

TWITTER SENTIMENT ANALYSIS TOOL
BY TEAM 5

SLACK/DISCORD CHANNEL: [CSC-483 GROUP PROJECT](#)

MEMBERS: KARIM ELAGAMY, SHAUN MATHES, COREY VANCURA, WADE
JOHNSON, AND THOMAS JESS

SOFTWARE FINAL DOCUMENT
FOR: PROFESSOR BRIAN MCBRIDE

ON: 12/10/2021

TABLE OF CONTENTS

Abstract	3
1.1 Purpose of System	5
1.2. Scope of System	5
1.3. Development Methodology	5
1.4. Definitions, Acronyms, and Abbreviations	6
2. Current System	6
3. Project Plan	7
3.1. Software and Hardware Requirements	7
3.2. Work Breakdown	8
4. Requirements of System	9
4.1. Functional and Nonfunctional Requirements	9
4.2. Identified Personas	10
4.3. Use Case Diagram	11
4.4. Requirements Analysis	12
4.5. Monetization Model	12
4.6. Risk Analysis	13
5. Software Architecture	13
5.1. Overview	13
5.2. Subsystem Decomposition	14
Figure 2: Class Diagram of Tweet Analyzer for Subsystem Decomposition	14
5.3. Persistent Data Management	15
Figure 3: Architecture Diagram for Object Interaction	15
6. Object Design	16
6.1. Overview	16
6.2. Object Interaction	16
6.3. Detailed Class Design	17
7. Testing Process	18
7.1. User Experience Tests	18
Time Taken for sentiment analysis: Instant	21
Errors Occurred: None	21
8. Glossary	22

ABSTRACT

The Twitter Sentiment Analysis Tool is a data mining tool that allows an organization to get feedback from the social media platform Twitter based upon a specified search request. This feedback is then analysed via a lexicon based approach to determine if the sentiment of the tweet is positive or negative in nature and display the results back to the organization.

1. INTRODUCTION

These days, data science and data mining is becoming more and more popular and advanced than ever before. With that in mind, we thought of creating a sentiment analysis tool to allow companies and corporations to make use of the data available to everyone online on social media platforms like Twitter to optimize their businesses. With our tool, marketing departments will be able to download and parse tweets from a specific hashtag (for example one regarding their brand or new product) and ascertain the sentimentality of their customers as well as the general public about their product or company. This can be useful for the company to decide what next steps to take, allowing them to respond to their consumers and make them feel heard.

The application can also be used to judge the effectiveness of a recent marketing campaign or advertisement, allowing companies to determine which parts and aspects of your advertising or marketing campaign users responded to, and which aspects they did not like. This can help them then tailor their future marketing strategies to be more effective, improving their conversion rate and allowing them to reach the customers they are trying to get. These are just a few of the motivations for creating this application, as we identified a clear need or demand for the product we are developing, so we started the planning and development process refining our application down to what it is today.

1.1. PURPOSE OF SYSTEM

The application will be able to import datasets composed of tweets made under a specific hashtag, such as a brand or news event, and parse through them. It will then assign each tweet a sentimentality score using a lexicon-based approach to Data Mining, and return an overall sentimentality report on the dataset. The aim is to achieve this with an acceptable level of accuracy, while also allowing the application to perform efficiently enough to handle large datasets with thousands of tweets.

1.2. SCOPE OF SYSTEM

The aim of the project is to develop a fully automated sentiment analysis tool, which will be able to process tweets from the social media website twitter and return a sentiment analysis score indicating the positivity or negativity of each tweet as well as the entire dataset. The application's main two objectives are performance as well as accuracy, in order to ensure users can process as many tweets as possible with the highest level of accuracy possible to receive reliable results. We have preliminary hopes to achieve this with at least 80% accuracy, however we will strive to improve that number as we go along.

1.3. DEVELOPMENT METHODOLOGY

This software was developed with the intention of a single user within an organisation to gain required results of sentiment analysis. As this program was designed in a previous class and built upon within this class, the requirements were first developed based upon the users needs. The design process created the requirements for the program to have a way to accept specified inputs, run them through a lexicon, and report the results back to the user. The team then developed the program to accept inputs, created the lexicon, and display the results. The testing of the program was then performed to verify that the program is functioning. The program was then deployed as the final product was deemed successful.

The requirements have since been changed since the creation of the program. The new requirements were to be more efficient, have a user interface, and an improved lexicon in order to move away from a high value of neutrality. We employed an agile development strategy to target different modules in the code, incrementally creating more and more efficient code to improve performance. We also incrementally worked on improving the lexicon, so that we may achieve the highest accuracy level possible. These systems were then tested by each member within the class and the results were recorded. After deciding that the program is fully functional and improved, the Twitter Sentiment Analysis Tool is ready to be deployed.

1.4. DEFINITIONS, ACRONYMS, AND ABBREVIATIONS

Actors: External entities that interact with the system.

2. CURRENT SYSTEM

This project is a carryover of Karim's final project for the class CSC-530, Information storage and retrieval. Originally the application needed the user to enter each tweet independently, and each tweet was processed individually until the user indicated they were done with their input then the overall result of the dataset was outputted. The application was in a very rudimentary stage at the beginning of this semester, with accuracy scores being very hit and miss depending on the tweet's phrasing.

Now, the user has the ability to enter the tweets as a dataset, providing the application with a csv file containing all the tweets they would like analyzed, and then the result is outputted after processing is complete. This allows the user to analyze hundreds or thousands of tweets at a time, instead of having to enter them one by one. We have also developed a separate auxiliary tool written in Python to create these dataset files, prompting the user to enter the hashtag they would like to analyze and then specify the amount of tweets they would like downloaded. The application then writes these to a csv file, formatted in a compatible way with the analysis tool. We have also improved the accuracy of the analysis tool considerably, by working on our lexicon. We spent time testing the project and adding slang terms, new vocabulary, and defining the Twitter emoticons in our lexicon so that we may factor them into our analysis. This way we can fully analyze a tweet to get a better understanding of its sentimentality, instead of just the words. We have also fixed a few bugs and errors we encountered, including an infinite loop in the analysis tool when a tweet longer than 256 characters was entered. This was due to the fact that Twitter recently raised the maximum character limit per tweet, and as such we must keep up with the platform to ensure the application runs smoothly.

3. PROJECT PLAN

3.1. SOFTWARE AND HARDWARE REQUIREMENTS

Hardware:

Minimum System Requirements:

OS: Windows 7 or Later / Mac OS X 10.8 or Later

Processor: Intel Core i3-2120 or AMD Phenom II X4 905e

Memory: 4 GB RAM

Graphics: Integrated Graphics or better

Storage: 10 GB available space

Recommended System Requirements:

OS: Windows 7 or Later / MAC OS X 10.8 or Later

Processor: Intel Core i5-2300 or AMD Ryzen 3 1300

Memory: 8 GB RAM

Graphics: Geforce GTX 760 or Radeon HD 7950 or better

Storage: 20 GB available space

Software:

Windows 7 or greater, Mac OS X 10.8 or Later

3.2. *WORK BREAKDOWN*

Task #	Task	Description	Duration	Dependencies
1	Project Plan	Get to know team members, brainstorm, assign roles, decide project topic	3 days	
6	Create use cases	Identify use cases, assign use cases to team members, each team member develops their assigned use cases	8 days	1
10	Review and completion of use cases	Present use case diagrams to professor, correct use cases	5 days	6
13	Development of Personas	Complete	12 days	10 (M1) (D1)
17	Software Requirements Document	Present SRD to class, submit SRD to professor	1 day	13
18	Software Architecture	Divide project into subsystems, identify objects, complete design document,	2 days	17
22	Object Design	Transition of software models into source code.	18 days	17
30	Implementation	Database design. Interface Layer, Application Layer and Storage Layer coding.	40 days	27 (M3)
35	Testing Process	Subsystem, System and Evaluation tests. Creation of the User's Guide.	16 days	27
40	Creation of FD	Complete FD, create Power Point presentation.	23 days	30, 35 (M4) (D3)

45	Presentation of FD	Present and submit the FD.	1 day	40
----	--------------------	----------------------------	-------	----

M – Milestone

D – Deliverable

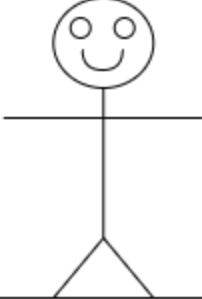
4. REQUIREMENTS OF SYSTEM

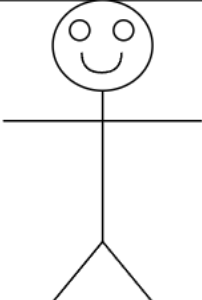
4.1. FUNCTIONAL AND NONFUNCTIONAL REQUIREMENTS

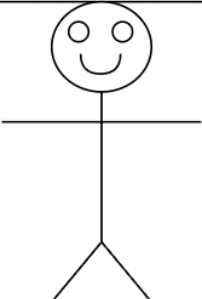
The Twitter Sentiment Analysis Tool™ must adhere to the following requirements:

1. Must be able to handle a properly formatted dataset of tweets, identify them, and separate them into separate strings to analyze individually.
Verification Method: Demonstration, Testing, and Inspection
2. Must be able to process each individual tweet and "rate" it, accurately giving it a sentimentality score from the lexicon.
Verification Method: Demonstration and Testing
3. Must be able to process large quantities of tweets at a time and analyze them within a reasonable amount of time.
Verification Method: Demonstration and Testing
4. Must be at least 80% accurate over a sample of 1000 randomly chosen tweets.
Verification Method: Testing
5. Must display or print out a report to the user about the tweets it processed, clearly indicating the sentimentality of the overall dataset as well as the positive or negative composition of the dataset.
Verification Method: Demonstration, Inspection, and Walk-Through
6. Must be intuitive and easy to use, with an appealing user interface.
Verification Method: Demonstration/Walk-Through
7. In case of failure, the program should output an error message as follows: "Oh no! It seems something went wrong! Please try again and make sure to format your data properly, or reach out to us at support@website.com"
Verification Method: Demonstration, Testing, and Walk-Through

4.2. IDENTIFIED PERSONAS

	<p>Data Analyst / Data Scientist</p> <p>Is 35 years old and hold a degree in Computer Science with a focus in Data Mining or Data Engineering.</p>	<p>Is used to Twitter as a social media platform and has experience and knowledge of Data Mining principles.</p>
<p>Employee</p>	<p>Prefers a data oriented scientific approach to his marketing campaigns.</p>	<p>Wants to analyze the feedback from a specific hashtag on Twitter to tailor his marketing campaigns to their consumer base, or judge the effectiveness and results of an existing or recent campaign.</p>

	<p>Social Media Influencer</p> <p>Is 28 years old and has a communications degree.</p> <p>Likes to use any tool that can provide audience metrics, uses the feedback frequently</p>	<p>Very experienced with Twitter and all social media platforms</p> <p>Is curious how different types of posts perform, and wants to run a number of tweet collections containing responses to individual posts they have made in the past.</p>
<p>Influencer</p>		

	<p>Undergraduate research assistant</p> <p>Is 22 years old and working on a project that has to do with how people behave online.</p>	<p>Very familiar with Twitter.</p> <p>Wishes to download every response to a certain tweet, and run that collection through the tool. Then repeat this process for multiple tweets that have been responded to.</p>
<p>Research Assistant</p>		

4.3. USE CASE DIAGRAM

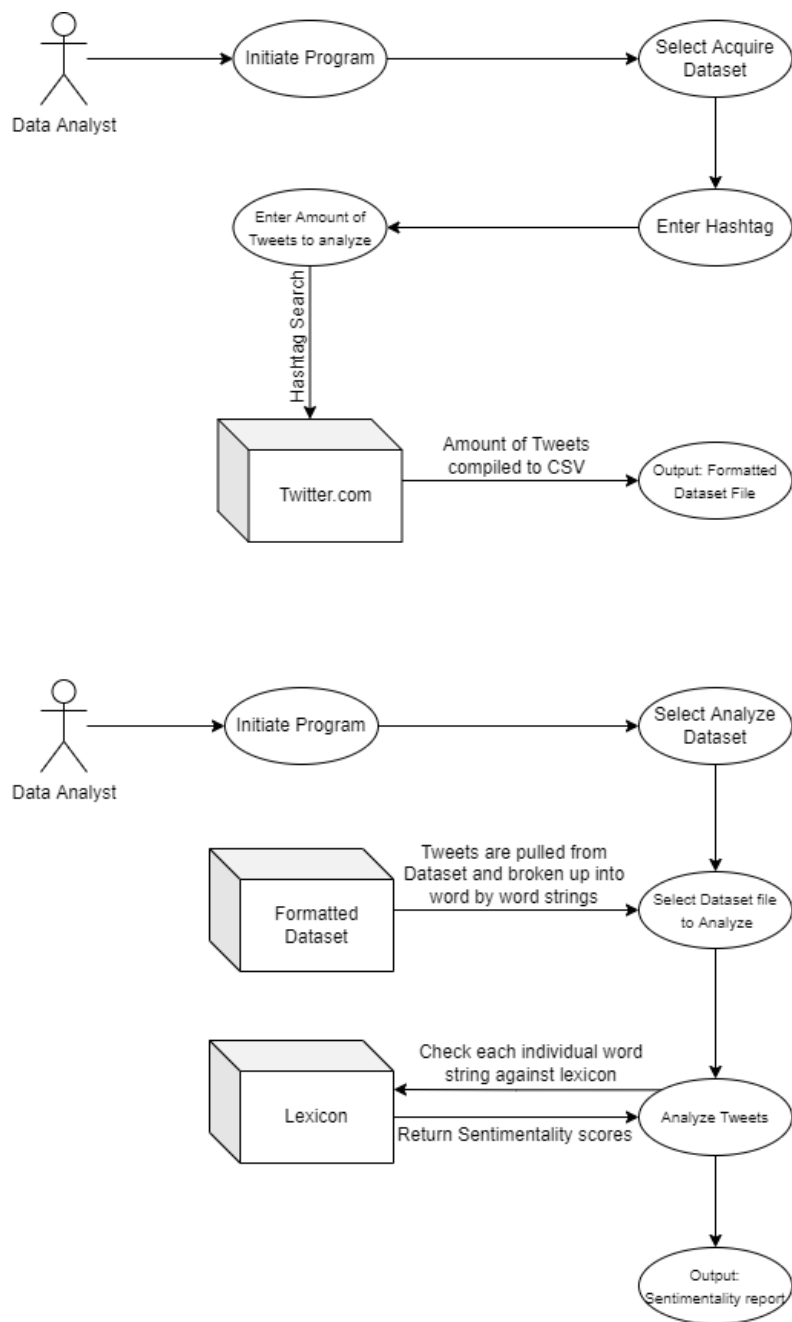


Figure 1: Use Case Diagrams

4.4. REQUIREMENTS ANALYSIS

In order to ascertain all of our clients' requirements, we made use of use case diagrams. We decided to go through what our clients expect to happen through every path in the application, in order to make sure they don't miss anything or forget to mention any requirements they may have. This then allows us to develop the most suitable software architecture and product to meet our clients' needs, ensuring customer expectations are met and our final product has a good customer satisfaction ratio.

4.5. MONETIZATION MODEL

The monetization strategy that will be implemented for this software will be a multi-tiered pricing system, with a timeline for when each price tier will become available. The initial launch of the product will utilize a corporate software licensing model, allowing only companies to purchase corporate licenses priced based on their expected number of users. The license will renew annually based upon the initial purchase date for each company, allowing corporations to have some sort of flexibility in their purchase and renewal dates rather than being limited to the standard business quarters calendar. Below is an example of our pricing tiers for our corporate clients:

Corporate Pricing Guide

	<u>Starter</u>	<u>Midsize</u>	<u>Large</u>	<u>Unlimited</u>	<u>Custom</u>
Number of Employees:	1-15	16-50	51-100	∞	As specified by Client
Pricing per Employee:	\$6.66	\$5.49	\$3.99	N/A	Flexible
Final Cost:	\$99.99	\$274.99	\$399.99	\$799.99	Final Offer made after Negotiations

As the company grows, the program would be opened up to personal accounts and individual users with a variety of pricing options. We are planning to offer individual consumers two versions of this product, one which they can purchase through a one time fee at the beginning, as well as a package deal with a monthly subscription model that will provide them with our continued assistance and support as well as access to custom

functionality upon request. The corporate version of our software will come with this feature included, since it is the industry standard. We will also offer free trial periods for our product upon request to all of our clients, available upon request through our support team. This will allow potential clients to try out our application and see for themselves the upside to integrating it into their business, and hopefully help entice them to become paying clients. You can see our pricing model for these packages below:

Individual Pricing Guide

	<u>Starter</u>	<u>Midsize</u>	<u>Large</u>	<u>Unlimited</u>	<u>Assisted Support Package</u>
Number of Systems:	1	2-5	6-10	10+	10+
Pricing per System:	\$59.99	\$49.99	\$44.95	N/A	N/A
Final Cost:	\$59.99	\$249.99	\$449.95	\$799.99	\$999.99

4.6. RISK ANALYSIS

At this stage we do not believe there to be any risk incurred by us by developing this application, seeing as the information and data it uses is already in the public domain. As such, we've skipped the risk analysis section

5. SOFTWARE ARCHITECTURE

5.1. OVERVIEW

The architecture of a software application is a plan for the application's main structure and functionalities, and as such it is an important decision. After evaluating our customers' requirements and needs, we chose a three tier architecture model for our application. These tiers include our lexicon and sentiment analysis dictionary, our tweet holder and parsing classes, and our tweet grabber and the formatted datasets they create. This allows us to build our application in a modular manner so that we can target errors or bugs more efficiently, since one issue in the tweet grabber module will not affect the performance of the rest of the application.

5.2. SUBSYSTEM DECOMPOSITION

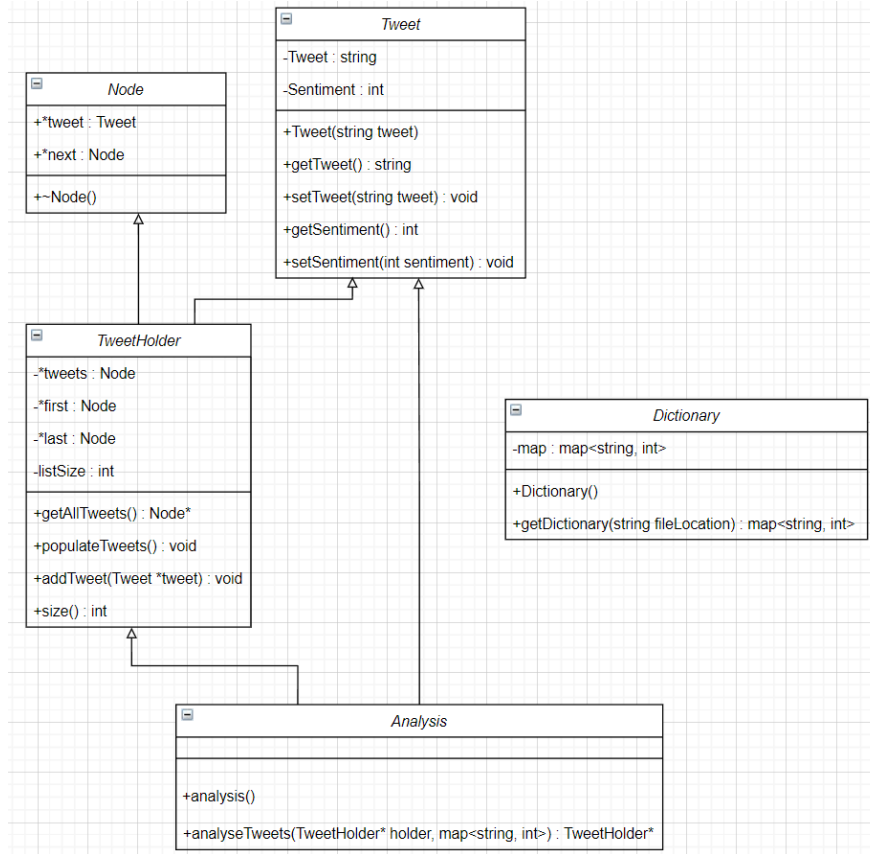


Figure 2: Class Diagram of Tweet Analyzer for Subsystem Decomposition

5.3. PERSISTENT DATA MANAGEMENT

Our persistent data includes the formatted datasets that are created by the auxiliary tweet grabber tool as well as the lexicon. The lexicon is managed by the dictionary class and is stored in a .txt file, while the formatted datasets are printed to .csv files that will then be managed by the tweet and tweet holder glass.

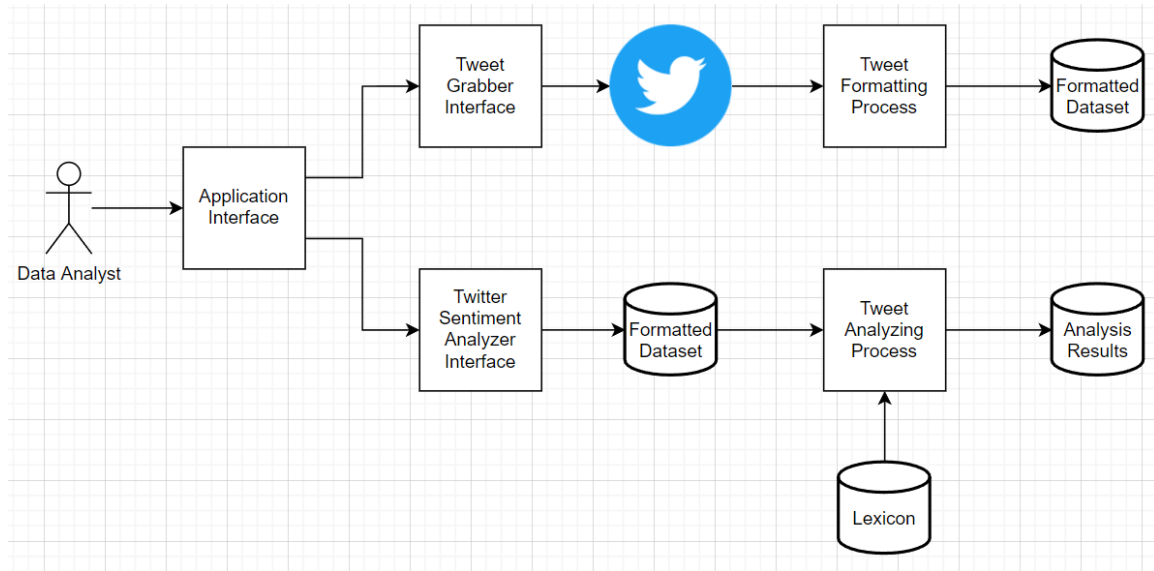


Figure 3: Architecture Diagram for Object Interaction

6. OBJECT DESIGN

6.1. OVERVIEW

The Architectural design of a software application is important during the development process, as it represents the structure of both the software and data involved, as well as the intended interactions between the users and the application. In this chapter, we will go over the different aspects of our application, as well as the inner workings of our classes and objects and how they interact with each other.

6.2. OBJECT INTERACTION

This is our sequence diagram, highlighting the process of actions taken by the user as well as the application throughout the whole process of analyzing a hashtag, from the user selecting the hashtag and grabbing the tweets via the auxiliary tool to putting them through our sentiment analyzer.

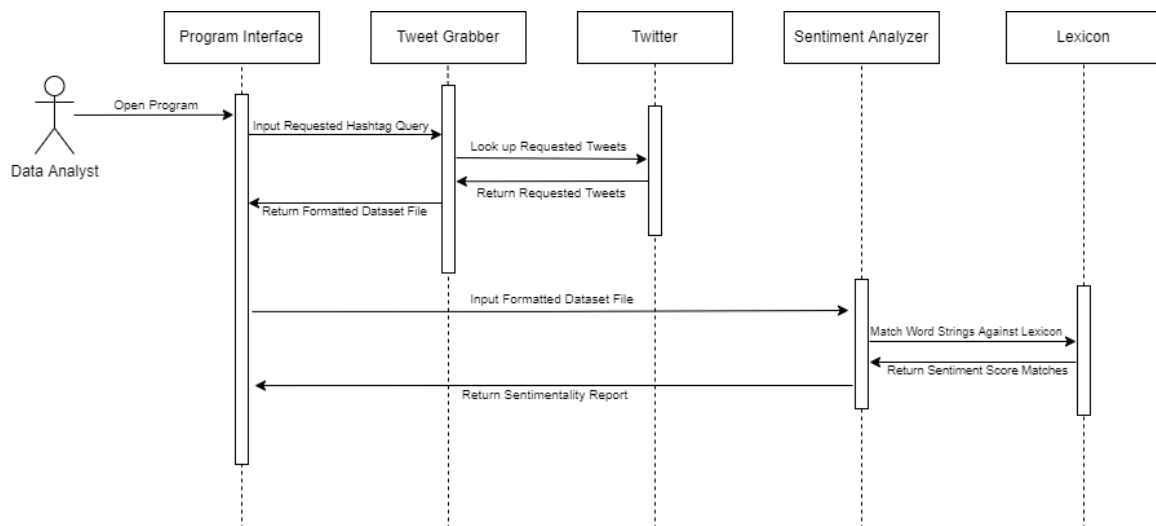


Figure 4: Sequence Diagram for Use Case Program Interaction

6.3. DETAILED CLASS DESIGN

This diagram is a detailed class design of our Tweet Analyser function, breaking down how we are taking in the tweets, breaking them up into strings or Nodes, and parsing them to be analyzed by our dictionary.

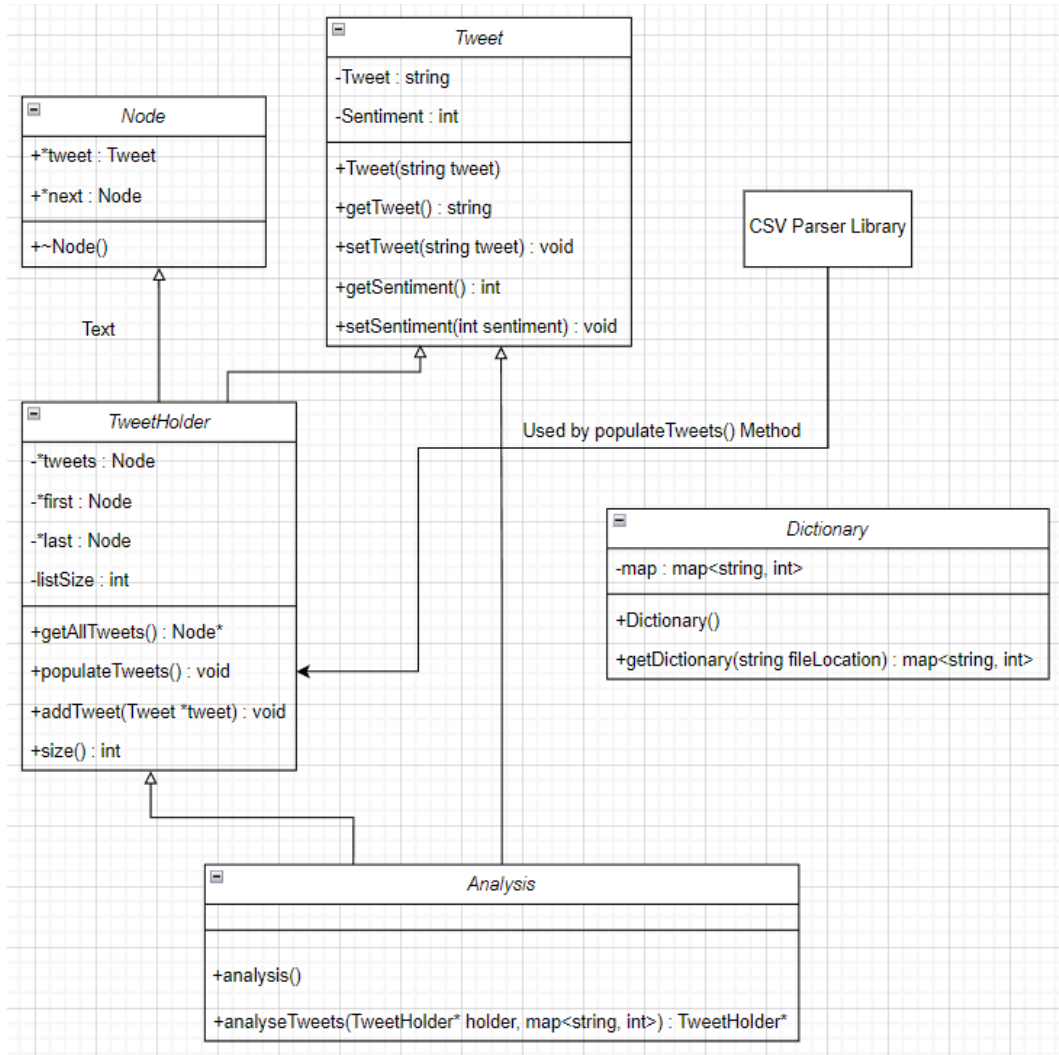


Figure 5: Class Diagram for Detailed Class Design

7. TESTING PROCESS

7.1. USER EXPERIENCE TESTS

Shaun Mathes' Tests

PC Specs:

Motherboard: B550AM Gaming

Ram: 16 GB

CPU: AMD Ryzen 3700X 8 Core Processor

GPU: XFX Radeon RX 6800 XT

Runtime User Experience Test 1:

Hashtag Analyzed: #Razer

Number of Tweets Processed: 85

Results:

```
sentiment for tweet:
Total Tweets: 85
Positive %: 28.2353
Negative %: 2.35294
Neutral %: 69.4118
General Consensus is Positive
Press Enter to exit
```

Time taken for sentiment analysis: Instantaneous

Runtime User Experience Test 2:

Hashtag Analyzed: #SteelSeries

Number of Tweets Processed: 51

Results:

```
sentiment for tweet:
Total Tweets: 51
Positive %: 33.3333
Negative %: 5.88235
Neutral %: 60.7843
General Consensus is Positive
Press Enter to exit
```

Time taken for sentiment analysis: Instantaneous

Errors Occurred: Program ran infinitely when input was above 256 characters

Karim Elagamy's Tests

PC Specs:

Motherboard: LNVNB161216

Ram: 16 GB

CPU: AMD Ryzen 5 4500U 6 Core Processor

GPU: AMD Radeon Graphics

Runtime User Experience Test 1:

Hashtag Analyzed: #FordMustang

Number of Tweets Processed: 71

Results:

```
sentiment for tweet:  
Total Tweets: 71  
Positive %: 43.662  
Negative %: 15.493  
Neutral %: 40.8451  
General Consensus is Positive  
Press Enter to exit
```

Response Time: Instantaneous

Errors Occurred: None

Runtime User Experience Test 2:

Hashtag Analyzed: #Hooters (Their new uniform was trending on Twitter)

Number of Tweets Processed: 153

Results:

```
sentiment for tweet:  
Total Tweets: 153  
Positive %: 50.9804  
Negative %: 39.8693  
Neutral %: 9.15033  
General Consensus is Positive  
Press Enter to exit
```

Response Time: Instantaneous

Errors Occurred: None

Corey VanCura's Tests

PC Specs:

Motherboard: OEM Motherboard

Ram: 12 Gb

CPU: Intel Core i7-5600U

GPU: AMD Radeon R7 M260

Runtime User Experience Test 1:

Hashtag Analyzed: #NBA

Number of Tweets Processed: 1,033,980

Results:

```
"C:\Users\karee\OneDrive\Documents\Assignments\CSC-483\Final Project - Twitter Sentiment Analysis Tool\Sentiment Analysis\bin\Deb...  
Welcome to my Twitter Sentiment Analyzer! CSC-430/530 Independent Programming Project by Karim Elagamy  
-----  
This program utilizes a lexicon-based approach to Sentiment Analysis, taking in any Tweet and analyzing it word by  
word to come up with a general consensus regarding whether it is a Positive or Negative Tweet. You can enter as  
many tweets as you would like, with each being up to 256 characters long in accordance with Twitter formatting.  
  
Enter the file name of your dataset with the extension:  
Dataset-Mil.csv  
sentiment for tweet:  
  
Total Tweets: 1.03398e+06  
Positive %: 40.1764  
Negative %: 4.27784  
Neutral %: 55.5457  
General Consensus is Positive  
  
Process returned -1073741571 (0xC00000FD) execution time : 36.887 s  
Press any key to continue.
```

Response time: 36.887 seconds

Errors occurred: None

Runtime User Experience Test 2:

Hashtag Analyzed: #College

Number of Tweets Processed: 58

Results:

```
Enter your Tweet [Enter a 0 on its own to exit Collection]:  
0  
sentiment for tweet:  
  
Total Tweets: 58  
Positive %: 32.7586  
Negative %: 13.7931  
Neutral %: 53.4483  
General Consensus is Positive  
Press Enter to exit
```

Response time: < 1s

Errors Occurred: None

Wade Johnson's Tests

PC Specs:

Motherboard: ASRock Z270 Gaming K6

Ram: 16 GB

CPU: i5-6600K

GPU: MSI GeForce 1070 ti

Runtime User Experience Test 1:

Hashtag Analyzed: #Hawkeye

Number of Tweets Processed: 60

Results:

```
sentiment for tweet:  
  
Total Tweets: 60  
Positive %: 41.6667  
Negative %: 15  
Neutral %: 43.3333  
General Consensus is Positive  
Press Enter to exit
```

Runtime User Experience Test 2:

Hashtag Analyzed: #N64 and Genesis

Number of Tweets Processed: 60

Results:

```
sentiment for tweet:  
  
Total Tweets: 60  
Positive %: 36.6667  
Negative %: 28.3333  
Neutral %: 35  
General Consensus is Negative  
Press Enter to exit
```

Time Taken for sentiment analysis: Instant

Errors Occurred: None

Summary of Findings:

- Fix infinite loop if a Tweet entered by the user is over 256 characters. (Fixed)
- Add lexicon entries for the twitter emoticon alternative texts.

8. GLOSSARY

Actors: External entities that interact with the system.

Architecture (in software): The architecture of a software system or application is the process of defining a collection of hardware and software components as well as data structures that will make up the system and its components.

Architecture Diagram: An architecture diagram shows the architectural layout of the application, defining its different components and how they interact with each other in a visual manner.

Class Diagram: Class diagrams are used to show a systems structure and classes, attributes, operations, and relationships between them.

Data Mining / Data Science: Searching and finding data to use for statistics and future analysis.

Hashtag: A word or phrase preceded by the pound sign (#), used on social media websites like Twitter, to identify specific topics or events and categorize tweets and users posts under them.

Influencer: Someone that uses social media status to influence other users or consumers in order to grow their own brand or promote businesses.

Lexicon: A Sentiment analysis lexicon is a compilation of words, phrases, or slang terms that make up a language along with sentimentality scores attached to them all, rating the positivity or negativity of a word, phrase, or slang term based on its most commonly used contexts and common occurrences as well as any connotations associated with the actual word, phrase, or term.

Licensing Model: A licensing model is a financial model for software products or services, where client companies pay an annually agreed upon fee for the right for all their employees to use our software or have it on their computers.

Persona: Personas are fictional characters or models created to represent a user type or entity that might make use of a website, product, brand, or application in a similar way. They allow developers to account for as much of their potential customers as possible.

Sentiment: The Sentiment of a phrase or sentence is the view or attitude of that phrase or sentiment towards a particular situation, event, or item.

Sentiment Analysis: Sentiment Analysis is a process by which datasets can be methodically analyzed and compared to existing standards and models to ascertain the sentimentality of that data, whether it indicates a positive or negative sentimentality.

Sequence Diagram: A Sequence Diagram shows object interactions arranged in time sequence in any given module of the application. It depicts the objects involved in the scenario and the sequence of messages or data exchanged between the objects and application processes that are needed to successfully complete the functionality of the application module.

Tweet Grabber: The Tweet Grabber is an auxiliary tool that comes provided with our Twitter Sentiment Analysis Tool, allowing users to search for tweets under a hashtag and create perfectly formatted datasets for the Twitter Sentiment Analysis process.

Use Case: A use case is a software development and planning tool which provides guidance to developers during the planning stage. Use cases are a compilation of the list of actions and interactions a user will have with the application for every possible function path through the application, including other parts of the system that the function is dependent on, other systems or platforms being utilized by the action, and the outcome or deliverable from this action. This forces developers to consider all those aspects during their planning stage.