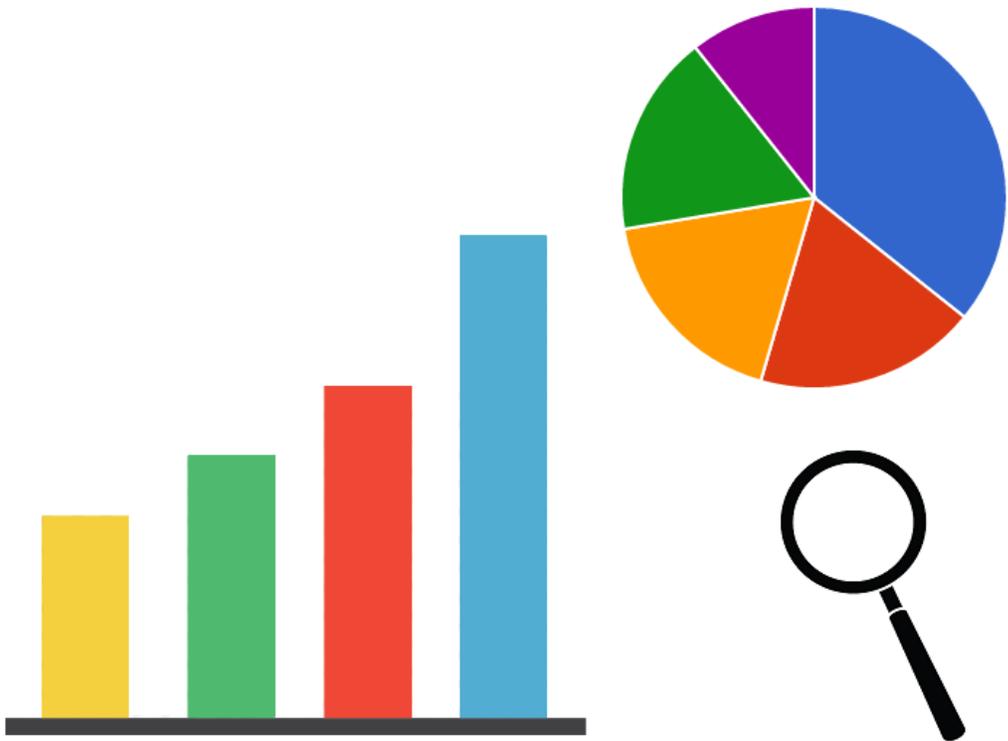


Sequence Control Panel

Technical Report



Mina Gaid - 13729979
BSHC4 – Data Analytics
10/05/2017

Declaration Cover Sheet for Project Submission

SECTION 1 *Student to complete*

Name: Mina Gaid
Student ID: 13729979
Supervisor: Dr Eugene O'Loughlin

SECTION 2 Confirmation of Authorship

The acceptance of your work is subject to your signature on the following declaration:

I confirm that I have read the College statement on plagiarism (summarized overleaf and printed in full in the Student Handbook) and that the work I have submitted for assessment is entirely my own work.

Signature: Mina Gaid Date: 10/05/17

NB. If it is suspected that your assignment contains the work of others falsely represented as your own, it will be referred to the College's Disciplinary Committee. Should the Committee be satisfied that plagiarism has occurred this is likely to lead to your failing the module and possibly to your being suspended or expelled from college.

Complete the sections above and attach it to the front of one of the copies of your assignment,

What constitutes plagiarism or cheating?

The following is extracted from the college's formal statement on plagiarism as quoted in the Student Handbooks. References to "assignments" should be taken to include any piece of work submitted for assessment.

Paraphrasing refers to taking the ideas, words or work of another, putting it into your own words and crediting the source. This is acceptable academic practice provided you ensure that credit is given to the author. Plagiarism refers to copying the ideas and work of another and misrepresenting it as your own. This is completely unacceptable and is prohibited in all academic institutions. It is a serious offence and may result in a fail grade and/or disciplinary action. All sources that you use in your writing must be acknowledged and included in the reference or bibliography section. If a particular piece of writing proves difficult to paraphrase, or you want to include it in its original form, it must be enclosed in quotation marks and credit given to the author.

When referring to the work of another author within the text of your project you must give the author's surname and the date the work was published. Full details for each source must then be given in the bibliography at the end of the project

Penalties for Plagiarism

If it is suspected that your assignment contains the work of others falsely represented as your own, it will be referred to the college's Disciplinary Committee. Where the Disciplinary Committee makes a finding that there has been plagiarism, the Disciplinary Committee may recommend

- that a student's marks shall be reduced
- that the student be deemed not to have passed the assignment
- that other forms of assessment undertaken in that academic year by the same student be declared void
- that other examinations sat by the same student at the same sitting be declared void

Further penalties are also possible including

- suspending a student college for a specified time,
- expelling a student from college,
- prohibiting a student from sitting any examination or assessment.,
- the imposition of a fine and
- the requirement that a student to attend additional or other lectures or courses or undertake additional academic work.

Table of Contents

Sequence Control Panel.....	0
Technical Report.....	0
Installation & Configuration Manual.....	8
Option 1.....	8
Option 2.....	9
Option 3.....	10
Site address & Information	11
Notes to consider:	11
Introduction	12
Executive Summary	12
Background.....	12
Objectives	14
Technical Approach	15
Special resources required	16
Anaconda	17
Technical Details.....	17
Project Plan	18
Evaluation.....	19
Requirements Specification (RS).....	20
Document Control	20
Revision History	20
Distribution List.....	20
Related Documents	20
System	21
Requirements	21
Purpose	21
Project Scope.....	21
Definitions, Acronyms, and Abbreviations.....	22
User Requirements Definition	22
Requirements Specification.....	23
Functional requirements	23

Functional requirement - 1	23
Functional requirement - 2	24
Functional requirement - 3	25
Functional requirement - 4	26
Functional requirement - 5	27
Functional requirement - 6	28
Functional requirement - 7	29
Functional requirement - 8	30
Functional requirement - 9	31
Documentation requirements.....	31
Non-Functional Requirements	32
Performance/Response time requirement	32
Availability requirement.....	32
Recover requirement	32
Robustness requirement.....	32
Security requirement	33
Reliability requirement.....	33
Maintainability requirement	33
Portability requirement.....	33
Extendibility requirement	33
Reusability requirement.....	34
Interface requirements	34
GUI	35
Application Programming Interfaces (API)	36
System Architecture	36
System Evolution	36
System Architecture - Class Diagram.....	37
Testing Plan	38
Customer testing	38
Monetization	38
Conclusions.....	39
Further development or research.....	40
Testing	41

Testing of the code.....	41
Testing of the Forecasts	41
Customer Testing	43
Appendices	44
Use Case - 1.....	44
Scope.....	44
Description.....	44
Use Case Diagram	44
Flow Description	45
Use Case - 2.....	46
Scope.....	46
Description.....	46
Use Case Diagram	46
Flow Description	47
Use Case - 3.....	48
Social authentication.....	48
Scope.....	48
Description.....	48
Use Case Diagram	48
Flow Description	49
Use Case - 4.....	50
Scope.....	50
Description.....	50
Use Case Diagram	50
Flow Description	51
Use Case - 5.....	52
Scope.....	52
Description.....	52
Use Case Diagram	52
Flow Description	53
Use Case - 6.....	54
Search Organisations.....	54
Scope.....	54

Description	54
Use Case Diagram	54
Flow Description	55
Use Case - 7	56
Search Codes.....	56
Scope.....	56
Description	56
Use Case Diagram	56
Flow Description	57
Use Case - 8	58
Create Forecast	58
Scope.....	58
Description	58
Use Case Diagram	58
Flow Description	59
Use Case - 9	60
Charts	60
Scope.....	60
Description	60
Use Case Diagram	60
Flow Description	61
Mock-up 1: Index page	62
Mock-up 2: Login page	63
Mock-up 3: Signup page	64
Mock-up 4: Dashboard page	65
Mock-up 5: Charts page	66
Mock-up 6: Tables page	67
Reflective Journal - September	68
My Achievements	68
My Reflection	68
Intended Changes.....	69
Supervisor Meetings.....	69
Reflective Journal - October.....	70

My Achievements	70
My Reflection	70
Intended Changes.....	71
Supervisor Meetings.....	71
Reflective Journal - November.....	72
My Achievements	72
My Reflection	72
Intended Changes.....	73
Supervisor Meetings.....	73
Reflective Journal - December	74
My Achievements	74
My Reflection	74
Intended Changes.....	75
Supervisor Meetings.....	75
Reflective Journal - January	76
My Achievements	76
My Reflection	76
Intended Changes.....	77
Supervisor Meetings.....	77
Reflective Journal - February	78
My Achievements	78
My Reflection	78
Intended Changes.....	79
Supervisor Meetings.....	79
Reflective Journal - March	80
My Achievements	80
My Reflection	80
Intended Changes.....	81
Supervisor Meetings.....	81
Bibliography	82
NCI Invention Disclosure Form.....	83

Installation & Configuration Manual

The following is a manual on the installation and configuration of the code.

Please note that the application was developed on a Windows machine and has not been tested on a Mac/Linux OS. The steps below are for Windows OS. Errors might occur if on a Network.

There are multiple ways to configure your machine to run the project's source code. Option one is the recommended method as its tested and easier than the alternative method.

Option 1

1. Download and install Anaconda for Python 3.6 from the web. Make sure to select the correct version for your machine. - <https://www.continuum.io/downloads>
2. After the installation of Anaconda, open CMD (AKA: command prompt – Terminal).
3. Run the following command – Be patient and wait for the installation to be complete –

```
pip install django
```

4. Once Django is installed, Run the following command – Be patient and wait for the installation to be complete –

```
pip install quandl
```

5. Once everything is installed, open the project folder, and hold the shift button then right click and choose the “open command window here”
6. Once the command window opens, type in the following – Be patient and wait for the installation to be complete –

```
python manage.py runserver
```

7. Text will then display as follows –

```
System check identified no issues (0 silenced).  
March 30, 2017 - 12:18:45  
Django version 1.10.5, using settings 'scp.settings'  
Starting development server at http://127.0.0.1:8000/  
Quit the server with CTRL-BREAK.
```

Copy and paste the URL, as seen in the example above (<http://127.0.0.1:8000/>), into your browser and the app should be working locally on your machine.

DO NOT – close the command window while running the app.

To close, head back into the open command window and simply hold Ctrl + C.

Option 2

Note: This option is not recommended for users with little knowledge in programming as errors are very common.

1. Download and install Pip from the web. - <https://pip.pypa.io/en/stable/installing/>
2. After the installation of pip, open CMD (AKA: command prompt – Terminal).
3. Run the following commands, run each one after the other – Be patient and wait for each installation to be complete –

```
pip install Django

pip install quandl

pip install --user numpy scipy matplotlib ipython jupyter pandas sympy
nose
```

4. Once everything is installed, open the project folder, and hold the shift button then right click and choose the “open command window here”

Alternatively, you can open the command window and type in –

```
cd (insert project path here)
```

Then simply drag in the project folder into command window and it will generate the path. Hit enter and command window will be set to the project directory.

5. Once the command window opens, type in the following – Be patient and wait for the installation to be complete –

```
python manage.py runserver
```

6. Text will then display as follows –

```
System check identified no issues (0 silenced).
March 30, 2017 - 12:18:45
Django version 1.10.5, using settings 'scp.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Copy and paste the URL, as seen in the example above (<http://127.0.0.1:0000/>), into your browser and the app should be working locally on your machine.

DO NOT – close the command window while running the app.

To close, head back into the open command window and simply hold Ctrl + C.

Option 3

Note: This option is not recommended for users with little knowledge in programming as errors are very common.

1. Download and install Pip from the web. - <https://pip.pypa.io/en/stable/installing/>
2. After the installation of pip, open CMD (AKA: command prompt – Terminal).
3. Once installed, open the project folder, and hold the shift button then right click and choose the “open command window here”

Alternatively, you can open the command window and type in –

```
cd
```

Then simply drag in the project folder into command window and it will generate the path. Hit enter and command window will be set to the project directory.

4. Type the following into the command window –

```
pip install -r /path/to/requirements.txt
```

/path/to/requirements.txt being the directory to requirements.txt found in the project folder.

Its recommended that you drag in the requirements.txt into the command window to generate the path.

5. Once done, restart the command window and type in the following – Be patient and wait for the installation to be complete –

```
python manage.py runserver
```

6. Text will then display as follows –

```
System check identified no issues (0 silenced).
March 30, 2017 - 12:18:45
Django version 1.10.5, using settings 'scp.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Copy and paste the URL, as seen in the example above (http://127.0.0.1:0000/), into your browser and the app should be working locally on your machine.

DO NOT – close the command window while running the app.

To close, head back into the open command window and simply hold Ctrl + C.

Site address & Information

The following URLs are for accessing the live (hosted) version of the application and the code.

Live Site – 144.76.202.171

GitHub – <https://github.com/mina-gaid/scp>

Notes to consider:

- The live site can be accessed by inputting the IP address (144.76.202.171) in a browser search bar. It's hosted on a dedicated private server.
- Due to the charts being shown as images, caching might be an issue. If you encounter an issue where the chart stays the same and doesn't change, it's likely because it got cached.
- When running the code locally on your machine, please consider that Social login will not work. Social login is set to work only on the domain of the live website. This is because Facebook and Google both require a site domain for their authentication system to work. Please sign up and login using the email option.
- When Signing up using email, the sign-up forum will ask for an email address. However, an email is not required as email verification will not work when running the code locally. Sign up and login will use your Username instead.
- If there's an issue with the login, the main user dashboard can still be accessed using the following path –

When running locally - <http://127.0.0.1:8000/dashboard/control-panel/>

When using the live site - <http://144.76.202.171/dashboard/control-panel/>

- This has been done for testing only. The finished product will not feature this.

Introduction

Executive Summary

Big data is becoming increasingly popular within the business world. It's being used in a variety of ways within corporations looking to better enhance and expand their business. This is especially true when it comes to the financial world. Particularly, regarding banking and stock trading where there has been an unprecedented growth of data analytics.

I intend to capitalize on this by building a KPI dashboard that looks to provide stock analytical data. I will implement this using statistical formalisms, an API and machine learning algorithms that will provide the user with useful data. A key focus will be forecasting of results and an emphasis on data visualization.

My development stack is basic and minimal. I will develop the application using Python as a backend programming language and will intend on using PyCharm as my main IDE aside from the terminal. Django will be used as a backend framework while Bootstrap will be used for front-end development. A range of packages will be used for visualising the data, and creating forecast modules such as Matplotlib and SciPy.

Background

Specializing in Data analytics, I planned on developing a web app that extracts data in real-time from an API and outputs it through mathematical formulas where the results are then displayed to the user. This being the core of my idea, I wanted to find a topic in which it can be done.

Given the use in mathematics, I chose to base my project on financial data as opposed to that of text based data. There are many reasons for this.

The first is to do with the dataset(s) I planned on using. There are many resources and tools for financial data in which I can use. The same is not as true regarding other forms of data as financial data is the most popular in analytics.

The second reason for using financial data is more so to do with the use of mathematics in Data analytics which happens to be a big part. As such, I want to integrate it as much as possible in any given way. With numbers and figures, I can do more than with text based datasets. I can use mathematical probabilities with the given data and provide a unique source of useful information to the end user.

Lastly, the ease of use while developing the web app with financial data. As it consists mostly of numbers, it can be easily incorporated, cleaned and managed efficiently than that of text based data. Being able to forecast results is a big part of the application, this is much easier to do with financial data than that of others.

Aside from this, data analytics / big data is a considerable part of the financial industry and my project would thus be ideal given its use. This is particularly accurate when I researched the use of big data in the financial world. The use of data is exceptionally important in banking and investing. One distinct area is that of the stock market in which big data is being used for setting an appropriate price of a new stock to tracking and predicting its trend on the market. As I looked more into this I was intrigued by the use and employment of data as well as its importance within the field.

One peculiar area that aroused my curiosity was how data is being used aside machine learning to provide accurate market predictions. I found this to be extraordinarily fascinating and as such looked more into it.

My research has shown that major banks use algorithms to invest within the market. Algorithms which are fed market data to make predictions based on non-random market trends. These algorithms are not 100% accurate but are useful for identifying market trends and are valuable when making market investment based decisions. Of course, the algorithms cannot predict random market events but it can be useful for outlining trends. Even if the algorithms are accurate six out of ten times, they are deemed profitable.

Of course, these algorithms are extremely advanced as they are fed into a neural network. It would be unrealistic to develop something of that kind but using statistical probability based formulas I can maybe make predictions on different parts of the stock which on its own would be useful for the end user.

With the given research, I had along with my interest and limited knowledge in the area, I chose this to be base of my project. Immediately thereafter I researched the resources and methods to achieve what I had planned. I came up with a rough plan and different ways to approach given tasks and requirements.

I have since built onto it, mainly on a technical basis of what I plan on using for development and different approaches in which I can take to accomplish different objectives. I hope to add more onto it as I begin development of what I plan to be a web based app.

I am confident that my project will be an enjoyable and educational experience in the world of big data and mathematics.

Objectives

The main objective is to design and build a stock analytics dashboard which will feature real time data using a stock market API. The aim of the KPI dashboard is to provide the user with basic stock information from the market with statistical probabilities based predictions and / or recommendations.

Possibility: Machine learning may be added in to provide predictions and identify trends in the market.

The following is a summary of the key objectives to be carried out through development of the web app:

- Design the wireframe of the User interface and a UML of the back-end architecture.
- Develop the front-end of the web application using Django and bootstrap. (important for reaching mobiles and tablets)
- Enable user registration and social authentication using the Facebook and the Google API
- Implement the Quandl API and retrieve real-time and historical data
- Add statistical probabilities and formulas into the data (using historical data from Quandl API)
- Implement some sort of machine learning algorithm or function for market predictions or recommendation.
- Output results using charts
- Site SEO and Google analytics (Not required but useful)
- Deploy the application using a Django supported host such as Heroku or a private server

Testing will be done between each task and at the end of development. Deployment of the web app will be implemented in the end of development.

Technical Approach

The following is a brief description of the research carried out and its results. This research was used to help in the planning and development of the proposed stock KPI dashboard.

The implemented language of choice for the development of the application is to be Python. Python is a general-purpose, versatile and popular programming language. The reason for choosing Python as the main programming language is mainly because it is so commonly used for data mining and scientific applications. It is also concise and easy to read.

Another reason for choosing Python is that vast resources available to developers. Python, being a popular language as is, has a lot of open source tools and packages that are extremely useful for the purpose of analysing data. Such packages include SciPy.

From experience, it is practical to use frameworks when developing an application. Twitter's Bootstrap is the go to source for front-end development to ensure optimization across platforms and web standards. More so to maintain a friendly developer environment, make changes easy to implement and to insure future updates and maintainability.

Given that Python is to be used as the implemented language, Django was to number one choice for back-end framework. This is necessary to maintain a developer friendly environment and to incorporate future code. Django is also open source so future support for the framework is assured.

When researching an API to be used for the data source, the Bloomberg API came up as the best option. However, further research showed that the Bloomberg API was not free (at the time of the research) and was not open source, making it restricted in development. The next best option was the Quandl API which was both free and open source. The Quandl API was more developer friendly, did not have an excessive limit in API requests and featured historical and real-time data. Some of the data is however locked behind a paywall but this wasn't an issue as a lot of what I needed is free.

For machine learning, my research has led me to believe that an advanced neural network is required. I since decided to use mathematical formulas to make predictions instead of creating decision trees. However, certain algorithms and functions can be useful for making recommendations. There isn't a specific source but rather an approach to tackling the issue. Mainly, using statistical probabilities. Liner regression appears to be the best option for forecasting market prices based on the historical data made available to be by the Quandl API.

Special resources required

There are no set of resources required outside of software. Developer tools, frameworks and API's are the only required resource for the development of the project. The following is a summary of the resources necessary to develop and run the web application I plan on building.

The developer environment is to include the following:

- PyCharm (Python IDE)
- Notepad++ (with add-ons)
- SQLite Studio
- CMD (Terminal / Command prompt)

The following are assets to be downloaded for development and configuration of the source code:

- Python 3+ (comes with Anaconda)
- (Continuum) Anaconda -
- Pip (comes with Anaconda)
- Django 10+
- Quandl API module

Assets that are required and are to be used for development:

- Bootstrap
- Matplotlib (Comes with Anaconda)
- Django-Allauth (Comes with Anaconda)
- Numpy (Comes with Anaconda)
- Sklearn (Comes with Anaconda)
- Scikit (Comes with Anaconda)
- Scipy (Comes with Anaconda)
- Pandas (Comes with Anaconda)
- Facebook API (Authentication) (Comes with Django-Allauth)
- Google API (Authentication) (Comes with Django-Allauth)
- Quandl API module

Other:

IBM's SPSS – To be used for testing of statistical formulas.

Anaconda

Anaconda will be vital throughout development as it comes with all the essential packages used within the application.

Anaconda is a freemium open source distribution and package management system for the Python and R programming languages. It's used for large-scale data processing, predictive analytics, and scientific computing. Anaconda simplifies package management and deployment, thus making it extremely useful for development. Furthermore, it keeps packages and tools up to date which is important as most are frequently updated, making certain functions and Features deprecated.

Technical Details

The project will be implemented as a web app and is to be built using Python as the main programming language. The reason why I chose Python is mainly for its use in analytics and data mining. Python is also a robust, very powerful programming language. Its small quantity of lines of code make it less prone to issues, easier to debug, and more maintainable. In addition, Python is extremely flexible and offers a huge standard library that can be used through development.

Other languages that will be used include JavaScript and HTML/CSS for front-end development.

Django will be used as a back-end framework to provide a developer friendly environment and to ensure ease in adding and incorporating features and functions. Django is also an MVC (model, view and controller) framework. This allows for flexibility down the road such as supporting multiple file formats and thus making the development process easier. While Bootstrap will be used as a front-end framework for the same reasons and to ensure device optimization across all platforms such as Desktop, mobile and tablet.

Quandl API will be the source of the data. It's a free open source API that provides real-time and historic market data, making it suitable for what I have planned. The API is simple, consistent and completely free to use. I will be using Quandl's Python based API to better incorporate it into my codebase.

The data retrieved from the Quandl API will be assimilated with mathematical formulas to output relevant information to the end user. Google charts or Matplotlib will be used to display some of the information in the form of Charts for data visualization. This is done through the Google Charts API which is fed data and generates the appropriate charts.

Django has built in user registration and social authentication. This will be implemented to ensure a tailored user experience. The Facebook API and the Google API will be used for this.

Evaluation

The application developed will be used for monitoring stock in real time and verifying forecasts. The aim of the application is to simply provide a platform in which investors can better track and analyse the stock market by allowing industry and none industry users to better understand the data behind it.

The stack used for development is mostly open source. Open source development tools will be used for the development of the application to incorporate good development standards by way of frameworks.

I will regularly evaluate how the web application runs. I will monitor its performance with each added function. I will test different responses and look for ways to better improve the code base as well as things to add that would expand the user experience.

When I'm fully satisfied, I will be able to test the system. I will do this by allowing volunteers to test the application and provide different feedback regarding what I can do to enhance the app. Different appropriate changes will be done if deemed necessary and will be recorded and documented.

Requirements Specification (RS)

Document Control

Revision History

Date	Version	Scope of Activity	Prepared	Reviewed	Approved
14/10/2005	1	Create	AB	X	X
21/10/2005	2	Update	CD		

Distribution List

Name	Title	Version
Eamon Nolan	Lecturer	

Related Documents

Title	Comments
Title of Use Case Model	
Title of Use Case Description	

System

Requirements

Purpose

The purpose of this document is to set out the requirements for the development of a Stock KPI (analytics) dashboard by way of a Web Application.

The intended customers are any investor who is interested in investing in the stock market.

Project Scope

The scope of the project is to develop a web application that allows users to monitor stock data and keep track of trends on the market.

The system shall have statistical formulas built in to outlay useful information to the end user. Machine learning will be included in some way or forum to provide relevant information that can be used for market predictions.

I was involved in discussions with Eugene O'Loughlin, my project supervisor to elicit the following requirements that align with research I made.

- Maintain the source code using frameworks for effortless development and future proofing the application.
- Plan for monetization's and user customization by enabling user registration.
- Display information in a clear user friendly interface. Preferably using charts if applicable.
- Allow for the efficient updating and viewing of real time market data using an API. Stock data is only useful in real-time. API should simultaneously cater historical data alongside real-time data.
- Include statistical probability based formulas to retrieve appropriate and convenient information to the user.
- Include machine learning algorithms to provide the user with advantageous data such as best / top performing stocks of the week. This can be achieved using historical data from the API.

Definitions, Acronyms, and Abbreviations

The following list below is a list of Acronym and Abbreviations which are used across this report.

Acronym / Abbreviations	Definitions
SCP	Sequence Control Panel
Stock	Stock
GUI	Graphical User Interface
API	Application Programming Interface
UML	Unified Modelling Language
IPO	Intel Public Offering

User Requirements Definition

The requirements defined above based on research and consultations with my project supervisor are the main features that I would like to see in this project. Additional requirements will be realized and outlined to me as development of my project progresses should they prove to be necessary.

Machine learning and data retrieval from the API are the focus for my project.

The API being the source of data, it is imperative that it have the most attention in terms of architecture, requirements and use by users. Without data being fed by the API to the system, the application is meaningless.

Machine learning is crucial for the use of the application. As such, the requirements will consider how it will be included in the system and its effects on the application.

With regards to the user interface, it's important that I build it to be both aesthetic and clear in terms of how information is displayed to the end user. Data and navigation are to be more straight forward and user friendly. This contrasts with current stock market investment systems who are poorly designed from a UI prospective. They are barely usable by non-trained professionals. The application will host a user-friendly interface that can be understood by anyone.

Requirements Specification

Functional requirements

This section focuses on functional requirements that describe distinct actions or functions.

Functional requirement - 1

Use Case 1 - see Appendix

Requirement 1: Hosting & Accessibility

The following requirement concerns Hosting and Accessibility of the web application.

Description & Priority

With the application being developed as a web app, hosting will be included. The web application will be hosted at the end of the development process to be used by end users online.

The web application is using Django as a backend framework which means that hosting will require changes to be implemented in the source code before deployment of the application. Such changes will be done as Django requires static files to be shown differently in development mode then when live on the web. This is due to how Django operates in a development environment as opposed to that when hosted.

Another factor of hosting is that the host must have Django support for the site to be deployed on to the world-wide web and accessible to users.

Hosting is volatile if the web app is intended to reach customers across different platforms. It is important that the requirement is taken into consideration regarding the priority and the appropriate host is selected. Otherwise, customers (users) won't be able to access the site or will have a bad experience regarding speed, an important factor when dealing with real-time data.

Functional requirement - 2

Use Case 2 - see Appendix

Requirement 2: User registration

The following requirement concerns user registration and authentication within the web application.

Description & Priority

This requirement will cover registration and authentication of users and administrators. Registration is required to monitor individual user interactions and provide a unique custom experience. Registration is also required to give users and administrators different access to key parts of the web application.

The requirement is important to the overall system in terms of security, monetization, administrative control and monitoring of end-user and site interactions. It's additionally for future upgradability were features such as customizing a user's dashboard could later be added.

The registration and authentication system is built into the Django framework which I will be using. An administration system and dashboard is also included with the framework. The system is built in but it will need to be configured, enabled and customized to my web application.

With Django, a registration system can be developed using the Django documentation. Another option is that of using packages such as "Django Allauth" which requires minor configuration and changes to the HTML structure of the registration and login forms.

Functional requirement - 3

Use Case 3 - see Appendix

Requirement 3: Social authentication

The following requirement concerns Social registration and authentication within the web application.

Description & Priority

This requirement will cover registration and authentication of users using Social authentication. Social registration and authentication is required to give users with better, faster and easier ways to sign up and login.

Just as in requirement 2 for user local registration and authentication, this requirement is important to the overall system in terms of security, monetization, administrative control and monitoring of end-user and site interactions.

The registration and authentication system is built into the Django framework which I will be using. A Django opensource package called Allauth will be used to setup and configure the social login API(s) for both Facebook and Google. Administrators can still see and manage users who have signed up using their social accounts.

Functional requirement - 4

Use Case 4 - see Appendix

Requirement 4: Dashboard

The dashboard requirement looks at the main interface in which the user interacts with and the content it includes

Description & Priority

The dashboard is the main service area of the website. The user dashboard is where the user is redirected upon login. It is where content / data will be displayed and hosted. As such, it is the primary location upon which the user will spend much of their time on the site. Within the dashboard, the data pulled from the API will be displayed for the user to see and interact with (if using Matplotlib). The dashboard will contain all the necessary services provided by the site. The Dashboard section will be split up into three possible sections.

- The first section, is that of the main dashboard in which the information on stock will be displayed. It is the focus for the user and the source of interaction. Users will be redirected here when logged in.
- The second section of the dashboard looks at graphs. This section will display some sort of chart or graph as requested by the user. The chart / graph will contain visual information regarding the Stock.
- The third section of the dashboard will host a table in which it will display the top / best performing Stock of the week and / or month.

Functional requirement - 5

Use Case 5 - see Appendix

Requirement 5: Data Retrieval

The following requirement will consider the retrieval of data from the API when requested by an end user.

Description & Priority

This requirement will consider the process of searching for a Stock and retrieving the data from an API.

The system is powered to provide the user with real-time data on stock. The data is retrieved from an API once requested. The retrieved data is then outputted on the user interface of the application where it is then shown to the user in an informative manner.

The process for the user is to simply enter in the name of the desired Stock onto the designated search field. When executed, the data is then displayed on the dashboard.

In the backend, after the search query is executed, the request for the Stock is sent to the API service where it is then retrieved and outputted to the end user by way of the GUI of the application. This all occurs in real-time.

As this requirement is focused on retrieving the crucial data in which the application is dependent on, it is the main and highest priority requirement. All requirements are centralized around this requirement and the application is useless without it. It is essential to the overall system.

Functional requirement - 6

Use Case 6 - see Appendix

Requirement 6: Search Organisations

This requirement is regarding searching for an Organisations code (Stock listing).

Description & Priority

The requirement considers Search for stock codes as a function. The user will have pre-set choices to choose from, such as Google, Apple, Facebook etc. However, the user must be given the option to search for other publicly traded shares of a particular stock. The search option allows for this. This functional requirement focus on this issue.

Quandl is set to only recognise stock code. For example, Google, is traded as GOOGL so for the search to work, it must be written as WIKI/GOOGL. The function will look to returning the code WIKI/GOOGL when Google is inputted by the user.

The functional requirement is to allow the user to search and find the codes via a simple search box. The search box will allow the user to search for a company and the code to show as a result.

The functional requirement will require a list (CSV file) of possible searchable companies and the codes for those companies. It will then look to returning the codes from the list when a search query is inputted by the user.

Functional requirement - 7

Use Case 7 - see Appendix

Requirement 7: Search Codes

This requirement is regarding search for Stock data by the user using stock codes.

Description & Priority

The requirement considers Search for results based on stock codes as a function. The user will have pre-set choices to choose from, such as Google, Apple, Facebook etc. Users will have the option to search for a publicly traded company using stock codes which will generate results.

The Quandl API is set to only recognise stock code. For example, Facebook, is traded as FB so for the search to work, it must be written as WIKI/FB. The function will look to returning the results for the code WIKI/FB when inputted by the user.

The functional requirement is to allow the user to search for publicly traded stock using codes via a simple search box. The search box will allow the user to search for a company and the result for that said company to show in the forms of tables and charts.

Upon searching for the code, data will be retrieved from the Quandl API as per functional requirement 4 (Data retrieval). At which case, the data is fed into algorithms and the results outputted to the user.

The functional requirement will require the exact code for the company to be entered. An Error message will display in the case that the wrong code is entered by the user.

Functional requirement - 8

Use Case 8 - see Appendix

Requirement 8: Create Forecast

This requirement is regarding search for Stock data by the user using stock codes.

Description & Priority

The requirement considers creating forecasts based on search queries. The search queries entered by the user looks at retrieving the results of a particular stock listing via stock codes.

The forecast models created are dynamic in the since that forecasts will be produced for any given company. Linear regression is the main model used, the model looks at retrieving the predicted results based on historical data provided by the Quandl API. Data is pulled from the Quandl API upon the user entering a search query, fed into the forecast model and results are then outputted to the user by way of raw data in the form of a table and / or by charts.

The forecasts are set by default to output the data, using charts, dating from the date the stock was first listed. For example, Google was IPO was first listed in 2006. The charts will display data dating from 2006 to present. This is done for the purpose of allowing the user to compare the forecasted results with those of other services. The charts produced by other services should show similarities to those generated by the application.

Functional requirement - 9

Use Case 9 - see Appendix

Requirement 9: Charts

This requirement is regarding visualization of data displayed to the user.

Description & Priority

The requirement considers data visualization using charts. Charts will be generated using a visual tool / package known as Matplotlib to provide a visual overview of the data and provide a friendly user interface.

The visual aspect of the application is not of that important but for the sack of providing the data in a user friendly and aesthetic fashion, it's significant that charts are included. More so to also make the data more dynamic to the user and help average users (industry outsiders) better understand the data.

The charts will clearly indicate the stock price and whither it is set to increase in value or decline, when outputting forecast results.

Documentation requirements

Requirement 1:

The Technical Report requirement is that of a document that outlines and provides an overview of the development process.

Description & Priority

The Technical Report document will describe the progression and results of the project when it's complete. This document is a high priority document because it's essentially a manual of the development process and why everything was developed the way it is.

Non-Functional Requirements

The following is an in-depth explanation on the non-functional requirements of my web application project.

Performance/Response time requirement

With this project being centered around the stock market, it was important that the user gets the best possible response time. This is because the stock market revolves around real-time data. Using an API, data is grabbed and displayed to the user. The API that will be used will have to have low latency to get the fastest response from the market. Of course, with the API being free, stock data would update a few minutes late, other are updated every month.

Performance wise, the web app will be able to perform well across different platforms. Using the latest web standards and adapting to frameworks both on the front-end and the back-end, the system should be able to adapt on any device.

Availability requirement

As mentioned before, it is high priority that the application is available on as many platforms as possible. This is to provide end users with access anywhere at any given time. It was for this reason along with others that I have decided to build the application as a web app. With an internet connection, users will have access the site regardless of their device. Of course, it was important the experience remains the same regardless of platform, hence the use of framework.

Recover requirement

With the use of an API to extract real-time data, recovery is not an issue because data is not being stored in a database. The only data being stored is that of the end user accounts. There is no recovery protocol being taken aside from backups. The source code itself will be using some sort of version control for the sake of recovery. Bit-Bucket or Github will be used for this.

Robustness requirement

The system will be running using the Django framework so in case of errors such as that in registration, the framework will handle it. The only issue in terms of robustness is that of the API. Errors caused by the API will be out of my control if the issue isn't in my source code.

Security requirement

As perversely specified, data is not being stored because it's being pulled in real-time from an API. There is not much that can be done in terms of security. The security imbedded such as that to prevent SQL injections is handled by Django, the back-end framework.

Reliability requirement

For this requirement, the dependability of the web app will depend on the Reliability and condition of both the API and the web host. The only action that can be taken is to insure the source code of the web app is fined tuned and that the API and web host are reliable.

Maintainability requirement

Maintainability of the application and its system is essential especially for any future implementations. Frameworks provide the best form of maintainability for any application. Using Frameworks like Django for the back-end, I'm able to better maintain the source code and easily add, remove or change it. Protocols are handled by Django which itself is maintained by updates for different iteration.

Portability requirement

In computer programming, portability is the process of usability of software in different environments. As formerly acknowledged, the application is built as a web application to provide users with access at all time on numerous devices. Web applications are extremely flexible in terms of portability. So, long as the user is connected to the internet, they will have access to the site.

Extendibility requirement

Extensibility is a system's ability to have new functionality extended. It is a significant software design principle. As discussed in the maintainability requirement section, the use of frameworks provide the best possible developer environment in terms of maintainability and extensibility. With frameworks, new functionality can be easily added to the system with little to no hassle. Changes can be performed with less difficulty. In example of this in the front end is the grid system built into bootstrap. Changes to the user interface of the web application can be effortlessly implemented.

Reusability requirement

In software engineering, reusability is the adoption of existing system assets in some fashion within the product development process.

In terms of reusability within the code base, components from the framework(s) will be re-used for development of the application through the development life cycle. Different classes will be referenced and used for multiple parts of the system. Frameworks provide the necessary development blocks and as such different functions can be recycled throughout the code base.

Interface requirements

This section describes how the application's interfaces with other software products or users for input or output. Examples of such interfaces include APIs, web services, shared memory, data streams, and so forth. Most systems would have a GUI, for my one, it is decisive.

The application will run as a web application by way of a Graphical User Interface. User input will be carried out using keyboard and mouse on personal computers and in the case of mobile and tablets, by touch.

Regardless of platform, using twitter bootstrap, the application will be compatible for any device. This is to cater to all users at all time on multiple devices providing them with the same experience and giving them full accessibility always.

Data will be fed to the interface via an API. The data will go through algorithms to display the relevant information. Depending on the data, visualization tools such as Matplotlib may be used to generate significant charts.

Interface is to be clear and modern to provide the user with a good experience and display data appropriately. The interface will be clear and feature an aesthetic design so that it can be used by anyone without training or any prior experience.

GUI

In this section, I will discuss the mock-ups / wireframe of the application and explain how they are linked.

I will address the above requirements in the design and how the mock-ups are in line with the functional requirements above.

- **Index page (Appendices: Mock-up 1)**

The index page features a clean design. It will be the first page a user will see when they access the site. The page along with the rest of the site is built and optimized for mobile. Users don't require logging in to access it.

- **Login page (Appendices: Mock-up 2)**

The login page is where the user will be able to log into the site. As shown in the mock-up, the Google API and the Facebook API will be used for login. Login can also be done using an email and password.

- **Signup page (Appendices: Mock-up 3)**

The signup page is where the user will be able to sign up. As shown in the mock-up, the Google API and the Facebook API will be used for registration aside email registration.

- **Dashboard page (Appendices: Mock-up 4)**

The dashboard page is where the user will see all the information. Stock data will be outputted as shown in the mock-up. The design is aesthetic and very user friendly. When designing the page, mobile devices were kept in mind. Login is required to access this page.

- **Charts page (Appendices: Mock-up 5)**

The charts page will show charts generated by data. The charts will be displayed as referenced in the mock-up. Login is required to access this page.

- **Tables page (Appendices: Mock-up 6)**

The table page will accommodate a table that will contain content as displayed in the mock-up. The table will include a list of the top performing Stocks of the week. This will be done with machine learning. Login is also required to access this page.

Application Programming Interfaces (API)

My system will use an API to provide the application with data in which to use. The API will use real-time data that will be fed through algorithms to display the relevant / appropriate information. The information will then be showcased on a dashboard either in plan forum or through a visual charts such as those generated with Matplotlib.

The API also provides historical data which will be used for machine learning. This data will also go through algorithms before being presented in the user's dashboard.

System Architecture

The following Class diagram (Figure 9 – Next page) is a summary of the system architecture being adapted. The reason for the chosen architecture is solely based on how external data is being used for and throughout the system. The class diagram also touches upon the login system being implemented.

System Evolution

The system can and could evolve over time to incorporate other features as the need for them arises. There is potential to monetize the system by adding in a subscription model into the user accounts. Features that could be added include customization of user dashboards to integrate their preferred Stock.

System Architecture - Class Diagram

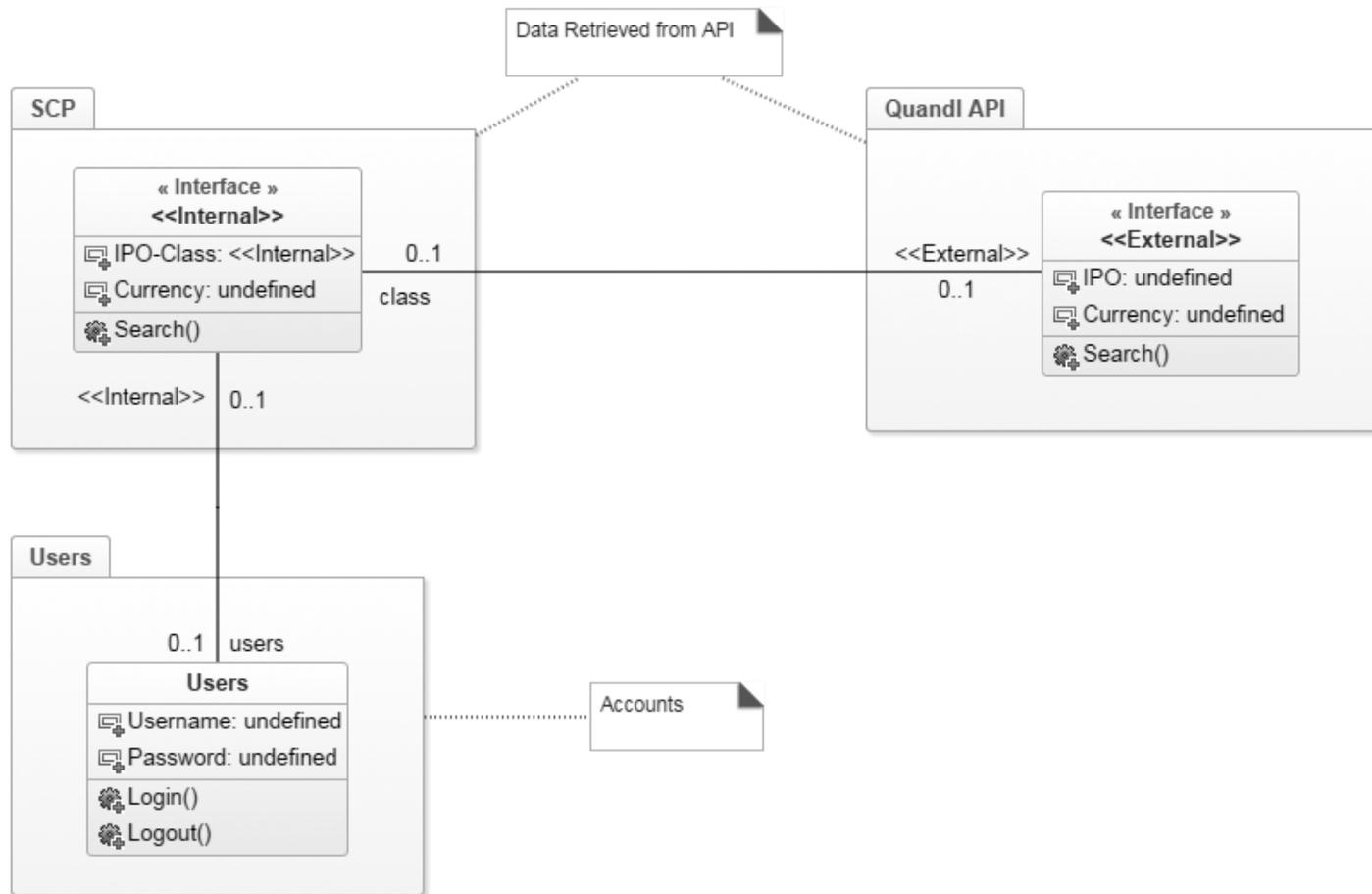


Figure 2

Testing Plan

At the end of the development process as well as in-between each completed tasks, system tests and integration tests will be carried out to evaluate the given tasks. A list is to be prepared of what tests are to be done and what the expected result should be. Each test will be carried out at least three times to ensure everything is functioning as planned. Errors will be recorded and fixed in which the same tests will be done as to confirm everything is working correctly.

There will be no one tool used for testing but rather built in testing tools within IDEs such as those within PyCharm.

Speed analysers will be used to test the speed of the live site. More than one will be used for this. This test might not seem necessary but it's important to test the site's loading time on different devices such as mobile and tablet.

Customer testing

Regarding evaluating the system with an end user, I plan on launching a live working version of the project and setup a survey in which I will question users who evaluate it in relation to their judgment and interaction of the web application. Appropriate changes will be done if deemed necessary and will be recorded.

Monetization

The application has the potential of being monetized for commercial use by adapting a freemium model. There could be different bundles. A user can pay for a bundle that best suits them. Each bundle will look at delivering specific results. Bundles could allow the user to have unlimited search and have access to more stock listings. The free option would have a limited-on search per day and is limited to the listings available to them.

Note that this is a potential option. The application delivered will not have monetization features.

Conclusions

In my opinion, the web application I developed will better allow investors to grow their portfolio as it will save time. By providing them with the necessary information they need on any stock, along with useful data, I'm improving their experience and giving them tools to invest smartly.

Advantages

- Clean minimalistic UI – User-friendly
- Can be used by anyone – New investors can adapt
- Helpful and useful information
- Machine learning – trends within the market
- Personalised user experience
- Security

Disadvantages

- Real-time data is delayed (data source is free)
- Not enough market data (limited by free API)
- Machine learning is not as advanced as competitors

Opportunities

- Monetization – Subscription service could be added to provide monetizing of the application. Features could be locked behind paywall while others are provided free through ad based revenue.
- Mobile App – While the web application is mobile compatible and optimized for such devices, a native mobile application is possible as the web app is developed with specific standards and frameworks that allow for such. This could better improve the performance of the app and provide a unique, much richer experience for users on their platform of choice.
- User Experience – As the application grows, the dashboard can include more functions that could better provide the user with a personalized experience. Users can customize the dashboard to their liking and track such things as companies, related news and stocks.

Limits

Since the application is powered by data provided by an API, the project is limited to it (the API) on what it can do. Multiple APIs could be added but this will require more resources and is unrealistic. Premium APIs are also not justifiable given the price tag.

Further development or research

Further development could lead to a lot of possible changes and improvements to the project. With more resources, this project could possibly match competing stock trading services. Such resources include manpower, time and money to develop a service that can provide experiences seen on similar products.

The service, if provided the resources could include real-time data that's not delayed through the use of a premium market API. It could include multiple APIs to provide access to more markets.

Native applications for different platforms can be developed as well for better experiences.

Another thing that could be added is actual stock trading within the site itself. This could be done through an API but would require a lot of work in terms of security and implementation of such a function.

However, as of now, these developments could not be achieved at the given time frame and resources.

Testing

Testing of the code

This section looks at testing of the code base.

When testing the code base, several tests were done to check for errors and vulnerabilities within the code of the application.

Tests were performed on the applications backend by checking for the output given. This was done by comparing the results with other services.

User authentication was tested by checking for account registration on the admin dashboard. The aim of the test was to see if all the user information was registered and saved to the database. Checking for login was done by simply testing the login system. Email was disabled as there were too many issues regarding the need for a mailing service such as Mail Chimp. To ensure that the appellation's dashboard can still be accessible in case a user can't login, the URL path /dashboard/control-panel/ can be added. This was done on purpose for the use of testing and running the application locally.

Testing of the GUI (Graphical user interface) was done by examining the application on multiple devices. The application was tested on Desktop, Mobile and Tablet. This was done by simply vesting the live hosted version of the application. The application was also tested throughout production using the built-in emulators found in Google Chrome. Furthermore, to test for browser capabilities, the app was tested on other browsers such as Firefox and Microsoft's Edge. All the main components were tested to make sure that everything is working across browsers.

Testing of the Forecasts

This section focuses on testing and verifying the results of the application's forecasting. The test is carried out by comparing the forecasted results of stock, for a specific period, shown in charts with the actual results found in alterative services such as Google Finance.

Of course, forecasted results for stock is never always accurate in the real world due to external factors but the compression will look for similarities between the forecasted price and the actual price.

Again, for the testing, the forecast is set to look to predict the price in a past period as to be compared with actual results.

Application forecast results for Google -

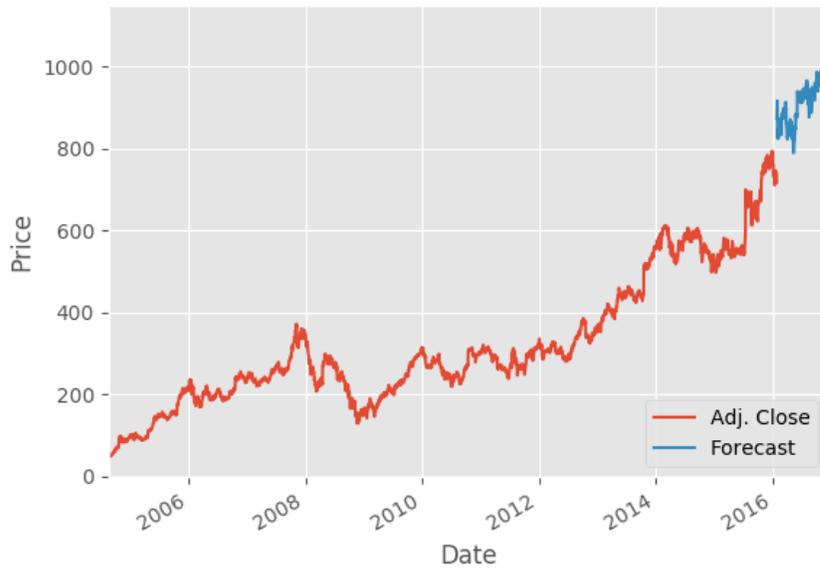


Figure 3

Results from Google Finance for Google -



Figure 4

Customer Testing

The application was shown to users who tested it and provided feedback. The following is the feedback collected.

User 1

I used the application and found it to be very clear, informative and friendly to use. The design makes it look modern and the use of charts makes the information shown to be very easily understood.

User 2

I was asked to test the application and provide feedback. I looked at the application and thought it was useful for people who are not all the familiar with investing. This was because of how clear the information was displayed. The particular use of predicting the value of a stock is valuable and something I think any investor would at least look at before investing their money. If there was one negative I found, it's the charts which aren't all that interactive. I also feel more information can be shown but given the time it took and the resources to develop what's already there, I believe this on its own is excellent and is a building block for something more in the future.

User 3

I found the idea of trying to estimate the value of stock to be very interesting and if I were to invest, this would be something I'd use before making the finale decision. I see potential in this application mainly because of the way information is shown. The design is great and it operates very fast. I feel as though there could be more information on stock though. Another thing that would be useful is to be able to set the date of the forecast.

Appendices

Use Case - 1

Hosting and accessibility

Scope

The scope of this use case is the process of redirecting the user to then dashboard upon login.

Description

This use case describes the process for authenticating and redirecting the user to the dashboard. It highlights the necessary steps taken by the end-user to request authorization from the system to access the dashboard.

Use Case Diagram

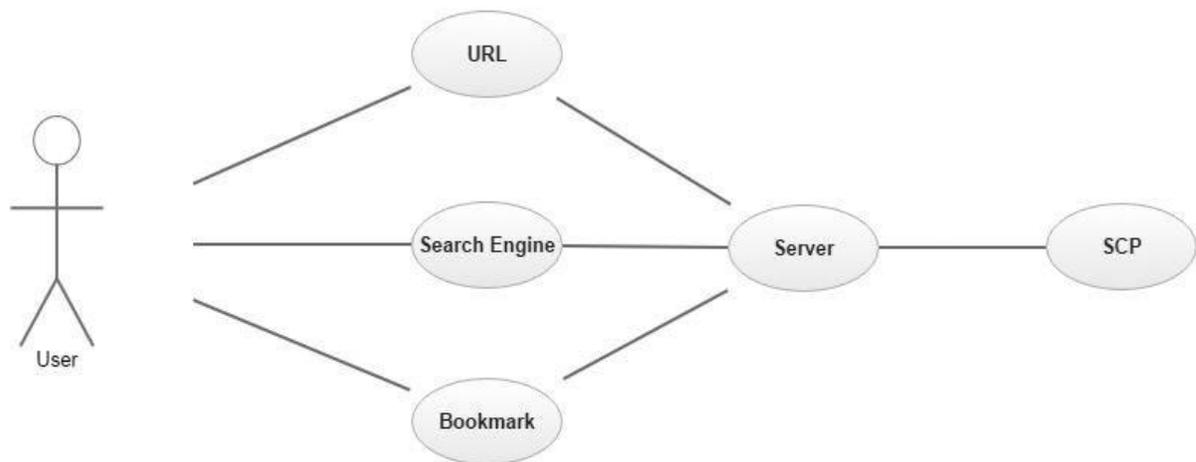


Figure 5

Flow Description

Precondition

The system is live and operational. The user has internet access on a suitable device (mobile, tablet, desktop / laptop).

Activation

This use case starts when the user wants to visit the site

Main flow

1. The user opens web browser
2. The user navigates to the website by way of search engine, URL or Bookmark
3. The host's server receives request
4. The server recognizes request (See A1)
5. The user gets redirected to the website (See E1)

Alternate flow

A1: Site not found

1. User tries another search query for the site
2. The use case continues at position 2 of the main flow

Exceptional flow

E1: 404: system not found error – site offline / temporarily unavailable

1. User receives 404 error
2. Site visitor tries reloading page
3. If access not granted, wait for issue to be resolved

Termination

The system's server provides access and redirects the user to the site.

Post condition

The system continues being operational for other users.

Use Case - 2

Registration and authentication

Scope

The scope of this use case is the registration process for users and authentication of both users and administrators.

Description

This use case describes the registration process for users while also highlighting the differences and authorization privileges of those of an administrator.

Use Case Diagram

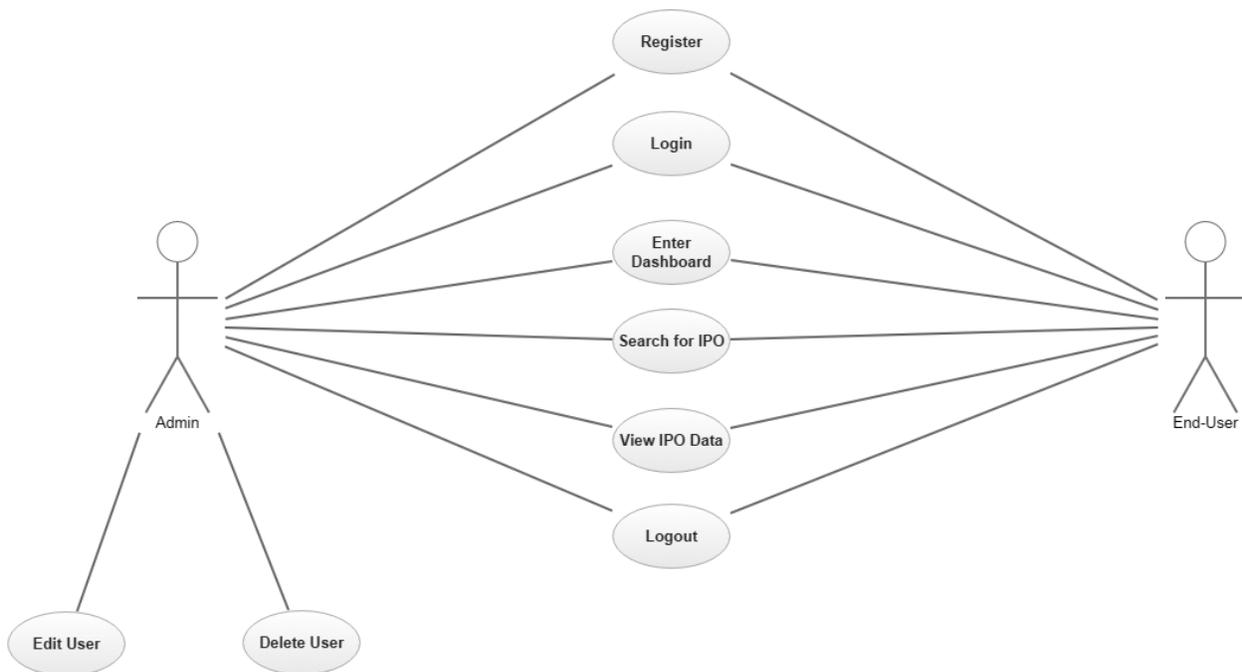


Figure 6

Flow Description

Precondition

The system is in initialisation mode and the user is on the site.

Activation

This use case starts when the user signs up.

Main flow

1. The system identifies the action of registration
2. The user enters in their information (See A1)
3. The system stores it into the database
4. The user confirms signup – Activates account (See E1)
5. The user is logged in

Alternate flow

A1: User Already exists

1. The system identifies the correct user
2. The use case continues at position 5 of the main flow

Exceptional flow

E1: Account not activated

1. The system identifies invalid account activation
2. The system notifies end user
3. The use case continues at position 5 of the main flow

Termination

The system presents the dashboard page when successful. It also saves all entered information to the database.

Post condition

The system goes into a wait state until the user enters his login details.

Use Case - 3

Social authentication

Scope

The scope of this use case is the process of registering / authenticating the end user using Social authentication.

Description

This use case describes the process for registering and / or authenticating the user using third party services such as their Facebook or Google account. It highlights the necessary steps taken by the end-user to request authorization from the system to access the dashboard.

Use Case Diagram

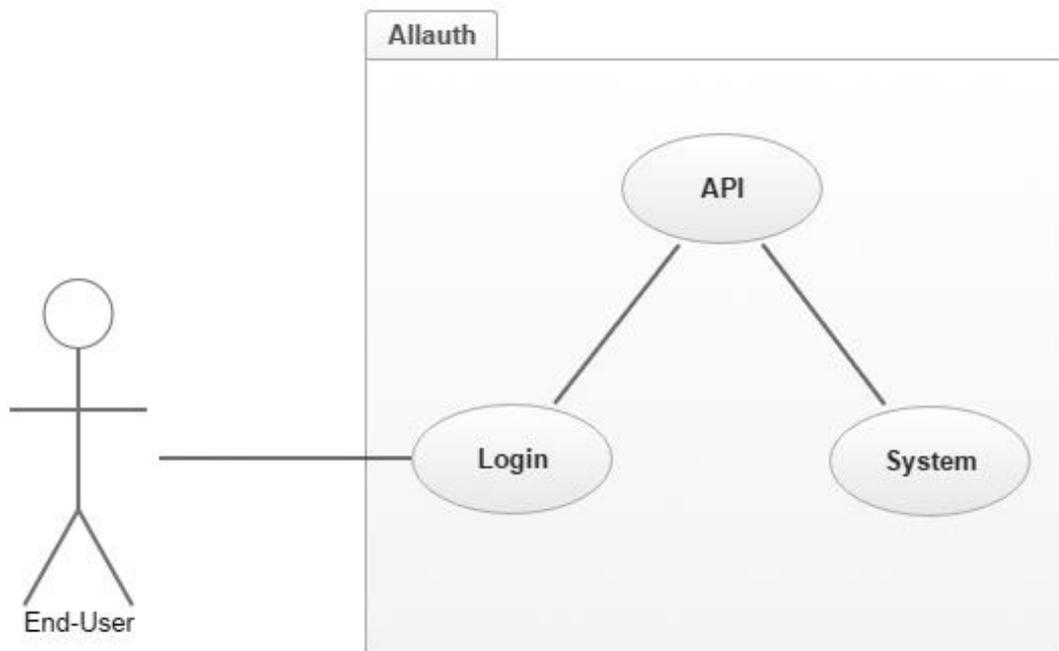


Figure 7

Flow Description

Precondition

The system is in initialisation mode and the user is on the site.

Activation

This use case starts when the user signs up or logs in.

Main flow

1. The system identifies the action of registration or login
2. The user selects the desired social account to register or login with (See A1)
3. The system requests information from the API of the selected social service
4. The API identifies the user (See E1)
5. The system pulls the information and stores it into the database
6. The user is logged in

Alternate flow

A1: User Already exists

1. User did not login to the social account and authorise usage
2. The use case continues at position 1 of the main flow

Exceptional flow

E1: Account not activated

1. The system / API does not identify the user (service might be down)
2. The system notifies the end user
3. The use case continues at position 1 of the main flow

Termination

The system presents the dashboard page when successful. It also saves all entered information to the database.

Post condition

The system goes into a wait state until the user logs out.

Use Case - 4

Dashboard

Scope

The scope of this use case is the process of redirecting the user to then dashboard upon login.

Description

This use case describes the process for authenticating and redirecting the user to the dashboard. It highlights the necessary steps taken by the end-user to request authorization from the system to access the dashboard.

Use Case Diagram

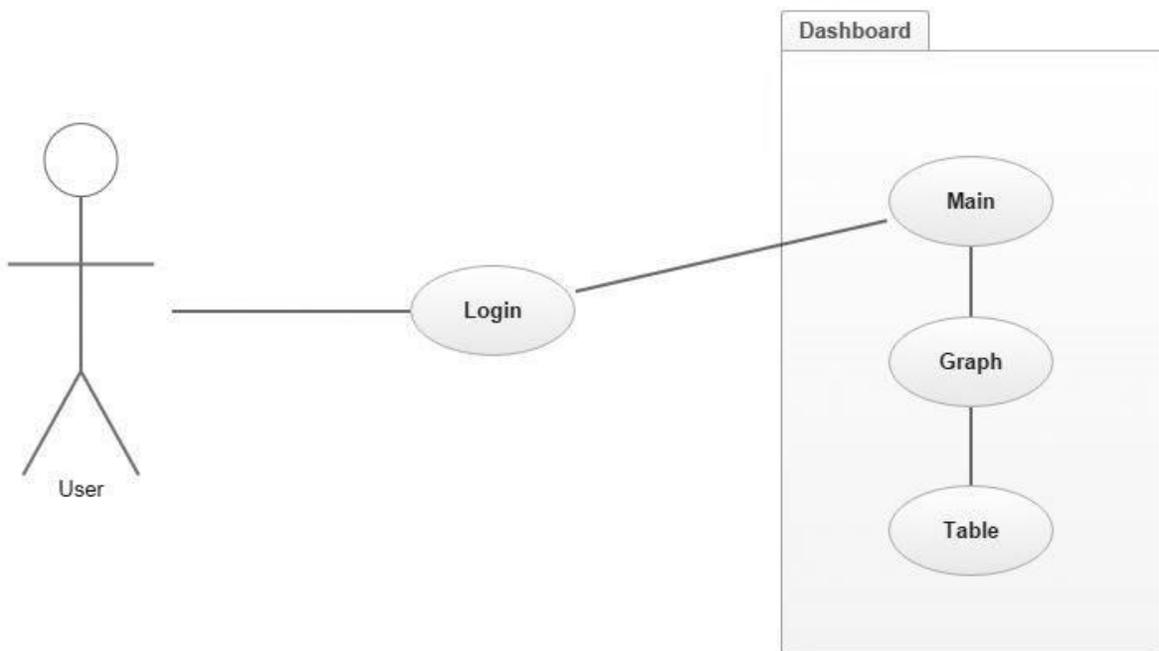


Figure 8

Flow Description

Precondition

The system is in initialisation mode and the user is logged on to the site.

Activation

This use case starts when an end user is logged in and on the dashboard.

Main flow

1. The user logs on to the site
2. The system identifies the user's profile (See A1) (See E1)
3. The system signs in the user
4. The user redirects to the dashboard
5. The user can navigate the three sections of the dashboard

Alternate flow

A1: The user was already logged in

1. End user visits the site
2. The system detects browser cookie
3. The use case continues at position 2 of the main flow

Exceptional flow

E1: User not identified

1. The system prints "invalid login – signup?" message
2. The user redirects to signup page
3. User registers for an account
4. The use case continues at position 1 of the main flow

Termination

The system provides access and redirects the user to the dashboard if / when the user is authenticated (signed in)

Post condition

The system goes into a wait state until the end user signs in.

Use Case - 5

API Data Retrieval

Scope

The scope of this use case is the API Data Retrieval process.

Description

This use case describes the process for Data Retrieval from the API. It highlights the necessary steps taken by the end-user to request the data and the way in which the system retrieves it from the API.

Use Case Diagram

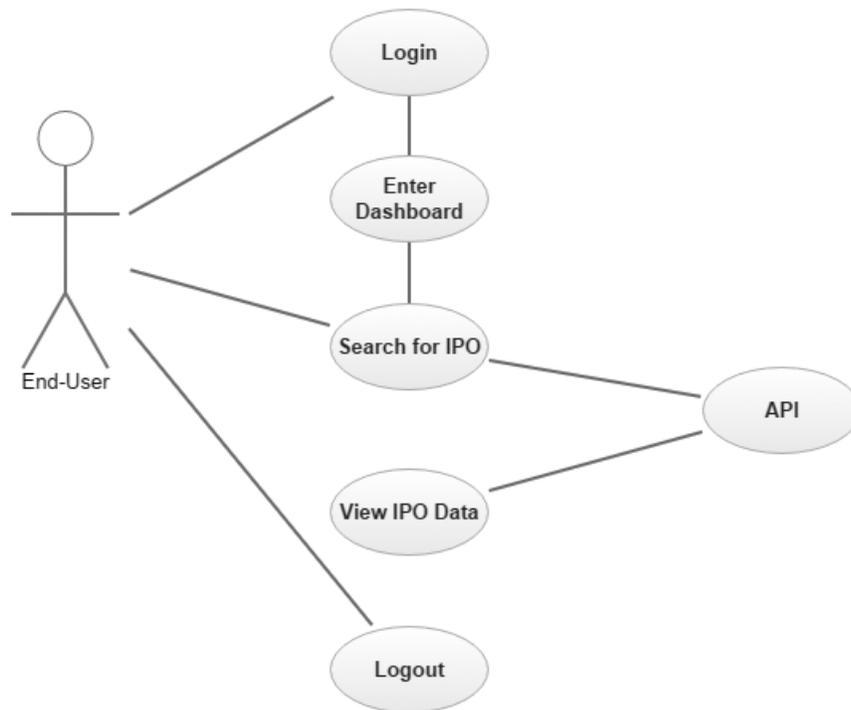


Figure 9

Flow Description

Precondition

The system is in initialisation mode and the user is on the site.

Activation

This use case starts when an end user is logged in and on the dashboard.

Main flow

1. The user inputs the desired stock name into the search field
2. The system identifies the input request in the search field (See A1)
3. The system requests the data from the API
4. The system retrieves the data from the API (See E1)
5. The data is presented to the user

Alternate flow

A1: stock name is not identified

1. The system prints "invalid stock" message
2. The user enters another stock
3. The use case continues at position 2 of the main flow

Exceptional flow

E1: API does not retrieve data

1. The system prints "invalid stock" message
2. The user enters another stock
3. The use case continues at position 2 of the main flow

Termination

The system presents the requested stock (data), when successful, to the user dashboard of the application.

Post condition

The system goes into a wait state until the end user requests another stock.

Use Case - 6

Search Organisations

Scope

The scope of this use case is the Search process for an organisation listing code.

Description

This use case describes the process for search and finding stock codes for companies. It highlights the necessary steps taken by the end-user to request the data and the way in which the system retrieves it from the CSV file (SP500) where a list of codes per company is kept.

Use Case Diagram

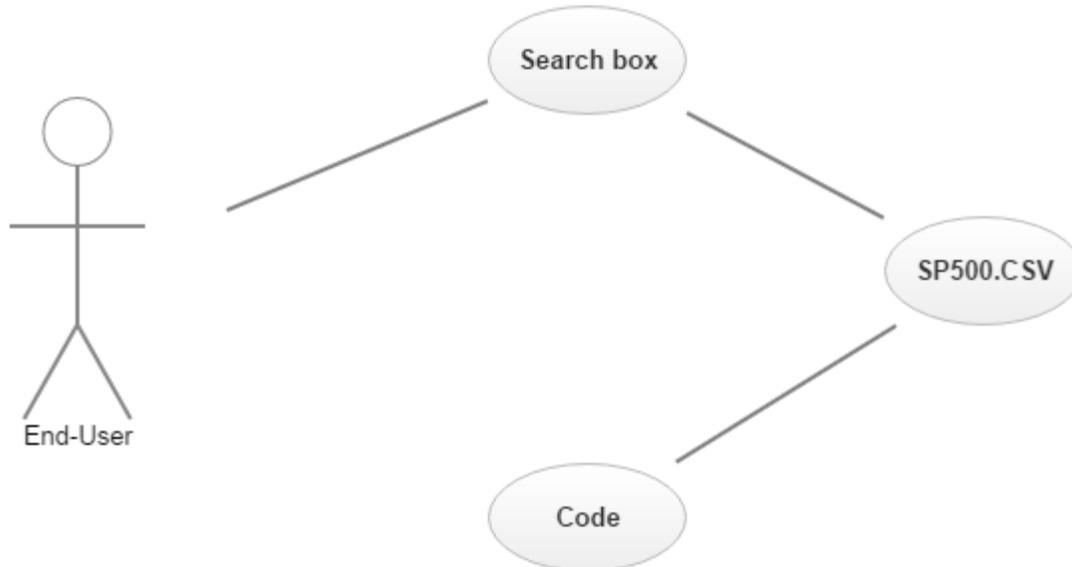


Figure 10

Flow Description

Precondition

The system is in initialisation mode and the user is on the site, logged in.

Activation

This use case starts when an end user is logged in and on the dashboard.

Main flow

1. The user inputs the desired Company name into the search field
2. The system searches for the Company name within the CSV file (SP500)
3. The system identifies the input request in the search field (See A1)
4. The system retrieves the code from the CSV file (SP500) (See E1)
5. The code is presented to the user

Alternate flow

A1: Company name is not identified

1. The system prints "Error!" message
2. The user enters another Company name
3. The use case continues at position 4 of the main flow

Exceptional flow

E1: The system does not retrieve the code

1. The system prints "Error!" message
2. The user enters the same (or another) Company name
3. The use case continues at position 5 of the main flow

Termination

The system presents the requested stock code, when successful, to the user dashboard of the application.

Post condition

The system goes into a wait state until the end user requests another stock listing.

Use Case - 7

Search Codes

Scope

The scope of this use case is the Search process for stock data, using a company's listing code.

Description

This use case describes the process for search using stock codes for companies. It highlights the necessary steps taken by the end-user to request the data for a specific listing and the way in which the system retrieves it from the Quandl API.

Use Case Diagram

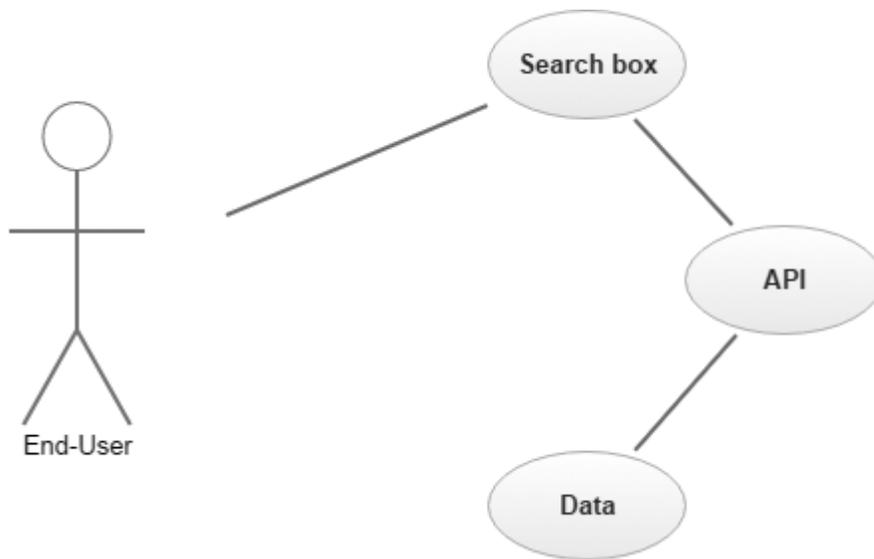


Figure 11

Flow Description

Precondition

The system is in initialisation mode and the user is on the site, logged in.

Activation

This use case starts when an end user is logged in and on the dashboard.

Main flow

1. The user inputs the stock code for the desired listing into the search field
2. The system sends a request to the Quandl API for the listing
3. The API identifies the request (See A1)
4. The system retrieves the data from the Quandl API (See E1)
5. The data is presented to the user

Alternate flow

A1: Stock code is not identified

1. The system prints "Error!" message
2. The user enters another stock code
3. The use case continues at position 4 of the main flow

Exceptional flow

E1: The system does not retrieve the code

1. The system prints "Error!" message
2. The user enters the same (or another) stock code
3. The use case continues at position 5 of the main flow

Termination

The system presents the requested stock data, when successful, to the user dashboard of the application.

Post condition

The system goes into a wait state until the end user requests another stock listing.

Use Case - 8

Create Forecast

Scope

The scope of this use case is the process for creating and generating forecasts based of data pulled from the Quandl API.

Description

This use case describes the process for creating forecasts based on data pulled from the Quandl API. The application's user dashboard features a search box for entering codes for stock listing. This search box is used to search for a specific stock. Upon entering a search query, the system send a request to Quandl using the API. Quandl then sends back the requested data for the stock listing to the application. At which point, the data is fed into out predictive Liner regression model. The results of the model are then outputted to the user by way of raw data within tables and / or through charts generated by Matplotlib.

Use Case Diagram

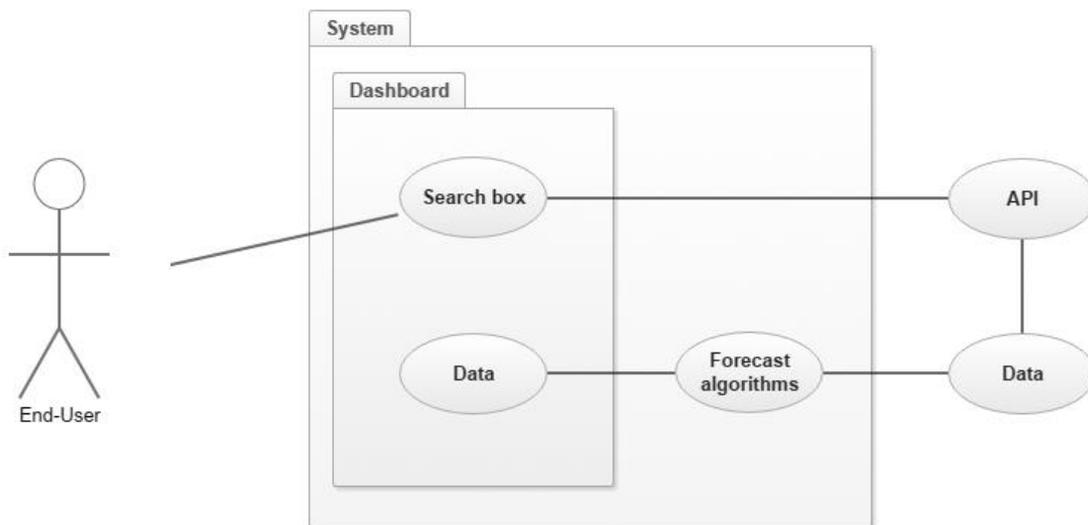


Figure 12

Flow Description

Precondition

The system is in initialisation mode and the user is on the site, logged in.

Activation

This use case starts when an end user is logged in and on the dashboard.

Main flow

1. The user inputs the stock code for the desired listing into the search field
2. The system sends a request to the Quandl API for the listing
3. The API identifies the request (See A1)
4. The system retrieves the data from the Quandl API (See E1)
5. The data is parsed (fed) into our Liner regression model (algorithm)
6. The results of the model are then presented to the user

Alternate flow

A1: Company name is not identified

1. The system prints "Error!" message
2. The user enters another stock code
3. The use case continues at position 4 of the main flow

Exceptional flow

E1: The system does not retrieve the desired data from the Quandl API

1. The system prints "Error!" message
2. The user enters the same (or another) stock code
3. The use case continues at position 5 of the main flow

Termination

The system presents the requested forecasts for the desired stock, when successful, to the user dashboard of the application.

Post condition

The system goes into a wait state until the end user requests another stock listing.

Use Case - 9

Charts

Scope

The scope of this use case is the process for creating and generating charts based of the forecast models (algorithms).

Description

This use case describes the process for creating charts based on data pulled from the forecast models. The application's user dashboard will allow the user to search for the desired company listing. When doing so, data will be pulled from the Quandl API and will be parsed by the forecast algorithms. The results of the forecast model are then outputted to the user visually using charts generated by Matplotlib. This all happens in real time from the moment a user submits his or her search query.

Use Case Diagram

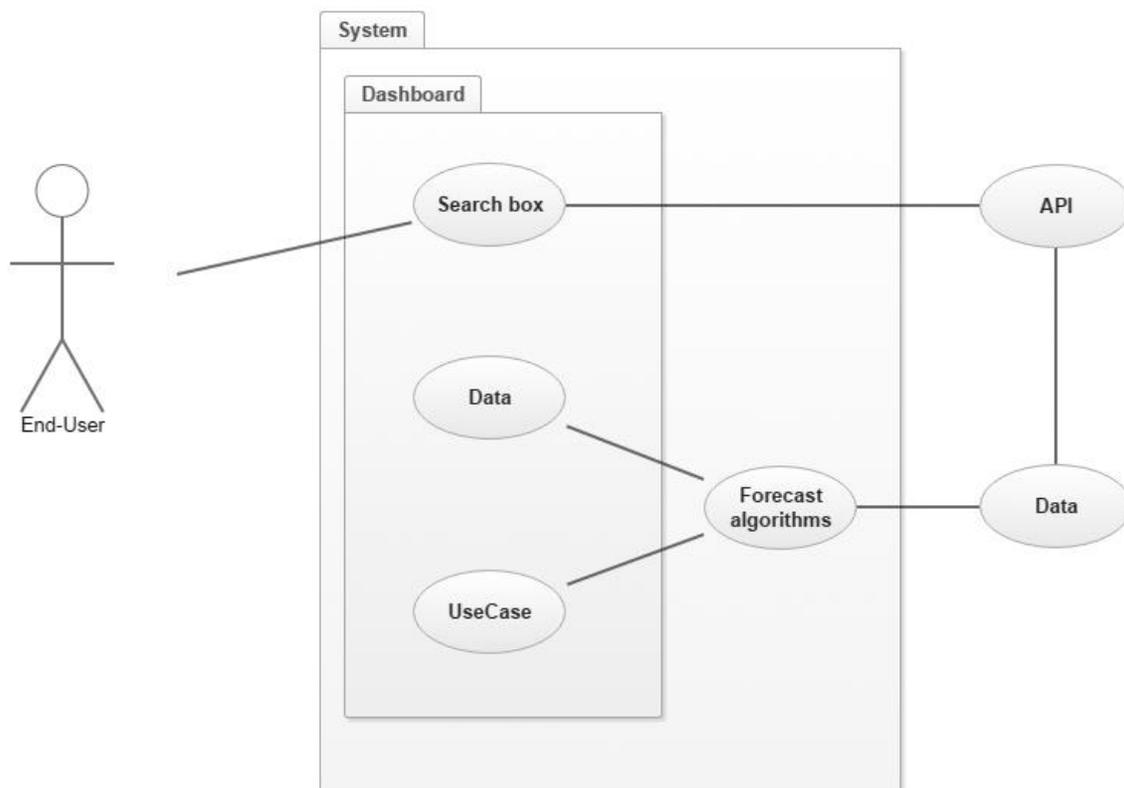


Figure 13

Flow Description

Precondition

The system is in initialisation mode and the user is on the site, logged in.

Activation

This use case starts when an end user is logged in and on the dashboard.

Main flow

1. The user inputs the stock code for the desired listing into the search field
2. The system sends a request to the Quandl API for the listing
3. The API identifies the request (See A1)
4. The system retrieves the data from the Quandl API (See E1)
5. The data is parsed (fed) into our Linear regression model (algorithm)
6. The results are then parsed to generate charts
7. The results of the model are then presented to the user visually in the form of a chart

Alternate flow

A1: Company name is not identified

1. The system prints "Error!" message
2. The user enters another stock code
3. The use case continues at position 4 of the main flow

Exceptional flow

E1: The system does not retrieve the desired data from the Quandl API

1. The system prints "Error!" message
2. The user enters the same (or another) stock code
3. The use case continues at position 5 of the main flow

Termination

The system presents the requested charts for forecasts of the desired stock, when successful, to the user dashboard of the application.

Post condition

The system goes into a wait state until the end user requests another stock listing.

Mock-up 1: Index page

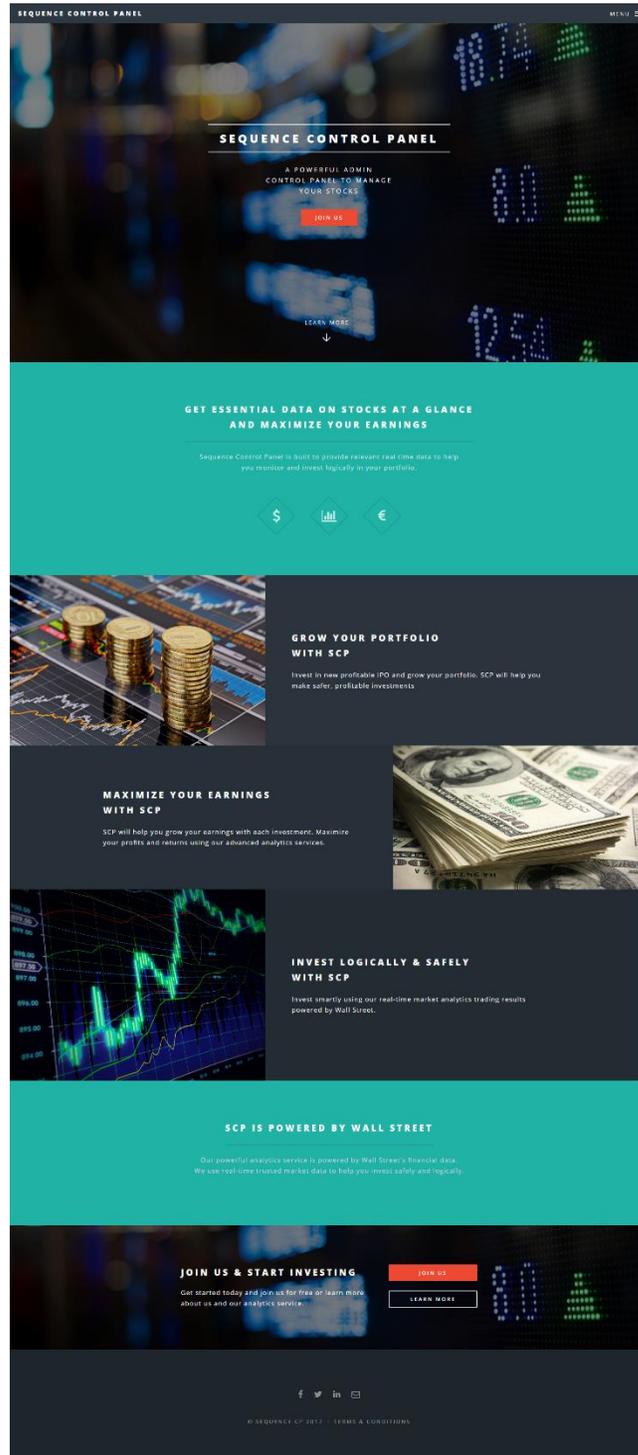


Figure 14

Mock-up 2: Login page

The image shows a web page mock-up for a login interface. At the top, a dark grey header contains the text 'SEQUENCE CONTROL PANEL' on the left and 'MENU' with a hamburger icon on the right. The main content area has a teal background. In the center, a white card titled 'LOG IN SECURELY' contains the following elements: two social login buttons for Facebook (blue with 'f') and Google+ (red with 'g+'); an 'Email address' section with a text input field labeled 'Your E-mail'; a 'Password' section with a text input field labeled 'Your Password'; a dark grey 'LOG IN' button; a link 'Forgot your password? Click here'; and a link 'Not a Member? Sign up here'. At the bottom of the page, a dark grey footer contains social media icons for Facebook, Twitter, LinkedIn, and Email, followed by the text '© SEQUENCE CP 2016 | [TERMS & CONDITIONS](#)'.

Figure 15

Mock-up 3: Signup page

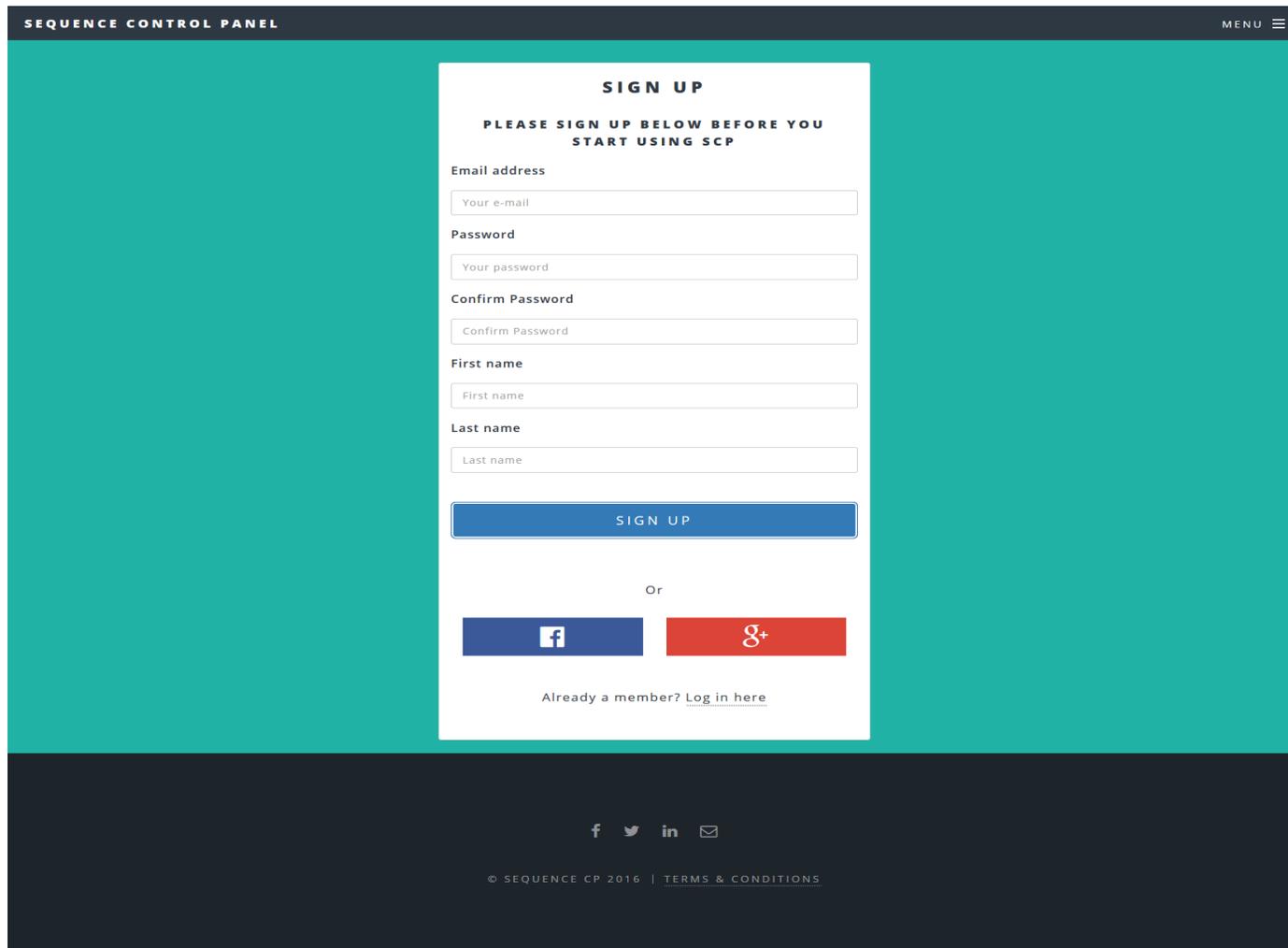


Figure 16

Mock-up 4: Dashboard page

SCP
👤

Search Organisations

Search organisations:

Suggestions: Google Apple Microsoft Facebook

Note: Use this to search for the Stock code of a publicly traded company

Latest stock prices...

Enter the stock code:

Suggestions: Google Apple Microsoft Facebook

Note: The tag "WIKI/" must be included when searching

WIKI/GOOGL

	Adj. Open	Adj. High	Adj. Low	Adj. Close	Adj. Volume
Date					
2017-04-24	868.44	879.9600	866.1100	878.93	1691896.0
2017-04-25	882.26	892.2500	879.2801	888.84	2020460.0
2017-04-26	891.39	892.9900	885.1500	889.14	1302032.0
2017-04-27	890.00	893.3774	887.1800	891.44	1817740.0
2017-04-28	929.00	935.9000	923.2200	924.52	3753169.0

Graphing price by date - Close and Forecast

Graphing price by date - Close

Figure 17

Mock-up 5: Charts page

SCP



Generate past stock price chart

Enter the stock code:

WIKI/GOOGL

Generate Chart

Suggestions: Google Apple Microsoft Facebook



Dashboard



Charts



Tables

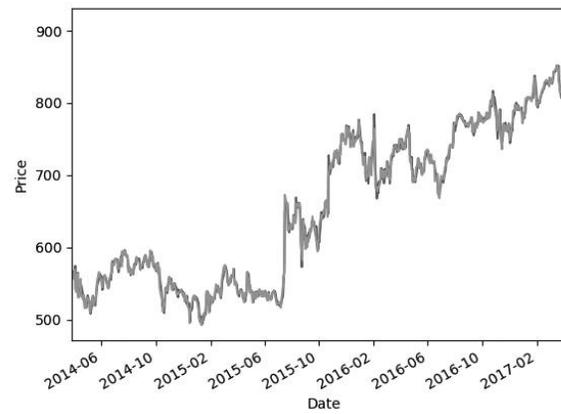


Figure 18

Mock-up 6: Tables page

SCP



Retrieve stock forecasts

Enter the stock code:

Suggestions: Google Apple Microsoft Facebook


Dashboard

Stocks Forecast for WIKI/GOOG

	Forecast
Date	
2017-03-20	878.162445
2017-03-21	889.226511
2017-03-22	886.102747
2017-03-23	889.746436
2017-03-24	931.623182


Charts


Tables

Figure 19

Reflective Journal - September

Student Name: Mina Gaid

Programme: BSc in Computing (Data Analytics)

Month: September

My Achievements

This month, I could successfully pitch my project idea to a panel of judges. Before doing so however, I had to come up with an idea, research it and roughly plan it out.

I specialise in Data analytics and I wanted my project to reflect this. Because data analytics involves a lot of statistics, I felt it was important to find an area that involves numbers such as that of the financial sector. This led me to the idea of building a web app that works with market and financial data. I ended up choosing to build a stock analytics dashboard which displays relevant statistical figures on IPO and to provide predictions on the stock value.

After choosing an idea, I went ahead and planned it out, choosing the technology stack to build the project with and researching the tools I'll be using along with what features are to be included.

When this was done, I was finally ready to pitch my proposed project. Due to the extensive planning and research that was done, the project was approved and I was given the go ahead to build it.

My Reflection

I felt, it worked well when it came to researching and planning the idea for my project. It helped me get an idea of what I wanted to build and what I should better do to improve it. The extensive planning will help me when I get started on building the dashboard and prepare me for any future changes.

However, I was not successful in when it came to pitching the idea of adding predictions within the dashboard. It was recommended that instead of predicting the stock's value (whether it will increase or decrease) I would do better to make predictions on a much smaller scale due to the complexity of the original idea.

Intended Changes

Next month, I will try to build the bases of my intended project. I plan on using frameworks both for front-end and back-end development. This is to ensure my tool is future proof, developer friendly and built based on web standards. I realised the importance of frameworks during my work placement and intend on implanting what I learned in the development of this project. I had already researched and picked a framework to work with but a few more were suggested to me by the panel of judges that I will consider.

Given that I will no longer be making a predication of the overall stock, I realised that I need to come up with relevant productions at a smaller scale such as calculating the mean of the IPO. This also means I must research and find a suitable algorithm.

Supervisor Meetings

Date of Meeting: Has not yet been set

Items discussed: Has not yet been set

Action Items: Has not yet been set

Reflective Journal - October

Student Name: Mina Gaid

Programme: BSc in Computing (Data Analytics)

Month: October

My Achievements

This month, I could come up with a wireframe for my intended web application. I spent a lot of time on this as I personally think the way in which data is displayed to the user is crucial. Creating a user-friendly GUI that plans for different interactions will make my project stand out more. It is also important that my design for the GUI is suitable for cross platform development (for different devices). It must be user friendly and display all relevant information whether on mobile, tablet, laptop or desktop. Lastly, the aesthetic is to be modern for branding.

After I completed the wireframes, I went about creating the project on Django. I set up all the main files and structures as well as the intended directories and the different planned files that will be included (index.html, dashboard.html, about.html). I then configured all the URL paths for the web app as Django requires URLs to be setup before the page's display. Even during development. Being new to Django, this was challenging as I couldn't get the URL paths linking to different pages but with research, I was able to figure it out.

Subsequently, I got a head start on the front-end development process. I began by creating the HTML structure of each file using (Twitter) Bootstrap and then commenced working on the custom CSS elements that will be used across all pages. It took about two weeks until I could start matching what I envisioned when designing my wireframe.

My Reflection

Getting the URLs of files and paths for the different directories such as images and CSS stylesheets was difficult. Django operates using a unique file architecture in which file paths must be linked in dynamically. This is done by placing the CSS file, for example, into the "Static" folder, which is referred to in the dynamic path. Because of this, it was important I get all the paths and URLs setup correctly to prevent any changes in the future. Being new to the Django framework, I questioned if the files were setup appropriately but given feedback and research from online boards such as Stack Exchange, I am sure that its perfect.

I am happy overall with the front-end of the app. Testing it on different devices and browsers I can see that only minor changes need to be done. The design is modern, user friendly, aesthetic and compatible with different platforms. This was my goal and I am proud of the results.

Intended Changes

I don't necessarily intend on making any major changes to what I have done this month but I hope to improve it further before progressing. Next month I will try and pinpoint any UI flaws that need to be fixed. I will look for changes that need to be made on the front end that require readjustment and get feedback from classmates who will assist in finding components that should be changed/moved/fixed. I will then add on to it for aesthetic purposes.

For development needs, I will create a GitHub repository to maintain my web app. This is mainly for the sake of using version control but also to manage the development of the project as I found myself shuffling between different versions, copies and backups of my app. Even more so when working between my Laptop and desktop.

Supervisor Meetings

Date of Meeting: Has not yet been set

Items discussed:

- Best approach for implementing the API
- Statistical formulas in which to use

Action Items: Has not yet been set

We received our supervisors at the end of the current month. Unfortunately, however, I was not able to book a meeting with him. This is due to reading week. I have nonetheless contacted him by email and planned a meeting for next month following reading week.

Through email, I gave a brief description on what I'm doing for my project and what I hope to achieve. I asked for guidance regarding the best approach for implementing the API I intended to use and the statistical formulas I wish to apply. I in turn received some advice and information which I plan to consider and will prepare more related detail questions for our planned meeting next month (November).

Reflective Journal - November

Student Name: Mina Gaid

Programme: BSc in Computing (Data Analytics)

Month: November

My Achievements

This month, I could start on the registration system which I plan on including as part of the system. The framework that I am using (Django) has a built-in user management system. Unfortunately, Django did a complete overhaul in their latest update in which made a lot of the online tutorials obsolete. Being new to Django I intended to follow a video or documents to enable and configure registration but this was difficult to do with their being a new system in place. The documentation for Django was difficult to comprehend for a beginner but I could follow it and partially in able the system to register users.

When I first started this project, I intended on using a Django app (library /package) which provided and enabled a generic registration system. Such packages are available in droves on GitHub. One example was Allauth, a package for local registration and social authentication. This package would only require configuration in the settings.py file in my application and the system would be complete. However, I can no longer use this because it was deprecated with the latest iteration of Django. What I have achieved so far with registration will take some time. I will have to work on it some more next month to get it fully functioning. The task has taken up all the time I intended to use for the project and I cannot lose anymore. I will continue researching and considering forums such as Stack Overflow for assistance in what steps to take or sources to consider.

My Reflection

I'm quite worried regarding time as I still have a lot to do regarding implementing the API and applying machine learning into the project. The difficulties I'm having with the registration system were not planned for as I am using a framework that's meant to ease such tasks in the development process. I will continue to work on it but if I continue having difficulties, I will shelve the registration task and look to work on it another time.

I am happy however happy that I managed to get some of it working. This means I'm on the right path. With a little more work, I can get it complete

Intended Changes

I might consider using different versions of authentication. Auth0 and Auth2 are the ones I'm currently working with. I keep having to change between them because of different methods I'm approaching. I did however look to make the login done through username instead of email. This was much easier to implement in Django and removed the hassle of dealing with email clients for confirmation of signup.

Supervisor Meetings

Date of Meeting: 15/11/16

Items discussed:

- Functional requirements
- Statistical formulas and machine learning

I met with my supervisor to review my requirements specifications. We discussed ways of improving it, adding more to it and removing certain things. I was informed of what is expected from an analytics student in terms of development. My supervisor advised me regarding machine learning and what I can do regarding different approaches.

I showcased a prototype of what I have achieved so far and was given feedback regarding my work. I was given positive response in respect to the user interface.

When discussing further steps that I should be taken, my supervisor advised me to hold off until next semester as more will be covered on descriptive statistics and machine learning.

Reflective Journal - December

Student Name: Mina Gaid

Programme BSc in Computing (Analytic Stream)

Month: December

My Achievements

This month, I had my midterm presentation in which I WAS TASKED WITH showcasing a prototype of my application. Before presenting, I took the liberty of cleaning up the application a little by moving certain things around, mainly in the front-end. I did this to better showcase what I have made in the past few months and what I intend to do in the remaining time I have.

When it came to the presentation, I talked about what my project is about, how it can be used in the real world, monetizing of the platform as a business and development. I presented a live demo of the web app and showed it off to the judges who provided me with feedback. The feedback was mostly positive. I was told to focus more on machine learning and to apply the API as soon as possible. I was given some advice on the wording that should be used when talking about the application which I have considered. Overall, I was proud of how the presentation went despite the tough questions that were asked.

Before the month ended, I took some time to test out the API on a new application. I was playing around with the functions and taking note of what I should apply to my web app. I now believe I have a clear and better vision of what will be included in the application and changes that are necessary.

My Reflection

The web API is more difficult to apply than I thought and work will need to go into it as soon as possible as I simply don't have the time. I have made some guide lines on what I should have for machine learning and I have looked up some tutorials on the subject online via YouTube.

The login system is still on hold and will continue until I get everything else out of the way. I won't have time on January as I will be busy with exams but I will use whatever time I can to work on the application and apply what I intend to do.

Intended Changes

The feedback I was given was mostly to do with my presentation of the data. I was told that it would be better to apply the data with less clutter as possible. I was also tolled to produce more machine learning in in a layout of that of a table. These were already in mind but the feedback better provided me with a clear vision.

Supervisor Meetings

Date of Meeting: 10/12/16

Supervisor meeting did not take place as supervisor was busy with exams.

However, the supervisor was present in the midterm presentation and provided me with feedback on the project and what I should do next.

The feedback he provided was that I consider applying the API as soon as possible and add more machine learning. He also told me to change the layout of the application by applying small changes.

There was not much to say after that other than to use the little time I have left to get it all done and working.

Reflective Journal - January

Student Name: Mina Gaid

Programme BSc in Computing (Analytic Stream)

Month: January

My Achievements

This month, I managed to work on the different algorithms in which my application revolves around. I managed to research a few videos on YouTube that helped me code the application with regards to Python packages that work off financial data. Using these tools, I managed to develop an algorithm in which the application gives a 30-day forecast (prediction) of the stock price based on the data I was receiving from the Quandl API. The data was retrieved and parsed before being displayed on my dashboard in the form of figures.

This data is extremely useful for investors and a key part of what the application is all about. It was of course important that I take care in getting the right figures. Based on my calculations, the figures are mostly accurate. However, it's important to keep in mind that I cannot predict the stock price exactly but can come close.

Other algorithms that were worked on included linear regression, a statistical formal in which we learned in Business analyses on our first semester. Likewise, the data is parsed using the algorithm and then displayed on the dashboard as figures.

My Reflection

I feel that it's impotent to output the data using charts as opposed to just plain figures on the dashboard. This was my original intended goal but I might not be able to do it based on the time I have left. I have looked at outputting some charts using Matplotlib, a Python plotting package. However, it's not as dynamic as Google charts which I consider to be better fitted. Regardless, the same results are outputted and the application development is on the right track.

The remaining "to do's" on my list is as follows -

- User Registration and Authentication.
- Enable Django Caching.
- Retrieve and output basic stock prices on the main Home dashboard.
- Deploy the application.
- Showcase poster

Intended Changes

As of now, I still haven't completed User Registration and Authentication. As such, the dashboard is accessible to everyone. When I eventually enable Authentication, users will need to login before accessing the dashboard.

Given more time, I might be able to use the Google Charts API to produce better looking dynamic charts for the dashboard. I won't replace Matplotlib but will include it as a secondary option if possible. For example, Matplotlib would output charts that can be saved as JPEGs while Google Charts would be used as dynamic charts for the user dashboard.

Supervisor Meetings

Date of Meeting: 31/01/17

A Supervisor meeting was scheduled in which I was given feedback regarding what I have achieved so far. The feedback I was given was positive regarding the 30-day forecast algorithm and linear regression. I was proud to be told that I'm on the right track but was also warned of the little time I have left to finish of the application.

I was also tolled to look more into producing charts and coming up with more information that can be utilised by investors.

It was also recommended that I consider referencing the YouTube videos that guided me through development as to avoid plagiarism if any.

Reflective Journal - February

Student Name: Mina Gaid

Programme BSc in Computing (Analytic Stream)

Month: February

My Achievements

I finally managed to enable user authentication on my Django application. I mentioned in my previous entries, I was having a lot of difficulty enabling registration and login as I'm new to the Django framework and there isn't a lot of information out there for beginners regarding the latest version of the framework. To add to that, there's hardly anything on Python 3.

I spent a weekend considering authentication and it paid off. I opted on using Allauth, a Django package that manages both local and social authentication. I began by reading through the documentation step by step and eventually, got it working with trial and error. After which, I designed all the templates, created the redirect link to the main user dashboard and restructured my code. I then enabled Facebook and Google social authentication to give users the option to quickly signup / sign in.

Other achievements this month included creating more charts to better visualise my data. I'm still having difficulties figuring out what more to add but I'm getting there. I've also considered cleaning up some of the front-end. Mainly the text and CSS. Finally, I fixed a few small bugs and changed some of the URLs to better integrate authentication.

My Reflection

I still have a lot to do regarding visualising my data and adding in more analysis. I'm still working on it but I'm confident that what I currently have will be enough as is. Visualising the data through dynamic charts is important and this is my focus. However, after getting around the authentication obstacle, I have hit a new one regarding rendering the charts I created within the actual HTML to be displayed to the user. I am familiar with MVC and rendering my code but I'm not quite there yet in terms of its use in Django.

The remaining "to do's" on my list is as follows -

- Enable Django Caching.
- Render charts on the main Home dashboard.
- Deploy the application on the web.
- Showcase poster

Intended Changes

I intend to add a lot more charts and render them to the main user dashboard where they will be displayed. Once this is done, my project is pretty much complete in terms of development. I will then proceed to host it online through a hosting service that support the Django framework. I will consider the process as deployment of apps built with frameworks is much different and involves a lot more work than basic applications.

Once the application is hosted, I will work to finish of my showcase poster and write a manual on the use of the application. The documentation will also feature some results from other services regarding forecasts made by my app. This document will be made to verify my results and explain them.

Supervisor Meetings

Date of Meeting: 08/02/17

A Supervisor meeting was scheduled in which I was given feedback regarding what I have achieved so far. The feedback I received was based on the results I obtained from my code in respect to forecasting. The mathematical approach taken was key to optioning the right results. My supervisor gave me positive feedback but advised me to have a document with results from alternative sources to verify my results are correct. It was also recommended that I consider visualizing more of my data. Lastly, I was told that it is important to have a systematic manual on how to run my code and to have the actual code hosted online.

The feedback received was noted and I'm in the process of working on it.

Reflective Journal - March

Student Name: Mina Gaid

Programme BSc in Computing (Analytic Stream)

Month: March

My Achievements

In the month of March, I began my reporting of the results my application outputs. I evaluate the output I receive from the models I created and compare them to those of other financial services online. I set out to reverse the output in SPSS to confirm I have the right results. I intend to include all the reporting within the final report that I am to hand up in May.

I began working on my dynamic charts. I'm still having difficulties rendering them to the front-end of my application. So far, I can only print out the charts on a separate window in the back-end. As mentioned before, I'm a beginner when it comes to using Django as a back-end framework so I'm finding it hard to display the charts on the user dashboard. I tend to do more research and look up a few videos that will help me out. Hopefully, I can have it done before the project deadline. This is the finale part of my code. After this, I will be done for good and would be ready to host it online on Heroku which happened to support deployment of Django 10 and Python 3 based applications. If Heroku doesn't work out, I'll host it on a private server.

Regarding my poster for the showcase, I designed it on Adobe illustrator and formatted to a PDF for printing as an A1 poster. All that's left is to finish off the report and try and get the charts working.

My Reflection

I still need to output my results to the user dashboard. So far, I only got them to output on the terminal. I need them to be viewable on the user dashboard as intended. Once, done I can host it online and my site will be live and ready to submit.

My biggest worry aside from the issues with the charts is that of reporting my results. I need to further discuss with my supervisor what I should include within the report and what I can do to better improve. I know my results and models are accurate but I'm having issues reporting it on paper. I should have this done though after exams are over in April so it's not that much of an issue.

I hope to be able to add appropriate screenshots of my charts and add them to the finale report. Visualization of data is extremely important and is a centre piece of my project. Screenshots must be included to better showcase and explain my findings. I will need to spend time and focus on the report after the code is done.

Intended Changes

I don't necessarily intend on making any changes to what I have done this month but I hope to finish the remaining tasks I have yet to finish. I still have the charts, the report, and the deployment of the application. I am hopeful I can get them all done within the final few weeks. There's not much left to do. More time on the project after the exams will help me a lot.

Supervisor Meetings

Date of Meeting: 29/03/2017

Items discussed:

- Reporting of results
- Feedback on midterm presentation
- Recommendation for final presentation

I met with my supervisor to receive feedback on my progress. I asked for help interpreting my results and recommendation on how to best report them. I asked to look at previous examples and asked for resources that I can use to help me out. I was recommended on using certain formats when reporting my results and examples to look at online.

I also received feedback on the midterm presentation which I hope to be able to use to better perform on my finale presentation. The feedback will defiantly help me avoid making mistakes that I made back in December.

I was very happy talking to my supervisor who gave me some great feedback and recommendation for the final presentation. Taking what was said in the meeting to note, I plan to get started preparing for the presentation. I should have no trouble so long as I finish off the report and get the charts working on the user dashboard. After which I am done. And will have nothing left but the showcase.

Bibliography

The following is a list of tutorials I used through development of the application. Most are centred around using the Quandl API. Code within the application was written with these tutorials.

Pythonprogramming.net. (2017). Python Programming Tutorials. [online] Available at: <https://pythonprogramming.net/machine-learning-python-sklearn-intro/> [Accessed 5th January 2017].

Connecting with Quandl for Annual earnings data - Programming and Fundamental Investing Part 11, video, sentdex, viewed 5th of January 2017, <https://www.youtube.com/watch?v=SHZjQoiJZKY>

Organizing Earnings Data - Programming and Fundamental Investing Part 12, video, sentdex, viewed 5th of January 2017, <https://www.youtube.com/watch?v=W16i4vYjN2o>

Graphing Finance Data - Programming and Fundamental Investing Part 13, video, sentdex, viewed 6th of January 2017, <https://www.youtube.com/watch?v=kq0TkGRZXHw>

Finishing the Graphing - Programming and Fundamental Investing Part 14, video, sentdex, viewed 6th of January 2017, <https://www.youtube.com/watch?v=JZdyG6qLSMA>

Adding our Historical Earnings to Chart Data Python - Programming for Fundamental Investing Part 17, video, sentdex, viewed 7th of January 2017, <https://www.youtube.com/watch?v=hTfPelEmLdI&>

Completing our fundamental investing stock screener - Programming for Fundamental Investing Part 18, video, sentdex, viewed 12th of January 2017, https://www.youtube.com/watch?v=b0nQ_cPN3Hk&

NCI Invention Disclosure Form

NCI INVENTION DISCLOSURE FORM

1. Title of Invention

Sequence Control Panel

2. Inventors

Name	School/Research Institute	Affiliation with Institute (i.e. department, student, staff, visitor)	Address, contact phone no., e-mail	% Contribution to the Invention
Mina Gaid	NCI	Student	mina.s.gaid@gmail.com	Inventor

3. Contribution to the Invention

Each contributor/potential inventor should write a paragraph relating to his/her contribution and include a signature and date at the end of the paragraph.

I invented and developed the application.

4. Description of Invention

(Please highlight the novelty/patentable aspect. Attach extra sheets if necessary including diagrams where appropriate). What is novel, the ‘inventive step’? For more information on patents, please look at <http://www.patentsoffice.ie/en/patents.aspx>

The application is built to help forecast stock prices using mathematical formulas. The novelty of it is based around how data is presented and the accuracy of the forecast.

5. Why is this invention more advantageous than present technology?

What is its novel or unusual features? What problems does it solve? What are the problems associated with these technologies, products or processes? Explain how this invention overcomes these problems (*i.e.* what are its advantages).

The application does not have many advantageous over that of “present technology” however, it does solve the problem of people being able to verify their investment by looking at relevant forecast. The freemium model for an application like this is rare and having the data displayed so that anyone can get an idea of what’s happening will be helpful to none industry users.

6. What is the current stage of development / testing of the invention?

The application has been developed and tested. It is now ready for launch.

7. List the names of companies which you think would be interested in using, developing or marketing this invention

AIB, Bloomberg,
Bank of Ireland,
KBC,
JPMorgan Chase,
Citigroup,
HSBC

8. Funding Partner(s)

Government Agency & Department	N/A
% Support	N/A
Contract/Grant No.	N/A
Contact Name	N/A
Phone No.	N/A
Address	N/A

Industry or other Sponsor	N/A
% Support	N/A
Contract/Grant No.	N/A
Contact Name	N/A
Phone No.	N/A
Address	N/A

9. Where was the research carried out?

The researched carried out was on the technology stack used for development and the approach taken for the forecast.

10. What is the potential commercial application of this invention?

The application can be monetized to include different sets of data for paying users.

11. Was there transfer of any materials/information to or from other institutions regarding this invention?

If so please give details and provide signed agreements where relevant.

There was no transfer of any materials/information to or from other institutions regarding this invention. However, algorithms shared online were used for forecasting.

12. Have any third parties any rights to this invention?

If yes, give names and addresses and a brief explanation of involvement.

Third parties do not have any rights to this invention. Quandl terms of services does not give them any rights to my application.

13. Are there any existing or planned disclosures regarding this invention?

Please give details.

There are no planned disclosures regarding this invention.

14. Has any patent application been made? Yes/No

If yes, give date: No Application No.: _____

Name of patent agent: _____

Please supply copy of specification.

15. Is a model or prototype available? Has the invention been demonstrated practically?

A prototype is available and will be demonstrated on the 16th of May

I/we acknowledge that I/we have read, understood and agree with this form and the Institute's *Intellectual Property and Procedures* and that all the information provided in this disclosure is complete and correct.

I/we shall take all reasonable precautions to protect the integrity and confidentiality of the IP in question.

Inventor: Mina Gaid Mina Gaid 10/05/17
Signature Date

Inventor: _____
Signature Date

Inventor: _____
Signature Date