**The Tweetables**

**Ada Bulgur, Amir Crutchfield, Sydney Johnson, Micaiah Steplight, Isaac Stinnette**

A blue bird in a circle with black text

AI-generated content may be incorrect.

**Design Specifications**

**Table of Contents**

[**1.0 Introduction 3**](#_heading=h.17dp8vu)

[**1.1 Goals and Objectives 3**](#_heading=h.1ksv4uv)

[**1.2 Statement of Scope 3**](#_heading=h.2jxsxqh)

[**1.3 Software Context 4**](#_heading=h.z337ya)

[**1.4 Major Constraints 4**](#_heading=h.3j2qqm3)

[**2.0 Product Design 6**](#_heading=h.1y810tw)

[**2.1 Overview 6**](#_heading=h.4i7ojhp)

[**2.2 User Interface 6**](#_heading=h.2xcytpi)

[**3.0 Architectural Design 8**](#_heading=h.1ci93xb)

[**3.1 Algorithm 8**](#_heading=h.3whwml4)

[**3.2 API Usage 9**](#_heading=h.2bn6wsx)

[**4.0 Restrictions, Limitations, and Constraints 13**](#_heading=h.qsh70q)

[**4.1 Restrictions 13**](#_heading=h.3o7alnk)

[**4.2 Limitations 13**](#_heading=h.2grqrue)

[**4.3 Constraints 13**](#_heading=h.vx1227)

[**5.0 Software and Performance 15**](#_heading=h.3fwokq0)

[**5.1 Classes of Tests 15**](#_heading=h.1v1yuxt)

[**5.2 Expected Software Response 15**](#_heading=h.4f1mdlm)

[**5.3 Performance Bounds 15**](#_heading=h.2u6wntf)

[**6.0 Revision Log 16**](#_heading=h.19c6y18)

# **Introduction**

Tweetables has been given the task of creating a complete Sentiment Analysis system. Sentiment Analysis applications work by measuring the emotional tone of certain products or subjects. Our program will work in the same fashion by using data from social media. By doing this we will be able to achieve our task of being able to measure the emotional tone of certain products. This document will outline all the requirements for this task.

## **1.1 Goals and Objectives**

The goal of the project is to create a running application that will help corporations determine the emotional tone of a selected product. The application will do this by sorting people’s feelings for a product into three different categories. These categories are positive, negative and neutral.

## **1.2 Statement of Scope**

Tweetables plans to use a Lexicon-based approach to build a Sentiment Analysis application. This will be done by using APIs from X to build a Lexicon-Based dictionary that will be used to measure people’s emotional tone towards a product. The application will include components that will allow it to gather data, clean data, analyze data and then display the results of the data. Our application will not be using any ML-based techniques. A synopsis of the tasks that must be completed to create the application are:

* Utilize X’s APIs to gather data.
* Implement data cleaning so that the analysis focuses only on meaningful words.
* Implement Lexicon-Based sentiment analysis within the application so that certain words get tagged with the correct polarity (positive, negative, or neutral).
* An installation script will be provided to install the app so that it can be executed with a single click on the installed file.

## **1.3 Software Context**

| **Software** | **Description** | **Type** |
| --- | --- | --- |
| Google Drive/Google Document | Cloud-based storage service that allows the team to store and collaborate on documents. | File Sharing |
| Visual Studio Code | An integrated development environment used for all of coding purposes for the project | Application |
| Python | The Programming language that the application will be created using. | Language |
| Jira | Allows tasks to be created and assigned to team members to be completed for each iteration. | Application |
| X (Twitter) | The social media platform where we will be using its APIs in order to build the program. | Application |

## **1.4 Major Constraints**

Implementation: Learning how to properly use X’s API. This is a major constraint because it is integral that we understand how the API works in order to get our application to run. It could take a lot of time to learn how to use X’s APIs which could lead to delays with our application.

Lexicon-Based Techniques: Not everyone is familiar with how to implement techniques related to Lexicon-Based searches. As a result this could cause delays as people have to take the time to properly learn how to use Lexicon- Based techniques.

# **2.0 Product Design**

## **2.1 Overview**

The design of our product will focus on these main features:

* Data acquisition
* Data cleaning
* Implementing lexicon-based sentiment analysis on cleaned data.
* Displaying the results.

Our product will work as follows: First, it will collect the raw text data from X, such as tweets. Then, it will clean this data by removing any unnecessary words, or symbols. It will also format the text as well for better analysis. After that, the system will search for keywords in the data that indicate an emotional tone. The data will then be compared to the lexicon dictionary and sorted into several different categories. These categories are positive, negative, or neutral. Finally, the results will be displayed to the user on the dashboard as the form of a graph. By doing this our application will be able to help organizations better understand the emotional tone of certain products.

## **2.2 User Interface**

The user interface (UI) will be designed with simplicity and ease of use in mind. This is to ensure that users can quickly access and understand the sentiment analysis results. The UI will have several features that will help users navigate the application. The core feature of the UI will be the chart that visually displays the sentiment distribution for a given product or subject. This graph will show how much the product or subject is associated with positive, negative, and neutral sentiments.

By keeping the results simple like this, it is easier for users to interpret the emotional tone immediately. Another feature that the UI will implement is allowing users to filter out certain words or symbols that they don’t want to be included in the sentiment analysis. This could be hashtags or user mentions. Additionally, the application will show the sentiment calculation for that product so that users will be able to understand how each tweet's sentiment is determined. To further enhance the user experience, the UI will include an additional filter option that allows users to select the language in which they wish to view the sentiment analysis results. This will enable the application to support multiple languages, ensuring that users from different regions and linguistic backgrounds can access the sentiment data in a language they are comfortable with.

## 

# **3.0 Architectural Design**

## **3.1 Algorithm**

The core of the sentiment analysis in this application will be a lexicon-based algorithm. This method involves the use of a predefined dictionary of words, each associated with a sentiment polarity (positive, negative, or neutral) and a sentiment strength. The application will implement data acquisition to gather data in the form of tweets. When a tweet is processed, it will be broken down into individual words or tokens, a process known as tokenization. After that, our algorithm will implement data cleaning. The process works by taking common words that do not contribute to the overall sentiment score, such as “and” or “the” and removing it from the data. This will be done through a process called stop-word removal. After the cleaning process is finished, the remaining words will be compared against the sentiment lexicon. A polarity score will then be assigned to each word based on its sentiment.

These scores will be aggregated to determine the overall sentiment of the tweet. The final step will involve calculating the sentiment for the entire dataset by aggregating the individual sentiment scores of all the tweets. The algorithm will also be capable of handling multilingual tweets, using language-specific lexicons where necessary to ensure accurate analysis regardless of the language of the tweet.

The algorithm can be broken down into the following:

* Tweets are broken down into individual words.
* Words are matched against the sentiment dictionary,
* A polarity score is then assigned to each word.
* Polarity scores of all the words in the tweet are combined to determine the overall sentiment of the tweet.
* The system calculates the sentiment for the entire dataset by aggregating the sentiment scores of all tweets.
* Results are then displayed to the user.

## **3.2 API Usage**

The application will utilize X’s Search Tweets API to gather real-time tweet data based on user-specified keywords, hashtags, tweets, or mentions. The API allows us to submit a couple different argument types such as string and integer. This gives us a large range of data from X in which we can search from. In order to utilize the API we have to format the call as the following: GET --url '<https://api.x.com/1.1/search/tweets.json?q=>[]'. In place of the bracket would be the data that we wish to find. Once the call is processed we expect the response to be a list of tweets related to the data that we entered. The format of what the returned data could possibly look like is:

"statuses": [

{

"created\_at": "string ",

"Id": int,

"id\_str": "string ",

"text": "string ",

"truncated": boolean,

"entities": {

"hashtags": [],

"symbols": [],

"user\_mentions": [],

"urls": [

{

"url": "string ",

"expanded\_url": "string ",

"display\_url": "string ",

"indices": [

int,

]

}

]

},

"metadata": {

"result\_type": "string ",

"iso\_language\_code": "string"

},

"user": {

"id": int,

"id\_str": "string ",

"name": "string ",

"screen\_name": "string ",

"location": "string",

"description": "string",

"time\_zone": "string",

"geo\_enabled": boolean,

"verified": boolean,

"lang": "string",

"contributors\_enabled": boolean,

"is\_translator": boolean,

"is\_translation\_enabled": boolean,

"geo": null,

"coordinates": null,

"place": null,

"contributors": null,

"is\_quote\_status": boolean,

"retweet\_count": int,

"favorite\_count": int,

"favorited": boolean,

"possibly\_sensitive": boolean,

},

The returned data will show all the different aspects of the tweet. These aspects include the user’s screen name, hashtags used, retweet count and the language of the tweet among other things. The returned data format itself could be larger or smaller depending on how we utilize the parameters of the API call.

By utilizing the API call the system will be able to analyze the current sentiment related to a specific product or subject. To efficiently manage the data retrieval process and stay within X’s rate limits, the system will implement request queuing and batch processing, enabling multiple tweets to be fetched at once and reducing the likelihood of exceeding API limits. Additionally, the application will leverage X’s language detection feature to identify the language of each tweet automatically. This allows users to filter and view sentiment analysis results in their preferred language, supporting a diverse, multilingual user base.

By including this language filtering option, the application ensures its accessibility to users across different regions and cultures. To maintain secure access to X’s API, OAuth 2.0 authentication will be used, ensuring the application can safely interact with X’s services while protecting user data. This approach will help provide reliable, real-time sentiment analysis based on the most up-to-date public opinions.

#### 

#### 

# **4.0 Restrictions, Limitations, and Constraints**

## **4.1 Restrictions**

* Due to the strict time limits, it is possible that some jobs will be completed hurriedly or incompletely to ensure that everything is finished by the planned delivery date.
* X’s API free tier: While X’s free API tier will allow us to access the data, we need to successfully complete a sentimental analysis. There are restrictions like a monthly tweet retrieval limit at 1500 tweets a month and there are also data storage limitations.

## **4.2 Limitations**

* Data limitations: There are limitations to the amount of data we can fetch due to the API’s free tier restrictions. Also, there is potential bias in data on X making it difficult to detect tweets that are serious or trolls.
* Model limitation: Our model could struggle sensing sarcasm in Tweets making it difficult to tell if certain words have a positive or negative tone.

## **4.3 Constraints**

* Incomplete Data: A major feature of X is threads which are connected tweets that allows users to add more context past the 280-character limit. When fetching data, only reviewing one part of a thread could lead to missing context which will negatively impact our sentimental analysis.
* Time: To reach our project's deadline, we're employing timesheets to keep our team organized and on track. To ensure the project is completed as quickly as possible, several team members who are proficient in Python are splitting up the coding responsibilities among themselves.

# 

# **5.0 Software and Performance**

## **5.1 Classes of Tests**

* **Unit Testing:** A type of software testing that verifies the behavior of isolated code units.
* **Integration Testing:** A type of software testing that verifies how well different parts of a software system work together.
* **System Testing:** a software testing process where the entire integrated system is evaluated to verify that it meets all specified requirements
* **Black-Box Testing:** A software testing method that evaluates an application's functionality without looking at its internal code.
* **White-Box Testing:** A type of software testing that gives testers full access to an application's source code and design documents

## **5.2 Expected Software Response**

* Initial integration issues but that will be fixed along with stability testing to ensure customer satisfaction.

## **5.3 Performance Bounds**

* Maximum performance output for our software will show correct sentiment analysis of tweets on X labeling them in their respective categories and showing the overall summary of a certain topic either positive, neutral or negative.

# **6.0 Revision Log**

| **Revision** | **By** | **Date** | **Description** |
| --- | --- | --- | --- |
| 1.0 | The Tweetables | 01/21/2025 | Design Specifications Document for Iteration 1 |
| 2.0 | The Tweetables | 02/13/2025 | Design Specification Document for Iteration 2 |
| 3.0 | The Tweetables | 3/6/2025 | Design Specification Document for Iteration 3 |
| 4.0 | The Tweetables | 3/27/2025 | Design Specification Document for Iteration 4 |
| 5.0 | The Tweetables | 4/20/2025 | Design Specification Document for Iteration 5 |