University of New South Wales
School of Computer Science and Engineering

COMP3900/COMP9900 20T1

Computer Science/Information Technology Project

Project Title: A Stock Portfolio Management System (Project 5)

Group: Yaaaahoo Finance

Team:

Anna Zhang
zID: z5114663
Email: anna.y.zhang@unsw.edu.au
Role: Developer

Anupam Chakraborty
zID: z5197405
Email: anupam.chakraborty@student.unsw.edu.au
Role: Developer

Mehri Amin
zID: z5113067
Email: mehri.amin@student.unsw.edu.au
Role: Developer

Oscar Fan
zID: z5114679
Email: oscar.fan1@student.unsw.edu.au
Role: Developer

Rachael Carson-Graham
zID: z5079853
Email: r.carson-graham@student.unsw.edu.au
Role: Developer, Scrummaster

Proposal submission date: 8 March 2020

Problem Statement

Background

Investors fall on a continuum from unsophisticated investors (do not understand financial products and are not confident buying/selling individual *investments*) to highly sophisticated investors (who have a deeper understanding of financial markets and are confident buying and selling a wide variety of *investments*).

"*Investments*" in this proposal could refer to any **one or more** of the following:

- Shares (either listed on the Australian stock exchange, or an international stock exchange)
- Bonds (again listed on either the Australian or an international stock exchange)
- Currencies (e.g. EUR - Euro, YEN - Japanese Yen, USD - US Dollar)
- Cryptocurrencies (e.g. Bitcoin, Ethereum, Dogecoin)
- Exchange traded funds (ETFs)[1]

We use the term "*portfolio*" to refer to the *investments* that an investor owns. E.g. an investor's *portfolio* may consist of shares listed on the Australian stock exchange, shares listed on one of the American stock exchanges, Bitcoin and multiple ETFs that are listed on the Australian stock exchange.

There are a wide variety of products that target the more sophisticated investors and provide fast access to both information and markets. These products tend to be more expensive and complex. Increasingly there are more products that target less sophisticated investors, but these tend to be very basic, and give a low level of control. These products are cheaper, but investors often do not know which individual *investments* they are adding to their portfolio, only a broad category (e.g. ASX 200 shares, dividends-paying shares). Both these types of products are discussed in the competitor analysis, below.

Problem Overview

When researching the products that are available to investors living in Australia, it became apparent that there is a gap in the middle of the two types of products listed above - there are no products that specifically cater to "mid-level" investors. These investors want some level of control over their individual *investments* (i.e. they don't just want to invest in ETFs), but they also don't want or need the higher complexity of more sophisticated investment products.

The characteristics of the mid-level investor (which is our user persona for this project) are:

- Has some level of financial understanding/ability
- Wants to invest more than $10,000
- May invest up to half their *portfolio* in ETFs, and trade individual *investments*
- Wants to be able to see an overall view of their *portfolio* in one place (e.g. ETFs, individual shares, cryptocurrencies)
- Wants to receive some recommendations on what *investments* to buy/sell, and when to buy/sell

[1] ETF (Exchange Traded Funds) is a term that is referred through throughout this proposal. For more information on what an ETF is, please refer to The Definitive Australian ETF Investing Guide (https://www.etfbloke.com/etf-investing/).

Competitor Analysis

Our competitor analysis focuses on 5 leading financial/investment products in the Australian market, and analyses how they fit the needs of the "mid-level" investor described above.

Company/Product	Overview	Advantages	Disadvantages
Raiz (formerly Acorns Australia)	Raiz is a financial service which aims to make investing more accessible for everyone, helping them save and invest their money. Raiz offers 6 *portfolio* options with varying levels of risk. These 6 *portfolios* are made up of 7 different ETFs (Exchange Traded Funds). Raiz does not offer the option to purchase *investments* - you can only invest in the *portfolio* they recommend. Raiz allows you to start investing from as little as $5.	• Very simple to use, with an easy setup, and clean, intuitive UI. • Recommends a *portfolio* based on the risk profile investors choose.	• Does not provide the ability to add individual *investments* to a *portfolio,* or modify the initial *portfolio* recommended by Raiz. • No recommendations on when to buy/sell *investments*. • No recommendations on individual *investments.* • Does not provide any additional information about *investments* or investing (e.g. no news or educational articles).
CommSec Pocket	Like Raiz, CommSec Pocket aims to make investing easy. CommSec Pocket offers 7 themed investment options and provides short educational videos/articles to help you learn about the share market. CommSec Pocket allows you to start investing from $50.	• Very simple to use, with an easy setup, and clean, intuitive UI. • Includes educational videos and articles to help users learn about investing.	• Only provides the ability to add one of 7 ETFs to the *portfolio* - no other *investments* are available. • Limited ability to modify the *portfolio* - investors need to add/remove the different ETFs to change their *portfolio*. • No recommendations on what kinds of *investments* are suited to an investor's risk profile, or when to buy/sell *investments.*
CommSec	CommSec is an online brokerage product that also provides research,	• Investors have a high degree of control over their portfolio and can add many different *investments*.	• A lot of features are unnecessary for mid-level investors, and end up being unused.

	recommendations and the ability to view live market updates.	• Provides recommendations on when to buy/sell *investments*. • Includes news that is relevant for investors.	This results in an over-complicated user experience. • Lots of clicks are required to find the information you are looking for. • Does not provide personalised recommendations - e.g. does not take an investor's risk profile into account.
CMC Markets	CMC Markets is an online trading platform that provides a large number of features relevant for sophisticated investors, including the ability to buy and sell multiple types of *investments* via different types of orders, access to news and research, and advanced charting capabilities.	• Investors have a high degree of control over their portfolio and can add many different *investments*. • Includes news that is relevant for investors.	• A lot of features are unnecessary for mid-level investors, and end up being unused. This results in an over-complicated user experience. • Lots of clicks are required to find the information you are looking for. • No recommendations on what kinds of *investments* are suited to an investor's risk profile, or when to buy/sell *investments.*
Yahoo Finance	Yahoo Finance provides incredibly comprehensive financial information and insights. The site includes information on Australian and international shares, currencies (including cryptocurrencies).	• Investors have a high degree of control over their portfolio and can add many different *investments*. • Provides detailed analysis and recommendations on whether to buy/sell/hold *investments*. • Includes news that is relevant for investors.	• A lot of features are unnecessary for mid-level investors, and end up being unused. This results in an over-complicated user experience. • Lots of clicks are required to find the information you are looking for. • Does not provide personalised recommendations - e.g. does not take an investor's risk profile into account.
Google Finance	Google Finance provides a very simple layout with minimal information. It provides users the ability to view a Market Summary, their own *portfolio*, and highlights (including news) of local and international markets.	• Investors have some degree of control over their *portfolio* - they can add a variety of *investments* to a watchlist. • Very simple to use, with an easy setup, and clean, intuitive UI. • Has a "You may be interested in" section of *investments* that (presumably) provides	• Provides no recommendations on when to buy/sell/hold *investments*. • The *portfolio* management functionality is limited. E.g. for shares, investors cannot add the number of shares that they own to the watchlist, only the company name/stock code. • Provides no information on how the "You may be interested in" recommendations

		recommendations on *investments* to research based on the data google has about users. • Includes news that is relevant for investors.	section is generated. E.g. it may be based on the investor's google search history rather than their risk profile.

System Purpose & Features

The main purpose of our system is to fill the gap in the market by building a robust, user-friendly system to help mid-level investors manage their investment *portfolios* and make better-investing decisions. Although there are many apps/websites available, most of them are either expensive or require in-depth knowledge of stock trading. Our system is targeted towards mid-level investors with some knowledge and experience in investing, who want to receive recommendations on which *investments* to make, and have the freedom to modify their portfolio to suit their individual needs.

The system has the following features:

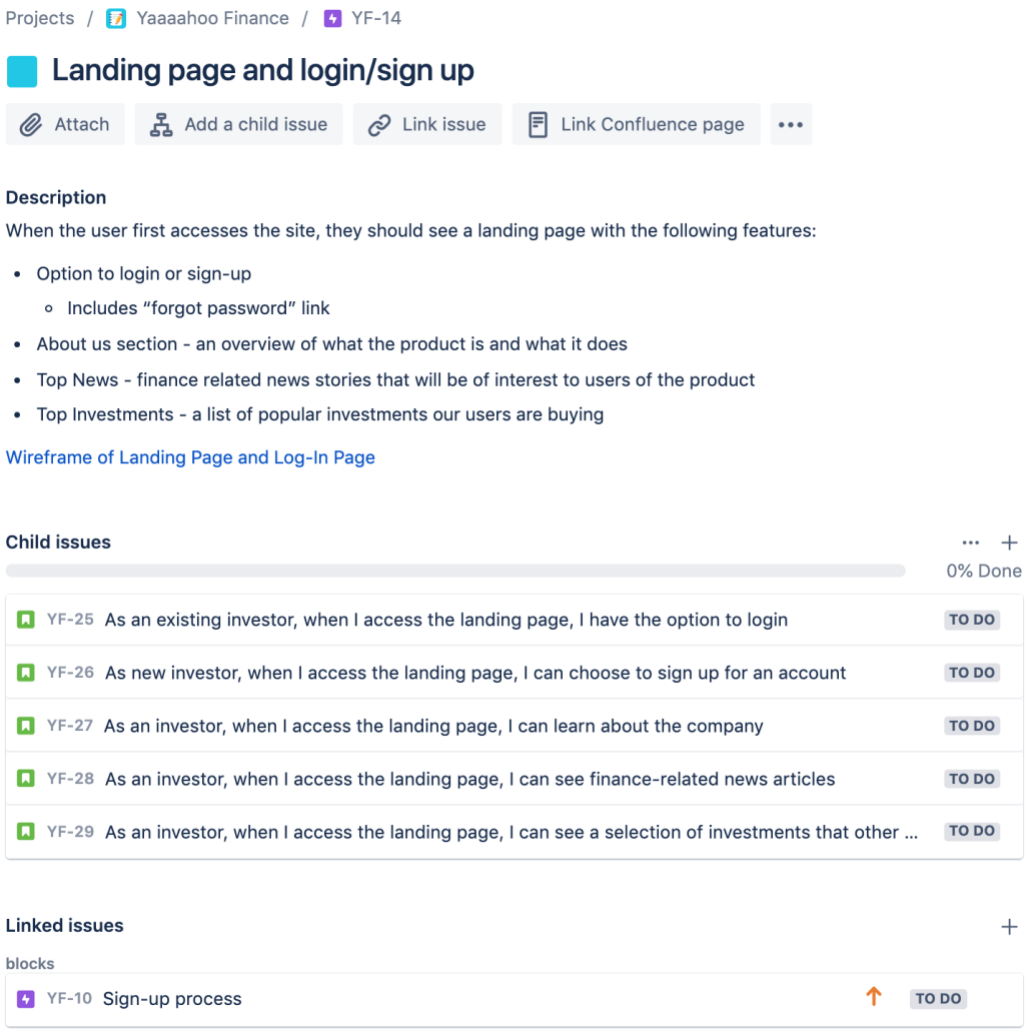
- A simple, easy to use website/Web-interface
- The ability to manage a portfolio of different investments, including:
    - Viewing the latest investment prices
    - Adding or removing investments to a portfolio
    - Viewing analysis of a particular stock
- News related to investments
- Share price predictions (using machine learning techniques), including recommendations on when to buy/sell shares
- Sentiment analysis – the outlook of a company from the public's standpoint
- A quick, easy sign-up process that provides users a recommended portfolio based on their profile and risk appetite.

Features in detail/Epics

Our first 3 epics will begin in Week 4. The epics are called Landing Page, Sign-Up Process, and Predict Future Share Prices. These epics will be further fleshed out in our next sprint planning meeting.

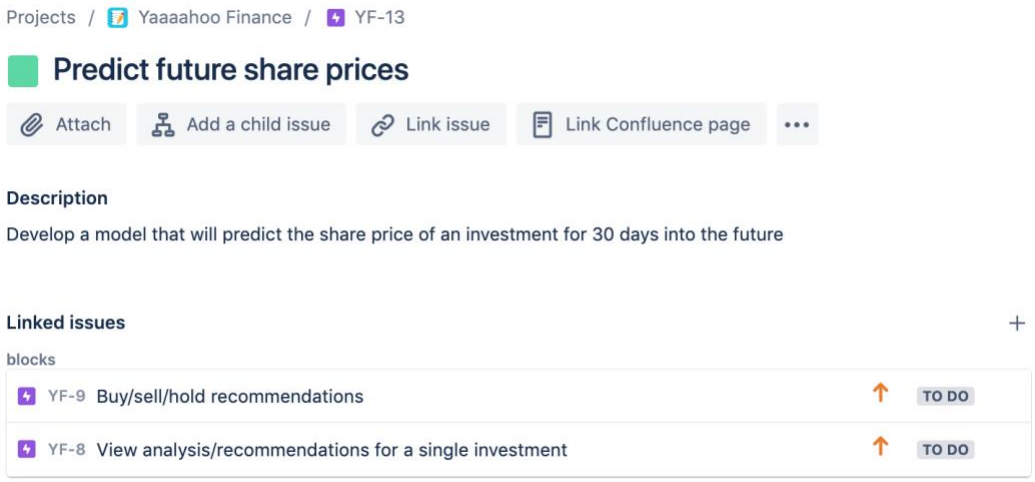
Landing page and login/sign-up

The landing page epic mainly requires front-end development. Back-end development included in this epic is functionality to sign-up and log-in. The purpose of this epic is to inform the user what our product is and incentivise them to sign up. Below details the scope of our project and the size of this epic is estimated to be one work week.



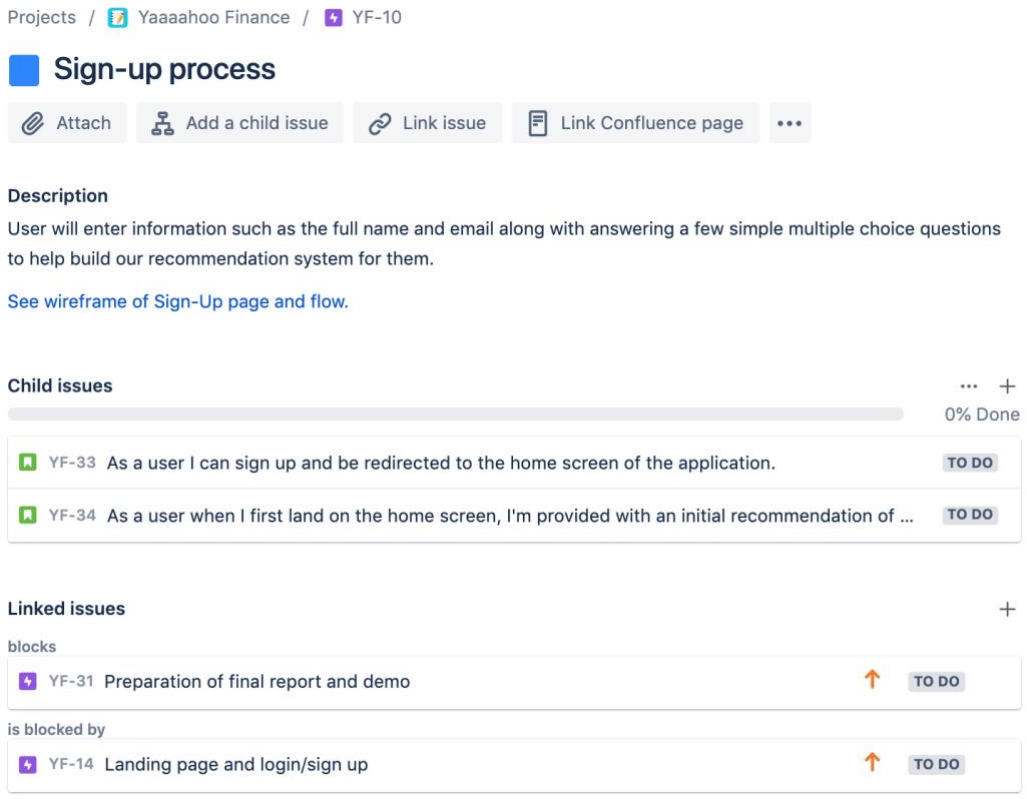
Predict future share prices

This epic is focused on building our prediction model to provide the user a 30-day prediction of a given share. This is a back-end development and data science heavy epic. Once back-end development is complete, front-end development is required to display this information to a user. The scope will be fleshed out further in our next sprint planning session and the estimated size of this epic is three work weeks.



Sign-up process

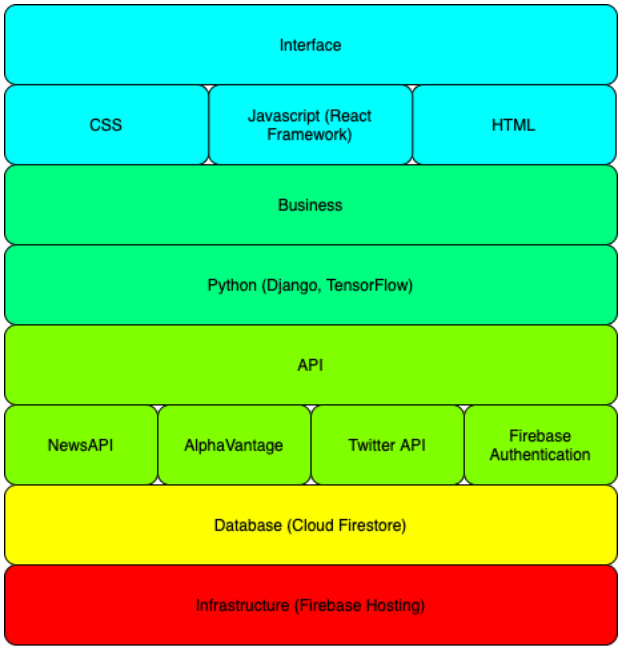
After the landing page and log-in/sign-up functionality is complete from our first epic, the next front-end development heavy epic is the sign-up process. This involves creating a questionnaire for users to complete which we then store to base our recommendation system. This scope of this epic is detailed in the image below and the size of this epic is estimated to be one work week.



Additional detail for the sign-up workflow:

- When a user successfully signs up, we will ask them questions about their age, income, age they want to retire at, what investments they already have and how risky they want their *portfolio* to be. The main purpose of the questionnaire is for us to better understand the users and the kinds of stocks they might be interested in.

Software Architecture



Interface Layer

HTML, CSS, and Javascript will be the fundamental tools we use to create our responsive application. The user interface will be built with a React framework. Aside from it being one of our team member's area of expertise, React ensures fast rendering due to its use of a virtual DOM, it has a fast learning curve, and enables component creation which makes code reusability and maintenance much easier. Integrating with other libraries will be easier. Some libraries we are considering using include Material Design for fast UI development and Chart JS for data visualisation.

Business Layer

Python will be our language of choice for the backend, as it is a simple and effective language that will enable us to utilise the Django framework to connect our backend to the frontend. Furthermore, Python is the most popular language for many machine learning frameworks, such as TensorFlow and PyTorch. During this project, we will be using TensorFlow to build our predictive models, as it is a library that is optimised for commercial performance, (which is highly important in high stakes environments such as stock market investing) and is compatible with Firebase.

API Layer

NewsAPI

NewsAPI is an API that provides international news from multiple sources. Although there is a 15-minute delay for new news articles, for the scope of our project, it provides the perfect functionality to show the user relevant news to their stock portfolio.

Alpha Vantage

An open source stock market API that will allow us to retrieve historic stock market data in real-time.

Alpha Vantage will be useful for displaying historical and current data to the user, as well as updating our predictive models. We will also be querying Alpha Vantage's search endpoint in order to provide a responsive search bar.

Firebase Authentication

Firebase Authentication will provide us with the ability to provide users login compatibility with all the popular social media platforms, such as Facebook and Twitter, as well as Google. Not only is Firebase Authentication highly customisable, it also utilises industry standard security (OAuth 2.0), so we can provide a secure and personalised experience to our users.

Twitter API

We will leverage Twitter's API in order to get the top tweets related to companies, which will be fed into our model to provide users with a real-time sentiment classification (Good, Neutral, Bad).

Database Layer

Cloud Firestore is the database of choice for Firebase. It is a NoSQL cloud storage system, which keeps data synced between client and server-side development. It also supports offline mode, so our web application won't be impacted by client-side issues with the Internet.

Cloud Firestore stores data in collections of JSON-like documents, and for our web application, we will be using three main collections: one for user authentication, one for storing historic market data, and one for the user profile which will store their portfolios.

Infrastructure Layer

Firebase is an app development platform that not only provides a cloud database solution (so we are not required to look for an alternative database solution) it also provides hosting. Firebase Hosting provides the ability to host web apps, content and microservices, and is fast, secure and easy to set up via the Firebase platform.

System Modules and Flow

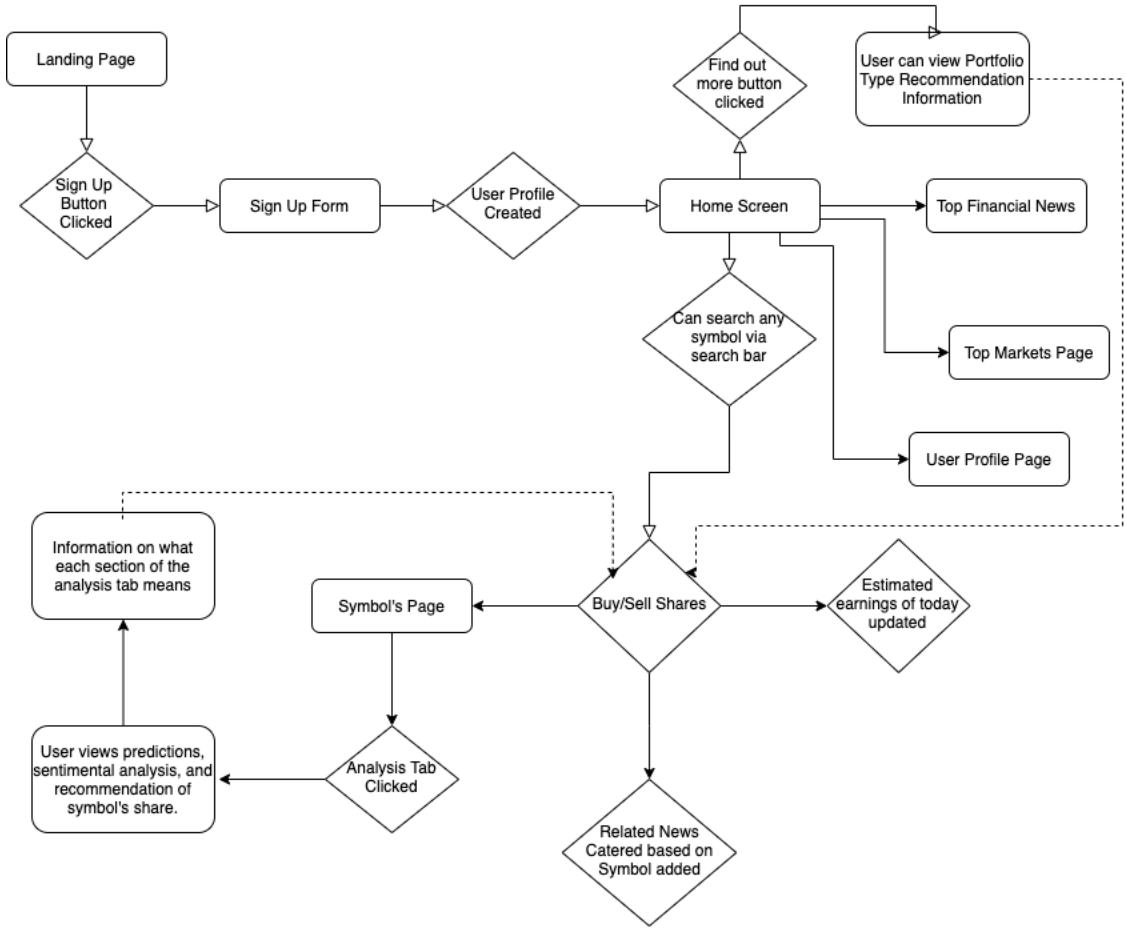
Throughout this section we use the term "symbol" which is simply a term used to represent a stock, option, mutual fund, or other security that trades on an exchange. Symbol is synonymous with the term *investment*, described in the Problem Statement above.

Overview of user flow

Below is an overview of the user flow from sign up to building their portfolio. The main pages of our application are the Landing Page, Sign-up Page, Home Screen and Symbol's Page. These pages will be discussed in more detail further below with wireframes. The flow is focused on returning the user back to either buying/selling shares to improve their portfolio, viewing their portfolio performance, or keeping informed with the market and the companies they have a share in.

A brief overview of the flow starts with the user seeing the landing page incentivising them to click a call-to-action button 'Sign Up'. This will initiate the sign-up flow where a user is asked a few questions such as their age or how aggressive/safe they want to be in their investments which we use to base our recommendations for them throughout the application. Once the sign-up process is complete our first recommendation is performed on the home screen with a suggestion of what type of portfolio a user

should build. A button ‘Find out More’ can be clicked to view more information about our suggested portfolio type and how to help the user build such portfolio type. On the home screen a user can search for any kind of symbol, start adding symbols to their portfolio, or view top financial news. They can view a symbol’s page which has a summary of their current price and historical performance and our own analytical assessments of said symbol such as our 30-day prediction, sentimental analysis, and recommendation to buy or sell a symbol’s share based on our user’s profile. Once a user adds shares to their portfolio, their estimated earning of the day are updated and the top news section on the home screen is updated to related news of the companies in their portfolio to keep them informed. Other pages such as Top Financial News, Top Markets and User Profile are accessible via a sticky docker at the bottom of the website.

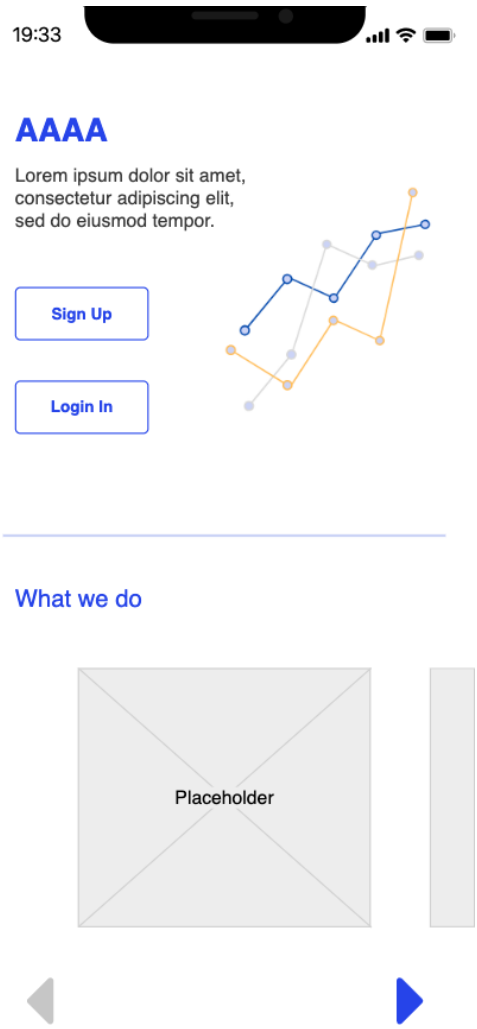


Our design is driven by mainly three UX Design Principles: meet the user’s need, usability first and designing with accessibility. These principles are held up by creating a minimal design that helps to reduce clutter and information overload which our competitor’s designs, such as Yahoo Finance, tend to create for mid-tier investors. Additionally, with our recommendation system basing off of our user’s profile we bring the focus back to providing a product that meets the users need.

To adhere to our goal of creating a responsive website, we are applying a mobile-first design[2] which means we start the product design from the mobile-end since it has more restrictions, then expanding our features to tablet and desktop versions. Hence, why our wireframes are in a mobile frame.

Landing page

Our landing page will provide information to users about what our product is, how it is different to our competitors, and why users should use our product. We'll have call-to-actions so users can sign up or log in.

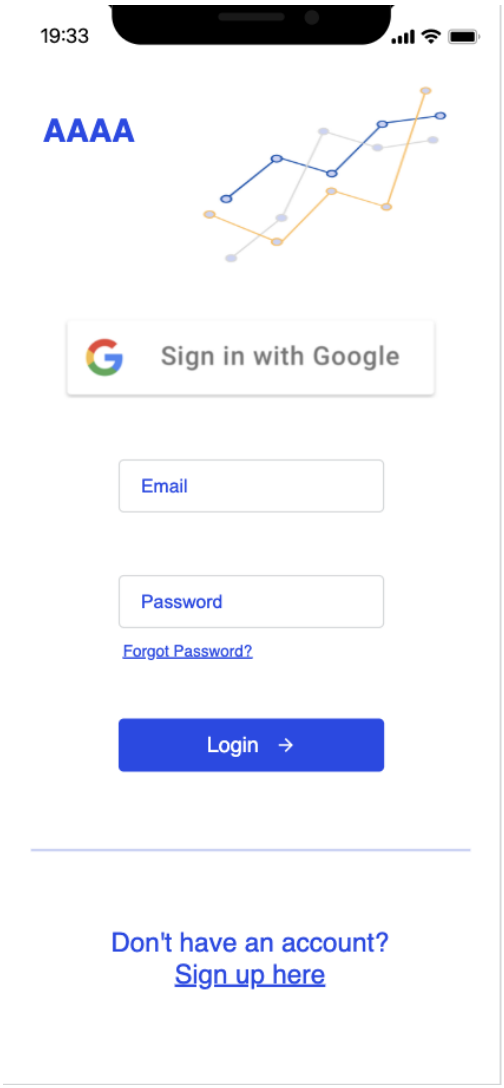


Login page

At the login page user can sign-in with Google, since our infrastructure is built on Google's Firebase platform, making signing in even easier. We will also have functionality to retrieve forgotten password,

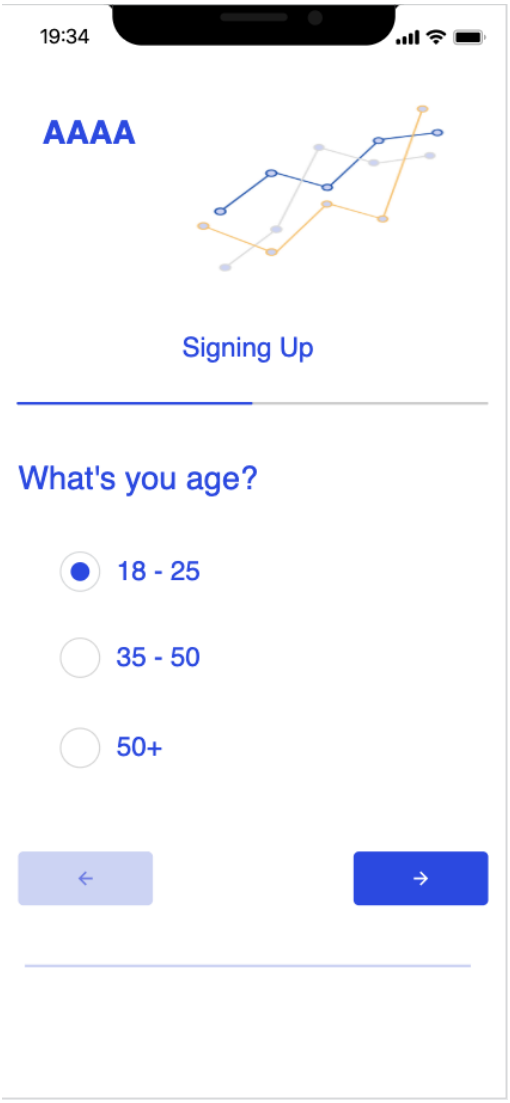
[2] Vincent Xia. (2017, December 21). What is Mobile First Design? Why It's Important & How To Make It [Web log post]. Retrieved from https://medium.com/@Vincentxia77/what-is-mobile-first-design-why-its-important-how-to-make-it-7d3cf2e29d00

and a redirect to our sign-up flow for users who do not have an account. New users who sign up with Google will be redirected to our sign-up flow as well.



Sign-up flow

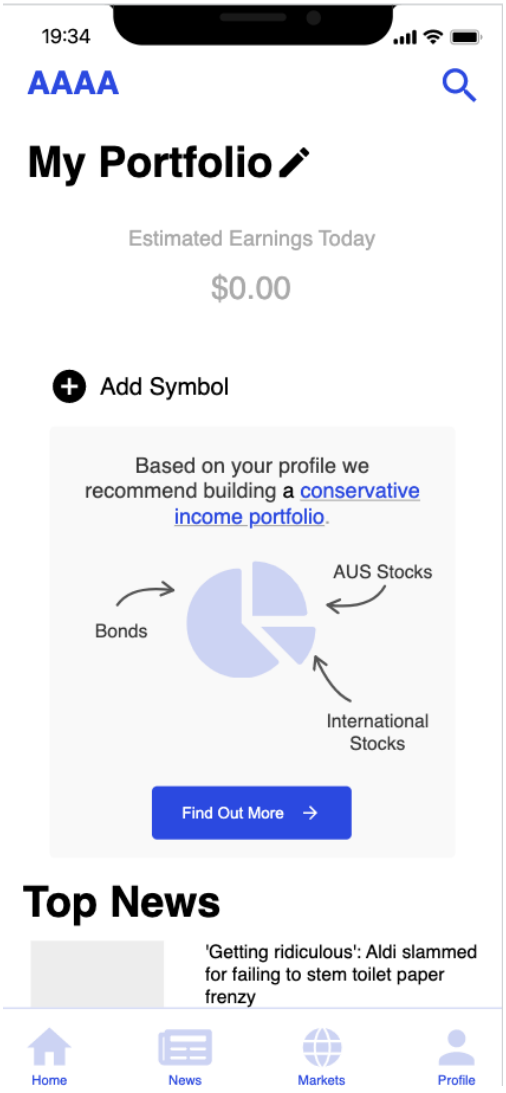
When signing up, we ask our users a few questions such as their age, how much they are looking to invest, and whether they want to be safe or aggressive with their investments. We use this information to calculate our recommendations for a user such as whether they should buy or sell a certain share, or what kind of portfolio they should build. We discuss this in more detail in the next frame.



First time logging in

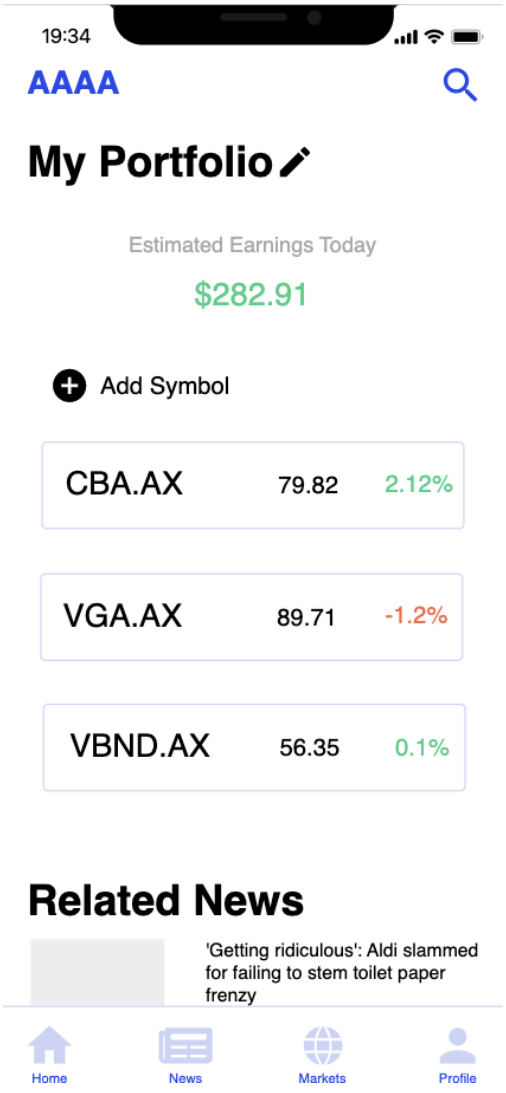
When a user first logs in, based on their profile we provide an initial recommendation on what kind of portfolio they should build. For example, for a user who is closer to retirement and not necessarily interested in growth but rather a safe, reliable income stream, we would recommend a **conservative income portfolio**. When they click to find out more about the portfolio type it will pop up a screen with a bit more information:

E.g. "The goal for a conservative income portfolio is to maintain the portfolio value and provide an income stream. To build this kind of portfolio we recommend investing 60% of your holdings to bonds, 25% in Australian stocks, and 15% in international stocks."



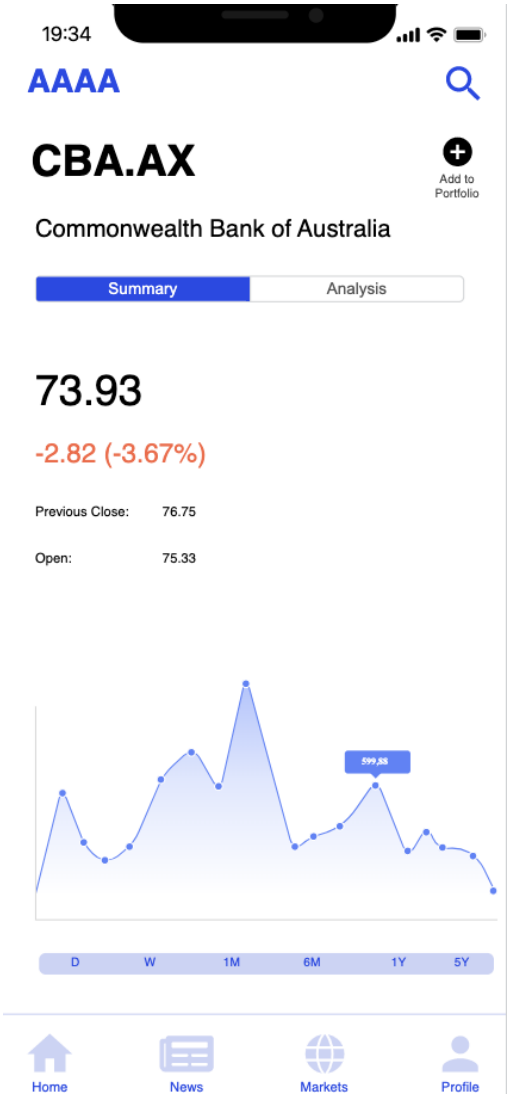
Home screen

The home screen is simple and focused on just showing the most important information required for an investor such as how much they earned today and viewing the current price of their shares. As they scroll down, they can view a list of catered top news related to the companies in their portfolio. Other actions a user can take on the home screen include searching symbols in the upper right corner, editing their portfolio, adding a symbol to their portfolio, navigating through the bottom dock to go to the top news page, top markets page, or profile page. When a user selects one of the symbols in their portfolio, such as “CBA.AX” it will navigate to the symbol’s page.



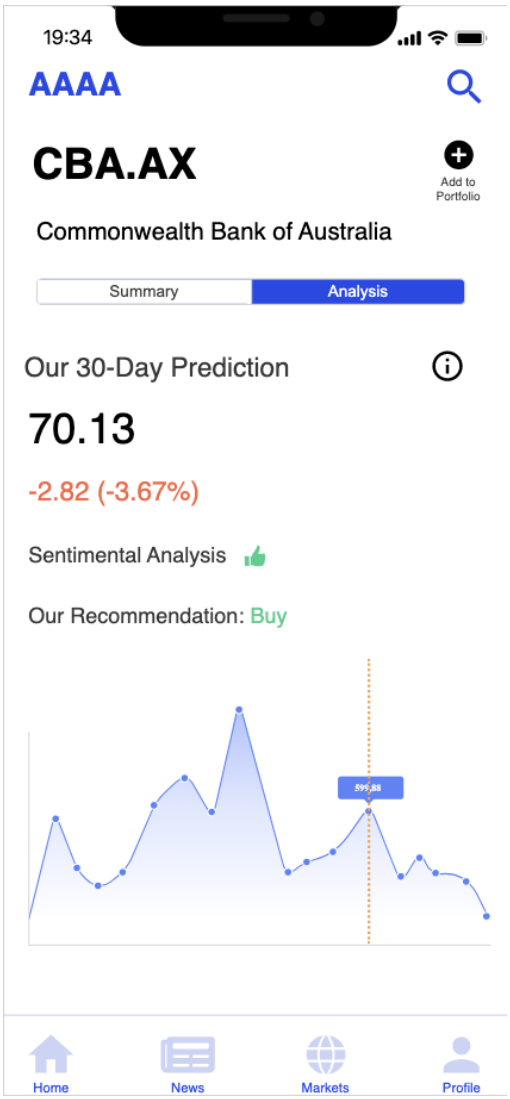
Symbols page

The default tab of the symbol page is the summary tab where a user can view a symbol's current price, previous close price, open price, the historical data in different time views e.g. day, week, month, or year view. User also can add the symbol to their portfolio if it is not already in their portfolio.



Symbols Analysis tab

The analysis tab provides our recommendation on whether to buy or sell a symbol's share, our 30-day prediction and sentimental analysis based on whether the current media coverage of a certain company is positive or negative. When the information icon on the right side of the screen is selected, a pop up appears with information about what each analysis on this page means for a user. Based on the user's profile, sentimental analysis and our 30-day prediction we will be able to provide our user a recommendation on whether to buy or sell a symbol's share. We will be drawing a graph that extends the current price to 30-days to visually show our 30-day prediction. The price displayed on this page will be the price we predict the share will be 30 days in the future.



Project Management and Stand-up/Sprint Schedule

Gantt Chart

Key

- Purple: Documentation
- Darker blue: Front-end and back-end development required
- Green: Machine learning model development
- Lighter blue: Front-end and back-end development required, will involve integration with third-party APIs
- Yellow: End to end testing, may involve small additional development work (e.g. bug fixes), but no new functionality should be added
- Lines between epics are dependencies:
    - Grey lines - subsequent epic cannot be started until previous epic is completed
    - Red lines - subsequent epic cannot be completed until previous epic is completed (development on the two can happen in parallel however).

Milestones and deadlines

Milestone	Owner	Deadline	Status
Complete project proposal	All	8 March 2020	Done
Develop model for predicting investment prices	Anupam Chakraborty / Oscar Fan	20 March 2020	Not started
Develop sentiment analysis model	Anupam Chakraborty / Oscar Fan	3 April 2020	Not started
Landing page, login and sign-up workflows	Anna Zhang / Mehri Amin / Rachael Carson-Graham	20 March 2020	Not started
View existing portfolio and add investments	Anna Zhang / Mehri Amin / Rachael Carson-Graham	3 April 2020	Not started
View details and analysis of shares	Anna Zhang / Mehri Amin / Rachael Carson-Graham	3 April 2020	Not started
Product complete and ready for testing	All	20 April 2020	Not started
End to end testing of product	All	26 April 2020	Not started
Final report and demo preparation	All	28 April 2020	Not started

Sprint schedule

Our team has agreed on the following sprint schedule, and the activities we will do in each session.

What?	When?	Description	Who is involved?
Sprint review	Every second Friday during the lab time, specifically: Mar 13, 2020 Mar 27, 2020 Apr 10, 2020	Before doing sprint planning, we will: • Review the stories completed in the previous sprint • Demo our code/functionality to each other and our mentor • Give each other feedback on any improvements that can be made • Celebrate what we have achieved	Whole team to demo what they have worked on and give each other feedback.

Sprint planning	Every second Friday during the lab time, specifically: Feb 28, 2020 Mar 13, 2020 Mar 27, 2020 Apr 10, 2020	Pre-requisite: Throughout the previous sprint, the entire team will add any additional stories/tasks we think need to be worked on for the project. During sprint planning: 1. Determine team capacity for the sprint based on velocity and team commitments (e.g. does anyone have anything coming up in the next two weeks that may mean their capacity is higher/lower than usual?) 2. Review backlog and determine the priority of stories 3. Estimate (using story points) the high priority stories 4. Based on the estimates, and our team capacity, determine how many stories we want to commit to this sprint 5. Move these stories into the sprint in Jira 6. Discuss and document acceptance criteria for the stories in the sprint, so we have a shared understanding of what work needs to be completed 7. Assign stories to team members to work on 8. Also leave some stories unassigned so team members can pick them up if they complete their assigned tasks	Scrummaster (Rachael Carson-Graham) to update Jira during the sprint planning meeting, e.g. making sure estimates/acceptance criteria are added to stories, moving stories chosen to be worked on into the sprint Whole team involved in discussion on capacity, priorities, estimates and acceptance criteria.
Retrospective	Every other Friday during the lab time, specifically: Mar 6, 2020 Mar 20, 2020 Apr 3, 2020 Apr 17, 2020	In scrum, retrospectives typically happen on the last day of a sprint (often a Friday), with sprint planning happening either on the same day, or on the next working day (often a Monday). This allows the team to reflect on the sprint and immediately implement any changes/improvements in the following sprint. We discussed this approach as a team, and decided that due to time constraints (only 2 hours in the lab each week) and other commitments, this approach wouldn't be feasible for us.	Scrummaster (Rachael Carson-Graham) to facilitate the retrospective and record the outcomes/action items Whole team involved in discussion and

		Using the lab session one week for sprint review and sprint planning, and then the lab session the following week for retrospectives will allow us to dedicate a decent amount of time to each of these meetings, without trying to rush and get through both of them. We anticipate that some retrospective outcomes may be able to be implemented immediately, while others may not be implemented until sprint planning the following week.	agreeing on outcomes/action items
Stand-ups	Every Friday during the lab, and at one other session during the week (dates/times TBD)	Due to other study/work commitments, we will not be working on the project all day every day, as a normal scrum team would. Because of this, we don't think we need a daily stand-up. Instead, we will do a stand-up twice per week, with each team member talking about: What they have done since the last stand-up What they plan to do before the next stand-up Any blockers or points of note they need to communicate to the rest of the team. We don't envisage this taking more than 20-30 minutes, so will do this at the start of each lab before sprint planning/retrospectives, and via video chat on another day during the week.	Whole team to present their updates.
Ad hoc updates/discussions	As needed via video chat or Facebook Messenger	We expect that as we are developing our project, we will have questions to ask the rest of the team, or need advice/opinions. Rather than waiting for the next regularly scheduled stand-up, we should try to resolve these as quickly as possible.	Anyone in the team can schedule a video chat or initiate a group conversation, all or some of the rest of the team can

		We already have a Facebook Messenger group set up, which we will use for discussions and to ask and answer any questions that come up throughout the week. We are also using when2meet (https://www.when2meet.com/) to set up ad hoc video chats for anything that needs a more detailed discussion than what can be achieved over Facebook Messenger.	participate as needed.

Team practices

In addition to agreeing on when and how we will meet to discuss the work, we have also agreed on best practices to follow when we are working on the project.

Practice	**Why**
Preference for automated testing if we can	Automated testing has many benefits, including: • Time saving - once set up, an automated test suite can run faster than a human can carry out manual testing • Reliability - the same test cases are run each time in the same way, regardless of who is running the testing. • Confidence - we can be confident that when we make changes, if the tests pass, then the software is still doing what we expect it to. We recognise the benefits of automated testing, however we also recognise there can be a significant time investment in learning, setting up, and maintaining automated test frameworks. As we have limited time to complete this project, we decided to take a pragmatic approach to try and use automated testing where we can, but also make sure we are meeting our deadlines/milestones so that we have a workable product at the end of term.

Follow industry standard coding styles for each technology/layer	Coding standards help ensure that code can be easily understood and maintained. A consistent code base should look like one developer has written it in a single session. This means that developers can focus on understanding the logic of the code, rather than trying to interpret its layout and style. For almost any language and framework, there are multiple companies using these technologies. Rather than trying to reinvent the wheel, we should use industry standard coding styles for each technology and layer. This will enable us to get the benefits of coding standards, without spending too much time agreeing on what those coding standards are. The coding standards we intend to use are as follows: • Python: PEP8 (https://www.python.org/dev/peps/pep-0008/ Python Software Foundation, n.d.) • Django: Django Coding Style (https://docs.djangoproject.com/en/dev/internals/contributing/writing-code/coding-style/ Django Software Foundation, n.d.) • Cloud Firestore (a JSON-based, NoSQL data store): MAPR Data Modeling Guidelines for NoSQL JSON Document Databases (https://mapr.com/blog/data-modeling-guidelines-nosql-json-document-databases/) • React: Airbnb's Style Guide (https://github.com/airbnb/javascript)
Peer review of work	Because the team has different skills and experiences, we don't all have the same level of ability in the technologies we plan to use. Doing strict "code reviews" in these circumstances may not be an effective use of time. E.g. if one person is an expert in React and writes some code for the front-end, another person who is less familiar with this technology may not be able to add much value by doing a code review, and may not detect any errors. We decided to take a pragmatic approach and ensure that each piece of work is reviewed to some extent. This may be an independent code review, a code walkthrough, or a discussion of what has been implemented. The types of questions we want to be able to answer in these reviews are based on Atlassian's best practice for doing code reviews[3], and include: • Are there any obvious errors in the logic being implemented?

[3] Atlassian. (2020). Why code reviews matter (and actually save time!). Retrieved from https://www.atlassian.com/agile/software-development/code-reviews

	• Can the person who implemented the code quickly explain it to a person who has not seen the code before, in a way they can understand? • Have the acceptance criteria in the story being implemented? • Have any automated tests been written? If not, why? What manual testing has been done instead? • If automated tests have been written, what different scenarios do they cover?
Develop on branches, and don't push to master until the code is tested, working and reviewed	We want to make sure that the master branch of our code always reflects the latest workable version of our project. If we were releasing regularly throughout the term, the master branch would be the version of our code that can be deployed to production at any time. Working on branches allows us to safely make changes to our code without worrying that they will break other people's changes. Similarly, when we are working on a branch, we know that we will not have other people's changes inadvertently change or break what we are doing. To ensure that the master branch is always stable and deployable, we will not push to master until the code on a branch is tested, working and reviewed. Branches can be merged to master after a pull request is raised and reviewed by another team member (in line with our review process described above).

References

Savona, B. (2018, April 24). What's in a name? Why micro-investment app Acorns changed its name to Raiz Invest. *Smart Company*. Retrieved from https://www.smartcompany.com.au/marketing/branding/whats-in-a-name-why-micro-investment-app-acorns-changed-its-name-to-raiz-invest/

What is Raiz? (2018). Retrieved from https://raizinvest.com.au/support/

What makes Raiz different? (2018). Retrieved from https://raizinvest.com.au/support/

How do I choose an Investment portfolio? What do I own? (2018). Retrieved from https://raizinvest.com.au/support/

What is my investment portfolio? What do I own? (2018). Retrieved from https://raizinvest.com.au/support/

Can I invest in individual stocks, bonds or other securities of my choice through Raiz, or opt out of some of Raiz' investments? (2018). Retrieved from https://raizinvest.com.au/support/

Is there a minimum investment amount required to open an account? (2018). Retrieved from https://raizinvest.com.au/support/

An app so simple, it turns a non-investor into an investor. (n.d). Retrieved from https://www.commsec.com.au/products/pocket.html

Zucchi, K. (2019, June 25). How to use Yahoo Finance. *Investopedia*. Retrieved from https://www.investopedia.com/articles/investing/091714/how-use-yahoo-finance.asp

Why switch to CommSec? (n. d.). Retrieved from https://www.commsec.com.au/why-join-commsec/making-the-switch-to-commsec.html

CMC Markets Stockbroking. (n.d.). *Stockbroking standard user guide* [Brochure]. Retrieved from https://assets.cmcmarkets.com/pdfs/stb_standard_quick_start_guide.pdf

What service will I receive as a customer of CMC Markets? (2019). Retrieved from https://www.cmcmarkets.com/en-au/support/faqs/switching

Alpha Vantage Inc. (n.d.). Alpha Vantage API Documentation. Retrieved from https://www.alphavantage.co/documentation/

Twitter Inc. (2020). Docs. Retrieved from https://developer.twitter.com/en/docs

News API Documentation (n.d.). Retrieved from https://newsapi.org/docs

Cloud Firestore (n.d.). Retrieved from https://firebase.google.com/products/firestore

Firebase (n.d.). Retrieved from https://firebase.google.com/

Tensorflow Core (n.d.). Retrieved from https://www.tensorflow.org/overview/

Encode OSS Ltd. (n.d.). Django REST framework. Retrieved from

https://www.django-rest-framework.org/

Atlassian. (2020). Scrum Overview. Retrieved from https://www.atlassian.com/agile/scrum

Ely Hechtel. (2015, July 9). Top 10 Benefits of Automated Testing [Web log post]. Retrieved from https://saucelabs.com/blog/top-10-benefits-of-automated-testing

Multidots (2020). Importance of Code Quality and Coding Standard in Software Development. Retrieved from https://www.multidots.com/importance-of-code-quality-and-coding-standard-in-software-development/

Gienow, M. (2018, May 18). Don't Mess with the Master: Working with Branches in Git and GitHub. *The New Stack*. Retrieved from https://thenewstack.io/dont-mess-with-the-master-working-with-branches-in-git-and-github/