Criterion B: Design Overview

Test Plan

Criteria	Testing Method	Input	Expected Result
This system is able to collect tweets about the top 5 popular cryptocurrencies on Twitter.	Collect tweets using filters and Twitter API to gather the tweets of the latest hour	Query of the cryptocurrency (ex. bitcoin)	All the most recent tweets in an hour will be collected and stored
This system is able to analyze whether or not each tweet is neutral, bullish, or bearish.	Use a sentiment analysis model to analyze the sentiments of tweets.	A tweet posted regarding bitcoin	Output a decimal number, which converts into the sentiment of either positive, neutral, or negative.
This system has a clear Graphical User Interface (GUI) in python.	A homepage with buttons for each specific cryptocurrency.	Click on the tab named ‘bitcoin’	Open the tab for in-depth bitcoin analysis.
This system has options to view each specific tweet collected under each type of cryptocurrency and their corresponding sentiment.	In each in-depth analysis, there will be a section where it shows 100 tweets and their corresponding sentiments and has an option to expand for more tweets.	Click on the refresh button	Display another 100 tweets and corresponding sentiment.
This system can display the percentage of neutral, bullish, or bearish of a cryptocurrency using a machine learning algorithm.	On the homepage, it will display the sentiments for all 5 cryptocurrencies, using a pre-trained machine learning model.	Tweets collected for bitcoin	Display the percentage of the sentiment of bitcoin in general, and each tweet with its corresponding sentiments.
This system can display the total tweets collected.	On the homepage, it will display the total number of tweets collected & also display the number of tweets collected for each cryptocurrency	Tweet collected for bitcoin	Display the number of tweets collected for bitcoin
This system can display the most discussed	On the homepage, the 5 different cryptocurrencies will be displayed based on popularity	Refresh for more tweets	Display cryptocurrencies based on the total amount of tweets collected in an hour

cryptocurrency over an hour.			
This system can display basic information about each cryptocurrency (cautions).	On the homepage page, there will be information displayed and not financial advice.	Open up homepage	Links & information about cryptocurrencies in general, and display the total tweets collected for each cryptocurrency.

Extensibility

To ensure the code is extensible, readable, and editable for others, I will be using the PEP 8[1] style to program my code in python. PEP stands for the Python Enhancement Proposal which is the most popular style guideline for the implementation of Python created on July 05, 2001. Some implementations of the PEP style include: naming variables & functions with lowercase letters separated by underscores to increase readability; including inline, block comments to enhance the understanding of each section of code; and having two blank spaces after each class definition, while only one blank space after a logical separator, functions, or method definition in a class to follow along with logical steps.

[1] "PEP 8 – Style Guide For Python Code | Peps.Python.Org". 2022. *Peps.Python.Org*. https://peps.python.org/pep-0008/.

Flowcharts

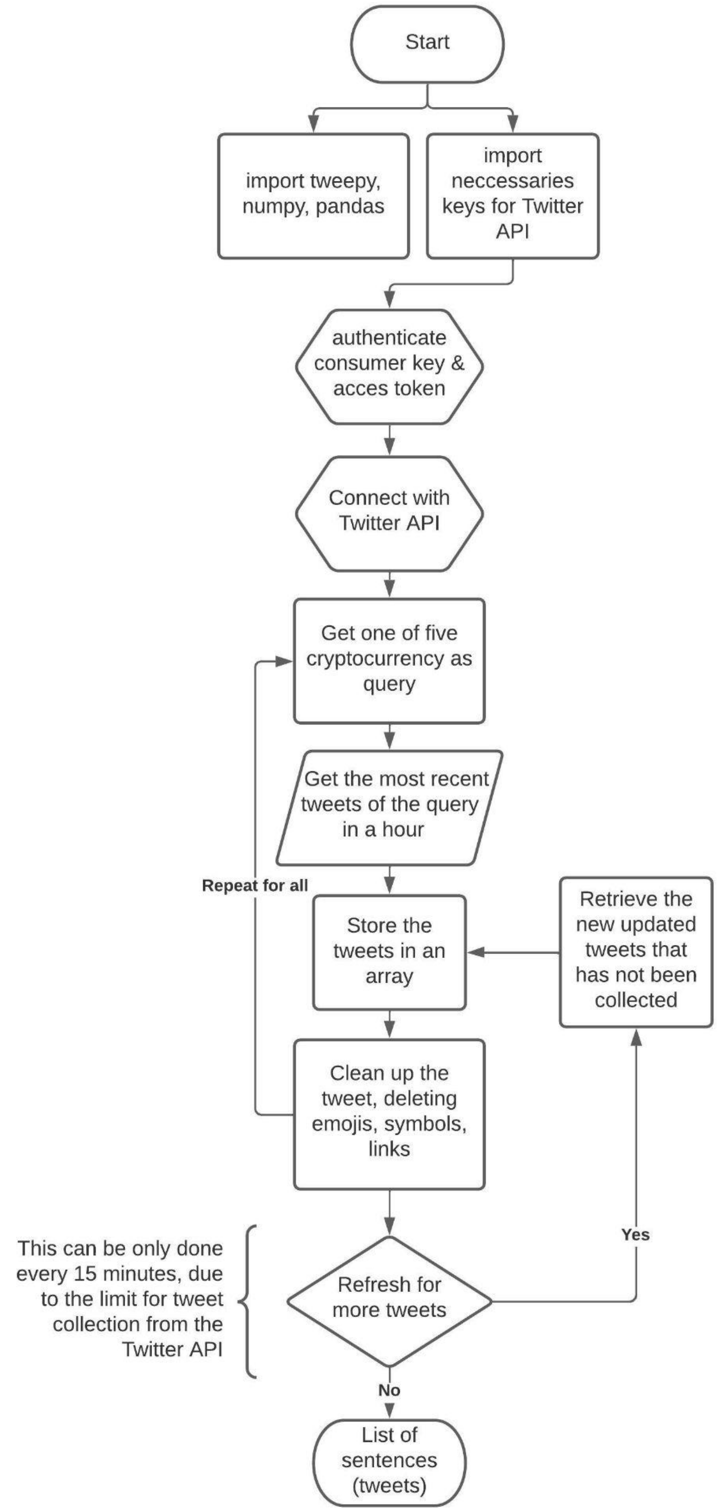

Start
import tweepy, numpy, pandas
import neccessaries keys for Twitter API
authenticate consumer key & acces token
Connect with Twitter API
Get one of five cryptocurrency as query
Get the most recent tweets of the query in a hour
Repeat for all
Store the tweets in an array
Retrieve the new updated tweets that has not been collected
Clean up the tweet, deleting emojis, symbols, links
This can be only done every 15 minutes, due to the limit for tweet collection from the Twitter API
Refresh for more tweets
Yes
No
List of sentences (tweets)

Fig 01. Tweet Collection

Fig 01 is a flowchart illustrating the process of using the Twitter API to retrieve and save tweets. The search function in the Twitter API requires a query, the cryptocurrency, and a maximum of 100 tweets per one request and 180 requests per 15 minutes; hence this process will be repeated several times for each cryptocurrency to ensure an extensive data collection.

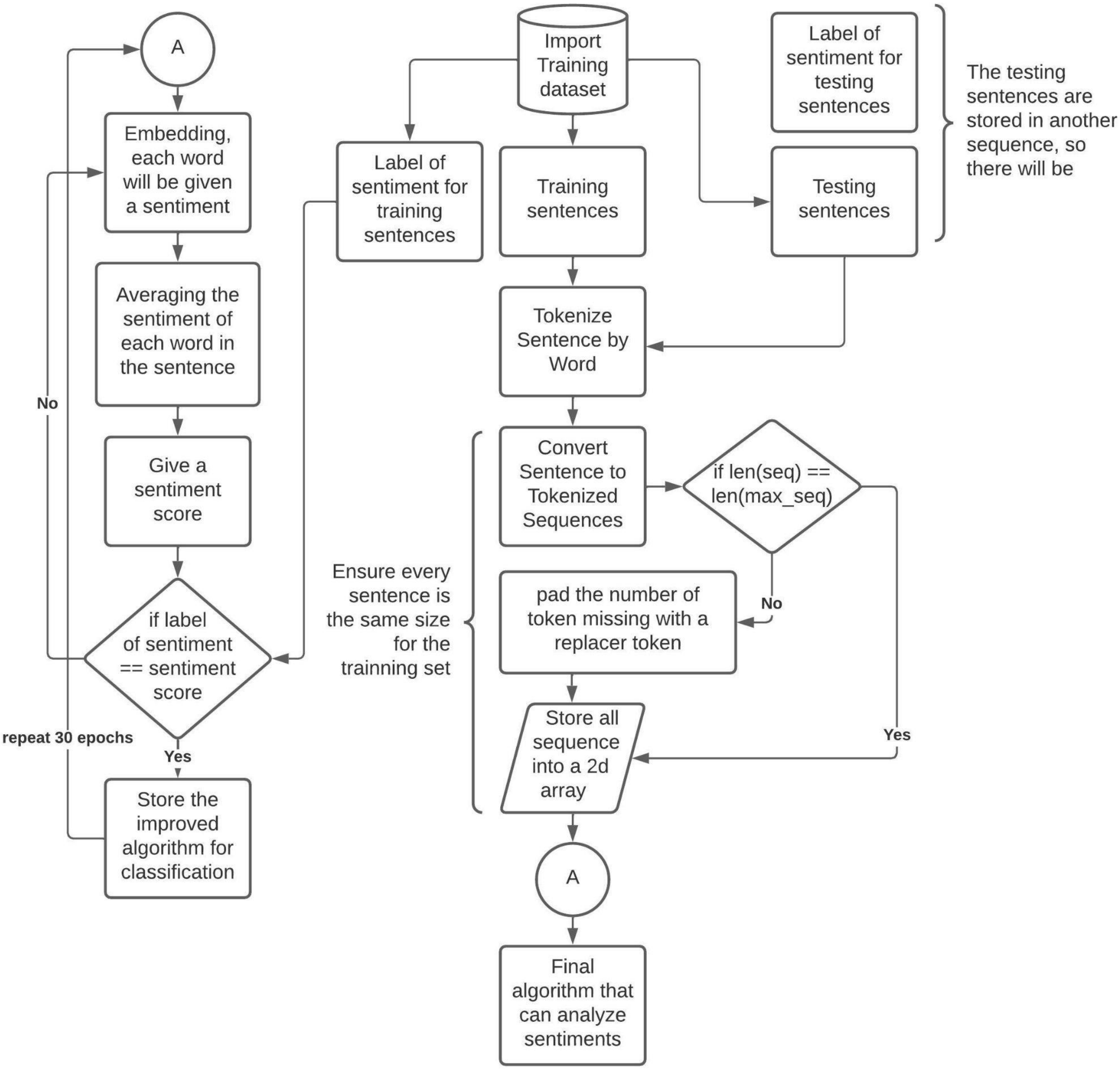

A
Embedding, each word will be given a sentiment
Averaging the sentiment of each word in the sentence
Give a sentiment score
if label of sentiment == sentiment score
No
repeat 30 epochs
Yes
Store the improved algorithm for classification
Import Training dataset
Label of sentiment for training sentences
Training sentences
Label of sentiment for testing sentences
Testing sentences
The testing sentences are stored in another sequence, so there will be
Tokenize Sentence by Word
Convert Sentence to Tokenized Sequences
if len(seq) == len(max_seq)
No
pad the number of token missing with a replacer token
Yes
Store all sequence into a 2d array
Ensure every sentence is the same size for the trainning set
A
Final algorithm that can analyze sentiments

Fig 02. Training Model

Fig 02 is a flowchart[2] illustrating the process of tokenization of words so that the neutral network can operate with numbers instead of characters.[3] Then, convert the token of terms into sequences of sentences. The neutral network prefers having the sequences of the same length, so through the padding, the sequence will be the same length. After the preparation, the padded sequences go into a neural network that trains with the training sentence and tests the testing sentences.

[2] "Intelligent Diagramming | Lucidchart". 2022. *Lucidchart*. https://www.lucidchart.com/pages/.

[3] "Text Classification With An RNN | Tensorflow". 2022. *Tensorflow*. https://www.tensorflow.org/text/tutorials/text_classification_rnn.

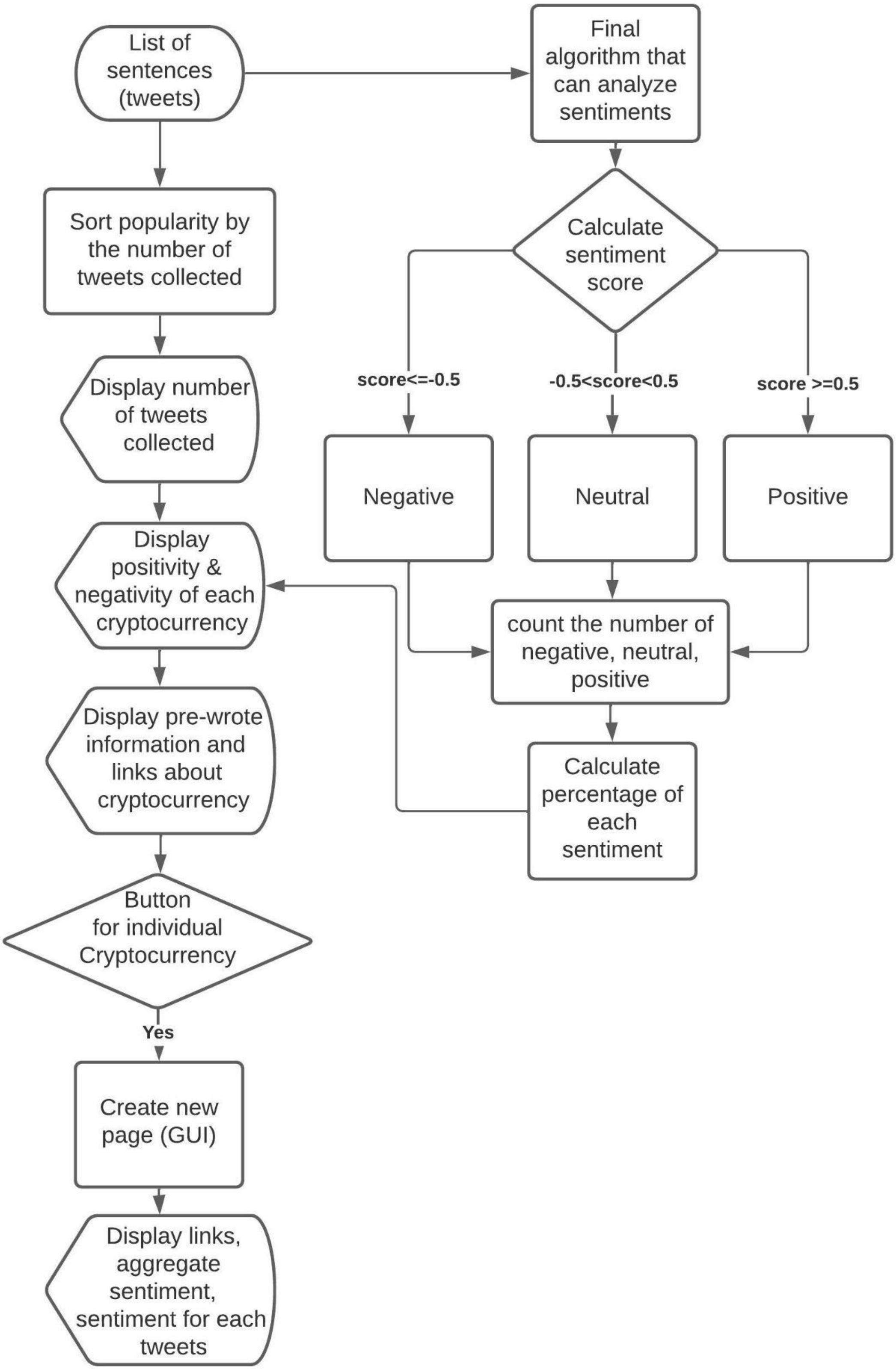

List of sentences (tweets)
Final algorithm that can analyze sentiments
Sort popularity by the number of tweets collected
Calculate sentiment score
score<=-0.5
-0.5<score<0.5
score >=0.5
Display number of tweets collected
Negative
Neutral
Positive
Display positivity & negativity of each cryptocurrency
count the number of negative, neutral, positive
Display pre-wrote information and links about cryptocurrency
Calculate percentage of each sentiment
Button for individual Cryptocurrency
Yes
Create new page (GUI)
Display links, aggregate sentiment, sentiment for each tweets

Fig 03. Analyzing and Displaying Sentiments

Fig 03 is a flowchart that illustrates how the application will use the algorithm in Fig 02 to analyze tweets, display general sentiments on the homepage, and in-depth analysis of individual pages.

Structure Diagram

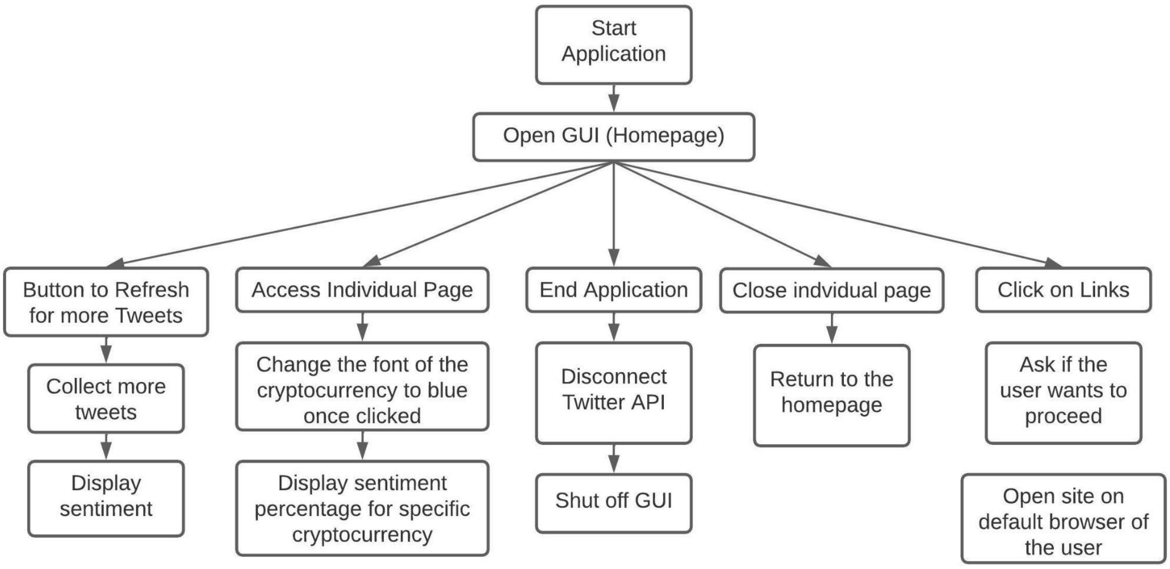

Start Application
Open GUI (Homepage)
Button to Refresh for more Tweets
Collect more tweets
Display sentiment
Access Individual Page
Change the font of the cryptocurrency to blue once clicked
Display sentiment percentage for specific cryptocurrency
End Application
Disconnect Twitter API
Shut off GUI
Close indvidual page
Return to the homepage
Click on Links
Ask if the user wants to proceed
Open site on default browser of the user

Fig 04. Structure Diagram of the homepage and individual pages

ER Diagram

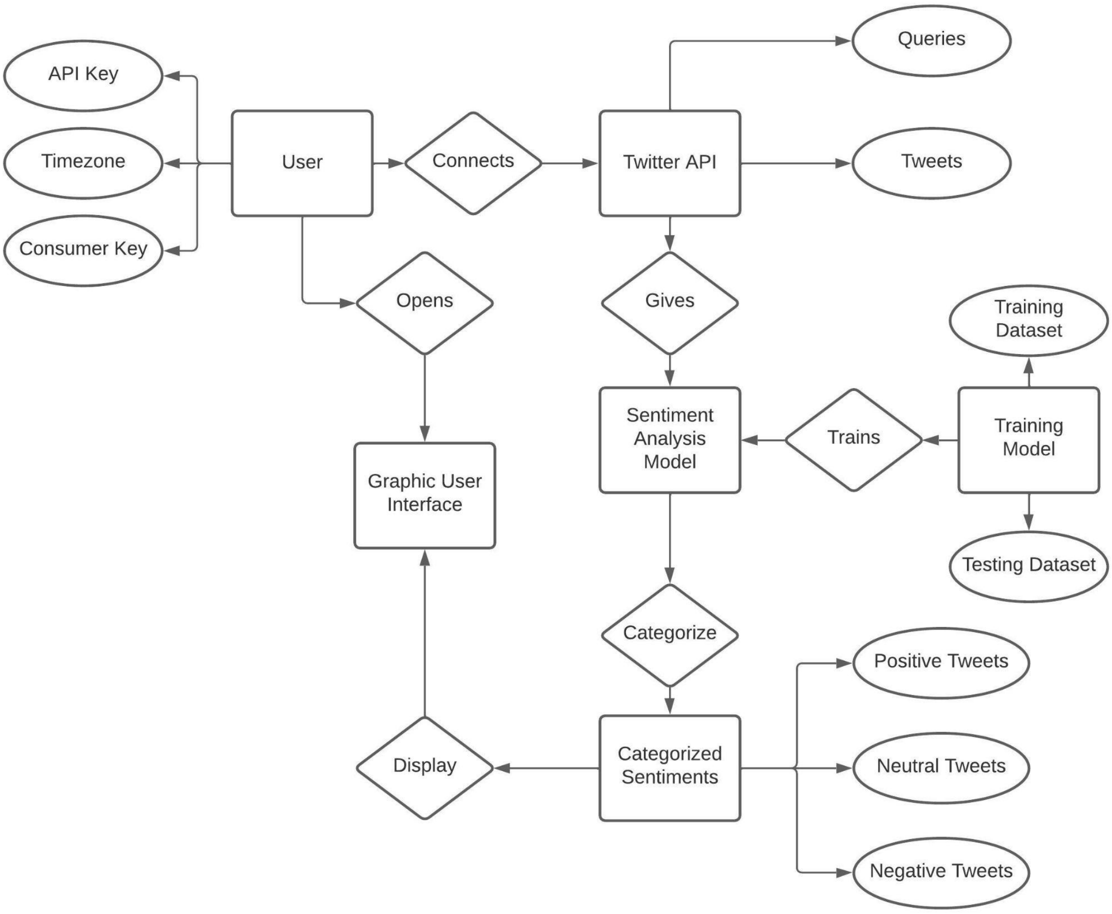


Fig 05. Entity-relationship

Fig 04 demonstrates the relationship between entities in this program and summarizes how the data will be collected, analyzed, and displayed.

Data Flow Chart

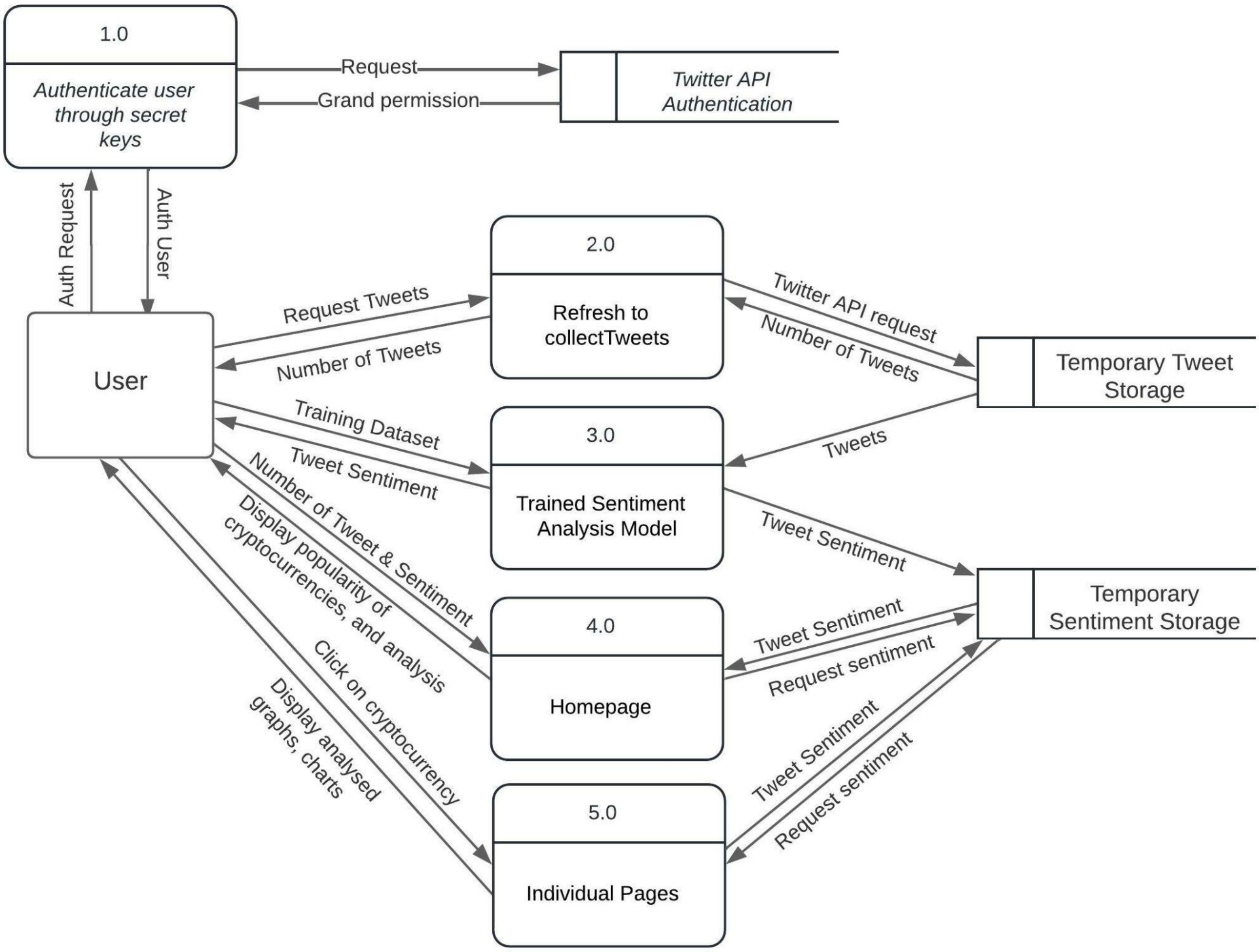


Fig 06. Data Flow chart of the user interface

UML Diagram

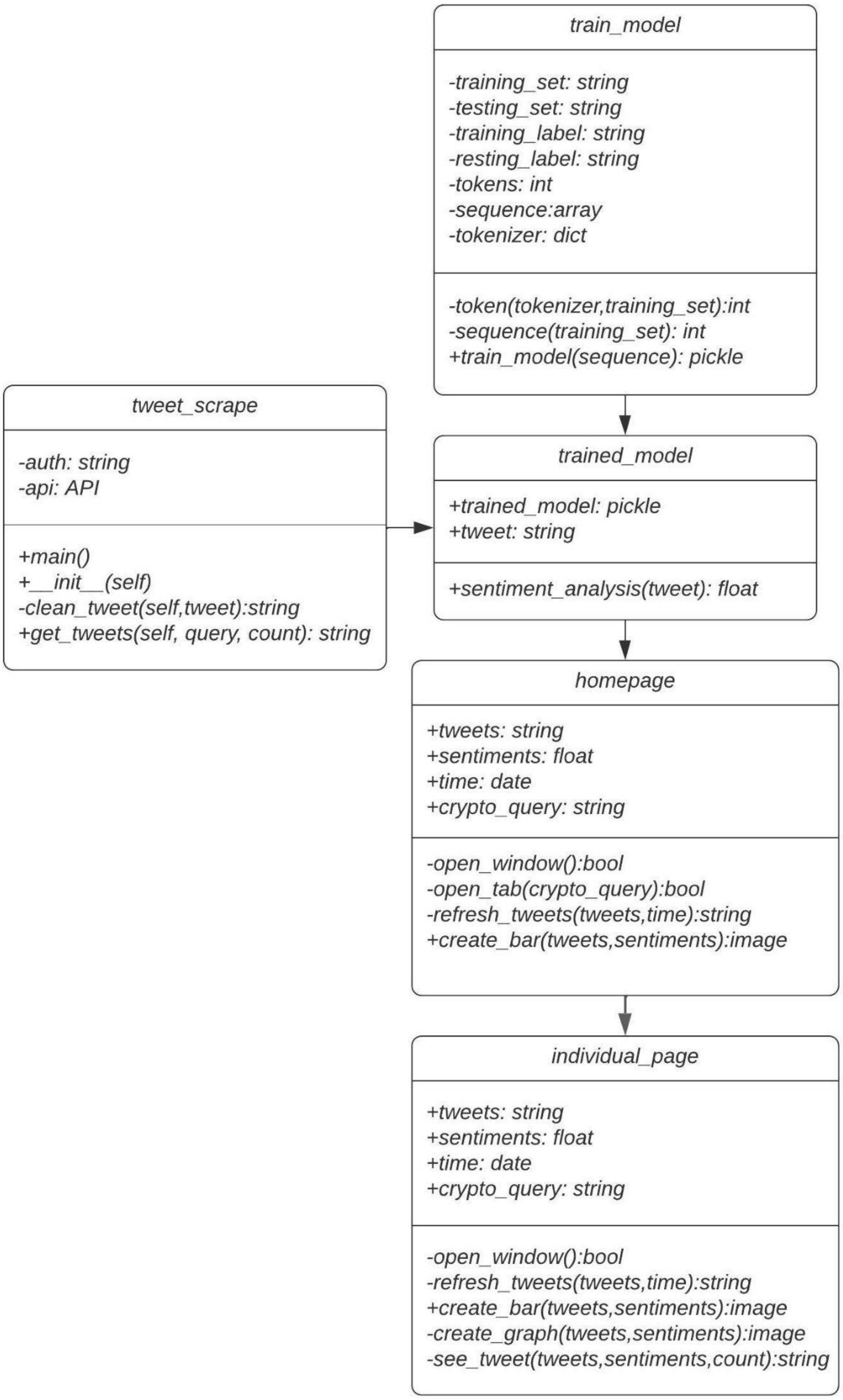
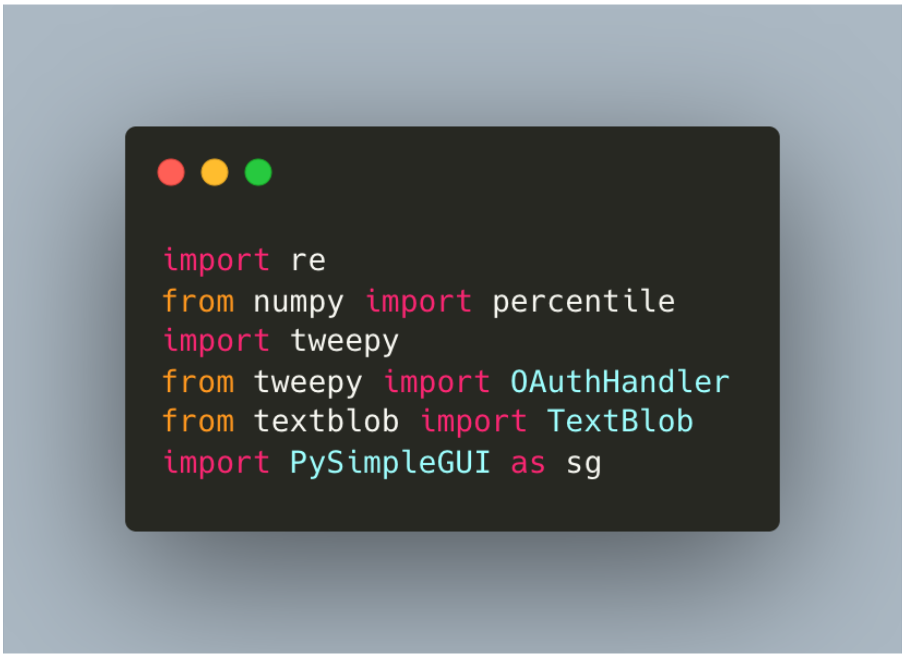

train_model
-training_set: string
-testing_set: string
-training_label: string
-resting_label: string
-tokens: int
-sequence:array
-tokenizer: dict
-token(tokenizer,training_set):int
-sequence(training_set): int
+train_model(sequence): pickle
tweet_scrape
-auth: string
-api: API
+main()
+__init__(self)
-clean_tweet(self,tweet):string
+get_tweets(self, query, count): string
trained_model
+trained_model: pickle
+tweet: string
+sentiment_analysis(tweet): float
homepage
+tweets: string
+sentiments: float
+time: date
+crypto_query: string
-open_window():bool
-open_tab(crypto_query):bool
-refresh_tweets(tweets,time):string
+create_bar(tweets,sentiments):image
individual_page
+tweets: string
+sentiments: float
+time: date
+crypto_query: string
-open_window():bool
-refresh_tweets(tweets,time):string
+create_bar(tweets,sentiments):image
-create_graph(tweets,sentiments):image
-see_tweet(tweets,sentiments,count):string

Fig 07. Class, attributes & functions

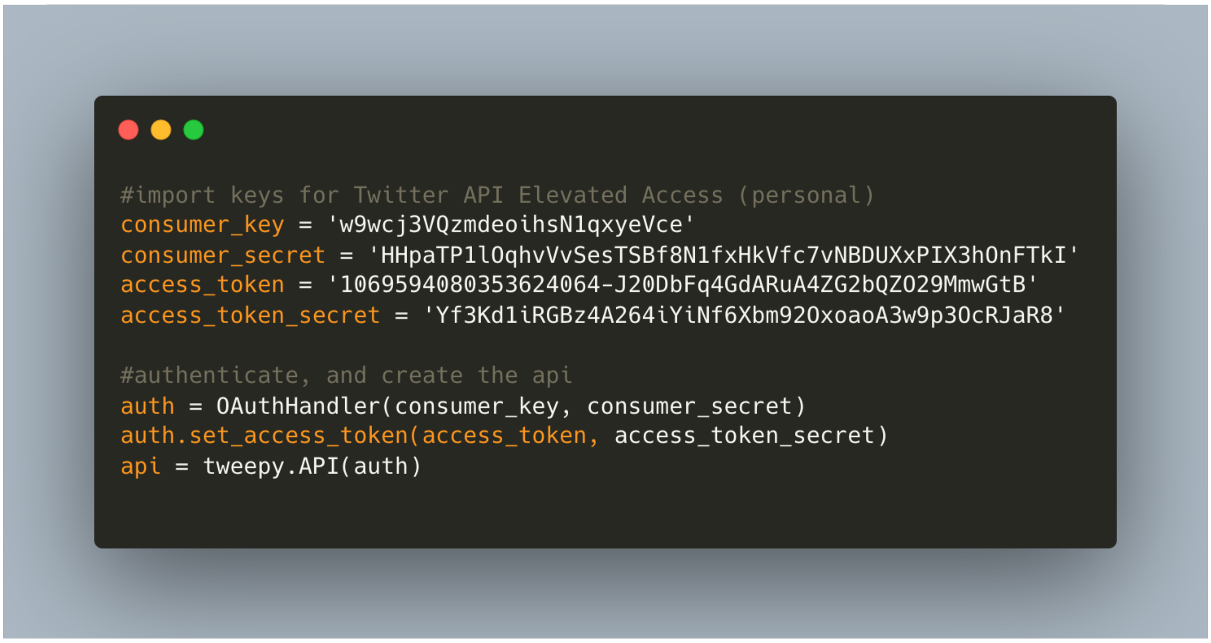
Prototype



```python
import re
from numpy import percentile
import tweepy
from tweepy import OAuthHandler
from textblob import TextBlob
import PySimpleGUI as sg
```

Fig 08. Import libraries

In the final product, the library TextBlob[4] will be replaced with the personally trained machine learning model.

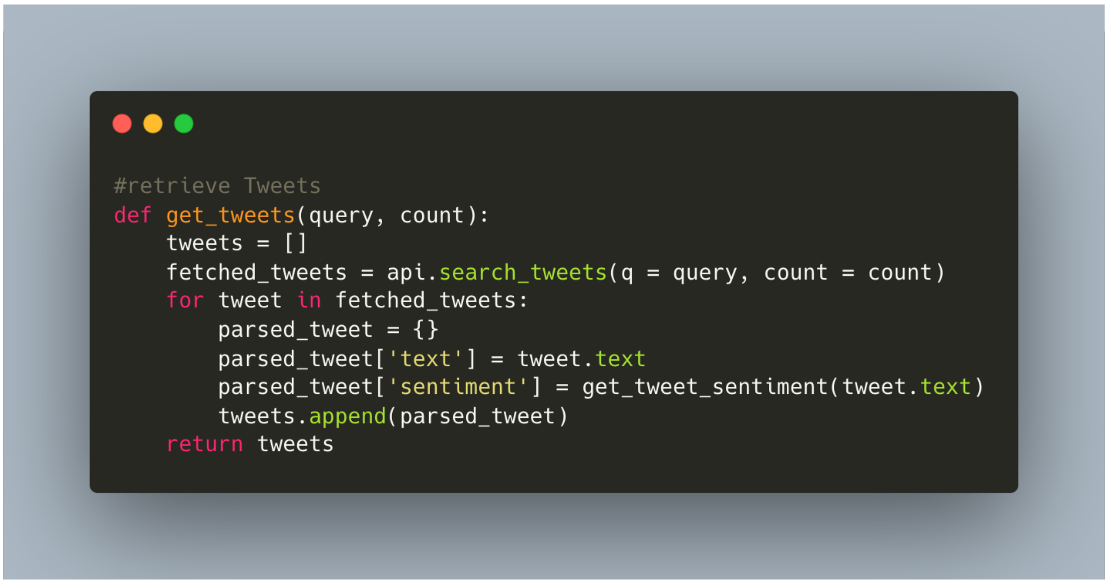


```python
#import keys for Twitter API Elevated Access (personal)
consumer_key = 'w9wcj3VQzmdeoihsN1qxyeVce'
consumer_secret = 'HHpaTP1l0qhvVvSesTSBf8N1fxHkVfc7vNBDUXxPIX3h0nFTkI'
access_token = '1069594080353624064-J20DbFq4GdARuA4ZG2bQZ029MmwGtB'
access_token_secret = 'Yf3Kd1iRGBz4A264iYiNf6Xbm920xoaoA3w9p30cRJaR8'

#authenticate, and create the api
auth = OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)
api = tweepy.API(auth)
```

Fig 09. Twitter API Connection

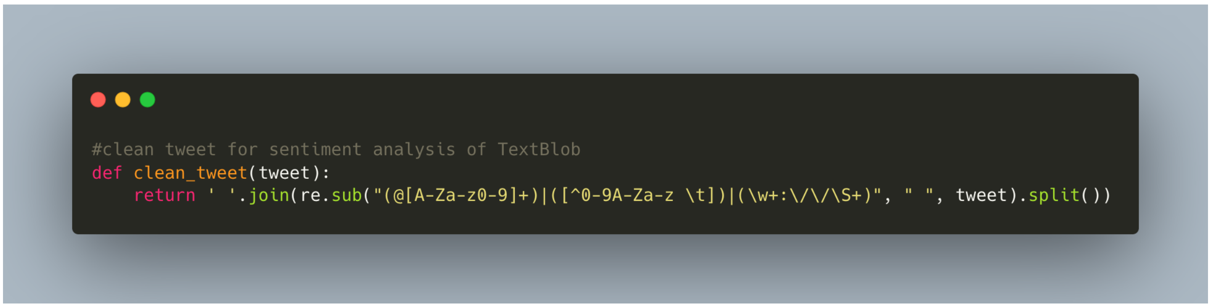
The libraries and keys in the final product, will be exported to the client through a text file for better package management.

[4] "Textblob: Simplified Text Processing — Textblob 0.16.0 Documentation". 2022. *Textblob.Readthedocs.Io*. https://textblob.readthedocs.io/en/dev/.



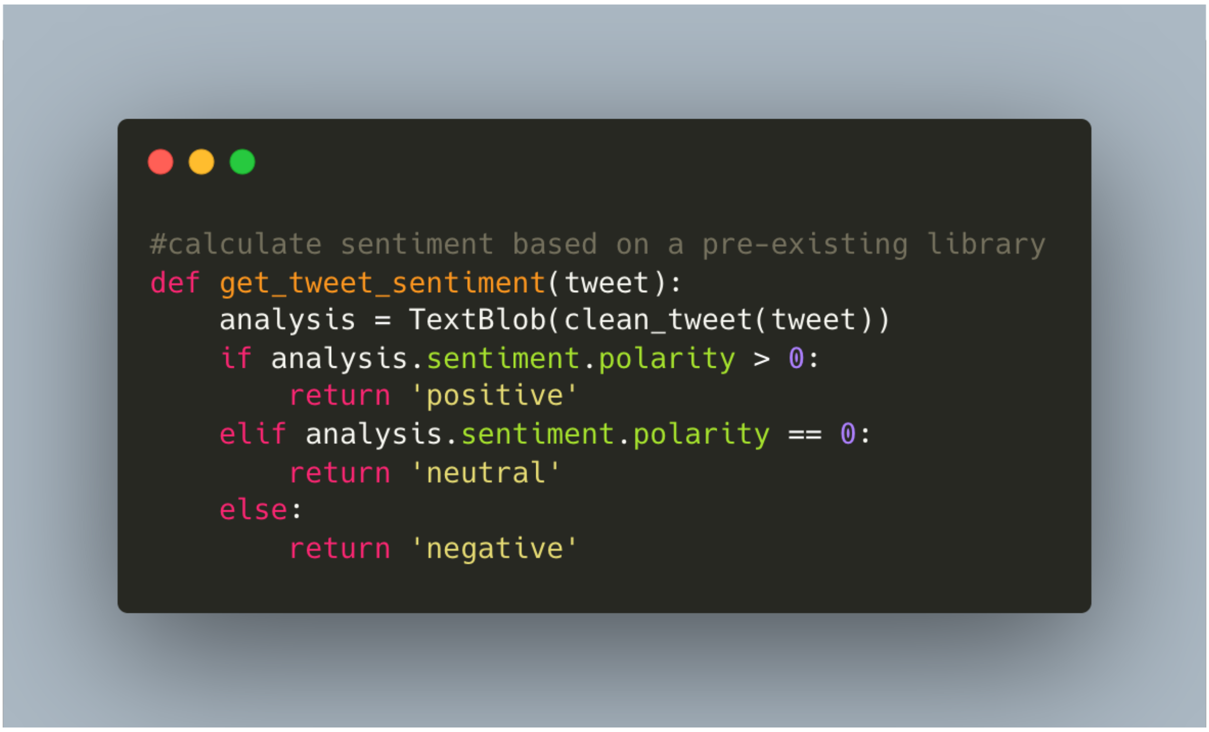
```
#retrieve Tweets
def get_tweets(query, count):
    tweets = []
    fetched_tweets = api.search_tweets(q = query, count = count)
    for tweet in fetched_tweets:
        parsed_tweet = {}
        parsed_tweet['text'] = tweet.text
        parsed_tweet['sentiment'] = get_tweet_sentiment(tweet.text)
        tweets.append(parsed_tweet)
    return tweets
```

Fig 10. Tweets Collection



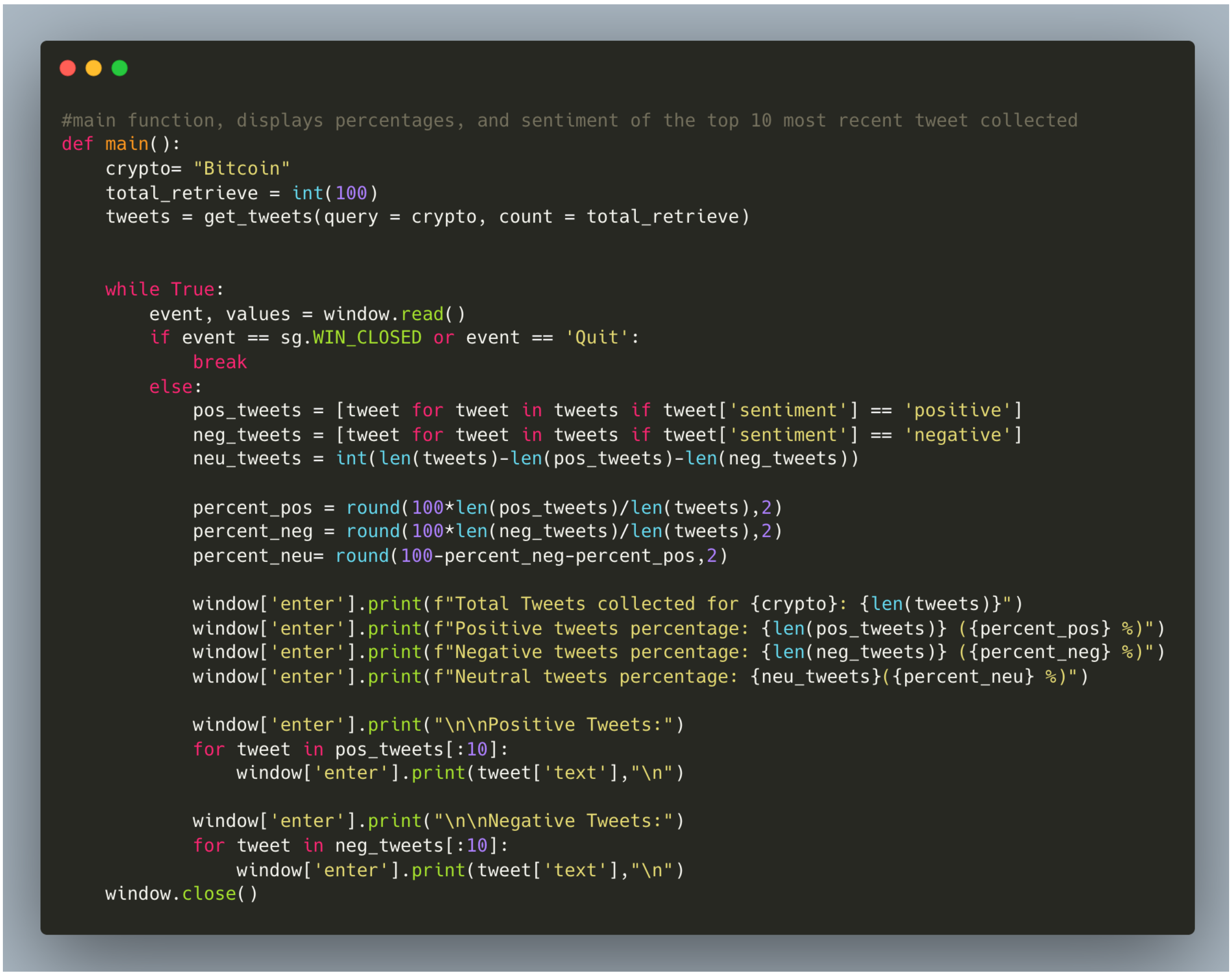
```
#clean tweet for sentiment analysis of TextBlob
def clean_tweet(tweet):
    return ' '.join(re.sub("(@[A-Za-z0-9]+)|([^0-9A-Za-z \t])|(\w+:\/\/\S+)", " ", tweet).split())
```

Fig 11. Tweet Cleanup



```
#calculate sentiment based on a pre-existing library
def get_tweet_sentiment(tweet):
    analysis = TextBlob(clean_tweet(tweet))
    if analysis.sentiment.polarity > 0:
        return 'positive'
    elif analysis.sentiment.polarity == 0:
        return 'neutral'
    else:
        return 'negative'
```

Fig 12. Sentiment Analysis using Textblob

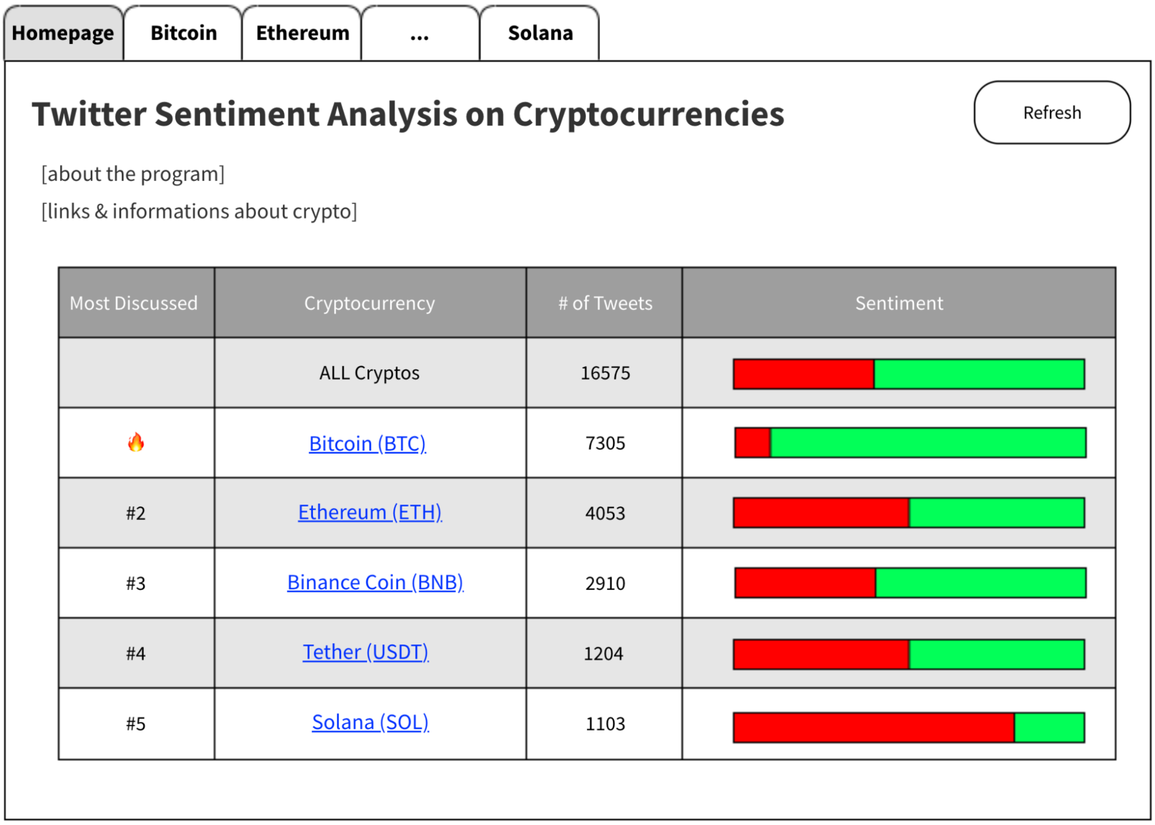


```python
#main function, displays percentages, and sentiment of the top 10 most recent tweet collected
def main():
    crypto= "Bitcoin"
    total_retrieve = int(100)
    tweets = get_tweets(query = crypto, count = total_retrieve)


    while True:
        event, values = window.read()
        if event == sg.WIN_CLOSED or event == 'Quit':
            break
        else:
            pos_tweets = [tweet for tweet in tweets if tweet['sentiment'] == 'positive']
            neg_tweets = [tweet for tweet in tweets if tweet['sentiment'] == 'negative']
            neu_tweets = int(len(tweets)-len(pos_tweets)-len(neg_tweets))

            percent_pos = round(100*len(pos_tweets)/len(tweets),2)
            percent_neg = round(100*len(neg_tweets)/len(tweets),2)
            percent_neu= round(100-percent_neg-percent_pos,2)

            window['enter'].print(f"Total Tweets collected for {crypto}: {len(tweets)}")
            window['enter'].print(f"Positive tweets percentage: {len(pos_tweets)} ({percent_pos} %)")
            window['enter'].print(f"Negative tweets percentage: {len(neg_tweets)} ({percent_neg} %)")
            window['enter'].print(f"Neutral tweets percentage: {neu_tweets}({percent_neu} %)")

            window['enter'].print("\n\nPositive Tweets:")
            for tweet in pos_tweets[:10]:
                window['enter'].print(tweet['text'],"\n")

            window['enter'].print("\n\nNegative Tweets:")
            for tweet in neg_tweets[:10]:
                window['enter'].print(tweet['text'],"\n")
    window.close()
```

Fig 13. The main function that displays analyzed information

In the final product, there will be a homepage, a specific page dedicated to each cryptocurrency, and display a visual representation of the data.

Graphical Illustration



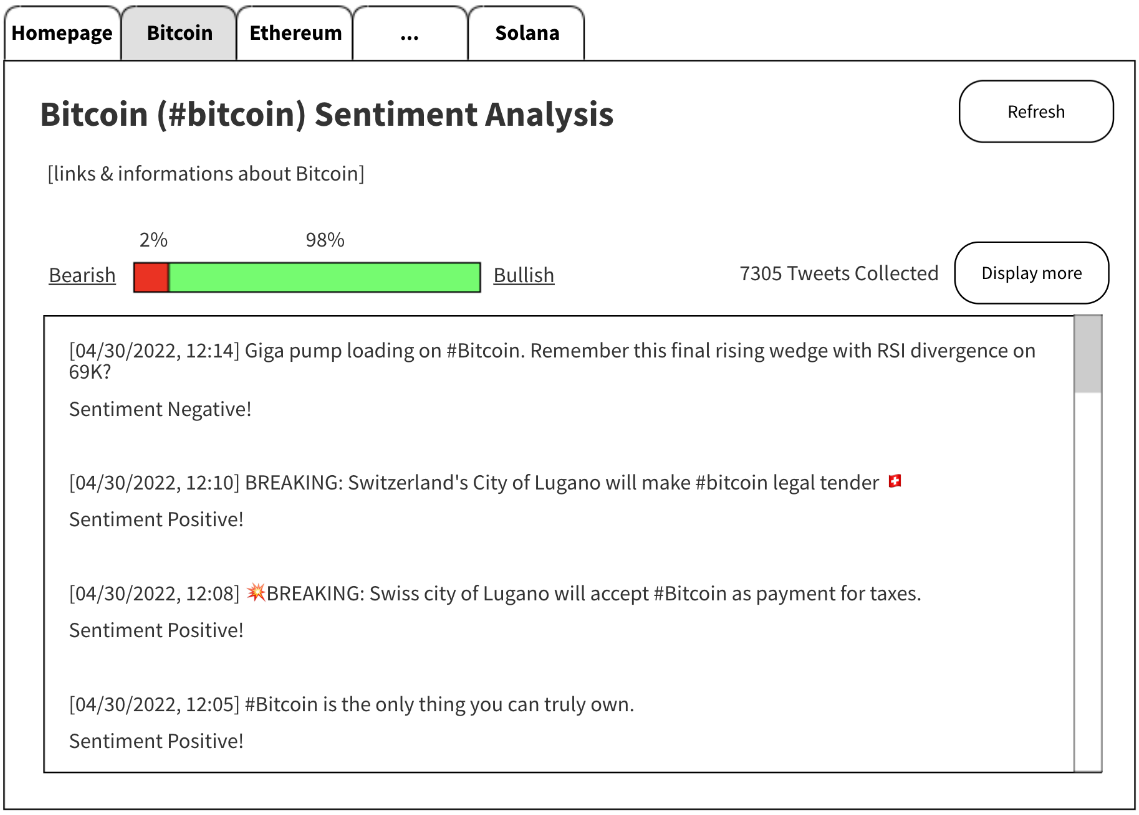
Twitter Sentiment Analysis on Cryptocurrencies

[about the program]

[links & informations about crypto]

Most Discussed	Cryptocurrency	# of Tweets	Sentiment
	ALL Cryptos	16575	🟥🟩
🔥	Bitcoin (BTC)	7305	🟥🟩
#2	Ethereum (ETH)	4053	🟥🟩
#3	Binance Coin (BNB)	2910	🟥🟩
#4	Tether (USDT)	1204	🟥🟩
#5	Solana (SOL)	1103	🟥🟩

Fig 14. Homepage



Bitcoin (#bitcoin) Sentiment Analysis

[links & informations about Bitcoin]

7305 Tweets Collected

[04/30/2022, 12:14] Giga pump loading on #Bitcoin. Remember this final rising wedge with RSI divergence on 69K?

Sentiment Negative!

[04/30/2022, 12:10] BREAKING: Switzerland's City of Lugano will make #bitcoin legal tender 🇨🇭

Sentiment Positive!

[04/30/2022, 12:08] 💥BREAKING: Swiss city of Lugano will accept #Bitcoin as payment for taxes.

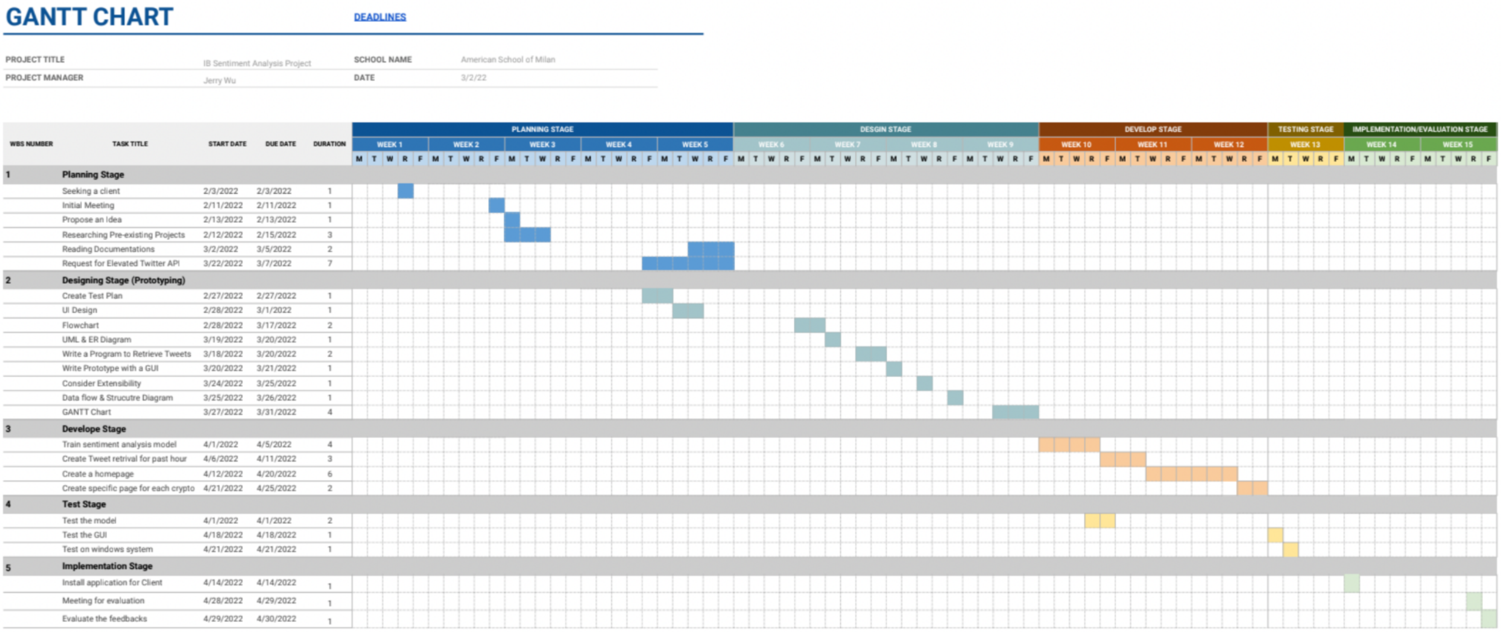
Sentiment Positive!

[04/30/2022, 12:05] #Bitcoin is the only thing you can truly own.

Sentiment Positive!

Fig 15. Individual Page (Bitcoin)

Gantt Chart



GANTT CHART

DEADLINES

PROJECT TITLE	IB Sentiment Analysis Project	SCHOOL NAME	American School of Milan
PROJECT MANAGER	Jerry Wu	DATE	3/2/22

WBS NUMBER	TASK TITLE	START DATE	DUE DATE	DURATION
1	**Planning Stage**			
	Seeking a client	2/3/2022	2/3/2022	1
	Initial Meeting	2/11/2022	2/11/2022	1
	Propose an Idea	2/13/2022	2/13/2022	1
	Researching Pre-existing Projects	2/12/2022	2/15/2022	3
	Reading Documentations	3/2/2022	3/5/2022	2
	Request for Elevated Twitter API	3/22/2022	3/7/2022	7
2	**Designing Stage (Prototyping)**			
	Create Test Plan	2/27/2022	2/27/2022	1
	UI Design	2/28/2022	3/1/2022	1
	Flowchart	2/28/2022	3/17/2022	2
	UML & ER Diagram	3/19/2022	3/20/2022	1
	Write a Program to Retrieve Tweets	3/18/2022	3/20/2022	2
	Write Prototype with a GUI	3/20/2022	3/21/2022	1
	Consider Extensibility	3/24/2022	3/25/2022	1
	Data flow & Strucutre Diagram	3/25/2022	3/26/2022	1
	GANTT Chart	3/27/2022	3/31/2022	4
3	**Develope Stage**			
	Train sentiment analysis model	4/1/2022	4/5/2022	4
	Create Tweet retrival for past hour	4/6/2022	4/11/2022	3
	Create a homepage	4/12/2022	4/20/2022	6
	Create specific page for each crypto	4/21/2022	4/25/2022	2
4	**Test Stage**			
	Test the model	4/1/2022	4/1/2022	2
	Test the GUI	4/18/2022	4/18/2022	1
	Test on windows system	4/21/2022	4/21/2022	1
5	**Implementation Stage**			
	Install application for Client	4/14/2022	4/14/2022	1
	Meeting for evaluation	4/28/2022	4/29/2022	1
	Evaluate the feedbacks	4/29/2022	4/30/2022	1

Fig 16. Gantt Chart

Planned schedule for the project.