**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.

**Functional Specifications**

**Table of Contents**

[**1.0 Introduction 3**](#_heading=h.3dy6vkm)

[**1.1 Goals and Objectives 3**](#_heading=h.17dp8vu)

[**1.2 Statement of Scope 3**](#_heading=h.26in1rg)

[**1.3 Software Context 3**](#_heading=h.35nkun2)

[**1.4 Major Constraints 4**](#_heading=h.44sinio)

[**2.0 Usage Scenario 5**](#_heading=h.2jxsxqh)

[**2.1 User Profiles & Use-Cases 5**](#_heading=h.z337ya)

[**2.3 Special Usage Considerations 5**](#_heading=h.1y810tw)

[**3.0 Data Model and Description 6**](#_heading=h.4i7ojhp)

[**3.1 Data Description 6**](#_heading=h.2xcytpi)

[**3.2 Word Dictionary 6**](#_heading=h.2bn6wsx)

[**4.0 Functional Model and Description 13**](#_heading=h.qsh70q)

[**4.1 Functional Diagram 13**](#_heading=h.3as4poj)

[**4.2 Functional Diagram explanation 13**](#_heading=h.1pxezwc)

[**4.3 State-Flow Diagram 14**](#_heading=h.m4rbucm69tua)

[**4.4 State-Flow Description 14**](#_heading=h.nlpx039xkg2x)

[**4.5 Software Interface Description 14**](#_heading=h.8swq6fv82z0r)

[**5.0 Risk Assessment 15**](#_heading=h.147n2zr)

[**5.1 Cybersecurity Attack 15**](#_heading=h.3o7alnk)

[**5.2 Ethical & Legal Risks 15**](#_heading=h.1hmsyys)

[**5.3 Data Accuracy & Integrity 16**](#_heading=h.41mghml)

[**5.4 Performance Issues 16**](#_heading=h.ni4t5070dax6)

[**5.5 Data Collection 16**](#_heading=h.1i3l878e998m)

[**6.0 Restrictions, Limitations, and Constraints 18**](#_heading=h.19c6y18)

[**6.1 Restrictions 18**](#_heading=h.2lwamvv)

[**6.2 Limitations 18**](#_heading=h.111kx3o)

[**6.3 Constraints 18**](#_heading=h.3l18frh)

[**7.0 Validation Criteria 19**](#_heading=h.206ipza)

[**7.1 Classes of Tests 19**](#_heading=h.4k668n3)

[**7.2 Expected Software Response 19**](#_heading=h.1egqt2p)

[**7.3 Performance Bounds 19**](#_heading=h.3ygebqi)

[**8.0 Appendices 20**](#_heading=h.2dlolyb)

[**8.1 Product Strategies 20**](#_heading=h.sqyw64)

[**9.0 Revision Log 21**](#_heading=h.3cqmetx)

# 

# **1.0 Introduction**

## **1.1 Goals and Objectives**

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. 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 in order 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 Usage Scenario**

## 

## **2.1 User Profiles & Use-Cases**

**Primary Users:**

**Market Research Analyst:** A market research analysis would greatly benefit from our lexicon-based sentiment analysis algorithm as it can help analysts gain further insight on brand perception from the public. It allows researchers to analyze tweets over time, determining trends and overall customer feelings towards a company. This can serve as customer feedback allowing companies to adjust their based on the feedback, they received from the sentiment analysis

**Product Managers & Developers:** Our algorithm can assist product managers helping them understand how customers feel about different features and services that their product provides. Our product can create a benchmark against competitors as product managers can compare the sentiment score from their product and competitor’s product to better identify their products strengths and weaknesses. Developers can constantly monitor feedback through real time sentiment score updates keeping them updated on user satisfaction.

**Secondary Users:**

**Journalist:** Our sentiment analysis algorithm can support journalists helping them track the public opinion on impactful events and news. Additionally, our analysis tool can help craft ideal headlines that grab the audience’s attention by understanding which headlines users best engage with based on their sentiment score. Our tool isn’t just for businesses as it can help journalists analyze public opinion, track trends, and craft more engaging content.

**Lawmakers:** Our algorithm can support policymakers analyzing public opinion on different laws and policies put in place. They can leverage sentimental insight to craft laws that align with the public and respond to societal needs. Sentiment analysis helps policymakers make data-driven decisions, while improving communication, and tracking public sentiment in real-time. This will ensure that policies align with public needs while minimizing backlash received after creating them.

## **2.3 Special Usage Considerations**

The application won’t be limited to just English. Users will have the ability to use the application in multiple languages. As a result, the application can be used across different cultures. The application will also be compatible with multiple operating systems. By doing this the application can become more accessible to a wider range of people.

# 

# **3.0 Data Model and Description**

## **3.1 Data Description**

Our Lexicon-based sentiment analysis algorithm will collect data on X to answer the question “What's the best action movie of 2024?” This creates the timeframe of data being collected from January 1st, 2024, until the present. The type of data being collected includes tweets, retweets, hashtags, mentions, and engagement metrics such as likes, shares, and replies. Once the data is collected, it will go through an extensive data cleaning process to eliminate filler words like “and the, or” and other irrelevant content. Next, the text will be standardized, converting all text into lowercase to prevent inconsistent data formatting. Then, our sentiment scoring system ranging from 5 to -5 is assigned to each keyword listed in our word dictionary to create a total sentiment score for each film. By leveraging our sentiment analysis tool, we will determine the best action movie of 2024 based on public sentiment and engagement rather than just box office numbers or critic reviews. This results in data-driven insights into audience preferences and movie popularity.

## **3.2 Word Dictionary**

Our word dictionary contains a collection of all keywords that our lexicon-based sentiment analysis algorithm detects to determine the emotion and tone from each tweet. Each word is paired with a value either positive, negative, or neutral with the higher value representing the most positive words and the lowest representing the most negative words.

**Table 1: Positive Words**

| Word | Value | Word | Value |
| --- | --- | --- | --- |
| amazing | 4 | iconic | 5 |
| artistic | 3 | immersive | 4 |
| award-worthy | 5 | impressive | 3 |
| awesome | 4 | incredible | 5 |
| balanced | 3 | inspiring | 4 |
| beautiful | 4 | intense | 3 |
| blockbuster | 5 | joy | 4 |
| brilliant | 4 | joyful | 4 |
| brilliant-performance | 4 | laugh-out-loud | 3 |
| beautiful | 5 | legendary | 5 |
| breathtaking | 5 | likeable | 3 |
| captivating | 4 | love | 5 |
| charming | 3 | masterfully-directed | 4 |
| cinematic | 3 | masterpiece | 5 |
| clever | 3 | mind-blowing | 4 |
| cool | 3 | mind-expanding | 4 |
| cult-classic | 3 | motivating | 4 |
| emotion-filled | 3 | moving | 4 |
| emotional | 4 | must-watch | 5 |
| emotional-journey | 4 | oscar-worthy | 5 |

| emotionally-powerful | 5 | outstanding | 4 |
| --- | --- | --- | --- |
| elegant | 4 | peak-cinema | 5 |
| electrifying | 4 | perfection | 5 |
| engaging | 3 | solid | 3 |
| enjoyable | 4 | spectacular | 5 |
| entertaining | 3 | strong | 3 |
| enthusiastic | 4 | stunning | 5 |
| epic | 4 | stylish | 3 |
| excellent | 4 | superb | 3 |
| excited | 4 | thrilling | 4 |
| exciting | 3 | timeless | 5 |
| fantastic | 4 | touching | 3 |
| feel-good | 3 | unforgettable | 5 |
| flawless | 5 | unique | 3 |
| fun | 3 | uplifting | 4 |
| fun-ride | 3 | visually-pleasing | 3 |
| funny | 4 | visually-striking | 4 |
| genius | 4 | visually-stunning | 4 |
| great | 3 | watchable | 3 |
| great-dialogue | 3 | well-acted | 4 |
| gripping | 4 | well-done | 3 |
| groundbreaking | 5 | well-paced | 4 |
| heartwarming | 4 | well-written | 3 |
| high-octane | 3 | witty | 3 |
| hilarious | 4 | wonderful | 4 |
| phenomenal | 5 | worthy | 3 |
| powerful | 4 |  |  |
| refreshing | 4 |  |  |
| remarkable | 4 |  |  |
| revolutionary | 5 |  |  |
| rewarding | 4 |  |  |
| riveting | 3 |  |  |
| satisfying | 4 |  |  |
| smart | 3 |  |  |
| solid | 3 |  |  |

**Table 2: Neutral Words**

| Word | Value | Word | Value |
| --- | --- | --- | --- |
| acceptable | 1 | passable | 1 |
| average | 0 | plain | 1 |
| basic | 1 | plain-jane | 0 |
| decent | 1 | predictable | -1 |
| expected | 1 | regular | 1 |
| forgettable | -1 | scary | -1 |
| formulaic | -1 | serviceable | 1 |
| meh | 0 | simple | 1 |
| middle-of-the-road | 0 | standard | 1 |
| moderate | 1 | straightforward | 1 |
| neutral | 0 | surprised | 1 |
| normal | 1 | typical | 0 |
| okay | 0 | uncomplicated | 1 |

**Table 3: Negative Words**

| Word | Value | Word | Value |
| --- | --- | --- | --- |
| angry | -5 | over-the-top | -3 |
| annoying | -3 | overrated | -3 |
| anxious | -2 | overused | -2 |
| atrocious | -5 | painful | -5 |
| awkward | -2 | plot holes | -3 |
| bad CGI | -3 | poorly-executed | -3 |
| boring | -4 | predictable | -1 |
| cheesy | -3 | repetitive | -2 |
| cliché | -2 | ridiculous | -3 |
| clunky | -2 | safe | -2 |
| confusing | -2 | scary | -1 |
| cringe | -4 | shaky | -2 |
| cringeworthy | -4 | shocking | -4 |
| devastating | -5 | simple | 1 |
| disappointing | -4 | slow | -2 |
| disgusting | -5 | standard | 1 |
| disjointed | -3 | straightforward | 1 |
| dragging | -3 | surprised | 1 |
| dry | -2 | tearjerker | 4 |
| dull | -3 | tense | -1 |
| empty | -3 | terrible | -5 |
| exaggerated | -3 | terrifying | -3 |
| flat | -2 | thin | -2 |
| flat-characters | -3 | typical | 0 |
| flop | -4 | underdeveloped | -2 |
| forgettable | -1 | underwhelming | -3 |
| forced | -3 | uninspired | -2 |
| forced-dialogue | -4 | unoriginal | -3 |
| formulaic | -1 | unrealistic | -2 |
| frustrating | -3 | unwatchable | -5 |
| garbage | -5 | waste | -4 |
| hate | -5 | waste-of-time | -4 |
| horrible | -5 | weak | -2 |
| horrific | -5 | wooden acting | -3 |
| insulting | -5 | worst | -5 |
| lackluster | -2 |  |  |
| lifeless | -3 |  |  |
| meaningless | -3 |  |  |
| mediocre | -1 |  |  |
| messy | -3 |  |  |
| nauseating | -5 |  |  |
| nonsensical | -3 |  |  |
| nostalgic | 1 |  |  |

**Table 4: Common Words**

| Word | Value | Word | Value |
| --- | --- | --- | --- |
| action-packed | 4 | leak | 1 |
| animation | 2 | light-hearted | 3 |
| announcement | 2 | lit | 4 |
| attack | -3 | melodramatic | -2 |
| banger | 4 | mid | -2 |
| banger | 4 | mystery | 2 |
| breaking | 2 | noir | -1 |
| buzz | 2 | on-point | 4 |
| campy | -2 | over-indulgent | -2 |
| character-driven | 3 | overhyped | -3 |
| chef’s-kiss | 5 | peak | 4 |
| cheesy-dialogue | -3 | performance | 2 |
| cinematography | 3 | psychological | 2 |
| controversial | -3 | rage | 3 |
| corny | -3 | raw | 3 |
| crisis | -3 | reaction | 2 |
| dark | -1 | rom-com | 2 |
| dead | -3 | rumor | -2 |
| debate | -1 | scandal | -3 |
| directorial-debut | 2 | sci-fi | 2 |
| documentary | 1 | score | 2 |
| drama | -2 | self-aware | 3 |
| editing | 2 | slaps | 4 |
| elevated | 3 | slept-on | 3 |
| ensemble-cast | 2 | story-driven | 3 |
| exposed | -3 | sus | -2 |
| family-friendly | 3 | thriller | 2 |
| fire | 4 | twist | 2 |
| flashy | 2 | under-rated | 3 |
| flop | -4 | understated | 2 |
| genuine | 3 | underrated | 3 |
| goat | 5 | urgent | -2 |
| goofy | -2 | vibes | 3 |
| gory | -2 | visionary | 4 |
| hard | 4 | wack | -3 |
| heavy-handed | -2 | yawn | -3 |
| horror | 1 |  |  |
| insane | 3 |  |  |

# 

# **4.0 Functional Model and Description**

## **4.1 Functional Diagram**



## **4.2 Functional Diagram explanation**

Our users first interact with our user interface (UI) to begin searching movies for sentimental analysis. After completing their search, the request goes to the backend which collects the data and cleans the data allowing the data to be in a consistent format. Then the Lexicon-Based sentiment scoring is given to each tweet based on keywords stored in our word dictionary. These scores are stored in a log and presented in a graph to help our users visualize the analysis.

## **4.3 State-Flow Diagram**

## 

## **4.4 State-Flow Description**

Our state-flow diagram represents the flow of user interaction as they use our movie sentimental analysis system. Our system starts at an idle state until a user initiates an interaction with the system. Once the user opens the system, the search button transitions users into the next phase of the analysis process where a search engine is available. After the user enters a movie title, the system fetches data based on that title and displays the results with the sentiment score. Users can examine these results and form conclusions on the general public opinion on any given movie from 2024. Then the user will return to the home screen where they have the option to close the software or perform another sentimental analysis search.

## **4.5 Software Interface Description**

Our user interface starts with a main screen that features our logo and a log in screen. The user will create a username and password to their liking to create an account. This will store the credentials for future log-ins. The button “fetch tweets” will provide all clean data associated with the keyword. During testing the keywords are “horror” and “fairytale”. The sentiment analysis button will provide the scoring associated with that certain tweet. The bottom of the screen will contain the total score of all the tweets along with the total neutral and negative scores that appeared in the search. This screen will also have a home button allowing them to restart the search for a different movie.

# 

# **5.0 Risk Assessment**

## **5.1 Cybersecurity Attack**

**Risk Level:** High

**Justification:** Using X it is a large and popular platform. Many data breaches and accounts have been hacked. This can affect the user's account, and they will not be able to post or for the team to retrieve tweets.

**Prevention:**

* Users enable two-factor authentication
  + This extra security layer will require users to verify their access with two forms of credentials. The first form will be user created secure username and password. The second form will be a QR scanner similarly to Google Authenticator.
* Limit API key exposure
  + Prevents unauthorized access to the X API by keeping keys secure and limiting their exposure and scope. This can be done by storing keys in environment variables to make them more discrete and regularly generating them to prevent misuse.
* Enable encryption for usernames and passwords
  + This protects our users.txt file that contains usernames and passwords after signing up. With our encryption tool this file becomes encrypted protecting account information even if hackers can get into our systems.

## **5.2 Ethical & Legal Risks**

**Risk Level:** Medium

**Justification:** Going over the data limit X has. If the data limit or time refresh is met then we will have to wait for the new timeframe.

**Prevention:**

* Optimize API calls
  + Minimize unnecessary API requests by caching results to avoid repeatedly fetching the same data. Also use efficient queries to request only the needed fields instead of pulling excessive data.
* Distribute Requests over time
  + Our check rate limit function ensures that there aren't excessive data requests. This also displays the amount of request left and the time until the usage resets allowing users to adjust how often they fetch data.
* Have multiple API keys
  + Potentially use multiple authorized accounts with separate API keys to distribute the request load. Rotate API keys responsibly to ensure that requests are balanced across different keys.

## **5.3 Data Accuracy & Integrity**

**Risk Level:** High

**Justification:** Our data will be cleaned to filter out stop words but not bots and misinformation.

Prevention:

* Do not use accounts that might look suspicious
  + Our system will require a username and password to strengthen our system and our multi-factor authentication will prevent suspicious accounts.
* Filter words that might be misinformation
  + Our sentimental analysis tool searches for keywords located in our word dictionary. Our word filtering removes stop words and focuses on positive, neutral, and negative words to ignore misinformation and unimportant data.

## **5.4 Performance Issues**

**Risk:** High

**Justification:** Overloading of tweets will not only increase caching issues of data but also lead to inaccurate sentiment analysis scores

**Prevention:**

* Sectioning groups of tweets
  + Implementing a safety measure to limit how many tweets the program will pick up during one queue will ensure a moderate level of tweets each time
* Fail-Safe Operation
  + By implementing a fail safe for this occurrence, the program will be able to return to its normal state and collect more tweets without shutting down

## **5.5 Data Collection**

**Risk:** High

**Justification:** API’S transmit data over the internet making them VULNERABLE to man-in-the-middle attacks. If API keys are hardcoded, they can be stolen allowing UNAUTHORIZED access to data and exploit API Rate Limit

**Prevention:**

* Secure API keys
  + Store API Keys in ENVIRONMENT variables instead of hardcoding API credentials
* API Limit Handling
  + Implement Rate limit Aware requests in code
    - Monitor API Usage and regularly refresh keys.
* HTTPS Encryption
  + Always send API requests over secure HTTPS connections to prevent interception.
* Monitor API Request Activity
  + Regularly check X developer account to track when API is used

# **6.0 Restrictions, Limitations, and Constraints**

## **6.1 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 1,500 tweets a month and there are also data storage limitations.

## **6.2 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.

## **6.3 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.

# 

# **7.0 Validation Criteria**

## **7.1 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

**White-Box & Black-Box Testing:** Black box testing is a software testing method that evaluates an application's functionality without looking at its internal code. White box is a A type of software testing that gives testers full access to an application's source code and design documents

## **7.2 Expected Software Response**

The optimal output of the response will be presented in a visual chart format. The user will enter the name of the desired action movie, and a chart will display all relevant tweets associated with that movie. Additionally, a summary section at the bottom will provide a breakdown of the total number of positive, negative, and neutral sentiment ratings, offering a clear overview of audience sentiment.

## **7.3 Performance Bounds**

If the movie has been mentioned on X, all relevant tweets will be retrieved. However, due to our basic-level access to the X API, the system is limited to displaying up to 1,500 tweets per month.

# **8.0 Appendices**

## **8.1 Product Strategies**

Developing a strong product strategy is essential for ensuring the success and growth of our team. We start by defining a clear value proposition, identifying our target audience, and aligning product development with our overall team goals. Conducting market research helps us understand customer needs, analyze competitors, and stay ahead of industry trends. We carefully consider our pricing strategy, whether it’s premium, competitive, or cost-based, to position our product effectively. To differentiate our offering, we focus on innovation, quality, and exceptional customer service, giving us a competitive edge. Managing the entire product lifecycle, from development to launch and beyond, ensures long-term sustainability. We also prioritize customer feedback and data-driven decision-making, allowing us to refine our strategy, improve user experience, and enhance customer satisfaction.

## 

# 

# 

# 

# 

# 

# **9.0 Revision Log**

| Revision | By | Date | Description |
| --- | --- | --- | --- |
| 1.0 | The Tweetables | 01/21/25 | Functional Specification revitalization for Iteration 1 |
| 2.0 | The Tweetables | 2/13/2025 | Functional Specifications revitalization for Iteration 2. |
| 3.0 | The Tweetables | 3/6/2025 | Functional Specification revitalization for Iteration 3 |
| 4.0 | The Tweetables | 3/27/2025 | Functional Specification revitalization for Iteration 3 |
| 5.0 | The Tweetabless | 4/20/2025 | Functional Specification revitalization for Iteration 4 |