

University of New South Wales

# STOCK PORTFOLIO MANAGEMENT SYSTEM PROPOSAL(IT CAPSTONE PROJECT)



Name	Role	sID	Email
Rui Liang	Scrum Master	z5174445	rui.liang1@student.unsw.edu.au
Zhicheng Tao	Developer	z5148738	z5148738@unsw.edu.au
Chuguan Tian	Developer	z5145006	z5145006@unsw.edu.au
Hao Jiang	Developer	z5146830	z5146830@ad.unsw.edu.au

A project proposal submitted for COMP9900/19T3

October 7, 2019

## **Abstract**

Since the living standards have improved so much in recent years, more and more people tend to spend their spare money on stock investment as an extra income source. However, due to a lack of financial background and techniques about stock data analysis, many people buy stocks blindly which causes great loss. There are a lot of stock analysis software on the market, however, they don't have the ability to help users with intelligent analysis or predict the trend of stock price, and most of them are hard to use. Therefore, we propose to develop a smart stock portfolio management system that equipped with modern machine learning prediction tools as well as factor analysis tools to help investors to buy stock wisely. In this proposal we first introduce modern software development background, existing systems and our major objectives. Then we elaborate on a variety of epics and features that we plan to develop. Finally, a project development methodology will be introduced as well as the project schedule timeline.

# Contents

<b>1</b>	<b>Background Introduction</b>	<b>1</b>
1.1	Background . . . . .	1
1.2	Existing systems and drawbacks . . . . .	1
1.3	Aim and objectives . . . . .	2
1.4	Application architecture . . . . .	3
<b>2</b>	<b>Epics</b>	<b>4</b>
2.1	User administration . . . . .	4
2.2	Portfolio management console . . . . .	5
2.3	Stock search . . . . .	6
2.4	Stock detail information . . . . .	7
2.5	Stock price prediction . . . . .	8
2.6	Factor analysis . . . . .	9
2.7	Final epic selection . . . . .	10
<b>3</b>	<b>Methodology</b>	<b>12</b>
3.1	Project management . . . . .	12
3.2	Software and library . . . . .	13
3.3	Team members and responsibility . . . . .	13
3.4	Schedule . . . . .	14
	<b>Bibliography</b>	<b>15</b>

# Chapter 1

## Background Introduction

### 1.1 Background

Today, more and more people participate in the stock market, and it becomes a significant part of investment. However, most of people cannot benefit from investing their money on the stock market and even loss a large amount of money. Stock market then becomes a gambling place for them. There are a few reasons why more people tend to be losing money rather than earning in the stock market:

Firstly, many people are lack of basic knowledge of financial investment theory. Most of people do not understand the basic concepts of stock market. They cannot utilise stock data and technical indicators for analysis. Many people buy stocks blindly, and rely only on feelings and suggestion from friends.

Secondly, many people cannot realise the background and financial status of listed companies. When choosing stocks, people do not have time or a good channel to collect information about the fundamentals of a listed company, business performance and other information. However, stock price and the fundamentals of a listed company are closely related. By investigating the company's industry analysis, business analysis, and operational capability analysis, it is possible to predict the long-term trend of a stock.

Under this background, stock portfolio analysis system has emerged, guided by the value investment philosophy. Through the scientific calculation and analysis of a large amount of stock market data, it is designed to help investors to grasp the evaluation of stock price in a short period of time and guide them to invest their money more wisely.

### 1.2 Existing systems and drawbacks

Stock portfolio management systems has been well developed for many years, there is a variety of portfolio software that enable us to keep tack of our investments. For example, some popular stock portfolio management software are *Personal Capital*, *Quicken*, *Investment Account Manager*, *Morningstar.com*, *YahooFinance Portfolio Tracker*, etc[1]. Most of them share some common features such as:

- Investment portfolio tracking.
- Investment performance checking (net worth, capital gains, asset allocation, etc).

- Charts for visual analysis of price movement.
- Basic portfolio analysis.

However, most of them have the following drawbacks:

- Many software needs to be installed in the computer so that we cannot use them in the mobile devices conveniently.
- Many existing application is not free and cost a lot of money for premium membership.
- Some of them work slowly and lack of optimisation.
- Portfolio analysis tools are hard to use.

All in all, it is hard to find an easy to use application that not only support portfolio tracking, stock searching, but also support state-of-the-art financial analysis tools.

### 1.3 Aim and objectives

Nowadays, smart phones, tablets and smart TV are becoming more and more popular and powerful, it will be beneficial and convenient if we can access the stock application management console through these devices anytime at anywhere. Our goal for this project is to create a web stock portfolio management application that support many modern electronic devices via web browser.

The reason why we choose to develop a web application rather than a computer/smartphone software is that different platform tend to have different framework for running the software and it is hard to maintain software consistence, in addition, it is time consuming and costly to develop application in different platforms. For example, an IOS application that runs on Apply devices cannot be run directly in Android devices or computers, we have to develop them separately using different techniques and libraries/packages. On the other hand, a web application can run on any device that has a web browser, so that we do not need to bother developing multiple software that support different electronic device platforms.

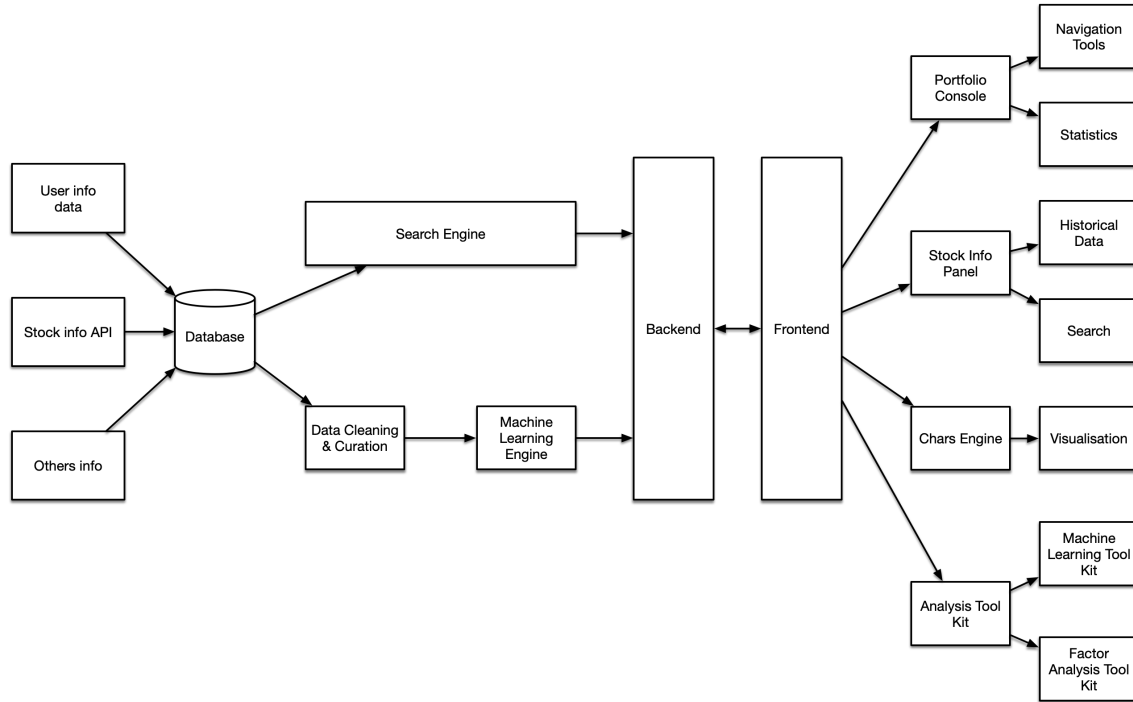
Moreover, we aim to tackle the existing system drawbacks and develop a web application that is easy to use, and do not need much finance background to easily choose a set of stocks to create a portfolio with built in analytic tools such as factor analysis tools and the state-of-the-art machine learning price prediction tools. The detailed epics and features will be discuss in the next chapter.

In a nutshell, our objectives are:

- Easy to use
- Support multiple electronic devices (IOS, Android, Mac, Windows, Linux, Smart TV, etc)
- Responsive and fast
- Rich functions and equipped with state-of-the-art financial analysis tools

## 1.4 Application architecture

The high level application architecture can be illustrate as the following figure:



**Figure 1.1:** *High level application architecture*

As the graph shows, the web portfolio management application can be divided into two major parts, namely backend and frontend. We may develop backend using Django[2] REST web framework and use React[3] as the frontend framework. Backend consists of the logic part behind the scene, when it receives a query or request from the frontend, it will pull the data from the main database, then go through different information process engine and finally give a response in commonly used json format. The application can periodically update the database from different APIs from the internet and fetch stock information as well as news data etc. Database also acts as a warehouse for user information such as user name, password as well as stock portfolio data.

As for the application frontend, it is the GUI (Graphical User Interfaces) for the web app which can interact with normal users. The frontend has been designed in four major components, namely Portfolio Console, Stock Info Panel, Charts Engine and Analysis Tool Kit. Each of which acts as a key feature of the application. Users can navigate through different components through the portfolio management console and easily see the statistics in the dashboard. If we are interested in a particular company, we can perform a fuzzy search on the company name. The Charts Engine will then render all the historical data for the stock info of that company. Moreover, a stock analysis tool kit which uses the-state-of-the-art machine learning algorithms will help investors to perform deep analysis of the stock price and portfolio combination.

## Chapter 2

# Epics

### 2.1 User administration

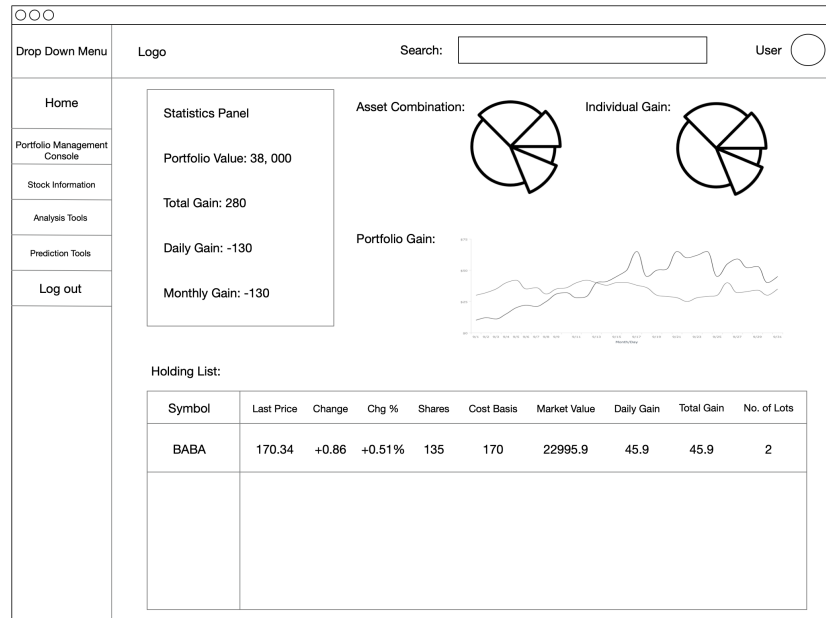
User administration includes user registration, authentication, administration and user profile editing. It is the basic component of the portfolio management application. A user can only use the application after authentication and edit his profile. Admin user are in charge of managing all the user data, which can be implemented in the backend. The difficulty score for this epic can be shown as follows:

Component	Feature	Major Challenge	Difficulty Score
Registration	Register a user.	Frontend design/implementation and update backend database.	5/10
Authentication	Authenticate a user.	Difficult to resolve cross-platform authentication.	7/10
Administration	Backend admin console.	Need to implement backend admin page.	7/10
User Profile	Edit user profile.	Frontend design/implementation and update backend data	6/10
Total Difficulty Score	6.25/10		

**Table 2.1:** *User administration difficulty score*

## 2.2 Portfolio management console

The preliminary design of portfolio management console can be illustrated as the following mockup figure:



**Figure 2.1:** *Portfolio management console*

Portfolio management console is the basic feature of the web application, it consists of three major components: statistic panel, charts component and stock holding list. The statistics panel mainly shows the user portfolio summary in order to be easily checked by users. Charts component acts as a visualisation tool for users to analysis portfolio components and historical profit. Stock holding list shows current holding list of shares of a company. Users can add or delete a stock, add or delete transactions and synchronise the stocks with the current prices.

We can break down this epic into several features together with individual difficulty score as follows:



Component	Feature	Major Challenge	Difficulty Score
Statistic Panel	Portfolio summary	Fetch historical portfolio data, calculation as well as frontend design/implementation.	5/10
Charts	Draw pie char and line graph for historical portfolio data.	Not easy to draw graph in frontend.	8/10
Stock Holding List	Add/delete stock	Difficult in implementing interactive frontend.	8/10
	Add/delete transaction	Difficult in implementing interactive web pages.	8/10
	Synchronise with current stock data.	Difficult in fetching data and calculate statistics.	7/10
Total Difficulty Score	7.2/10		

Table 2.2: Portfolio management console difficulty score

## 2.3 Stock search

Searching a stock is an essential part of the portfolio management application in order to help users to add a particular stock into the portfolio. As Figure 2.1 shows, the search bar can be located in the top of the application all the time so that users can search stocks easily through out all the pages. Users can input the full name (case insensitive) of the stock or its financial code in order to perform a search. Then the result would be sorted by the probability to be matched in ascending order and returned to users, if a user click on a particular stock, the application would be redirected to the stock information page. Apart from basic full text search, we may implement fuzzy search in this project. However, since it is difficult to implement fuzzy search in Django alone, we may need to combine Django with other framework such as Elasticsearch to work together in order to support fuzzy search[4]. Considering we only have around 5 weeks to implement the whole application which means we are short of time, so that fuzzy search will be added as an advanced feature. We decided to implement advance feature only after the basic features are implemented. The difficulty score for this epic can be shown as the following table:

Component	Feature	Major Challenge	Difficulty Score
Search	Search a stock.	Frontend drop down search result design/implementation and implement search feature in backend.	7/10
Fuzzy Search (advance)	Search a stock in a fuzzy manner.	Frontend drop down search result design/implementation and implement fuzzy search feature in backend.	9/10
Total Difficulty Score	8/10		

Table 2.3: Stock search difficulty score

## 2.4 Stock detail information

Stock detail information is also an essential part of the application which enable users to see the historical price movement and technical charts of a stock. The preliminary design of the stock detail view can be shown as the following figure:

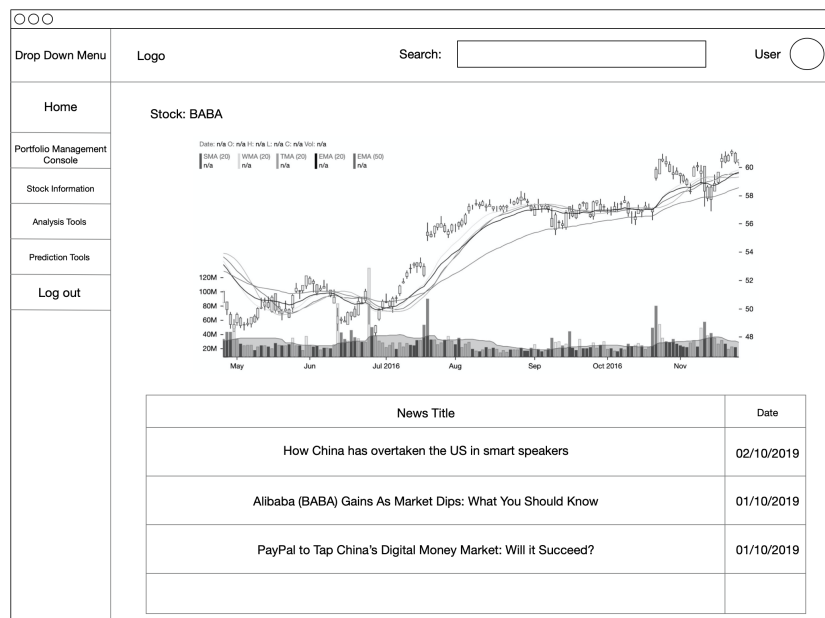


Figure 2.2: Stock detail view

The stock detail view consists of two major components, namely stock price movement/technical charts and news feed. Users can interactively see the historical price movement of a particular stock in daily/monthly/yearly view. In addition, users can also see different technical lines in the graph. As for the news component, we decided to mark it as an advance feature. The reason is that we have not yet found a free news feed for stocks so that we may need to crawl the news from internet periodically and classify them to different stocks accordingly. The classification part is very tricky and it requires advance machine learning text classification algorithms. Since we are short of time, we may consider giving priority to other features. The difficulty score of this epic can be shown as the following table:

Component	Feature	Major Challenge	Difficulty Score
Interactive Chart	Draw historical stock data and technical index charts.	It is difficult to draw interactive charts in frontend.	8/10
News Feed (advance)	Display news for a particular stock.	Difficult to acquire free news feed and need to manually classify different news topic.	10/10
Total Difficulty Score	9/10		

**Table 2.4:** *Stock detail information difficulty score*

## 2.5 Stock price prediction

Nowadays, machine learning and AI becomes more and more popular due to the ground breaking deep learning algorithm [5] and much faster cpu and gpu than the last few decades. Machine learning technics can be utilised on various applications, forecasting time-series data is one of them. However, we can rarely see stock price prediction products in the market. One of the reason for this is that stock data is generated by non-stationary time-series “windows” so that it is difficult to forecast a feature “window” according to past data [6]. Having said that, when using the right model and factor, we can still generate a probability that whether a stock price may go up or down to some degree by applying machine learning on time-series data. Another reason is that when a strategy which is making profit is used by most of the people, then this strategy will lose its ability to gain profit in the near future [7] because financial market can only be robust when people are trading using different strategies. This is why many stock price prediction products can only be accessed by premium users or be kept secrete (like a secrete recipe in a restaurant).

We planed to use traditional machine learning model such as linear model, SVM and random forest and possibly together with state-of-the-art deep learning models like LSTM to predict the stock price. Selecting, training and analysing different models as well as tuning parameters are time-consuming tasks and cannot guarantee the effectiveness of the algorithm. Since the development of our project only lasts for around 5 weeks and we cannot develop the project in full time (we have other courses in this term), so that we decided to mark this epic as advance and experimental. We planned to demo the final model prediction on the

web application, then show the model development procedure, analysis as well as evaluation on Jupyter Notebook. The difficulty score for this epic can be shown as follows:

Component	Feature	Major Challenge	Difficulty Score
Model Development (advance)	Developing, training, testing, analysing and evaluating different machine learning models on stock data and final prediction result demo.	Difficult to develop an effective machine learning model for stock price prediction in all aspect.	10/10
Total Difficulty Score	10/10		

**Table 2.5:** *Stock price prediction difficulty score*

## 2.6 Factor analysis

Factor analysis is a commonly used technic in stock selection and portfolio evaluation [8] [9] [10]. We can often see a variety of technical and financial indices of a particular stock or a financial market such as PE (Price-Earnings), PB (Price-to-Earning Ratio), EPS (Earnings Per Share) and ROA (Return on Asset), etc. These indices are correlated with the stock price by some means or other, with factor analysis, we can explore the most correlated factors (indices) and select them as the final stock selection factors. Then we can score each stock according to these factors and rank them in descending order, the stocks gets higher score then can be added into the stock portfolio.

Factor analysis requires a lot of concepts, formula and technical details to be digested, all the members in our team do not have a financial background, however, it is very exciting to learn something new and try to apply them in real world financial market. We decided to mark factor analysis as an advance challenging epic and try our best to build a useful analysis tool in this project. We planned to first develop and evaluate the effectiveness of applying factor analysis in Jupyter notebook, then we can build a small demo in the web application panel to show the results. The difficulty score for this epic can be shown as the following table:

Component	Feature	Major Challenge	Difficulty Score
Factor Analysis (advance)	Applying and evaluating factor analysis on stock data and generate final result demo.	Difficult to digest factor analysis concepts and mathematical formula.	10/10
Total Difficulty Score	10/10		

**Table 2.6:** *Factor analysis difficulty score*

## 2.7 Final epic selection

There are total 6 epics in our project, however, since we only have limited amount of time to develop the application, all our team members cannot participate in full-time due to heavy load of other courses in this term, so that we break the epics into different features and score them with regard to the relative difficulty as 0-6 (elementary), 6-8 (medium) and 9-10 (advance). In the final demo, we will deliver the **elementary** and **medium** features with **higher priority** and experiment with some of the advance features. Different features are ranked in the following table according to their difficulty score:

Component	Feature	Major Challenge	Difficulty Score
Registration	Register a user.	Frontend design/implementation and update backend database.	5/10
Statistic Panel	Portfolio summary	Fetch historical portfolio data, calculation as well as frontend design/implementation.	5/10
User Profile	Edit user profile.	Frontend design/implementation and update backend data	6/10
Authentication	Authenticate a user.	Difficult to resolve cross-platform authentication.	7/10
Administration	Backend admin console.	Need to implement backend admin page.	7/10
Search	Search a stock.	Frontend drop down search result design/implementation and implement search feature in backend.	7/10
Charts	Draw pie char and line graph for historical portfolio data.	Not easy to draw graph in frontend.	8/10
Stock Holding List	Add/delete stock	Difficult in implementing interactive frontend.	8/10
	Add/delete transaction	Difficult in implementing interactive web pages.	8/10
	Synchronise with current stock data.	Difficult in fetching data and calculate statistics.	7/10

Component	Feature	Major Challenge	Difficulty Score
Interactive Chart	Draw historical stock data and technical index charts.	It is difficult to draw interactive charts in frontend.	8/10
Fuzzy Search (advance)	Search a stock in a fuzzy manner.	Frontend drop down search result design/implementation and implement fuzzy search feature in backend.	9/10
News Feed (advance)	Display news for a particular stock.	Difficult to acquire free news feed and need to manually classify different news topic.	10/10
Model Development (advance)	Developing, training, testing, analysing and evaluating different machine learning models on stock data and final prediction result demo.	Difficult to develop an effective machine learning model for stock price prediction in all aspect.	10/10
Factor Analysis (advance)	Applying and evaluating factor analysis on stock data and generate final result demo.	Difficult to digest factor analysis concepts and mathematical formula.	10/10

**Table 2.7:** *Portfolio management application features rank*

## Chapter 3

# Methodology

### 3.1 Project management

The project will be developed using Agile Scrum Methodology which is considered to be people-based rather than plan-based due to the trade off among time, cost and performance/scope. The major advantages of scrum methodology are [11]:

- It can produce a relatively high quality product in the shortest time.
- Team members can choose the best way to deliver the highest priority features.
- Develop software in an iterative manner so as to enhance quality.

Since the project is held as one of the courses in the university, and all the team members are students who also need to undertake other courses, so that we cannot satisfy the full-time member requirement for the real world scrum team. As a result we need to make some adjustment accordingly. We planned to make progress in a series of “sprints”, the project will be designed, implemented and test during each sprint. The work flow of the project can be break down as follows:

1. Each team member will maintain a working diary which keeps track of individual progress, plans and finished tasks. The diary will be store in Github so that all the team members can see the progress.
2. The general project progress including project backlog, finished features will be tracked via Github front page so that it can be seen easily.
3. Project code and references are stored in Github repository so that we can develop the project in a collaborative manner.
4. We planned to hold a weekly face-to-face sprint retrospective meeting and report daily problems via Wechat group.
5. The target duration of the sprints are two weeks and we should make a progress meeting with our mentor during these time.

### 3.2 Software and library

The stock portfolio management software will be developed in the form of web application, so that we planned to separate the development into backend and frontend. Backend will be developed using Python and Django, together with other machine learning libraries such as Pandas, Numpy, Scipy, Sklearn and Prophet, etc. Frontend will be develop using React, Google Material-UI and react-stockcharts, etc.

### 3.3 Team members and responsibility

1. Rui Liang (Scrum Master, frontend developer)
  - Maintain Github repository, project issues, communicate with mentors.
  - Write project proposal, project demo and project report.
  - Develop project frontend.
  - Help backend developer develop project backend.
  - Ensure project quality and on time delivery.
  - Experiment with advance features of the project.
2. Hao Jiang (backend developer)
  - Develop project backend.
  - Responsible for Epic 2, 3 and 4.
3. Zhicheng Tao (backend developer)
  - Develop project backend.
  - Responsible for Epic 5 and 6.
4. Chuguan Tian (backend developer)
  - Develop project backend.
  - Responsible for Epic 1.



### 3.4 Schedule

The overall preliminary project schedule can be illustrated as the following gantt chart:

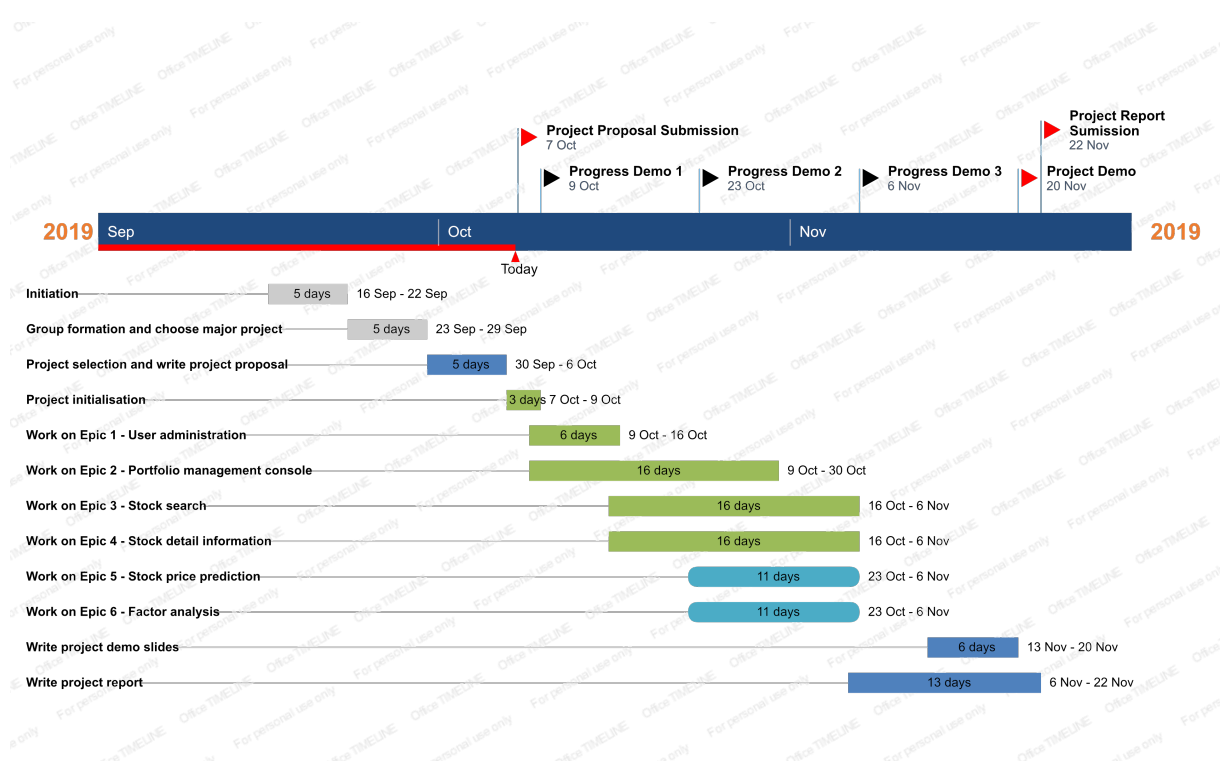


Figure 3.1: *Project schedule*

# Bibliography

- [1] 7 best portfolio management software tools for investors 2020. <https://www.roboadvisorpros.com/best-portfolio-management-software-for-investors/>, 2019. [Online; accessed 01-Oct-2019].
- [2] Django. <https://www.djangoproject.com>, 2019. [Online; accessed 06-Oct-2019].
- [3] React - a javascript library for building user interfaces. <https://reactjs.org/>, 2019. [Online; accessed 06-Oct-2019].
- [4] Tutorial: How to create a fuzzy search-as-you-type feature with elasticsearch and django. <https://www.freshconsulting.com/how-to-create-a-fuzzy-search-as-you-type-feature-with-elasticsearch-and-django/>, 2019. [Online; accessed 03-Oct-2019].
- [5] Ritika Singh and Shashi Srivastava. Stock prediction using deep learning. *Multimedia Tools and Applications* 18569, 76:18, 2017.
- [6] Can machine learning techniques be used to predict stock prices? [https://www.researchgate.net/post/Can\\_Machine\\_Learning\\_Techniques\\_Be\\_Used\\_To\\_Predict\\_Stock\\_Prices](https://www.researchgate.net/post/Can_Machine_Learning_Techniques_Be_Used_To_Predict_Stock_Prices), 2019. [Online; accessed 29-Sep-2019].
- [7] Lecture 16: Portfolio management. <https://ocw.mit.edu/courses/mathematics/18-s096-topics-in-mathematics-with-applications-in-finance-fall-2013/video-lectures/lecture-16-portfolio-management/>, 2019. [Online; accessed 27-Sep-2019].
- [8] Jeffrey Donaldson and Marcus Allan Ingram. Applying multi-factor models of stock returns: Student exercises and applications. *Journal of Financial Education* 1, 40(3/4):1–21, 2014.
- [9] Christophe Morel. Stock selection using a multi-factor model - empirical evidence from the french stock market. *The European Journal of Finance*, 7(4):312–334, 2001.
- [10] Lecture 15: Factor modeling. <https://ocw.mit.edu/courses/mathematics/18-s096-topics-in-mathematics-with-applications-in-finance-fall-2013/video-lectures/lecture-15-factor-modeling/>, 2019. [Online; accessed 01-Oct-2019].
- [11] Comp9900 lecture slides. <https://webcms3.cse.unsw.edu.au/COMP9900/19T3/resources/32998>, 2019. [Online; accessed 05-Oct-2019].