Smart Predict

Trading Platform

CA400 Functional Specification

Ben Kelly - 15337716

Table of Contents

1 Introduction

1.1 Overview

SmartPredict is a Web application trading platform which aims to help users make intelligent decisions on buying and selling of currencies (foreign exchange & cryptocurrency markets). This is achieved through machine learning and chart analysis. The application will provide the user with predictions for the market based on knowledge it has gained from past and present data. Recommending whether the user should buy, sell or hold their specific currency relative to its current value. A user will be able to trade, view the current market sentiment and review the prediction of the machine learning algorithm all within the application.

Foreign currency exchange (Forex) remains the largest financial market in the world, the amount traded on Forex exceeds all the world's equity markets combined. Traders participating in this market make wagers based on the values of currencies, they profit when their predictions are accurate this is where SmartPredict's prediction element thrives.

Cryptocurrency exchange is something extremely new to the trading environment. It has been the fastest growing method of capital trading in recent history and has captured the imagination of millions of people. Attracting new people to the industry who are perhaps less financially experienced than the typical trader and more software/tech influenced. This is where SmartPredict will appeal to both experienced and novice traders.

1.2 Business Context

SmartPredict's target audience is anyone who has an interest in trading. Whether they are complete novices or experienced traders, they should be able to use SmartPredict from the get-go. My goal is for SmartPredict is to be as intuitive as possible while having complex functionality behind the interface. In essence, this app aims to level the playing field for all traders, so that even beginners can make intelligent trades using SmartPredict's prediction algorithm to make informed decisions. Traders will also be able to learn visually through live graphs about the various currency data.

Generally speaking, the Forex market is highly volatile and auction-based. It is open to investment professionals as well as amateur traders, particularly through online and app-based brokerages specialising in this area. One other important aspect of the Forex market which factors into the SmartPredict experience is the fact that it is open continuously, due to time differences around the world. The combination of the worlds largest financial market, as well as the world's fastest growing one with the aid of a prediction algorithm all in the same place, will be what sets SmartPredict apart from the competition.

As of last year, one in every 7 adults in Ireland made money from trading online. This approximates to nearly 1 million people in Ireland (North and South). This number drastically

increases when we compare the number of active traders in other regions particularly Asia (3.2 million) and North America (1.5 million). Amounting to a total of 13.9 million traders worldwide. SmartPredict can, therefore, be used by an extremely large group of people.

1.3 Glossary

Forex - Foreign currency exchange, a market in which the foreign currencies of the world are traded.

LSTM - Long Short-Term Memory, an extremely powerful algorithm that can classify, cluster and make predictions about data, particularly time series and text.

MVC - The Model-View-Controller (MVC) is an architectural pattern that separates an application into three main logical components: the model, the view, and the controller. Each of these components is built to handle specific development aspects of an application.

Candlestick chart - a style of financial chart used to describe price movements of a security, derivative, or currency. Each **candlestick** typically shows one day, thus a one-month chart may show the 20 trading days as 20 **candlesticks**.

Open - Refers to the time at which a particular market opens.

Close - Refers to the time at which a particular market closes.

Price pullback - also referred to as a retracement or consolidation, is the falling back of a currency's price from its peak.

2. General Description

2.1 Product / System Functions

User Registration

If a user wishes to access SmartPredict they need only create their account which will be accessible on the web. Upon clicking the register button a user will be asked to enter their email (which will be used as a username), their age and create a password. Once a short email verification process is completed a user can begin trading. When the user completes verification their details will be stored in SmartPredict's database and they can log in from anywhere and trade as normal.

Note: A user must be older than 18 years of age to use this application.

User Login

A user will login in with their unique email and password which they have created earlier. When a user logs in for the first time they will be asked a series of questions, what currency the would like to use, what markets they are interested in and what their trading experience is. User's who are relatively new to trading will be provided with the option of informational articles and videos to broaden their knowledge of the topics. As the app will contain private information to the user such as their balance and past trades it is essential that this

information is kept confidential. To combat fraud, the app will have a timeout feature, if the user is inactive for an arbitrary amount of time (set by the user with a default of 5 minutes) the system will log the user out.

Trading

Once a user has successfully logged in they will be able to trade on Cryptocurrency and forex markets in real time. A user will have all of the expected functionality they would have on any other platform like buying, selling and holding etc. Making decisions on which market to trade in will be aided by graphical representations in the data visualisation element of the app as well as predictions generated by the application mentioned below with the current sentiment of the market also calculated.

Prediction Algorithm

The algorithm will use machine learning (particularly chart analysis and LSTM), to make accurate predictions. It will also use sentiment analysis which will be mentioned below. The algorithm will be provided with datasets of past data for each currency and market to give it specific information about potential rises or drops in the future.

The app will look to predict:

- If the closing price of a certain currency will be higher/lower than the previous day
- If the closing price will be higher/lower than the opening price.

Sentiment Analysis

Accurately predicting currency prices solely on past data is a challenge, this is why sentiment analysis will be utilised, it will add another dimension to the overall prediction. This will be implemented into the actual Django app (prediction algorithm), it will use Twitter's API and other popular sites to gain insight into each currency and their respective market in general to aid the prediction algorithm.

Chart Analysis and Data Visualisation

On the user's personalised home page they will be met with graphs of the currencies they are interested in along with analysis to go alongside these graphs. Questions the user answered upon initial login will be used to aid these graph suggestions to make them as useful as possible for each individual user. The output from the prediction algorithm will be seen on screen specific to the market being shown as well as the current sentiment of that market generated from the sentiment analysis. The algorithm will generate its own graph of expected prices to give the user a clearer picture of what is expected to happen.

Deleting an Account

SmartPredict recognises that a user's needs may change in the future, therefore every user has the option to delete their account. This can be found it the user's settings page.

2.2 User Characteristics and Objectives

As stated previously, SmartPredict is primarily aimed at people who have some experience in trading but can also be used by anyone who wishes to trade for the first time. As anyone over the age of 18 can trade this gives us an extremely large target audience. It is safe to assume that this application will be used by anybody with an internet connection. One of my highest priorities for the app is to make it intuitive enough so that anyone can use it. I intend to create a user interface which can be navigated quickly and intuitively, there should be a minimal learning curve to using this application and most tasks should be almost instinctual.

A typical user of SmartPredict would be someone with a full-time job in trading or simply someone who has an interest in trading in their free time who wishes to improve their success by using emerging technologies such as those utilized by SmartPredict. If the app is to be a success it will need to perform the menial tasks like analyzing graphs and looking at market sentiment at least as good as a talented trader can, if not better. This will help all users make smarter investments, additionally, save them time and effort.
This will be achieved in the following ways:

- Make the front-end clear and efficient to stop users having to scroll through multiple pages for information.
- Make all processes fast and have minimal loading time
- Make the prediction algorithm accurate to the point of much more than 50% (likelihood of an investment being good or bad based on no knowledge of market)
- Make the sentiment analysis a true reflection of the general opinion of the public
- Displaying the graphical data in a clear but useful manner (not too simple but also not too difficult to understand)
- Perform tasks such as sentiment analysis, market predictions, chart analysis and execute subsequent trades more accurately, faster and easier than a user can manually perform them.

This poses some challenges that will be discussed in the constraints section below.

2.3 Operational Scenarios

SmartPredict uses MVC (Model-View-Controller) architecture, when ‘System’ is mentioned below it refers to these three components, not all components are used at all times. Whichever components are used will be indicated within the brackets beside ‘System’.

User Login & Registration

The user accesses SmartPredict via a web browser. Then they can create an account with their email & password. Invalid or incomplete login details will be notified to the user using the “failed to register” View. Once a user successfully registers they are taken to the login page. Upon successful login, they are met with the home screen of the application. Incorrect or incomplete login details will also be notified to the user using each specific View.

USE CASE 1	Create a SmartPredict account	
Goal in Context	User creates an account on the WebApp	
Scope & Level	System(Model, View, Controller), database	
Preconditions	User has a valid email address and has not already registered before.	
Success End Condition	User's account is created	
Failed End Condition	User's account is not created	
DESCRIPTION	**Step**	**Action**
	1	User opens the WebApp in a browser
	2	User is presented with the login/sign up page
	3	User presses sign up
Extended	4	User enters their email address and creates a password. Controller checks the Model to see if the email is unique
Extended	5	User is prompted to confirm password
	6	Client validates email and sends a confirmation request to the email address
Extended	7	User opens an email and presses the link to confirm
	8	Account is created, storing the User's data within the database. User is then taken to the login page.
EXTENSIONS	**Step**	**Branching Action**
	4a	An account with this email address already exists. Controller returns unsuccessful register to the View
	4b	User is asked to re-enter a valid email address or sign in
	4c	User is prompted with the sign-up screen again
	5a	Passwords do not match
	5b	User is told to re-check that passwords are correct and try again
	7a	User does not receive email, the account is not created
	7b	User does not click the link in the email, the account is not created

USE CASE 2	User logs in to the Application	
Goal in Context	User logs in to SmartPredict	
Scope & Level	System(Model, View, Controller), database.	
Preconditions	User has accessed the WebApp online and has an existing account	
Success End Condition	User progresses to log into their account	
Failed End Condition	User fails to log in	
DESCRIPTION	**Step**	**Action**
	1	User opens the WebApp
Extended	2	User enters their email address
Extended	3	User types their password
	4	User's details are checked within the database
	5	User is now logged in
EXTENSIONS	**Step**	**Branching Action**
	2a	Email does not exist in the database
	2b	User is not logged in and is prompted to enter a valid email address or sign up
	3a	Password is incorrect
	3b	User is told that this password is incorrect for this account
	3c	User is encouraged to enter a valid email and password combination

USE CASE 3	User adds a trading API key to their account
Goal in Context	User wishes to add an API key to their already existing trading account so that they can trade on SmartPredict and not have to leave the app
Scope & Level	System(Model, View, Controller), database, server.
Preconditions	User is logged in, has created an account on either Cryptocurrency or Forex exchange and has access to their valid API within that exchange
Success End Condition	User adds an API key to their account and can start trading

Failed End Condition	User fails to add an API key to their account	
DESCRIPTION	**Step**	**Action**
	1	User chooses between Cryptocurrency and Forex market exchanges
	2	User selects ‘Add API Key’ on their specific currency Dashboard
	3	User is met with their existing API Keys and a text box in which to fill with a new one
	4	User enters their new API key and submits
	5	Controller takes API key from View and passes it to the Model to be stored in the Metadata
Extended	6	API key is compared to the stored ones in that market exchange
	7	Controller notes successful pair and returns message to the View
	8	View returns ‘API key added Successfully’ to the User, API key is added to Model
EXTENSIONS	**Step**	**Branching Action**
	6a	Entered API key cannot be found or paired
	6b	Controller cannot find online match, returns unsuccessful message to View
	6c	User is met with a pop up specifying that the API key entry was unsuccessful, informing them to ensure API key is valid
	6d	New API key is not added, Model is unchanged

USE CASE 4	User wants to make a trade	
Goal in Context	User wants to make a trade on the Forex or cryptocurrency market	
Scope & Level	System, product data	
Preconditions	User has an account, is logged in and has added a valid API key for the market they wish to trade in and has added it to their SmartPredict account.	
Success End Condition	User makes a trade	
Failed End Condition	User fails to make a trade	
DESCRIPTION	**Step**	**Action**

	1	User chooses which exchange they wish to trade on
	2	User presses ‘Start Trading’
	3	User selects which currency they wish to trade
	4	The user is taken to their trading dashboard (Controller changes View) where they are met with the Data analysis page, they can graphical data of what price the currency is at and past price points for that currency
Extended	5	The user enters the amount they wish to invest and confirms
Extended	6	Controller takes this number and passes to the Model where it is processed using the relevant market exchange.
	7	The user is met with information regarding the trade they just made, the success message and their new current balance is displayed
EXTENSIONS	**Step**	**Branching Action**
	5a	User has insufficient funds, in order to trade that amount the user must have a balance equal to or more than the amount they wish to trade.
	5b	Controller passes insufficient funds message to view where it is displayed to the user, User’s balance is unchanged
	6a	The trade fails, controller takes the error code from the model calculates what happened and returns the reason back to the View along with the Trade Unsuccessful message

USE CASE 5	User wishes to gain information regarding current Market state	
Goal in Context	User views the market sentiment and prediction for that currency and gains insight on what they should trade on and how	
Scope & Level	System(Controller, View)	
Preconditions	User has created an account and is logged in	
Success End Condition	User’s dashboard is successfully loaded and prediction features and visualization methods are working	
Failed End Condition	User fails to gain any insight into the current market state	
DESCRIPTION	**Step**	**Action**
	1	User opens their dashboard
	2	User selects either Cryptocurrency exchange or Forex exchange

	3	User selects a specific currency
	4	User is met with graphical data and chart analysis of past prices for that currency
	5	User is met with current sentiment of that currency as well as what the Machine Learning algorithm believes will happen in the future and how they should invest
	6	User gains valuable insight into that currency exchange and can make trades accordingly

Users Dashboard
Once logged in a user sees their Dashboard, where they can select their market for trades, add API keys, change their settings, gain insight into the overall market sentiment.

A user wishes to Delete Account
When a user decides to delete their account they can do so by entering the account settings screen. This will be accessible from the Dashboard provided the user has an existing account and is logged in.

2.4 Constraints

Associated Risk Constraints
Like any Trading Platform, a user can be susceptible to losing money. This is especially due to the volatile nature of both Cryptocurrencies and Forex. SmartPredict will take no responsibility for the loss of capital due to unsuccessful investments, this will be outlined when a user creates an account, users will have to agree to the Terms & Conditions of the application before they begin trading.

Licensing Constraints
The main constraint I've experienced in the early development of this project has been the inability to create a standalone trading platform in the true sense of the word. Initially, I wanted SmartPredict to be a centralised place where traders could have only one app which handled all of their trading needs. This was naive on my part because there is a trading license needed in order to achieve this, which, for the scope of this project wouldn't be feasible. To combat this I've suggested that users create an account on one of the supported trading platforms for either Cryptocurrency or Forex markets (which I'll define and expand during the User Manual) and link it to SmartPredict via API keys. This will handle the trading back-end of the app while a user can still enjoy all of the other 'Smart' functionalities of SmartPredict.

Prediction Algorithm Constraints
Due to the volatile nature of both markets mentioned above, it is extremely difficult to predict future prices of these currencies. To combat this I will implement Sentiment analysis (which

was not originally proposed) to try to improve the accuracy of the prediction element. The aim for the algorithm is to be more accurate than a user could be with chart analysis alone. If the algorithm can consistently produce predictions of over 50% accuracy it will be deemed a success.
In order to predict future prices of currencies, I am relying on the datasets that I plan on using to be kept up to date. Failing this I will have to implement some sort of web scraper to get the daily price of each specific currency. As there are strict time constraints for this project this isn't ideal but may be necessary to not have to completely rely on datasets maintained by other people.

Browser & Hardware Constraints
As this is a web application, it will be accessible for anyone with an internet connection on any browser. However, due to the high space requirements and interactivity of the graphical data, we will be showing on screen it will be difficult to optimise for all browsers. As I will be using Google Chrome to develop and test I will optimise for this. The application will be accessible on other browsers it may run slower, however, this will be outlined in the User Manual and on the app itself. Users will also ideally be trading on a PC which will be powerful enough to run the browser efficiently. I don't see this as being a huge issue, in this day and age the majority of people have access to a PC with the requisite specification. The code for the application will mostly be written in Python 3 within a Django environment on Mac OS.

Time Constraints
As this is a 4th-year project, this application must be completed by Friday 19th of May 2019. I aim to complete the application and all supporting documentation by that date.

Security Constraints
We will be using an external database to store all user data such as emails, passwords and profiles. Therefore, we must respect a user's privacy and must take the security of
this data into consideration. All of the data entered into the MySQL Database will be encrypted and passwords will be hashed.

3. Functional Requirements

3.1 Trading
Description:
Once a user successfully logs in, he/she will be met with their specific Dashboard where they can choose which Market to trade on (Cryptocurrency or Forex) within each market they can specify which currency they are interested in (eg. Bitcoin, litecoin etc.). When a user selects a currency they wish to trade on they can continue to trade (providing they have a valid API key). The trading mentioned will refer to the buying and selling of specific currencies to order to gain capital. The list of currencies supported and APIs that will be used for connection purposes are still yet to be decided on. This will define how the user can

set up their trading account and enter their API keys. As it stands SmartPredict will be the only trading platform on the market that combines cryptocurrency and Forex markets.

Criticality:
Trading is one of the main functionalities of the application, without it a user wouldn't be able to turn a profit. It is vital that this element of the app not only works but works well. Trading should be fast and intuitive, with instant feedback of success and detailed analysis of the trade that was made. In contrast, if the trade is unsuccessful it is important to display to the user the reason(s) as to why the trade failed and suggest solutions.

Technical issues:
The issue with trading natively on SmartPredict is the licensing restrictions mentioned in the constraints above. The workaround through API keys to other trading platforms isn't a perfect solution but will enable the app to work as initially planned.

Dependencies with other requirements:
The trading aspect to the app relies on other trading platforms to make the actual trades. The system will communicate with the trading platform the user has already set up through an API key pair. Unfortunately, the user will need two API keys, one for the Forex market and the other for trading cryptocurrencies. The trading platform will thus be split in two. The trading element will not require anything from other functionalities of the app but will be aided on the user's end by the prediction algorithm and sentiment analysis who's output will be seen on each specific currency dashboard.

3.2 Prediction Algorithm

Description:
The prediction algorithm will forecast how a currency will be priced in the future. This will be calculated daily taking into account sentiment analysis of the specific market and currency using LSTM modelling. It will be shown in a visual format through the data visualisation window but will also be returned in a simple sentence and statistics to go alongside the graph. E.g "Bitcoin is currently being overbought this is expected to cause a price pullback, we don't recommend buying this stock currently". The algorithm itself will use machine learning in order to accurately determine if the price of a currency will rise, fall or remain level. The prediction algorithm will help users make decisions on what currency to buy/sell and why. It is not intended to be the only source of information for the user which is why sentiment analysis and chart analysis have been included. The user is expected to understand the market and make their mind up on what to trade with an influence from the before mentioned prediction factors.

Criticality:
It is essential to the application that the prediction algorithm can accurately predict the rise or fall of a currency. This is the main functionality of the app and will bring users to it. Without it, there is little stopping a user from using another trading platform. As a result of the system using AI in this way, it will, in theory, become smarter and more accurate as time goes on.

Technical issues:
As mentioned previously it is extremely difficult to accurately predict something as temperamental as cryptocurrency and Forex. This is why Sentiment analysis will also be used, the goal will be to consistently calculate whether a currency will rise or fall in value on a daily basis. It may happen that certain markets are more influenced by different factors more strongly than others. This may lead to having 2 different algorithms for the two markets or different weightings within each specific currency itself. This will be developed further later down the line and will be experimented with testing during the developmental phase.

Dependencies with other requirements:
Unfortunately, calculating whether a currency price will increase or decrease cannot be done much more often than daily. I am relying on datasets (Kaggle and others) to be updated. If this becomes an issue I can manually scrape websites in order to get more up-to-date price points to add to my dataset. The trading element will also have access to the most relevant prices so it might be useful to gain insight through this.

3.3 Chart Analysis & Data Visualisation

Description:
The user will be met with a graphical description of the price progression of a currency when they choose to trade. These graphs can also be seen from the user's dashboard once they have selected a market. The graphs will be in the form of candlestick charts, line charts and bar charts, all of which will be extremely useful for the user in their own analysis of the market. Along with the graph of past data the user will see the output of both the prediction algorithm and sentiment analysis, again giving the user as much information as possible before they make a trading decision. The aim for these charts is to display as much information as possible in a small window. It is much more valuable for a user to examine visual data such as chart compared to trying to understand lists of past price points and come to a conclusion if they should invest or not. Charts will also be interactive, users will have the ability to display greater or fewer data over longer or shorter periods of time and be able to the see the min or max price a currency has been at over time.

Criticality:
It is crucial that the graphical interface is easily readable and provides key information so that the user can gain a better understanding of the market. It will also visualise the data for the price prediction of the Machine learning algorithm, so if the graphs are not clear or unreadable it will devalue the prediction algorithm as well. All graphs need to be laid out in a consistent manner so that there is no difference (from a purely graphical perspective) between each currency and market. The user should be able to browse different currencies and see a consistent interface throughout the system. If the converse were to happen it would lead to confusion and the user may be deterred from the system altogether.

Technical issues:
Displaying multiple interactive graphs and charts is extremely taxing on the overall browser. It is essential that the system does to put too much of a load on the user's end while still

making the interface smooth and usable. This will be achieved through careful optimisation of the data visualisation element at the code level and efficient use of the back-end for server-side operations.

Dependencies with other requirements:
As this is one of the last steps of the system, it relies on the datasets collected for past price data and the prediction algorithm in order to push it's calculations to the front-end. Unfortunately, the datasets are not maintained by myself so if they were to stop being updated regularly I would have to manually scrape the data for each individual currency affected.

3.4 Sentiment Analysis

Description:
Sentiment analysis will be used to gauge an overall view of the market and what the overall feeling is towards a certain currency etc. The input of this element will be the Twitter API along with others (yet to be identified) to gain insight into the markets. The number crunching will be done in Python within the Django framework, this will allow me to utilise libraries like NumPy and Scikit-learn. The output will be a simple multiplier to the prediction algorithm, a floating point number from 0 - 1 will suffice, 0 being negative, 0.5 being neutral and 1 being positive. As well as being used within the prediction algorithm the result from the sentiment analyser will be shown on screen with the data visualisation and when a user is in their trading dashboard. E.g. "The public is really confident in the US Dollar/Euro currently, you should think about investing in it".

Criticality:
This element not only has an effect on itself but also acts as a guide to the prediction algorithm. It is essential that it is a true reflection of the general consensus and is not easily swayed by a small number of positive/negative connotations. To combat this, users will be weighted based on their following. E.g. Donald Trump has 55 million Twitter followers so what he thinks has more of an effect on the general market than someone who has less than a hundred followers. The algorithm will take this into account when calculating.

Technical issues:
As trading includes certain keywords and specific language a general sentiment analyser will not work optimally. Terms like long, haircut and bull have different meanings to their generic definition. To reduce the negative effect specific language might have on the analyser, the before mentioned terms will be redefined and inserted into the categoriser before calculating.

Dependencies with other requirements:
The sentiment analysis element will rely on the Twitter API and whichever ones are chosen as input during the developmental phase. These web tools are out of my control so APIs that are often maintained and well managed will be chosen as it is vital that there is a significant input to the algorithm to obtain accurate results.

3.5 Login/Registration

Description:

In order to use this application, the user must first register and then log in. A database is needed to store this information, a user will be given the option to register when he/she first opens the webpage. Once registered a user can then login on any web-accessible device.

Criticality:

As registration is critical to the use of this application, it is important for this function to work correctly and make sure that each user is given a unique ID to store their details and profiles. It is also vital to keep this data private, to aid this all user passwords will be hashed so that if the system is hacked their account won't be compromised.

Technical issues:

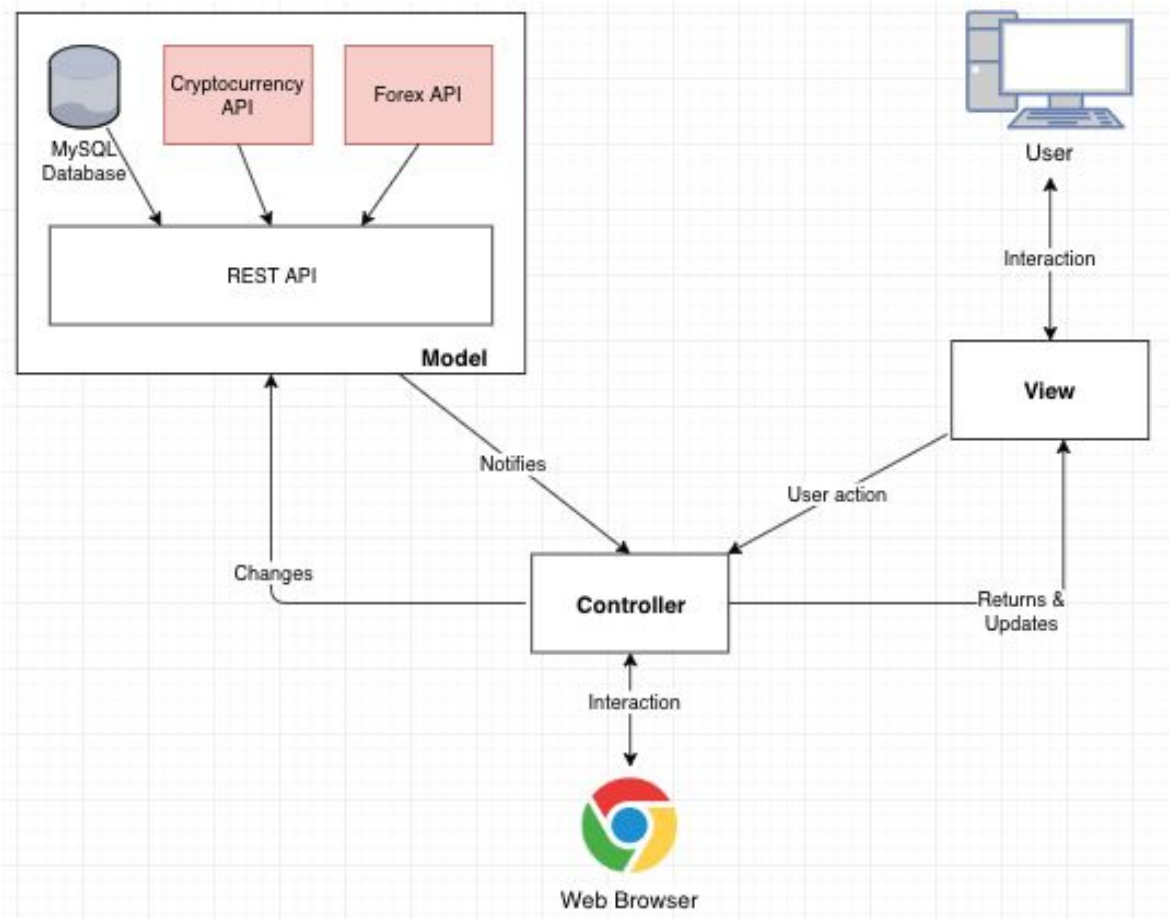
It is essential that all login details are securely added to the database, also it is important that they are added correctly so that there are no mismatches between multiple users. User details must only map to their own accounts.

Dependencies with other requirements:

Without a correct Login/Registration, a user will not be able to operate this application past the login page. All other functionality will not be available to the user. So while the user login is not dependent on any other feature of the application, all of the other features are reliant on it.

4. System Architecture

SmartPredict implements the MVC architecture within Django. The following system architecture diagram gives a better explanation of how each element interacts with one another than I could do explaining each permutation by text. 3rd party elements are outlined in red.

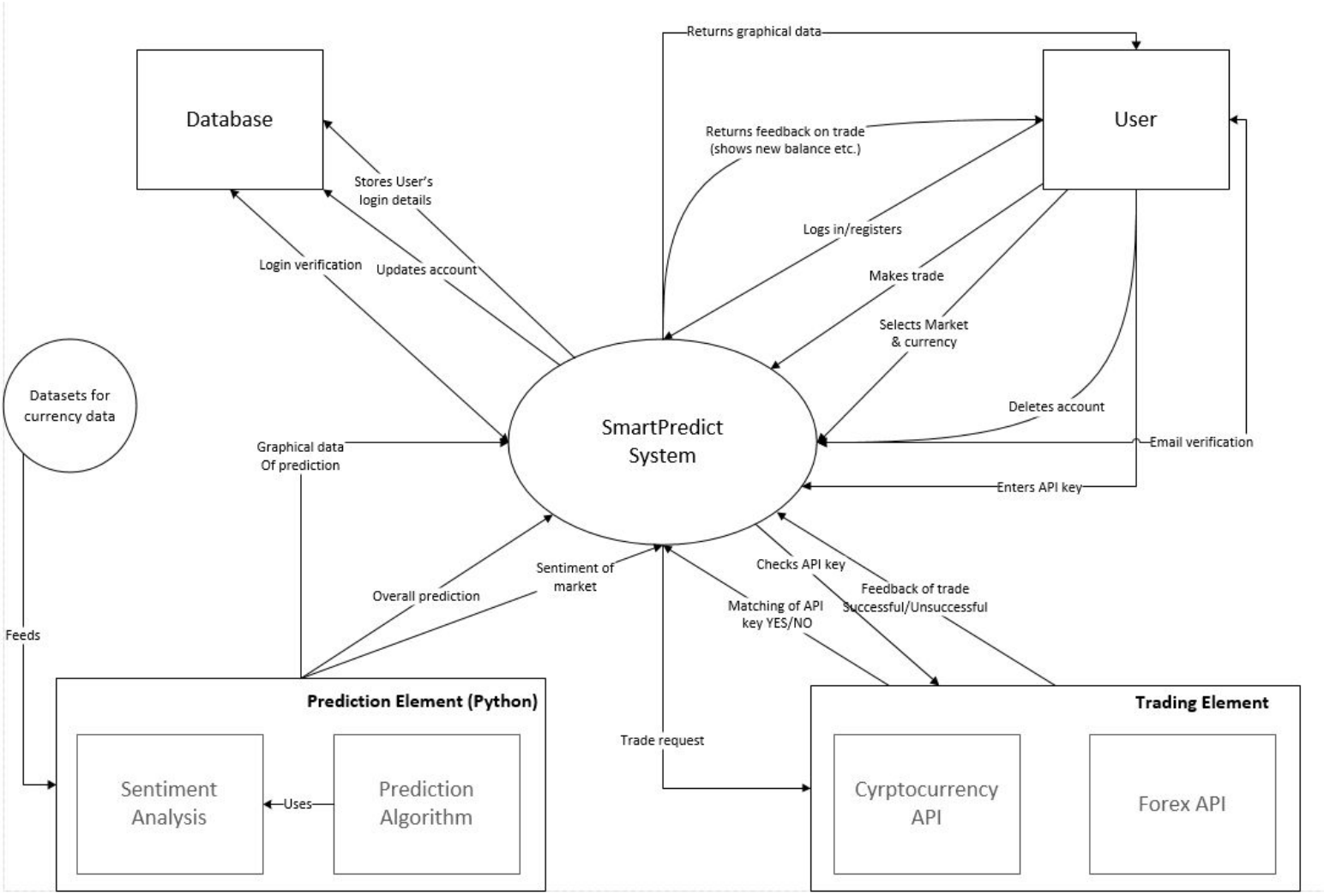


5. High-Level Design

5.1 Context Diagram

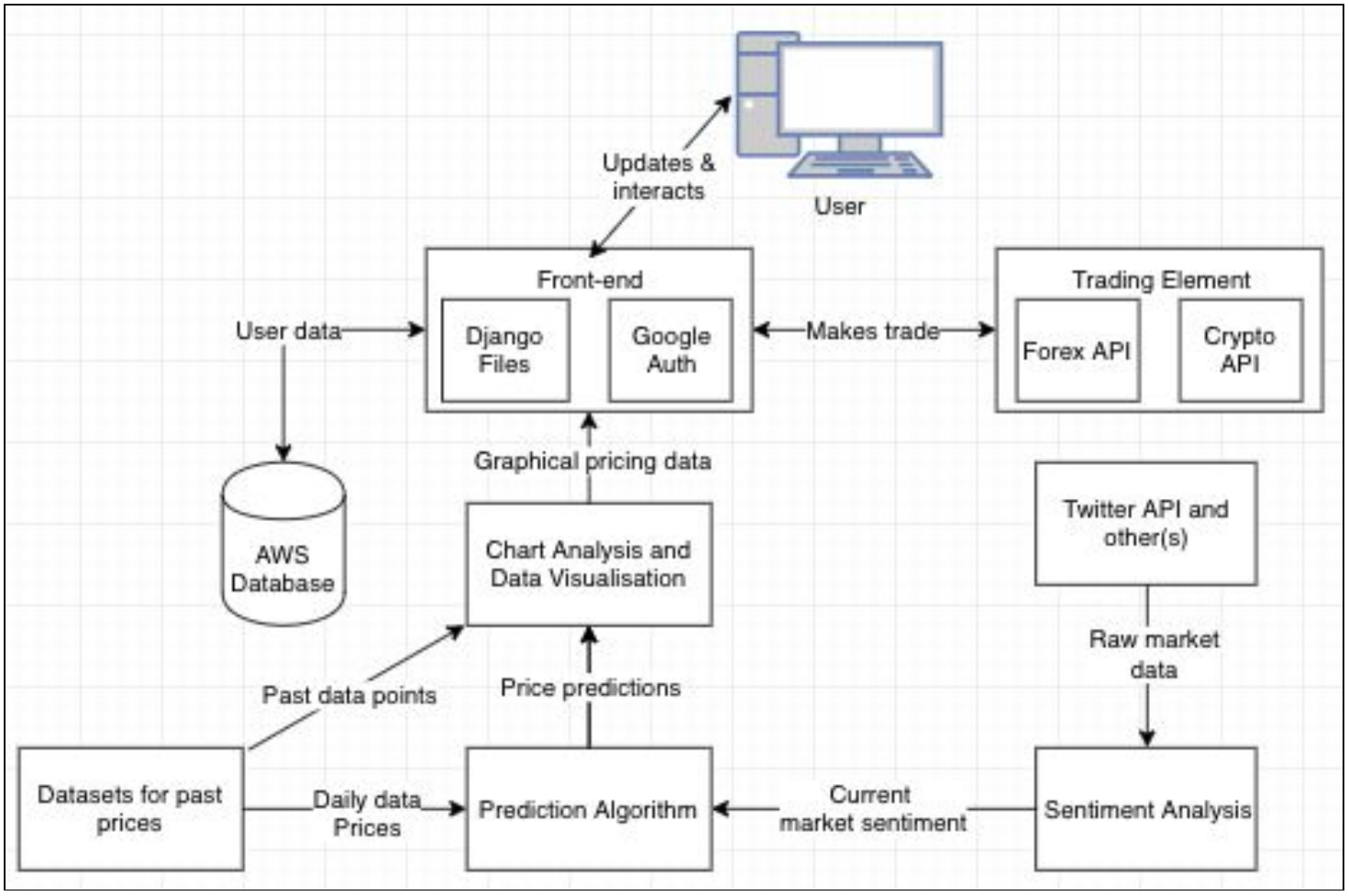
The context diagram gives us a great volume of detail in terms of the system itself and the entities it interacts with. For example, the system only interacts with the database when verifying a user's login or storing credentials within it. In contrast to this, the user can interact with the system in a multitude of ways at any instant as seen below.

For reference: the prediction element is within the application itself but I thought it was relevant to mention it within the Context Diagram as their data is collected from outside the app so they are listed as seen below.



5.2 Process Diagram

The process diagram below is a graphical representation of the movement of data through the system. It essentially shows how each process works at step by step intervals. For instance, a user cannot directly interact with the Database or the prediction algorithm but through the application, in various ways, the user can get different feedback from it through the front-end.

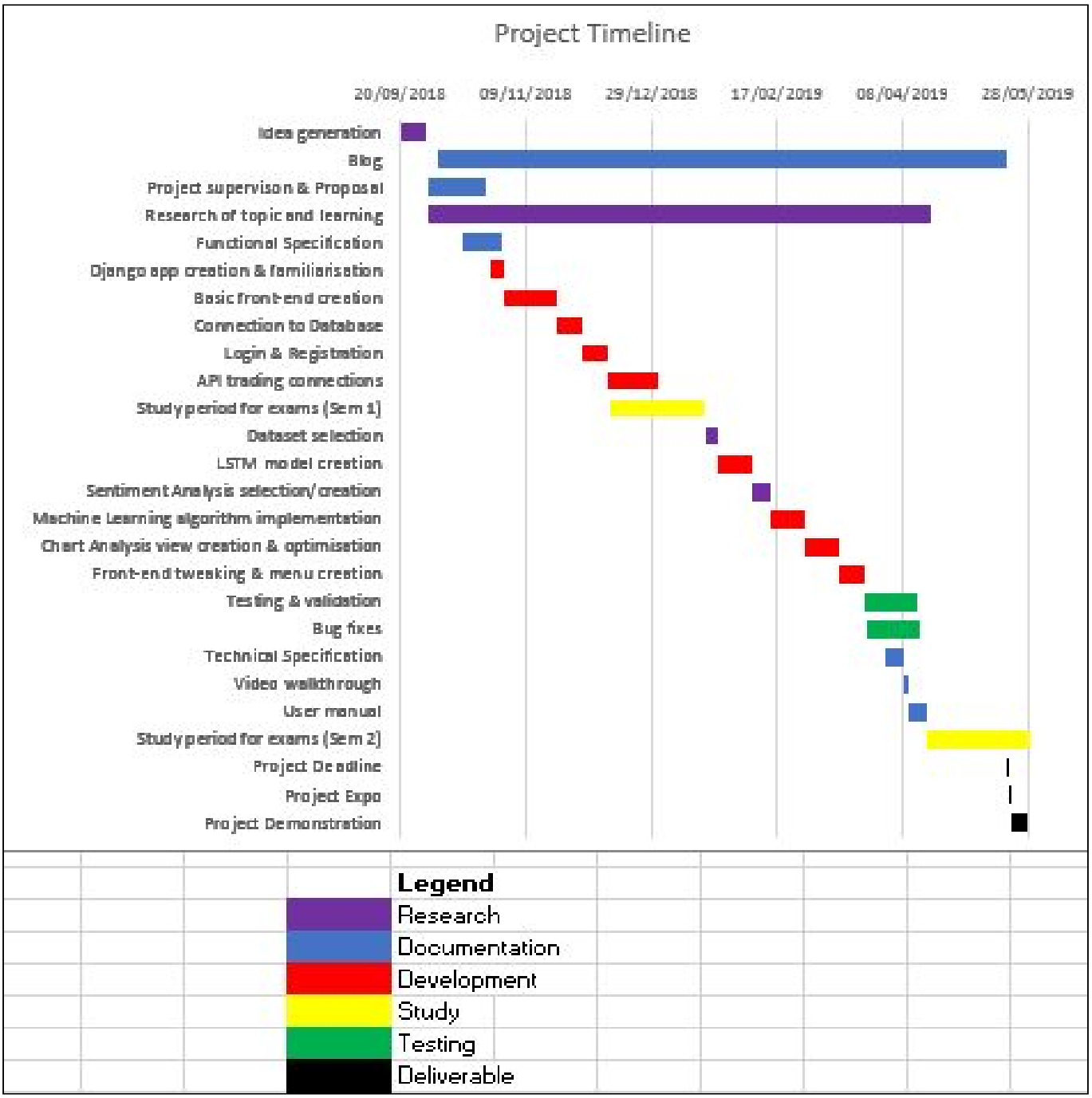

Updates &
interacts
User
Front-end
Django
Files
Google
Auth
Trading Element
Forex API
Crypto
API
User data
Makes trade
Graphical pricing data
AWS
Database
Chart Analysis and
Data Visualisation
Twitter API and
other(s)
Raw market
data
Past data points
Price predictions
Datasets for past
prices
Daily data
Prices
Prediction Algorithm
Current
market sentiment
Sentiment Analysis

6. Preliminary Schedule

6.1 Project Timeline

Task	Category	Start Date	No. Days
Idea generation	Research	20/09/2018	10
Blog	Documentation	05/10/2018	226
Project supervison & Proposal	Documentation	01/10/2018	23
Research of topic and learning	Research	01/10/2018	200
Functional Specification	Documentation	15/10/2018	15
Django app creation & familiarisation	Development	26/10/2018	5
Basic front-end creation	Development	31/10/2018	21
Connection to Database	Development	21/11/2018	10
Login & Registration	Development	01/12/2018	10
API trading connections	Development	11/12/2018	21
Study period for exams (Sem 1)	Study	12/12/2018	38
Dataset selection	Research	20/01/2019	4
LSTM model creation	Development	24/01/2019	14
Sentiment Analysis selection/creation	Research	07/02/2019	7
Machine Learning algorithm implementation	Development	14/02/2019	14
Chart Analysis view creation & optimisation	Development	28/02/2019	14
Front-end tweaking & menu creation	Development	14/03/2019	10
Testing & validation	Testing	24/03/2019	21
Bug fixes	Testing	25/03/2019	21
Technical Specification	Documentation	01/04/2019	7
Video walkthrough	Documentation	08/04/2019	2
User manual	Documentation	10/04/2019	7
Study period for exams (Sem 2)	Study	17/04/2019	42
Project Deadline	Deliverable	19/05/2019	1
Project Expo	Deliverable	20/05/2019	1
Project Demonstration	Deliverable	21/05/2019	7

6.2 GANTT Chart



7. Appendices

7.1 References

Volatile nature of Bitcoin:
https://www.marketwatch.com/story/bitcoins-haven-claim-is-destroyed-amid-broad-pummeling-of-riskier-assets-2018-11-24?siteid=yhoof2&yptr=yahoo

Trading terms:
https://www.timothysykes.com/blog/trading-terms-you-need-to-know/

Trading numbers & statistics:
https://brokernotes.co/modern-trader/
https://www.dublincity.ie/councilmeetings/documents/s13720/Report%2030-2017%20Impact%20of%20Trading%20Online%20-%20Stephen%20Brennan.pdf

Twitter API:
https://developer.twitter.com/content/developer-twitter/en.html

XML diagrams:
https://www.draw.io/ & Microsoft Visio

GANNT:
Microsoft Excel