Software Requirements Specification

for

Sentiment Analytics

Version 1.0 approved

Prepared by

Cao Qingtian

Huang Jingfang

Lin Jacky

Leon Gurtler

Kelly Wong Jie Yin

Nanyang Technological University

1 April 2022

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The purpose of this software requirements specification is to document the implementation of the desktop application Sentiment Analytics. This document covers the overall description of the application of the overview of the software system including both functional and non-functional requirements as well as user interfaces.

The system's purpose is to make it easier for companies to track the social media sentiment towards their brand or products and extract actionable insights from that data. The key scope covered are:

- pulling tweets
- sentiment analysis on tweets
- explanation of the sentiments via sentiment reasoning
- a dashboard that visualised the sentiment values and the corresponding reasoning

1.2 Document Convention

The priority level of all lower-level requirements are equal to those of the corresponding high-level requirement unless explicitly stated otherwise.

1.3 Intended Audience and Reading Suggestions

This document is intended for the developers, managers and sales staff. It includes a scope of project and a detailed description of the project, but is not limited to user-interface design, UML diagrams, and other analysis models.

Developers, project managers are strongly recommended to read the entirety of this document to get a clear understanding of the application and its requirements, in order to test or update this application. Users of Sentiment Analytics are also recommended to but it is not compulsory for

them to read the overall description and system features of the application to ensure the application serves their purpose.

The publication to our test customer was omitted, as they, business managers, are not very technologically affine. Therefore, to avoid unnecessary confusion, they may choose to skip this document but only refer to the use case document.

1.4 Product Scope

The abstract goal of the software is to offer actionable insights to companies in real time. This is achieved by providing an interactive dashboard that showcases the recent sentiment specific keywords received on twitter (i.e. their brand, products or competitors) as well as a plethora of tools for easier extraction of information from the raw data, such as technical indicators (Moving Average, Exponential Moving Average, etc.). Additionally, the dashboard includes a section with what we called “Sentiment Reasoning”, where a large language model was used to summarize tweets from a specific timeframe and sentiment range. The latter is arguably the largest value proposition we have, as it greatly alleviates time investment (i.e. reading thousands of tweets) and provides a real time explanation of the sentiment and therefore, a plethora of value-adds to the companies:

- Faster feedback for developers/engineers which can lead to a faster product cycle
- KPIs for managers and the marketing/PR team
- etc.

The software will be monetized by restricting certain functionality behind a paywall.

1.5 References

- Use case document
- Standard network protocols (https://docs.microsoft.com/en-us/azure/mysql/concepts-connectivity-architecture)
- TweePy (tweepy/tweepy: Twitter for Python! (github.com))

- Jurassic-5 (google-research/text-to-text-transfer-transformer: Code for the paper "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer" (github.com))
- Twitter API (https://developer.twitter.com/en/docs/twitter-api/rate-limits)

2. Overall Description

2.1 Product Perspective

The software is a new stand-alone system. However, it does depend on two already existing external subsystems. Namely, the twitter-api and the pre-trained "Jurassic-5" NLP model by Google.

- For the twitter-api connection a pre-existing library was used, called TweePy (tweepy/tweepy: Twitter for Python! (github.com)), which will not be further elaborated upon in this SRS.
- Jurassic-5 was taken from (google-research/text-to-text-transfer-transformer: Code for the paper "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer" (github.com)) and used to summarise tweets with some prompt editing but no further training of the model.

2.2 Product Functions

2.2.1 Main functions for Users

- Users must be able to input information for account registration
- Users must be able to input information for logging in to the system
- Users must be able to edit their login credentials (password) using the verification information
- Users must be able to enter the keyword at the search bar on the main interface.
- Users must be able to view the sentiments of a specific keyword entered by them, on a dashboard.
- Users must be able to interact with the graph by selecting the indicators at the side.

- Users must be able to generate a report in PDF format based on the current information they selected to view. The report has to include the sentiment graph, some statistical information and all of the sentiment reasonings for the displayed time frame.
- Users must be able to view the list of keywords that they have searched before, under the history section.
- Users must be able to add the keywords they searched before into the collection of their favourite keywords.

2.2.2 Main functions of System

- System must be able to store users' information.
- System must be able to verify users' input (login credentials).
- System must be able to display the information about the keyword that the user has searched for.
- System must be able to fetch and display the relevant information based on users' choices of indicators.
- System must be able to store the information of the keywords that the user has searched before.
- System must be able to display the progress of fetching tweets if the user has searched for a keyword that is not found in the database.
- System must be able to generate a report in pdf. format based on the information that the users selected to display.
- System must be interactive and support functionality that makes it easier to extract information from the raw data, such as technical indicators.
- System must allow the user to select a specific timeframe for which both the summary of the positive tweets and the summary of the negative tweets are displayed.

2.3 User Classes and Characteristics

User Class	Frequency of Use	Products Functions	Technical Expertise	Value To The Customer
Developers / Engineers	After releasing a new product or when the development for a new product / incremental improvement on a current product starts. If an agile development style is used, the system can also act as an instant feedback function for early releases.	Checking of the overall development of the sentiment w.r.t their products. Using sentiment reasoning as qualitative feedback w.r.t their products. And generating reports for easier communication during scrum/ development meetings.	High	Real-time feedback on the developed products and therefore a faster development cycle and a better product-market fit.
PR / Marketing	After executing a marketing campaign, the team can use the Dashboard to validate how the campaign has influenced the overall sentiment. Additionally, if the campaign addresses specific short-comings or recent bad publicity, the sentiment reasoning can be	Similar to the Developers, the key product functions used will be viewing the Dashboard, analysing the sentiment reasoning and generating reports for ease of inter-team communication.	Low	Real-time qualitative and quantitative feedback regarding marketing campaigns as well as an easy selection of what bad or good publicity

	used to check whether this bad publicity prevails or has been resolved.			and public sentiment to address.
Business Managers	The manager of the marketing or development team can use the quantitative and qualitative feedback generated by our dashboard as KPIs (Key Performance Indicators) for his/her teams.	The main functionality used will be the generate report function so that the quantitative and qualitative data can be discussed with the team.	Low / Medium	Reports with quantitative and qualitative feedback that can be used in 1-on-1s and/or team meetings.

2.4 Operating Environment

The software will operate on three different operating environments:

1. On the user's device. The interface is executed and viewed directly on the user's device in the form of a desktop application.
2. The Database is hosted online, so that the users device does not carry the burden of storing potentially gigabytes of data and that if multiple users view the same keyword, it does not have to be downloaded from twitter multiple times, but rather accessed in the centralised storage, which reduce financial and computational cost on our side, and speed up the user-experience on theirs.
3. All computationally expensive and asynchronous tasks (including downloading and preprocessing tweets as well as summarising the via the T5 language model) are run on our cloud server (which is connected to the edge user via a custom REST API).

2.5 Design and Implementation Constraints

The key limitations that will have to be considered are:

- **Twitter API**: The Twitter API has a rigorous limit for how many tweets can be pulled in a specific time-frame (https://developer.twitter.com/en/docs/twitter-api/rate-limits). This will not only have to be taken into consideration for when the system is deployed (i.e. including relevant loading-bars and making sure that the downloading can run in the background), but might also pose some limitations for testing during development, as it can slow down the progress. Therefore, it is advisable to reduce the maximum number of tweets to download during preliminary testing.
- **Sentiment Reasoning**: The sentiment reasoning (the summary of tweets for a specific timeframe and sentiment grouping) is done in the cloud with a non-scalable server (limited to 24GB V-RAM and 128GB RAM). Therefore, both during testing and after deployment, it is important to run the sentiment reasoning in sequence, rather than parallel. This is very important to point out, as, in contrast, the downloading of tweets should run in parallel.
- **Edge device**: All code built to be run on edge devices has to be lean and not require too much computation. Therefore, all "heavy" tasks have to be moved to the cloud.
- **User Data Security (GDPR)**: The only sensible user-data collected is the account data. It is of utmost importance that the data is kept as safe as possible, and adheres to Europe's GDPR (data has to be stored in a European country). Therefore, it is advisable to build the MySQL instance on a European server.

2.6 User Documentation

The user documentation including use case diagrams, use case tables as well as the corresponding sequence diagrams will be included in a separate document 'use_cases.pdf'. The dialog map and data dictionary can be found in the Appendix below.

2.7 Assumptions and Dependencies

The two assumptions and risk factors are:

- **Twitter**: Until the system has expanded to process data from other social media platforms, twitter and their goodwill towards API users present a single point of failure that is outside our control. Therefore, it is important to be up-to-date with any potential changes to their policies, create contingency plans and as soon as possible include other social networks.
- **Government Regulation**: There has been a recent trend, especially in Europe, to tighten privacy laws. Thus far, this has not affected the anonymous citing and collection of publically available data (i.e. social media), but it certainly isn't a given that this will continue to be the case. Therefore, contingency plans have to be in place and optimally the system should have functionality to fully exclude a specific geographical area.

3. External Interface Requirements

3.1 User Interfaces

The user interface of the application is designed by incorporating Shneiderman's Eight Golden Rules. These rules will hence aid to improve the users' experience while using the interface.

3.1.1 Strive for Consistency

- The layout for the interface is consistent with a navigation bar that is located on the top of every page.
- The colour scheme throughout the interface was consistent where the same few shades of grey were used between all the pages.
- Pages with similar functionality, such as the login page and signup page, have a similar layout of widgets.

3.1.2 Enable Frequent Users to Use Shortcuts

- The signup and log in features allow the users to log in into the interface. This allows choices made by them to be unique to their account.
- For instance, the users can choose to save their favourite words, and these words will be saved into the database. These favourite words can then be accessed from any interface whenever the users log into their account.

3.1.3 Offer Informative Feedback

- The progress bar displays the progress of how much data has been pulled for a keyword that was searched by the user.
- The bar is present such that users will be informed of the progress, and may be able to estimate the time remaining for the keyword.

3.1.4 Design Dialog to Yield Closure

- There are popups after interacting with certain widgets to inform users of their actions.
- For example, after generating a report, a popup with the message “Report generated.” will appear if the user performed the action correctly.
- Before creating an account, there will be a confirmation page at the end that contains all the information given by the users, such as their email address, username, etc.

3.1.5 Offer Simple Error Handling

- The interface will inform the users if there are mistakes in their inputs, and how they could rectify the problem.
- In the log in page, users will be informed if the login information is incorrect. The popup will inform the users if the username does not exist, or if the password given by them is wrong.
- For the sign up page, users will not be able to continue to the next page if there are fields that are left empty.
- If a user attempts to sign up an account with a username that already exists in the database, a popup will inform the users of this, and they will not be able to continue to the next page.

3.1.6 Permit Easy Reversal of Actions

- There are options in the settings of the interface that allows the users to change their plan or their password.

3.1.7 Support Internal Locus of Control

- There are no operations that will occur naturally. All operations in the backend or the frontend will be triggered by the users themselves.
- Search box on the top left of the dashboard page allows the user to search the data from the database while the "Update Tweets" button will fetch new data using the API. All these functionalities are hence controlled by the users.

3.1.8 Reduce Short-Term Memory Load

- The "Dashboard" button on the top left of the interface will bring the user back to the main page, whichever page the user might be on.
- The dashboard page is kept to a minimum where only the necessary functionalities are kept, so as to not distract the users. Other pages are also kept minimalistic so that it is clear what needs to be done on each of these pages.

3.2 Hardware Interfaces

The interface is a desktop application that runs on any devices that have Python and Tkinter installed. The device should also have a connection to the internet such that it can pull tweets with the API and query data from our cloud database.

3.3 Software Interfaces

Software used	*Category*	*Description*
Tkinter 8.6	Library - user interface	The tkinter package is the standard

		Python interface to the Tck/Tk GUI toolkit. Our group utilised this package to implement a user interface that includes all the functions available to our users.
MySQL 8.0.28	Database	Our group used mySQL database to store and manage the user information as well as the fetched tweets for further analysis.
Tweepy	Library - access API	Tweepy is an open-source python library that was used to simplify connecting with the twitter-api to fetch historic tweets.
Jurassic-5 (T5)	NLP Model	T5 is a google developed Text-to_text Transfer Transformer that has been open sourced. It was used for summarising tweets within specific timeframes and sentiment groupings.
Fpdf2 2.5.2	Library	Our group makes use of the fpdf2 package to generate a report in pdf. format if the user wishes to download a copy of the report that consists of information they are interested in.
Pandas, Seaborn, Matplotlib	Library - data visualisation	We mainly used these three libraries to plot the data and display in the user interface.

3.4 Communications Interfaces

3.4.1 Communication standards:

- **MySQL Database**: The MySQL Database utilised is hosted on Azure and follows all of the standard network protocols (https://docs.microsoft.com/en-us/azure/mysql/concepts-connectivity-architecture) and devices with access right to the database were whitelisted. Additionally, every SQL-Command executed is checked for potential security threats (mainly SQL-Injection attacks)
- **API Server**: The API server is on-premise and made available via a ngrok tunnel and uses HTTP.
- **Twitter-API**: Connection to the twitter-api is done via a standard http request call.

3.4.2 End-user device requirements:

The end-users device has to be connected to the internet when accessing new data (either via updating tweets or changing a keyword). However, once the user starts pulling a new keyword he/she can disconnect the device and the download will continue (as it is run on cloud). Additionally, any data currently displayed on the dashboard will continue to be displayed even if the user loses their connection.

4. System Features

4.1 User Registration, Login and Forgot Password

4.1.1 Description and Priority

This feature allows users to register for an account and choose a preferred plan which provides access to certain features of the application. The system will save the information of the user (e.g. email, chosen plan) in the database. Although users are not required to have an account before using the main feature (sentiment graph based on specific keyword) of the application, users are highly recommended to create an account as the database saves his/her search history and favourites (in detail in 4.2).

4.1.2 Stimulus/Response Sequences

	User action	System Response
1	Click the “Welcome!” button.	The login page is displayed.
2	Click the “Login” button after keying Username and Password.	Returns to the dashboard page (displaying sentiment graph) while the “Welcome!” button is replaced by “Welcome, {Username}!”, showing that the user successfully login to his/her account.
3	Click the “Forgot Password?” button	• The forget password page is displayed. • The interface will ask for the answer to the verification question, username and new password. • If the username exists and the user is

		verified using the verification answer, the password in the database will be updated.
4	Click the “Signup” button to create a new account.	The registration page is displayed.
5	Enter personal information (e.g. Username, Email, Password and Verification Question)	The user inputs are displayed.
6	Click the “Continue” button to finish up the account creation process.	● If the username is taken by other users, the system will throw an error message for the user to choose another username. ● Otherwise, the plans page is displayed
7	Click the “Select” button under the preferred plan (e.g. Starter, Professional, Organisation).	The confirmation page is displayed with personal details (e.g. Username, email, selected plan and answer to the verification question)
8	Click the “Create Account” button.	The user will automatically login after successfully creating a new account. (Refer to Step 2)

4.1.3 Functional Requirements

REQ-1: User must be able to input information for account registration

- User must be able to enter Username, Email Address, Password, Password Confirmation and Verification answer.

REQ-2: System must verify whether username already exists in the database.

REQ-3: System must check for password validity.

- System must check that password and password confirmation are identical.
- System must check that the password is at least 4 characters long.

REQ-4: Users' information must be stored in the database.

- System must be able to ensure that all fields are entered.
- System must be able to verify the user's input (username and answer to verification question) when updating his/her password.
- System must also tell the user whether the registration process is successful.
- System must also tell the user whether the update of password is successful.

4.2 Search History and Favourites

4.2.1 Description and Priority

When the user is signed in to his/her account and searches for a keyword, the keyword will be saved to the database as his/her search history. Furthermore, our interface has a function for the user to add the keywords that he/she is most interested in to his/her favourites section.

4.2.2 Stimulus/Response Sequences

	User Action	System Response
1	User enters a keyword into the search box.	System will add the keyword into the user's search history in the database. The keyword is displayed below the search box.
2	Clicks the "Add" button beside the keyword	• System will add the keyword into the

	the user has already searched for.	user's favourites in the database. ● The keyword is displayed in the "Favourites" section below the search history.

4.2.3 Functional Requirements

REQ-1: The user must sign in to his/her account for the keywords searched to be stored in the database as his/her search history.

- It is possible for the user to search for keywords when he/she did not sign in to his/her account, the keywords searched will be displayed in the "Search History" section. However, as the user did not sign in to his/her account, the keywords searched will not be stored in the database.
- Furthermore, if the user is not signed in to his/her account, they are not allowed to use the Favourites function, as the keywords will not be saved to the database as his/her favourite keywords.

REQ-2: System must be able to store the keywords (Favourites and Search history) in the database.

- This function assists users in easily finding the keywords they previously searched for and allows them to store the keywords that they will consistently search for.
- System must be able to record the username of the user and the keywords he/she specifically searched for in the database.

4.3 Plotting of the Sentiment Graph

4.3.1 Description and Priority

A scatter graph will be plotted based on the sentiments of tweets that contain keywords provided by the user. The graph will illustrate the change in sentiment over time for a fixed period of 7 days, which will be vital information to companies and the public who want to know the outlook of a topic.

4.3.2 Stimulus/Response Sequences

Users will search for a keyword in our application. In the backend, tweets that contain the relevant information to the keyword will be extracted from Twitter API. The data collected will then be analysed by our Sentiment Analysis model, which will output the sentiments of each tweet. These sentiments will be recorded to our database, and plotted as a scatter plot with respect to time. The scatter plot may be smoothed using indicators like Moving Average with various sliding widths (5, 10, 20) and Exponential Moving Average with smoothing factors (0.1, 0.2, 0.3) such that trends can be identified easily.

4.3.3 Functional Requirements

PLT-1: Searched keywords and results recorded in the database.

- Keywords that have been searched, together with their respective sentiments, will be saved in the database.
- When users search for identical keywords from our application, sentiments will be retrieved from our database, instead of retrieving the information from Twitter API and running the data through our model.
- As retrieving information from the database is faster than using the Twitter API, it will help to optimise performance and reduce the time taken to produce a result when users utilise our application.

4.4 Direct Manipulation of the Sentiment Graph

4.4.1 Description and Priority

The direct manipulation feature will ease information extraction by allowing users to customise the sentiment graph in real-time. This includes using indicators such as the moving average of various sliding widths (5, 10, 20), exponential moving averages of various smoothing factors (0.1, 0.2, 0.3) or number of positive, negative and neutral tweets each day in the form of bar plots.

4.4.2 Stimulus/Response Sequences

The users will directly manipulate the graphs in an intuitive way, as detailed below.

Using the indicators beside the plots:

- To understand the trends behind the sentiment graph, the users have the choice to use the indicators displayed beside the plot (as detailed in REQ-1). The users can choose to use the Moving Average indicators with different sliding widths (5, 10, 20) or Exponential Moving Average indicators with different smoothing factors (0.1, 0.2, 0.3) to smooth the sentiment plot such that trends could be easily identified.
- All currently used indicators are listed on the right hand side (as detailed in REQ-2). Some indicators include adding a horizontal line to show the values of 25th percentile, median and 75th percentile, and a line plot to show the values of 25th percentile, median and 75th percentile for each day. These indicators give important information about the trends of sentiment graph as they show how the values vary across the 7 days. Other indicators include adding a horizontal line to show the values of maximum and minimum sentiments overall and a line plot to show how the values of maximum and minimum sentiments vary across the 7 days.
- Another important indicator will be the bar plots to show the number of positive, negative and neutral tweets each day (as detailed in REQ-3). These indicators provide insightful information of how the number of positive, negative and neutral tweets vary across the 7 days, and also the variation of number of tweets related to the keyword each day.
- Lastly, the user can choose to look at the density plot of the sentiments instead of the scatter plot of sentiments. The density plot gives the user an overview of the distribution of sentiments over the range of -1 and 1, with -1 being the most negative sentiment of tweets and 1 being the most positive sentiment of tweets.

4.4.3 Functional Requirements

REQ-1: Indicators such as Moving Averages (MA) and Exponential Moving Averages (EMA).

- All the indicators are listed next to the graph to offer an intuitive path of action for users to take.

- The adjustments made to the sentiment graph in the centre of the dashboard page will be reflected in the graph in real-time. All adjustments will be made via indicators since indicator parameters like sliding widths for MA and smoothing factors for EMA are usually restricted in a numerical range. This will prevent unnecessary user error.

REQ-2: Indicators such as Quantiles (overall and for each day) and Maximum and Minimum sentiments (overall and for each day).

- These indicators will be positioned next to the graph below the MAs and EMAs.
- For overall quantile intervals and maximum and minimum sentiments, a horizontal dotted line will be plotted corresponding to each value, such that the user will be able to have a better glimpse of the position for each specific value.
- As for quantiles and maximum and minimum sentiments for each day, different markers are used to represent the type of value each point (values of quantiles, maximum and minimum sentiment of each day) belongs to. This helps users to easily distinguish the specific values which are joined by lines.

REQ-3: Indicators in the forms of bar plots to show the number of positive, negative and neutral tweets each day.

- These indicators are labelled as Count, Positive, Negative and Neutral.
- The indicator Count shows a stacked bar plot which joins the bar plots of positive, negative and neutral tweets each day, determined using a threshold value of 0.05, such that if the sentiment is more than 0.05, the tweet is positive and if the sentiment is less than -0.05, the tweet is negative, otherwise the tweet will be deemed as neutral.
- Positive, Negative and Neutral indicators are bar plots of the number of positive, negative and neutral tweets each day respectively.

4.5 Display sentiment reasoning (summary and statistics) for specific interval

4.5.1 Description and Priority

This feature will display the reasoning behind the sentiments for a selected day. The motivation comes from the assumption that users (e.g businesses, corporations) will not only care about the public sentiment, but are also interested in the reason for a certain public sentiment. This reasoning can be achieved by summarising all tweets within a certain day, effectively offering a super-brief summary of what the general public thinks of a firm. This summary can hence bring about a better understanding of the firm. Also, some additional statistics of the sentiments like the percentages of positive, negative and neutral tweets will be provided along with the summary of the tweets for a specific day below the sentiment graph.

4.5.2 Stimulus/Response Sequences

- The only stimulus required of the user is to select the specific day that he/she is most interested in. The two text boxes below the sentiment graph will display the reasoning for the positive and for the negative tweets. The specific day of the tweets that will be taken into consideration can be adjusted via a widget above the two boxes. (DIS-1)
- Furthermore, there is also another function that provides the overall summary statistics of the tweets related to the keyword chosen by the user. These statistics are calculated and displayed below the sentiment graph and users are not required to take any action for this function. (DIS-2)

4.5.3 Functional Requirements

DIS-1: Displaying sentiment reasoning

- It is important that the sentiment reasonings for the selected timeframe are pre-loaded, as they require substantial computation (running the DL model) to create.
- Additionally it is important that it is explained to the user that the reasoning summaries are created via a non-deterministic language model and therefore might not be correct every single time.

DIS-2: Summarised statistics

- The statistics of the sentiments like the percentages of positive, negative and neutral tweets will be computed and displayed below the sentiment graph based on the sentiment labels of each tweet determined by the NLP model, which is then stored in the database.

4.6 Report Generation

4.6.1 Description and Priority

A report of the sentiment graph, together with its summary for each day and statistical description, can be downloaded as a PDF such that it can be reviewed by companies or the general public. The report will be generated on a user's request. The report's format will be fixed and the user will only be required to press a button to download the report. This feature has medium priority as it is an additional feature that helps improve the overall usability of the application. However, this function is restricted to users who have paid for professional (Level 1) or organisation features (Level 2). In addition, there is a restricted number of reports that can be generated per month by users who only paid for the professional plan.

4.6.2 Stimulus/Response Sequences

After the sentiment graph is plotted, users may choose to keep a record of this data by clicking on a button to save all this information in a nicely formatted PDF. The PDF will be downloaded to the user's local device and can be accessed easily for further references.

4.6.3 Functional Requirements

PDF-1: Manual Generation of Report

- The user should be able to easily identify the element that needs to be interacted with, which is the "Generate Report" button at the bottom left corner of the dashboard page.
- By clicking on the widget, the report will be generated using the fpdf python module and will be downloaded to the user's local device.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

5.1.1 The system should not take longer than 3 seconds to include a loading sign.

5.1.2 The system must offer informative feedback to the user.

5.1.2.1 System should display a loading bar to inform the user about the fetching progress if the process takes longer than 10 seconds.

5.1.2.2 System should display an explanation to the user if one process is taking too long(eg. A process that takes longer than 15 seconds).

5.1.2.3 System should display relevant error messages to inform the users what went wrong when an error occurs.

5.1.2.4 System needs to remind users to have a stable internet connection when they are fetching new tweets.

5.1.3 The system must allow easy reversal of the user's actions.

5.2 Reliability Requirements

5.2.1 The system must not crash when the user opens the application and uses the functions.

5.2.2 The system must make sure the fetching is continued after the user chooses to fetch data and go offline.

5.2.3 The system must be able to work at any time of the day.

5.2.4 The system must be able to restart to full functionality within 10 minutes of crashing.

5.2.5 The system must not leak any user information.

5.3 Security Requirements

5.3.1 The system must remind the users not to make any important business decision solely based on the summary provided or other information displayed.

5.3.2 The system must be able to detect SQL injection attacks and prevent it.

5.3.3 The system must verify user' inputs and verify users' identities.

5.3.3.1 The system must ensure the username and password are correct when the user logs in.

5.3.3.2 The system must be able to verify the user's identity before it allows users to change or recover their password.

5.3.4 The system must ensure accounts' information is inaccessible by unauthorised personnel.

5.3.4.1The account information is encrypted in the database and only accessible by system administrators who have the login credentials.

5.4 Usability Requirements

5.4.1 The system must be easy and intuitive to use for a non-tech-savvy user.

5.4.2 The system must ensure consistency.

5.4.2.1 The system must have a consistent UI design. (e.g. colours, buttons, fonts, etc.)

5.5 Business Rules

5.5.1 The correct user is charged for each action.

5.5.1.1 The user is charged for adding a new key-word to the list and for any additional computation required.

6. Other Requirements

Appendix A: Data Dictionary

Term	Definition
User	A user is a person who uses the application to find the general sentiment towards a particular keyword he searches for and possibly the reason for the sentiment.
Plan	There are three plans that the user is able to choose from. The free version has limited functionalities, and number of keywords that the user can search in a month. It will be followed by the Professional version which will have a step up in the number of keywords and functionality. Lastly, the Organization version has all functionalities with an unlimited number of keywords.
Keyword	The keyword corresponding to the tweets that we want to run sentiment analysis on.
Raw tweet	The raw tweets pulled from Twitter for a specific keyword.
Pre-processed tweet	The preprocessed raw tweets (preprocessing includes removing “illegal” characters, translating the tweet if necessary, and removing unnecessary substring, such as links).
Timestamp	This timestamp corresponds to when the tweet was created.
Sentiment	Sentiment denotes the connotation related to a subject. It is given a score from -1 to 1, with -1 being a negative connotation and 1 being a positive connotation.
Tweet URL	The url of the tweet created by the user at the specific timestamp
Start time	The start timestamp of the specific range that the user choose for sentiment analysis
End time	The end timestamp of the specific range that the user choose for sentiment analysis
User Feedback	The users feedback regarding the accuracy of the sentiment summary is recorded here for ease of re-training the NLP model.
Moving Average of	The moving average of the sentiment for a specific time-frame. It is

sentiment	calculated by the sum of the sentiment over this time-frame divided by the number of days in this time frame.
Sentiment variance	The variance of the recorded sentiment over the specific time-frame. It is a measure of spread of sentiment over this period. $\sigma^2 = \frac{\sum_{i=1}^{n} (X_i - \mu)^2}{n}$ (μ is the mean of sentiment)
Sentiment standard deviation	The positive square root of sentiment variance.
Transparency	The transparency of the sentiment analysis graph displayed in the dashboard page of the application. (This is adjusted by changing the “alpha” attribute of a graph plot in matplotlib)

Appendix B: Dialog Map (State Machine Diagram)