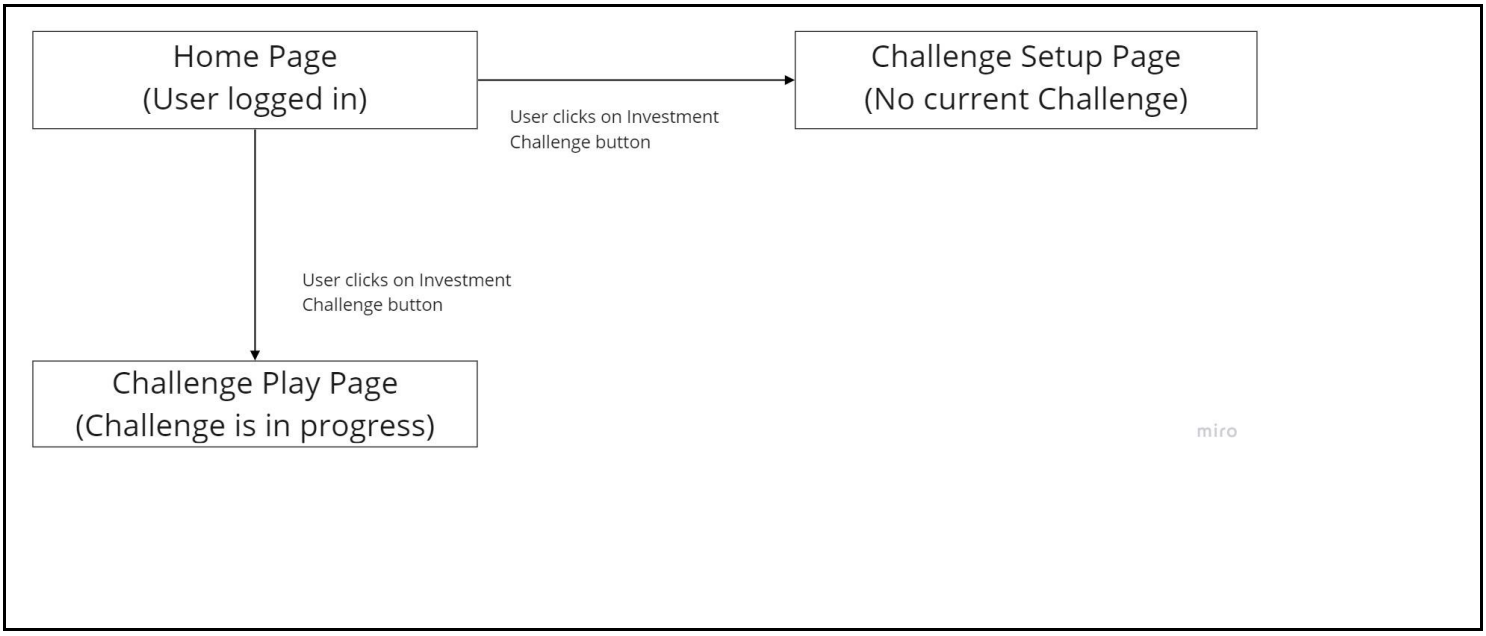
User Story 2.9



Registered users having access to the investment advisor features to guide them on stock investments.

Name	Investment Advisor
Acceptance Criteria	● Should have an advertisement link at the home page. ● Investment Advisor will suggest three stocks to invest in everyday for the user. ● Investment Advisor can suggest selling 1 or more owned stocks per day.

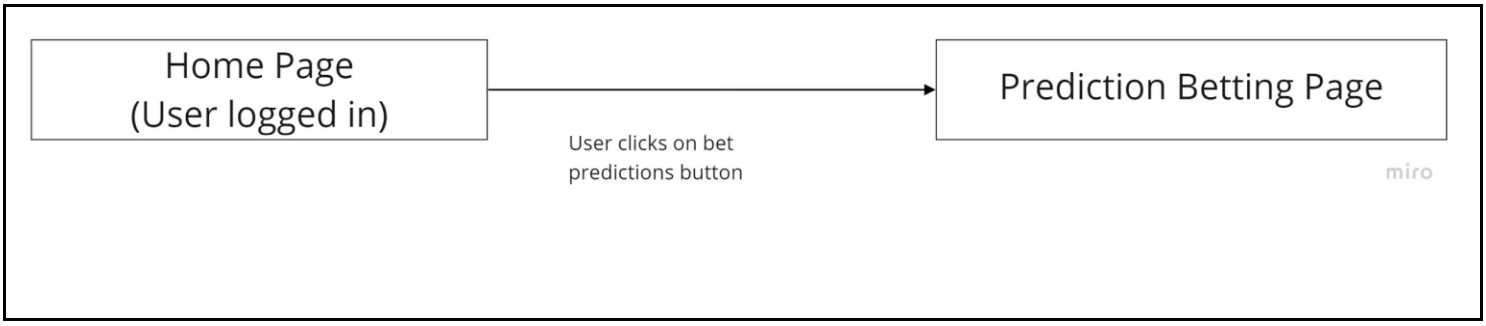
User Story 2.10



Registered users challenging the system for a chance to increase the profits for an investment.

Name	Investment Challenge
Acceptance Criteria	● Challenges are 1 week in length ○ Can start on any day ● Players/Users can issue a challenge to the system. ○ Player specifies a starting amount that can only be used within the challenge. ○ Players cannot choose less than 5 stocks to be used in the challenge ○ At the end of the challenge, profit is compared to determine the winner ● There should be a button in the home page to access the gamepage. ● Users can see their current challenge information.

User Story 2.11



Registered users are able to bet on a stock's performance for a chance to win money off the bet but they also lose their money if their predictions are wrong.

Name	Prediction Betting on Stocks
Acceptance Criteria	● There should be a button in the home page to access the gamepage ● Users can see their current bet information ● User predicts the range of percentage amount that a specific stock will increase/decrease at the end of the day/week/month ● User's can only make one concurrent bet at a time. ● User can place a bet on their prediction ○ User wins the bet if the increase/decrease falls within the selected range. ○ The reward is the initial bet multiplied by the reward multiplier dependant on the period selection ○ If they lose, they lose the money they betted.

4.0 System Architecture

4.1 Overview

To ensure quick delivery of the project within the limited timeframe, the software stack must balance developer familiarity against the suitability of the software in quick development and deployment. An overview of the software architecture can be seen in Diagram 4.1:

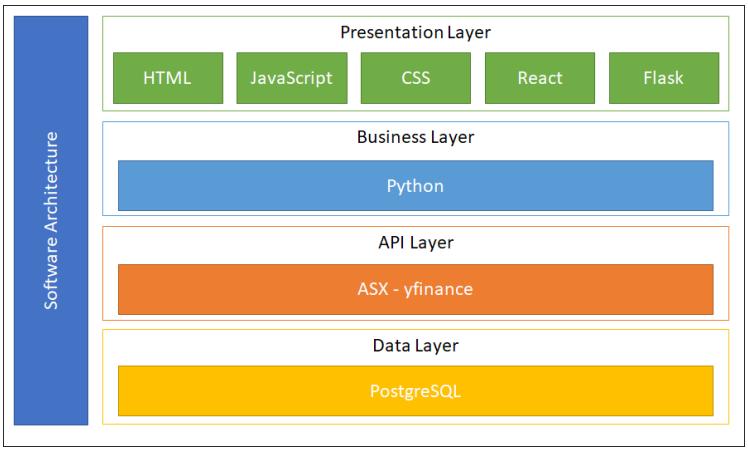


Figure 4.1: Basic software architecture

The presentation layer contains the components which external actors directly interface with to access the services provided by the web application.(Meier et al., 2010) This will be built on common web technologies: HTML, JavaScript, CSS with support from the React JavaScript library and the Flask web framework. The presentation layer is only concerned with relaying user input to the business layer, and rendering the content served back in a clear manner via the UI.

The business layer, also known as the application layer, contains the business logic that provides the web application's core functions.(Logi Analytics, Inc.) In the case of the Investment Simulator, all the user functionalities as shown in the Product Backlog, such as purchase/selling stocks and calculating current profits, will be processed in the business layer via Python.

The API layer contains third party API's which are necessary for the business layer to function. The main software package that will be used is yfinance(aroussi, 2019), which will allow the business layer to process and serve the relevant ASX stock information to the user via the presentation layer.

The data layer manages the required storage of user data, commonly through databases. As users can register and create an investment portfolio, it is important that their user data and stocks be tracked in a database to ensure persistence. This will be implemented with PostgreSQL. This user data is fetched by the business layer to perform functions such as profits calculation, storing records of stock purchase/sales and user validation for logging in.

4.2 Architecture Interactions

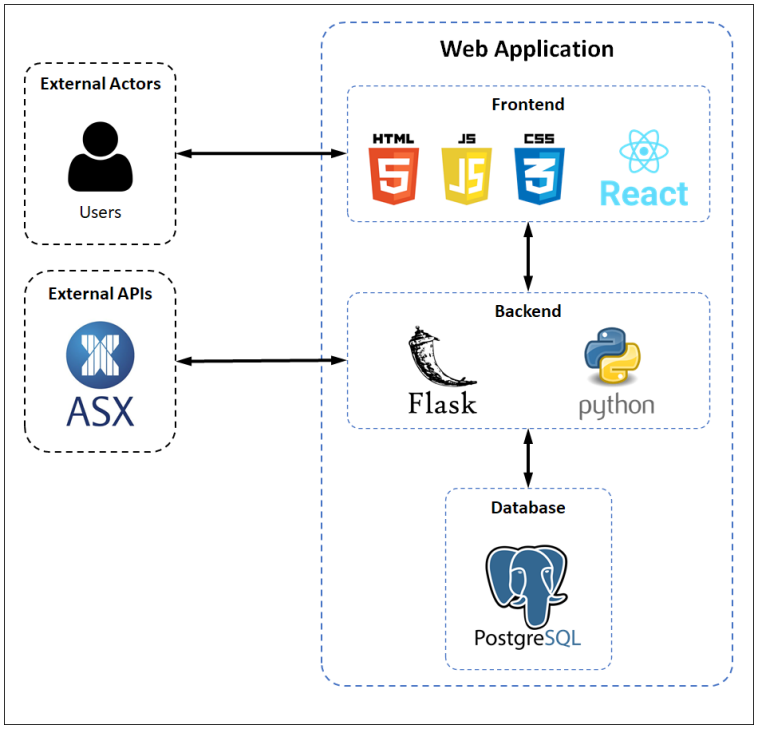


Figure 4.2: Architecture Interactions

4.2.1 External Actors

The only expected external actors interacting with the web application would be the users. This includes unregistered and unregistered users, as there is no distinction between them in the methods which they interact with the web application. Users interact with the frontend through their browsers via HTTP requests to our website.

Unregistered users only have access to a subset of the full functionality offered to registered users. The backend checks if the actors requesting access to certain pages have been validated before serving the page contents. Registered users have access to the full functionality of the web application.

4.2.2 Client Side/Front-end

Frontend leverages modern web application standards, including HTML, JavaScript and CSS(Cox, 2018). HyperText Markup Language (HTML) provides the basic structure and elements of the site. Cascading Style Sheets (CSS) modifies the presentation, formatting, style and layout of the HTML page, while JavaScript adds interactive elements, scripting, searching

and other dynamic behaviour. React is a JavaScript library that simplifies the process of creating interactive UI elements.

4.2.3 Server Side/Back-end

Flask is a web microframework that is designed to make development quick and easy, and is one of the most popular Python web application frameworks today(Ronacher, 2015). Flask does not place any restrictions on the structure of the code, however the Model-View-Controller (MVC) pattern is commonly used and is suitable for this web application. As Flask is written in Python, it will interact smoothly as part of a Python-based back-end.

Python is a popular, high level language designed to use simple syntax and allows for quick development.(w3.css) Python is also supported by a large variety of libraries, including numpy and SQLAlchemy which may be required to implement the web application's features. PostgreSQL is a free, open-source object-relational database management system (DBMS) that adheres closely to SQL standards. Part of choosing this DBMS is the familiarity our developers have with this package, which will ensure quick development and deployment of this web application.

4.2.4 External API's

In order to retrieve information on stocks listed on the ASX, yfinance will be used. yfinance is an external API that provides a Pythonic way of downloading historical market data from Yahoo! Finance.(aroussi, 2019) This indirect method of obtaining ASX information is due to ASX's fees requirements to use their Information Services Application, while yfinance is released on the Apache 2.0 License which allows for free use. yfinance also provides a level of abstraction that will allow easier deployment of the web application by placing less focus on learning an API.

5.0 References

- ASX, *ASX Homepage – Main Site* [online]Available at<https://www.asx.com.au/education/sharemarket-game.htm> (04/10/2020 22:15)
- SMG, *SMG Homepage - Main Site* [online]Available at<https://www.stockmarketgame.org/> (04/10/2020 22:17)
- NZX, *NZX virtual Trading Homepage - Main Site* [online]Available at<https://virtualtrading.nzx.com/> (04/10/2020 22:20)
- Meier, J.D., Homer, A., Hill, D., Taylor, J., Bansode, P., Wall, L., Boucher, R. Jr., Bogawat, A., (2010). Application Architecture *Guide - Chapter 10 - Presentation Layer Guidelines* [online]Available at<http://www.guidanceshare.com/wiki/Application_Architecture_Guide_-_Chapter_10_-_Presentation_Layer_Guidelines#Overview> (04/10/2020 21:40)
- Logi Analytics, Inc., *3-Tier Architecture: A Complete Overview* [online]Available at<https://www.jinfonet.com/resources/bi-defined/3-tier-architecture-complete-overview/> (04/10/2020 21:43)
- Aroussi, R., (2019). *yfinance 0.1.54* [online]Available at<https://pypi.org/project/yfinance/> (04/10/2020 21:46)
- Cox, L.K., (2018). *Web Design 101: How HTML, CSS, and JavaScript Work* [online]Available at<https://blog.hubspot.com/marketing/web-design-html-css-javascript> (04/10/2020 21:50)
- Ronacher, A., (2015). *The Pallets Project.Project.Flask* [online]Available at<https://palletsprojects.com/p/flask/> (04/10/2020 22:05)
- W3.CSS, *Python Introduction* [online]Available at<https://www.w3schools.com/python/python_intro.asp> (04/10/2020 22:09)