School of Computer Sciences

CAT400 Undergraduate Major Project

Final Report

MW20210153: WHATSON: Time Series Analysis of Social Media Content

MUHAMMAD ADAM FIKRI BIN ANUAR

137109

Supervisor: Dr. Noor Farizah Ibrahim

Examiner 1: Dr. Hazrina Yusof Hamdani

Examiner 2: Dr. Sukumar Letchmunan

Academic Session

2020/2021

DECLARATION

"I declare that the following is my own work and does not contain any ***unacknowledged*** work from any other sources. This report was undertaken to fulfill the requirements of the Undergraduate Major Project for the Bachelor of Science in Computer Science (Honors) program at Universiti Sains Malaysia".

Signature : 

Name : Muhammad Adam Fikri Bin Anuar

Date : 18th June 2021

ABSTRAK

Media sosial ialah platform untuk pengguna berkongsi kandungan dan telah menjadi sebahgian daripada kehidupan kita. Media sosial mempunyai pelbagai kegunaan seperti hiburan, berkongsi maklumat, menjalankan perniagaan dan melakukan kajian. Ini membuatkan media sosial dianggap sebagai tempat data besar. Belakangan ini, kebanyakan syarikat sedang bergelut untuk menadapatkan gambaran keseluruhan daripada data media sosial disebabkan isu data seperti maklumat terlalu sarat dan kepayahan untuk memahami dinamik tren di social media. Mereka mencari penyelesaian pada masa hadapan. WHATSON ialah peralatan data siri masa Twitter. Kepayahan pengguna memperoleh gambaran keseluruhan dari Twitter mencetus kepada idea project ini. Objektif projek ini adalah untuk menghasilkan laman sesawang yang menyediakan analisis siri masa media sosial Twitter terdiri daripada ciri-ciri pengenalpastian sektor, analisis sentimen dan analisis topic. Di akhir projek ini, laman sesawang dilengkapi analisis siri masa dan pangkalan data Twitter dihasilkan. Laman sesawng akan dihasilkan dalam HTML, CSS, JavaScript, PHP dan Python, dan menggunakan MySQL sebagai pangkalan data.

Kata kunci: Analisis Siri Masa, Ramalan Topik, Media Sosial, Data Besar

ABSTRACT

Social media is a platform for users sharing content and has become part of our life. Social media has many uses such as entertainment, sharing information, run a business or conduct research. This makes social media a place of big data. Nowadays, a lot of companies are struggling with gaining insight from social media data due to data issues such as information overload and difficulties understanding the dynamic trend on social media. They are seeking for solutions in the future. WHATSON is a time series analysis tool of Twitter. The difficulties of users to gain insight from Twitter leads to this project idea. The objective of this project is to develop a website that provides time series analysis of Twitter content consists features of sector identification, sentiment analysis and topic analysis. At the end of this project, a website with time series analysis and a database of Twitter data will be developed. The website will be developed in HTML, CSS, JavaScript, PHP and Python, and use MySQL as a database.

Keywords: Time Series Analysis, Topic Prediction, Social Media, Big Data

ACKNOWLEDGEMENTS

I would like to express my gratitude to God for making it possible for me to complete this analysis report especially during in the midst of Covid-19 pandemic. Next, a heartfelt gratitude I want to state to my supervisor, Dr. Farizah Ibrahim. Her knowledge in time series analysis and advises are matters in delivering a good quality final report. I also would like to express my gratitude to my examiners, Dr. Sukumar and Dr. Hazrina, for having their time in examining my project and giving feedback to improve my project. Their feedback is important in delivering a project that meets its demand. Next, to my friend, I want to say thank you for their belief and support during making this report. Last but not least, not to forget my own family, I am grateful to have them. They are supporting me from home during making this report.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS AND SYMBOLS

API - Application Programming Interface

TSA - Time Series Analysis

NLP - Natural Language Processing

ARIMA - Autoregressive Integrated Moving Average

SVM - Support Vector Machine

ML - Machine Learning

BERT - Bidirectional Encoder Representations from Transformer

HTML - Hypertext Markup Language

CSS - Cascading Style Sheets

WBS - Work Breakdown Structure

SWOT - Strength, Weakness, Opportunity and Threat

SSD - Sequence State Diagram

LDA - Latent Dirichlet Allocation

AR - Auto Regressive

MA - Moving Average

MAPE - Mean Absolute Percentage Error

1 INTRODUCTION

This chapter will give an overview of this project. Here, the background, problems, objectives, and other important details will be described. At the end of this chapter, the overview of the rest of the other chapters will be described.

1.1. Background

Time series is an arrangement of data points in time order (hourly, daily, weekly etc.). The characteristics of data points are continuous, same distance of time interval between two consecutive data, each time unit in the time interval has at most one data point and consists of successive measurements made over a time interval [1]. The instances of time series data are stock price, website traffic, temperature, quarterly sales and more. Time series analysis is a way to analyze time series to gain insight/overview/deep understanding [1]. The component can be observed in time series are trend (direction of time series), seasonality (predictable pattern), irregular fluctuation (sudden change in direction) and more. There is a crucial task in time series analysis comprise of modelling, forecasting, clustering and change detection [1].

Social media is a very well-known platform that runs both on a website and mobile application which allows users to create and share their contents with the public. Nowadays, social media has been used for a lot of purposes such as sharing information, entertainment and running businesses. Example of social media is Facebook, Twitter, Instagram, TikTok and Spotify. A lot of activities can be engaged in social media other than creating and sharing contents. All these activities are recorded. Therefore, social media can be a place of big data as it keeps various data (text, video, audio and picture) and in high volume.

The project named WHATSON: Time series analysis of social media content is a webpage to analyze Twitter data using a combination of time series analysis, sentiment analysis and topic modelling. It aims to help the users such as researcher, data analysts and social media marketer to gain more insight from Twitter. It helps to overcome the data issues on social media such as information overload and

understanding the dynamic trend of social media. What makes it is unique is that the system will have a function to predict topics that might appear in the future.

1.2. Problem Statements

The problem that concerns in this project are:

Problem 1: Difficulties to gain insights of social media data

Social media data is naturally high in volume. The ability to gain a deep understanding or insight of a thing is difficult to possess especially with data. With a great understanding of data, someone can execute the best action. According to [4], based on a survey conducted by Deloitte in 2018, a lot of companies are searching for means to solve their data issues in the future and only 33% of them can find the meaning of data that could help to achieve their goals. This means 67% of the companies are still struggling to gain insight into the data.

Problem 2: The importance of topic prediction

From a business perspective, marketing in social media is important as social media has become part of our life. According to [2], 71% of shoppers who have had a decent social media service experience in a brand are probably going to prescribe it to other people. Therefore, the marketers need to have creative ideas or inspirations to deliver a good and more advance content to the audience. One of the possible ways is to exploit trendy topics whenever necessary in order to create “contagious” content. This shows that there may be a need for topic prediction. Topic prediction is not just important for business only but can be applied in research to observer topics or themes from an important event such as election and Olympics.

1.3. Motivation

According to statistics given in [3], the number of social media users in 2019 is about 45% (3.5 billion) of the world population and time spent on social media per day in 2019 is an average of 3 hours. Well, those statistics are in 2019 and it is possible that there will be an increase in those numbers as the pandemic hit the world in 2020. This means the volume of data will be a lot and the users such as researchers, data scientists and social media marketers might lose the insights due to a rapid increase volume of data. This problem has been faced by many companies since 2018 as mentioned in [4]. This leads to this project idea to create a time series analysis of social media with a new feature hoping that the users can gain insights of data from other perspective.

1.4. System Objectives

The main objectives of this project are:

1. To identify the trend of domain areas that people often talk about on social media and help the users gain an understanding of tweets based of its sectors. (Sector Identification Module)
2. To examine the sentiment and users' perceptions of the tweets that indicate public opinion of whether they are positive, negative or neutral. (Sentiment Analysis Module)
3. To display the time series of current topics and perform topic prediction (uniqueness of this project). This could help (1) to predict and understand users experience, (2) businesses to plan ahead for their operations and (3) brands to fix the reputation. (Topic Analysis Module)

1.5. Proposed Solutions

This project will develop **a website** that runs time series analysis on Twitter content. The features including but **not limited to** sector identification, sentiment analysis and topic analysis. The explanation of features is as below.

Tweets Management

This feature will fetch and keep all the tweets in the database. It utilizes Twitter API to collect information needed from the tweets such as the text and time. This feature will do the tasks automatically every day.

User Management

This feature will enable users to have an account. The users can create account, login account and logout.

Sector Identification

This feature will give insights of the tweets based on the sector of tweets. This feature will classify the text of tweets into few sectors such as news, entertainment, sports, politics and others. Then the time series trends of the sectors are plotted and able to make prediction for the next few weeks/months. This feature also provides popular topics for each sector.

Sentiment Analysis

This feature will classify the sentiment of the tweets. This will identify the sentiment of tweets whether the tweets are positive, negative, or neutral. Then the time series for all sentiments will be plot and predict sentiment for the next few weeks/months. This feature also provides popular topics for positive and negative sentiments.

Topic Analysis

This feature will give insights of topics that are currently popular among Twittersphere. This feature will identify current popular topics on Twitter. Then the popular topics will show up and the link of news related to the current topics will be

provided. This feature also can make a forecast of the longevity of the topics that will be popular in the next few weeks/months.

The overall modules are shown in Figure 1.5.1 and Figure 1.5.2 until Figure 1.5.4 show wireframes of the proposed solution.

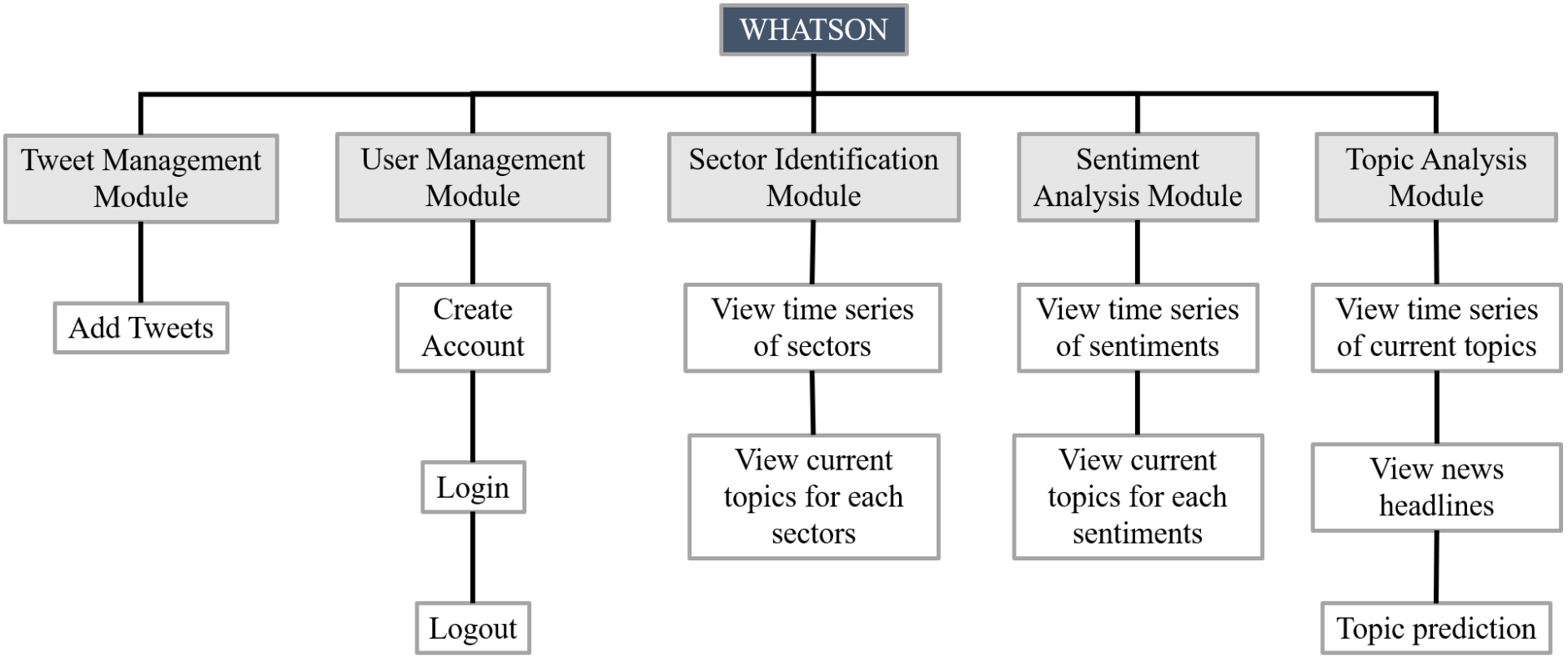


Figure 1.5.1 Overall Module of WHATSON

Figure 1.5.2 Wireframe of Sector Identification Module

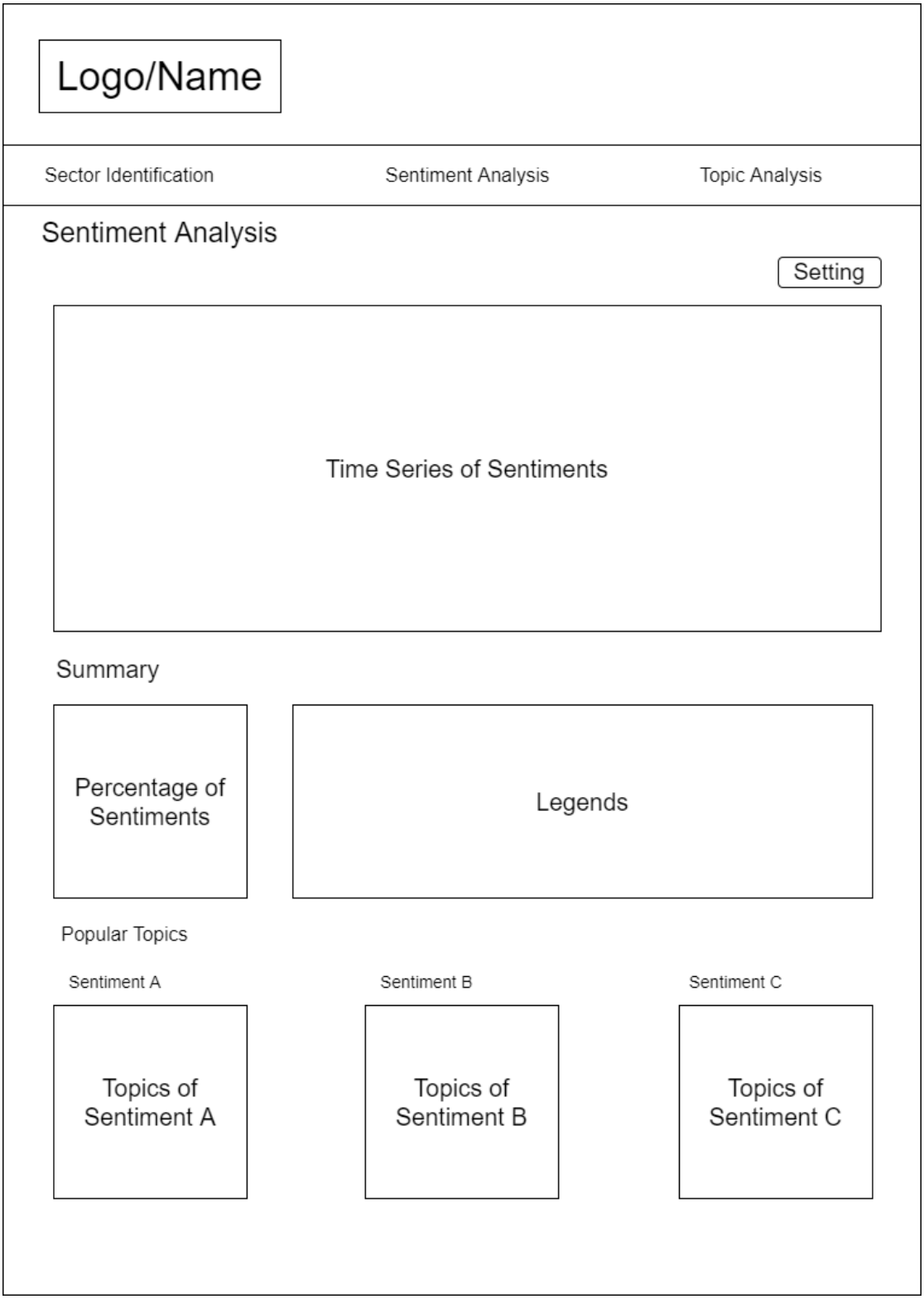


Figure 1.5.3 Wireframe of Sentiment Analysis Module

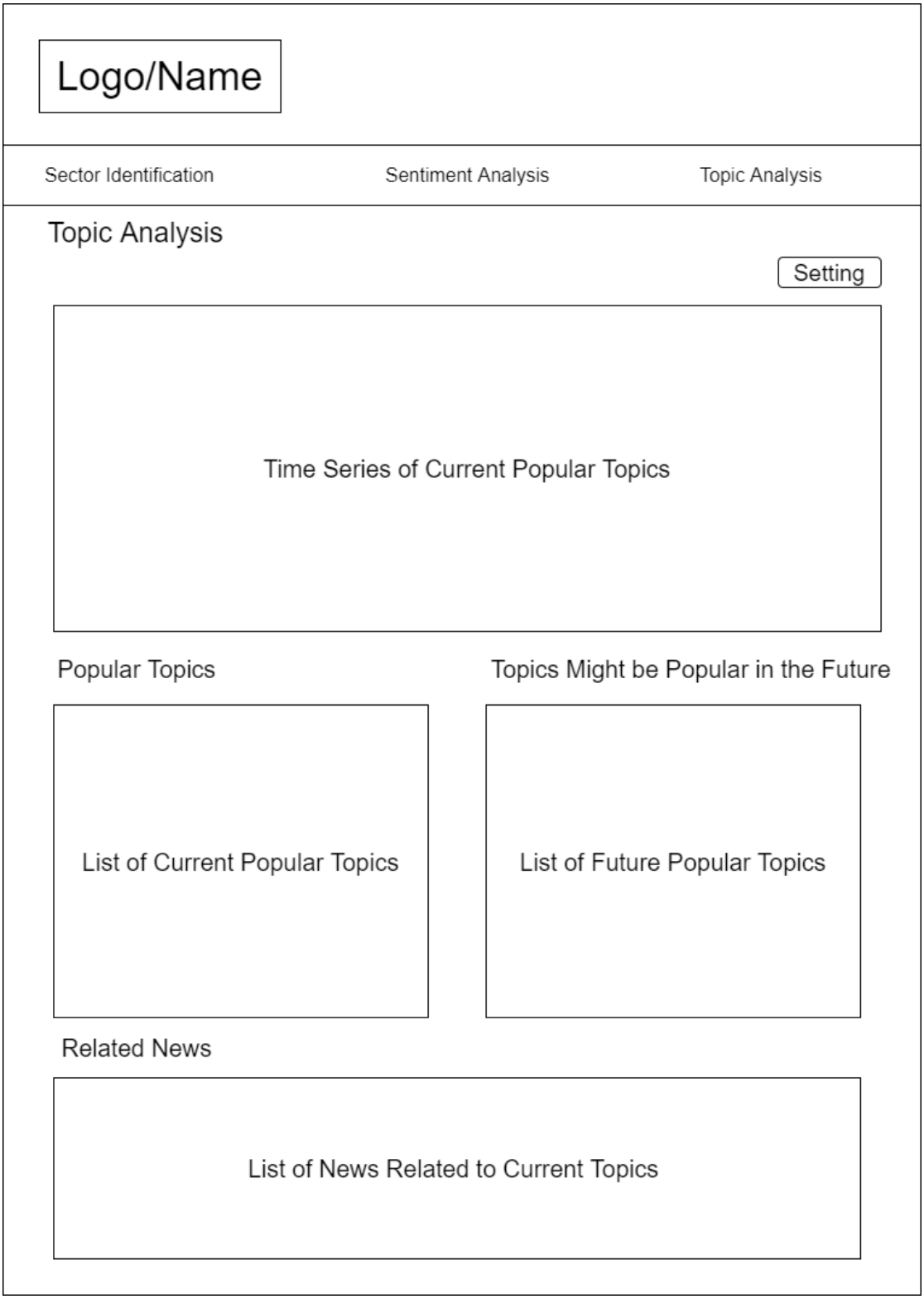


Figure 1.5.4 Wireframe of Topic Analysis Module

1.6. Benefits and Uniqueness of the Proposed Solutions

This social media time series analysis would be beneficial for users such as researchers, data scientists, social media marketers and others). Some of the benefits include:

1. Provide topics that can be trendy in the future.
2. Able to plan for future campaigns (for marketing, politics, customer segmentation etc.).
3. Allow users to get updated with current topics.

The uniqueness of this project is the topic prediction feature. This feature will identify viral topics among Twittersphere and predict the topics people will talk about in the future. There is no other platform that provides topic prediction feature. For example, in [5], the Brandwatch provides top topic feature but does not extend it to predict future topics. Another thing that makes this project unique is that it is exclusive to Twitter.

1.7. Expected Outcomes

In the end, this project will deliver two components:

1. A website

   The website will be developed in HTML, CSS, JavaScript, PHP and Python that have features comprise of sector identification, sentiment analysis and topic analysis.

2. Database

   A database that stores data from Twitter. The database used is MySQL.

1.8. Organization of the Report

This report comprises of 6 chapters namely, Introduction, Background & Related Work, System Requirements/Analysis, System Design and Implementation, System Testing and Evaluation, and Conclusion & Future Work. All this chapter has covered all core processes which consist of Identify Problem & Obtain Approval, Plan the Project, Discover & Understand Details, Design System Components, Build, Test and Integrate System Components, and Complete System Tests and Deploy Solution.

The first chapter provides the big picture of this project. Few details have been described such as Background, Problem Statements, Motivation, Objectives, Proposed Solution, Benefits and Uniqueness and Expected Outcomes. This chapter set the tune for the rest of the chapter.

Chapter 2 is the study of the domain. The existing systems same as the proposed solution (WHATSON) have been identified. Their strength and weakness of existing systems and proposed solution have been compared. The findings from the literature review have been explained.

Next, chapter 3 is the analysis of the requirements of WHATSON. The scope of the project, capabilities and limitations of the system have been identified. Project planning such as WBS, Gantt chart and SWOT analysis have been created. Few diagrams that describe the system have been created such as Overall Module Diagram, Use Case Diagrams, Use Case Descriptions, Wireframes. Details requirement, development methodology and technology deployed have been identified.

Fourth chapter is the designing and implementation of WHATSON. Few diagrams have been designed in order to develop the system. The architecture of the system has been well sketched with a lot of details such as the components involved and the flow of front end and back end processes. Design Class Diagram and Entities Relationship Diagram have been designed to help in understanding details and methods needed in database and the system. Package Diagram also described the file needed in the system. The implementation strategy also discussed in this chapter which helped in developing each module.

Chapter 5 explained types of system testing, how the testing is conduct, the results of each test. Each test comes different results. The test is not just focus on functional capabilities but also non-functional capabilities such as performance, usability and security. Problem faced during testing also explained in the last two sections of this chapter.

Last but not least, Chapter 6 provides conclusion and future work. It summarized the report and reflected of the flaws in the project. It also included some hopes and actions that could be accomplish in the future.

2 BACKGROUND & RELATED WORK

The study of the background and related work is the essence of understanding the domain of this project. This chapter will provide insight of the domain of this project. Information gathering technique has been conducted are literature review.

2.1. Status of the Project

This project status is **enhancement**. There is a lot a platform out there that provide social media analytics. For example, Google Analytics and Hootsuite. The thing that makes the difference is the topic prediction feature where the system can predict popular topics in the future.

2.2. Existing Systems

Few existing systems that work the same with the proposed solution has been identified:

1. Hootsuite

   Hootsuite is founded in 2008 by Ryan Holmes is a social media management tool. Hootsuite provides a dashboard that integrates with Facebook, Twitter, Instagram, YouTube and Pinterest. Based in Vancouver, Canada and has more than 16 million users over 175 countries [6].

Figure 2.2.1 Hootsuite Logo

2. Google Analytics

   Google Analytics is the most widely used web analytics tools provided by Google that tracks and reports website traffic. It launched 15 years ago on 14th November 2005 after acquiring Urchin Software Corp. in April 2005 [7].

Figure 2.2.2 Google Analytics Logo

2.3. Strength and Weakness of Existing Systems

Both competitors, Google analytics and Hootsuite are well-known platforms for social media analytics and have been developed a long time ago, 2005 and 2008, respectively. Hootsuite is a good social media analytic platform that personalized for a social media marketer while Google analytics is a platform that personalized for traffic tracking for social media and websites.

Although Google Analytics has been developed a long time ago, it still not providing topics analysis and sentiment analysis feature inside it. Google does it in a different way. For topic analysis, Google made another platform called Google Trends. For sentiment analysis, Google provides an API for it.

Below shows the comparison between the proposed solution, WHATSON and other platforms.

Platform	**Hootsuite**	**Google Analytics**	**WHATSON**
Social media can be integrated	Facebook, Twitter, Instagram, YouTube, LinkedIn, Pinterest	Facebook, Twitter, Instagram, YouTube, LinkedIn, Pinterest and more	Twitter only
Sector Analysis	✓	✓	✓
Sentiment Analysis	✓	✗	✓
User Activity Tracking	✓	✓	✗
Topic Analysis (trend)	✓	✗	✓
Topic Prediction	✗	✗	✓

Table 2.3.1 Comparison Between WHATSON and other platforms

The tick (✓) represents the feature is provided while the cross (✗) means the feature is not provided.

2.4. Literature Review

This is the only information gathering method conducted. This method helps to understand the domain and related topics. It can help in gathering the detailed requirements of the system that will be developed. Below show the findings from this method.

2.4.1. Time Series Analysis

According to [8], the author of 'Introduction of Time Series Modelling', Genshiro Kitagawa, has defined time series data as:

> *"A record of phenomenon of irregularly varying with time is called time series."*

This means time series data is a set of data taken by time order (yearly, monthly, daily, hourly, etc.). Few characteristics of time series data has been mentioned by [1], time series is an arrangement of data points that consists of successive measurement made over a time interval, the time interval is continuous, the distance in the time interval between any consecutive time is same and each time unit in time interval has at most one data point. Below is the example of time series shows in [9].

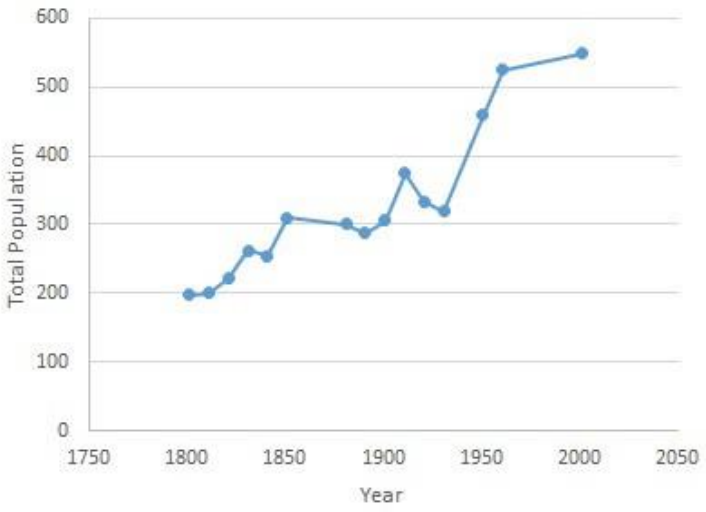


Figure 2.4.1 Example of Time Series in [6]

There are few components of time series that is observed in [9], namely:

- Trend

  Trend basically is the direction of the time series is developing. The direction of the can be developed to upward (uptrend) or downward (downtrend). The trend of a time series is not always developing in a consistent direction based on the given time period.
- Seasonality

  Seasonality is the predictable pattern that repeats in given time intervals. It usually can be observed in a year or less. For instance, the temperature is the lowest during winter.
- Cycle

  Cycle is an upward and downward pattern that is not seasonal. Cycle is usually observed in business. It has variations such as peak, recession, trough/depression and expansion.

Time series data can be seen as meaningless data at first. With time series analysis (TSA) the data can be transformed into something valuable. Time series analysis is the method of analyzing time series data with the purpose of extracting meaningful statistics and other characteristics of data [1,9]. This method is very useful in a lot of sectors, especially in business. It transforms data into actionable insight for business problem solving and competitive advantage.

TSA consists of few main jobs that can be done with the help of machine learning or certain algorithm. These jobs are:

- Modelling

  Modelling in a simple word is a process of describing the data. This method analyzes the time series and find a way to describe it. This method is also associated with regression in most cases [1]. This method is done before forecasting. There are 11 regression methods mentioned in [10]. Most of the methods use same Python package *statsmodel.tsa*.

- Forecasting

  Forecasting is the process of using a model to predict future values based on previous values [9]. The Python package *statsmodel.tsa* provides function to forecast. According to [1], the forecasting works the best in Autoregressive Integrated Moving Average (ARIMA) models.

- Classification

  Classification is the process of allocating the data to a predefined label [1]. It is a supervised machine learning approach. This means that the training data is labeled. Support Vector Machine (SVM) is an example of a conventional classification method stated in [1].

- Clustering

  Clustering is the process of grouping the data into certain group/cluster based on its features [1]. It is an unsupervised machine learning approach. The difference between classification and clustering is the presence of the label. Clustering does not need an annotation or data label.

2.4.2. Text to Data

This topic is covered due to the need of analyzing text in the tweets. The text must be converted into something tangible and meaningful. The is where Natural Language Processing (NLP) is needed. Here cover the process of text processing/mining explained in [11,12]. Most of the processes will use Python package NLTK (Natural Language Tool Kit)

Tokenization

Tokenization is the first process need to be done. This process is breaking down text string into a small unit called token. The next process will be stop words removal.

Stop Words Removal

Stop words can be described as the ‘noise’ in data because it is common and meaningless. These words must be removed. Example of stop words includes articles (a, an, the), conjunctions (and, or) and prepositions (with). Python package *nltk* has a list of 200 stop words.

Punctuation Removal

Punctuation in text such as symbols will not give any meaning to the text. This can be removed by using Regular Expression (Regex). In the end, the list will consist of alpha-numeric characters only.

Lemmatization and Stemming

Lemmatization and stemming are the process of reducing the words to their roots. The difference between those two is the outcome. Lemmatization will convert a word into something meaningful while stemming leads to incorrect meaning and spelling errors. There are variations of methods in stemming, namely, Porter Stemmer and Lancaster Stemmer. Lancaster Stemmer is an aggressive stemmer algorithm compare to Porter Stemmer. Below shows the difference between stemming and lemmatizing.

> ***Lemmatising:*** *considered, considering, consider → "consider"*
> ***Stemming:*** *considered, considering, consider → "consid"*

Figure 2.4.2 Difference between Stemming and Lemmatizing

Next, showing the difference between Porter Stemming and Lancaster Stemming.

```
waited:wait
waiting:wait
waits:wait
```

```
giving:giv
given:giv
given:giv
gave:gav
```

Figure 2.4.3 Difference between Porter Stemmer (Left) and Lancaster Stemmer (Right)

Figure 2.4.3 above shows the aggressiveness of Lancaster Stemming.

Part of Speech (POS) Tagging

This process is assigning each word to part of speech based on the word context. Part of speech include nouns, verbs, pronouns, adverbs, conjunction, adjective and interjection. There is a lot of Python packages that provide part of speech tagging such as NLTK, Spacy, TextBlob and Standford CoreNLP.

Name Entity Recognition

This is the step of detecting entity such as name, location, events, company name and more. This method will chunk or grouping words into bigger pieces.

2.4.3. Sentiment Analysis on Social Media

Since sentiment analysis is part of the proposed solution, this topic must be covered. Sentiment is a view or opinion of a certain topic. Sentiment analysis is an application of Natural Language Processing (NLP) that helps users to gain the public opinion of a certain topic. Usually, sentiment analysis is done by classification. The most popular Python library for text classification is Scikit-Learn. However, there is a lot of Python libraries mentioned in [13] that provide sentiment analysis. Figure 2.4.4 below shows a variety of Python libraries that can be used.

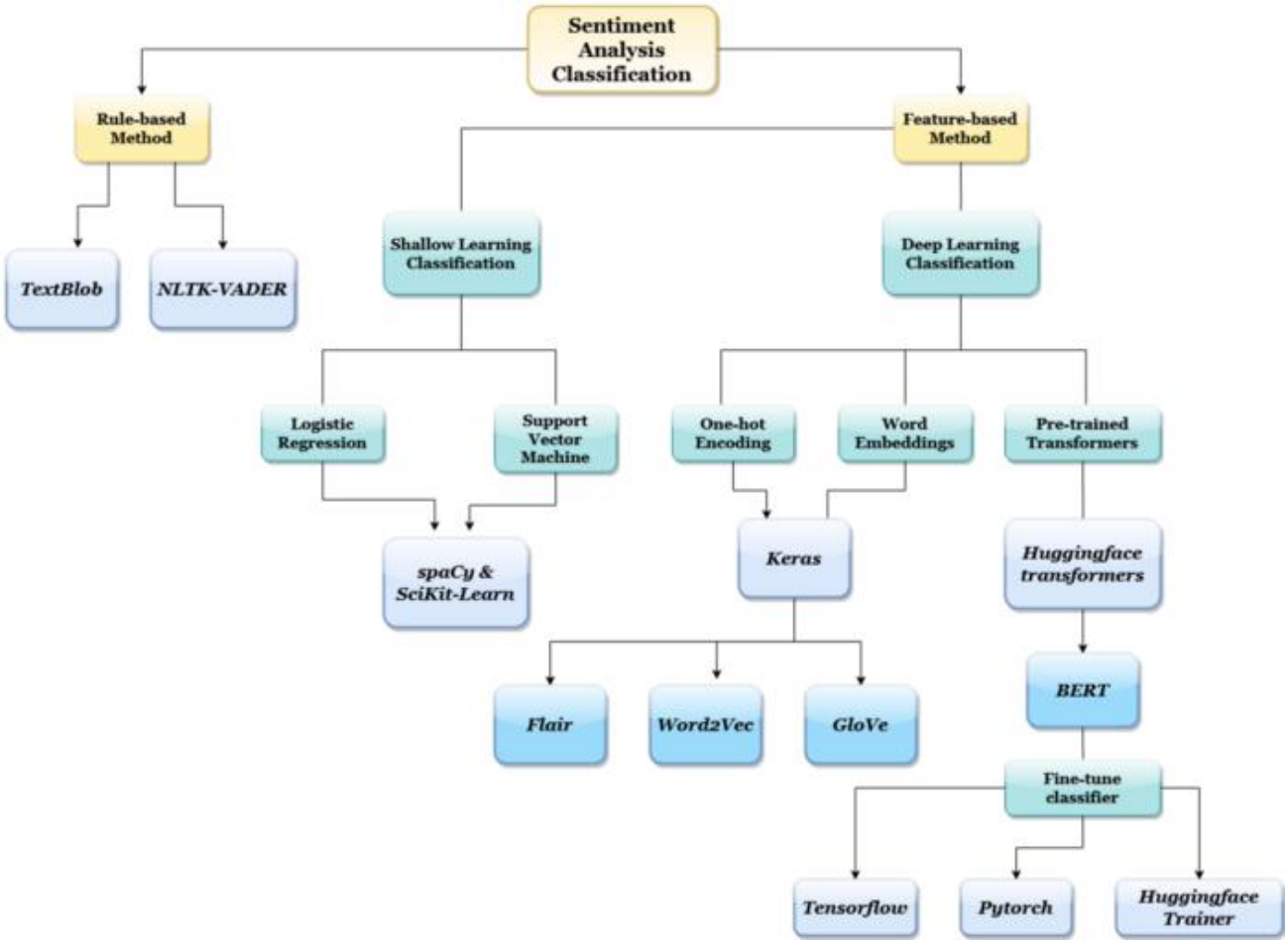


Figure 2.4.4 Variety of Python Libraries for Sentiment Analysis in [13]

In [13], there are few things can be summarized for each method. For rule-based method libraries, it is convenient but does not provide accuracy. It is a very good starting point for a beginner to NLP.

Next, shallow learning approaches such as Logistic Regression and Support Vector Machine (SVM) is more accurate than rule-based method. It also requires less work compared to deep learning. Besides, this method can automatically extract features with little or no work.

Deep learning classification provides more accuracy, but it come with a setback in early time [13]. In 2013, Google created Word2Vec algorithm as word embedding methods to be use along with GloVe algorithm as word embedding method. It used recurrent or convolutional neural networks. However, it is very slow in training due to convolutions and recurrence is difficult to parallelize. Attention mechanism improved the accuracy of the networks. Then, this problem is solved in 2017 when the transformer architecture introduced a way to use attention mechanism without recurrence or convolutional.

For Python deep learning libraries, Keras is recommended for beginners. However, it is not good to handle complex problems but still a good library if it combines with TensorFlow. Huggingface transformers is a good introduction to transformer. This library can be used effortlessly with other Machine Learning (ML) libraries such as Huggingface Trainer, Pytorch and TensorFlow. The most liked transformer is Bidirectional Encoder Representations from Transformer (BERT). It uses transfer learning that strengthens the power of pre-trained model weights. It allows nuances of contextual embedding to be shifted during the calibrating process.

2.4.4. TextBlob

According to [13], TextBlob is a one of the rule-based sentiment classifiers. This means there are a knowledge base and an inferencing engine exist in the classifier. Well, according to [14], there is a lexicon-based XML file that the describe the words and some of its measurement metrics such as polarity, subjectivity and confidence. That XML file acts as knowledge base. When determining sentiment with TextBlob, two measurements are given, namely, the polarity and subjectivity. The polarity determines the sentiment. If the polarity is negative, the sentiment is negative, if polarity is zero, the sentiment is neutral and if the polarity is positive, the sentiment is positive. The polarity is range from -1 to 1.

3 SYSTEM REQUIREMENTS / ANALYSIS

This is the most important element in developing a system. Analyzing the requirements is important as it helps the developers to understand what the system must meet before developing a system. Chapter 3 explained the requirements of the system.

3.1. Project Scope, System Capabilities and System Limitations

3.1.1. Project Scope

The overall scope of WHATSON is presented in Figure 1.5.1. The figure shows all modules and submodules.

3.1.2. System Capabilities

For system capabilities, it is divided into two parts, namely, functional capabilities and non-functional capabilities.

Functional Capabilities

- Manage Tweet
    - Add tweets
- Manage Users
    - Create account
    - Login
    - Logout
- Manage Sector
    - View time series of all sectors
    - View monthly sectors
    - View current topics related to sectors
    - View proportion/percentage of sectors
- Manage Sentiment
    - View time series of all sentiments
    - Show top tweets with polarity
    - View current topics related to sentiments

    - View proportion/percentage of sentiments
- Manage Topics
    - View time series of current popular topics
    - View current topics
    - Topic Prediction
    - Save topics
    - Read saved topics
    - View news headlines
    - Search news

Non-functional Capabilities

- Usability

  No user training will be given because the system is user-friendly, easy to use and can be self-learned.
- Performance

  The performance of the system is depending on few factors such as algorithm used, volume of data and internet connection.
- Security

  Although the system can be view by any users but the system still secure. This is due to fewer text inputs and most inputs are provided by the system. Most of the inputs are buttons and date pickers.

3.1.3. System Limitation

The only limitation of this system that it requires an internet connection. This is due to the system need to fetch the data from the database to process the data and print the output.

3.2. Project Management

The duration of this project is 248 days (35 weeks and 3 days) starting from 12th October 2020 until 16th June 2021. Figures below show WBS, Gantt chart and SWOT analysis.

WBS

Phase	List of Tasks	Days Allocated
Initial	**1. Project Bidding** 1.1.Placing bid on projects 1.2.Bidding review by lecturer and results	8 days
Proposal	**2. Project Initiation** 2.1.Carry out preparation for proposal 2.1.1. Identify the problem 2.1.2. Identify the system objectives 2.1.3. Describe proposed solution 2.1.4. Identify benefits of the project 2.1.5. Identify uniqueness of the project 2.1.6. Identify expected outcome 2.2.Complete the proposal	11 days
Analysis	**3. Project Analysis** 3.1.Carry out preparation for analysis report 3.1.1. Identify existing system strengths and weaknesses 3.1.2. Identify project scope, capabilities, and limitations 3.1.3. Define project management 3.1.4. Define development methodology 3.1.5. Define system requirements 3.1.6. Design system design and implementation	28 days

	3.1.7. Draw software design diagram 3.1.8. Identify hardware and software deployed 3.2.Complete analysis report	
Development	**4. Prototype Design and Development** 4.1.Design and set up server 4.2.Design and developing website 4.2.1. Design functions for tweets management 4.2.2. Develop functions for sector identification 4.2.3. Develop functions for sentiment analysis 4.2.4. Develop functions for topic analysis 4.3.Implement error handling function	110 days
Testing	**5. System Testing** 5.1.Carry out testing activities 5.2.Debug the system	35 days
Final	**6. Finalize the System** 6.1.Touch up the Graphical User Interface 6.2.Finalize the system 6.3.Complete final report	56 days

Figure 3.2.1 Work Breakdown Structure

Gantt Chart

ID	Task Title	Start Date	End Date	Duration	Oct 2020		Nov 2020					Dec 2020					Jan 2021					Feb 2021				Mar 2021				Apr 2021				May 2021					June 2021		
					18	25	1	8	15	22	29	6	13	20	27	3	10	17	24	31	7	14	21	28	7	14	21	28	4	11	18	25	2	9	16	23	30	6	13	20
1	**Project Bidding**	**12 Oct**	**19 Oct**	**8 days**																																			
1.1	Placing Bid on Project	12 Oct	13 Oct	2 days																																			
1.2	Bidding Review and Results	14 Oct	19 Oct	6 days																																			
2	**Project Initiation**	**20 Oct**	**30 Oct**	**11 days**																																			
2.1	Carry Out Preparation for Proposal	20 Oct	25 Oct	6 days																																			
2.2	Complete the Proposal	26 Oct	30 Oct	5 days																																			
3	**Project Analysis**	**31 Oct**	**27 Nov**	**28 days**																																			
3.1	Carry Out Preparation for Analysis Report	31 Oct	13 Nov	14 days																																			
3.2	Complete Analysis Report	15 Nov	27 Nov	14 days																																			
4	**Project Design and Development**	**28 Nov**	**17 Mar**	**110 days**																																			
4.1	Design and Set Up Server	28 Nov	7 Dec	10 days																																			
4.2	Design and Developing Website	8 Dec	17 Feb	72 days																																			
4.3	Implement Error Handling Function	18 Feb	17 Feb	28 days																																			
5	**System Testing**	**18 Mar**	**21 Apr**	**35 days**																																			
5.1	Carry Out Testing Activities	18 Mar	31 Mar	14 days																																			
5,2	Debug the System	1 Apr	21 Apr	21 days																																			
6	**Finalise the System**	**22 Apr**	**16 June**	**56 days**																																			
6.1	Touch Up Graphical User Interface	22 Apr	5 May	14 days																																			
6.2	Finalise the System	6 May	19 May	14 days																																			
6.3	Complete Final Report	20 May	16 June	28 days																																			

Figure 3.2.2 Gantt Chart

SWOT Analysis

Strength (S)	**Weakness (W)**
• Easy to use • No user authentication • Prior knowledge in developing website in HTML, CSS, JavaScript, PHP and Python • Provide an interactive data representation	• This platform only available on website • Need internet connection • Limited development time • Little knowledge in time series analysis • Lack of experience in machine learning and natural language processing.
Opportunity (O)	**Threat (T)**
• No other platform provides topic prediction feature • Topic prediction feature exclusives on Twitter • Some companies are seeking for solution to handle data of social media	• A lot of platforms provide social media analysis • Topic prediction is still a new concept when it comes on social media

Figure 3.2.3 SWOT Analysis

3.3. Development Methodology

The development methodology chosen to be implemented in this project is **Waterfall** development method. This method is considered traditional method but still a very suitable method in developing WHATSON. Waterfall development method is linear and sequential process makes it easier to understand and manage. Plus, it is suitable for someone with lack experience in developing projects.

Compared to other development methods, Waterfall development method is a slow process. This fact shows a weakness in this method but exploiting it can be so advantageous in this project. It provides ample time to explore new knowledge to be implemented in the system. This method is suitable with lack of knowledge in time series analysis. Plus, no urge from client/stakeholder makes this method recommended.

3.4. Detail Requirement

The requirement is gathered through a literature review. A lot of master thesis and web articles have been reviewed in order to gain knowledge that will help in making the proposed solution. A lot of components in time series analysis and text mining have been discovered. The results of the literature review can be found in chapter 2.

3.5. Analysis of New System

Overall Modules Diagram

Overall modules diagram can be view in Figure 1.5.1. The figure shows all the modules and submodules.

Use Case Diagram

Figure 3.5.1 shows use case diagram of WHATSON.

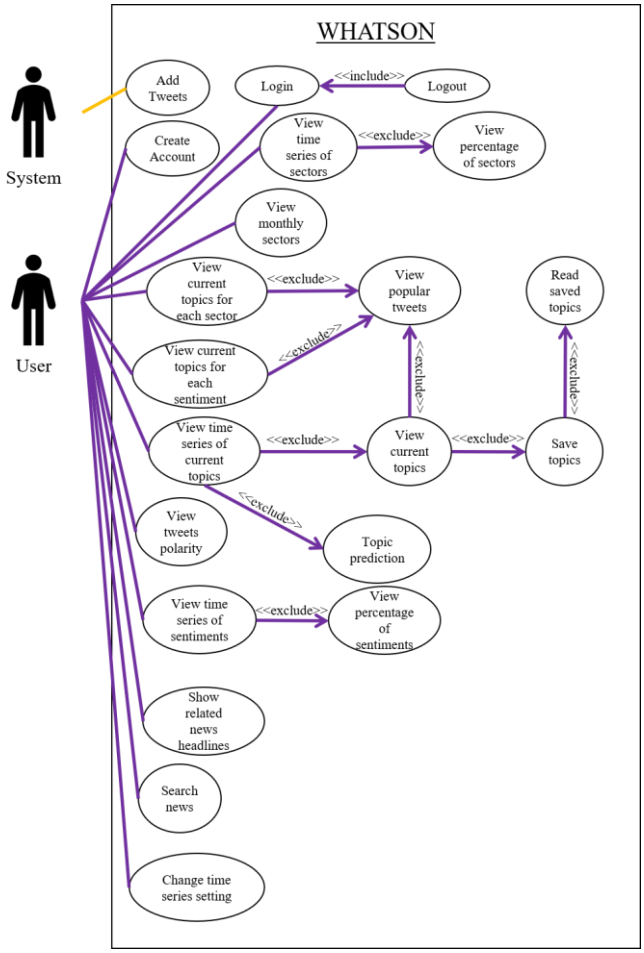


Figure 3.5.1 Use Case Diagram of WHATSON

Use Case Descriptions

Figure 3.5.2 until Figure 3.5.4 shows Use Case Description. The rest of the Use Case Description is in Appendix A.

<table>
<tr><td>Use Case</td><td colspan="2">Add tweets</td></tr>
<tr><td>Scenario</td><td colspan="2">Add tweets to database</td></tr>
<tr><td>Triggering Event</td><td colspan="2">-</td></tr>
<tr><td>Brief Description</td><td colspan="2">The system will automatically add tweets to database</td></tr>
<tr><td>Actor(s) Involved</td><td colspan="2">-</td></tr>
<tr><td>Related Use Case</td><td colspan="2">-</td></tr>
<tr><td>Stakeholder Involved</td><td colspan="2">-</td></tr>
<tr><td>Precondition</td><td colspan="2">-</td></tr>
<tr><td>Postcondition</td><td colspan="2">-</td></tr>
<tr><td rowspan="2">Flow of Activity</td><td>Actor</td><td>System</td></tr>
<tr><td>-</td><td>1. System add tweets to database</td></tr>
<tr><td>Exception Condition</td><td colspan="2">1. Unable to connect to database
2. Database having failure</td></tr>
</table>

Figure 3.5.2 Use Case Description of Add Tweets

Use Case	View time series of sectors	
Scenario	View time series of sectors	
Triggering Event	User wants to see time series of sectors	
Brief Description	The system will display time series of sectors to user	
Actor(s) Involved	User	
Related Use Case	Add tweets	
Stakeholder Involved	User	
Precondition	Tweet has been added	
Postcondition	-	
Flow of Activity	Actor	System
	1. User views time series of all sectors	1.1. The system fetch tweets from database 1.2. The system preprocessing text stream from tweets into data 1.3. The system classifies the data into its sector by using machine learning algorithm 1.4 The system groups the data by date and hour for each sector 1.5. The system plots the time series based on the frequency of each sector
Exception Condition	User has no internet connection	

Figure 3.5.3 Use Case Description of View Time Series of Sectors

Use Case	View time series of sentiments	
Scenario	View time series of sentiments	
Triggering Event	User wants to see time series of sentiments	
Brief Description	The system will display time series of sentiments to user	
Actor(s) Involved	User	
Related Use Case	Add tweets	
Stakeholder Involved	User	
Precondition	Tweets has been added	
Postcondition	-	
Flow of Activity	Actor	System
	1. User views time series of all sentiments	1.1 The system fetch tweets from database 1.2 The system preprocessing text stream from tweets into data 1.3 The system classifies the data into its sentiment by using machine learning algorithm 1.4 The system groups the data by date and hour for each sentiment 1.5 The system plots the time series based on the frequency of each sentiment
Exception Condition	User has no internet connection	

Figure 3.5.4 Use Case Description of View Time Series of Sentiments

Sequence State Diagram

Figure 3.5.5 until Figure 3.5.7 shows Sequence State Diagram (SSD) of WHATSON. The remaining SSD is in Appendix B.

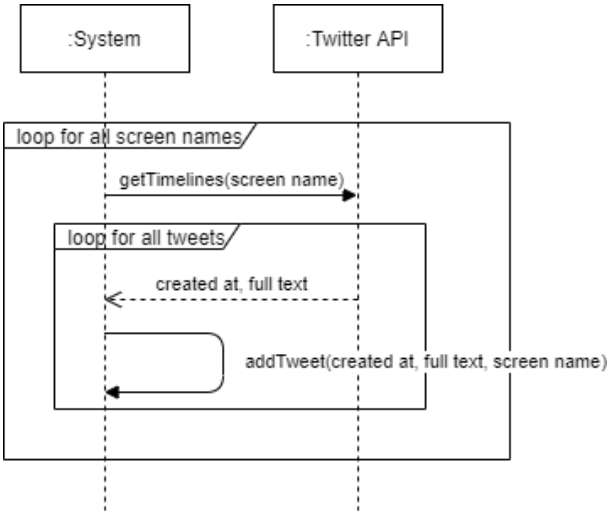


Figure 3.5.5 SSD of Add Tweets

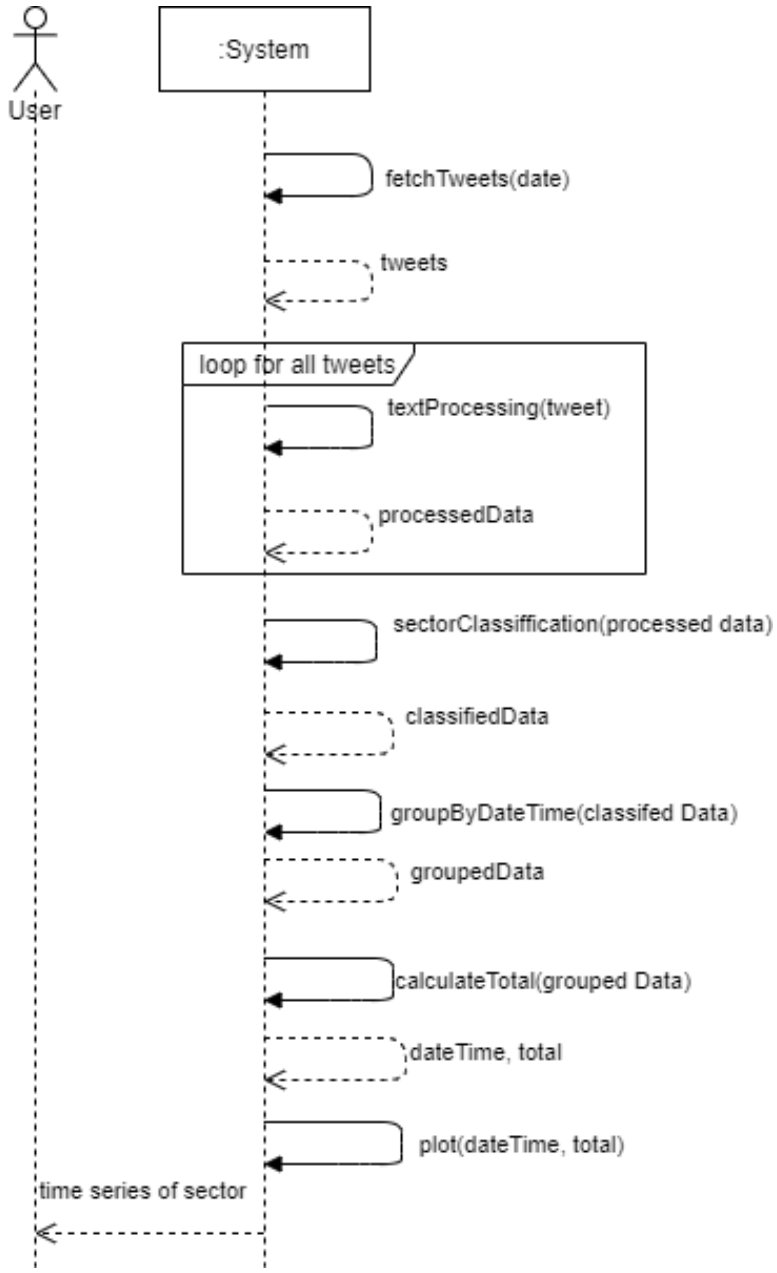


Figure 3.5.6 SSD of View Time Series of Sectors

Figure 3.5.7 SSD of View Time Series of Sentiments

3.6. Technology Deployed

Hardware

- Laptop
  - Model: Lenovo™ ideapad 320
  - Windows: Windows 10 64-bit, x64-based processor
  - Processor: Intel® Core™ i3-6006U
  - RAM: 8.00 GB
- Stable internet connection (LAN or Wi-Fi)

Software

- XAMPP Server
- Sublime Text 3 Editor
- PHP
- HTML
- JavaScript
- CSS
- Python
- Jupyter Notebook (anaconda 3)

4 SYSTEM DESIGN & IMPLEMENTATION

Once the system's requirements are known and analyzed, then the system needs to be designed and implemented. This includes the plethora of the processes such as designing system architecture, system modeling, database, user interface and more. The implementation strategy is also discussed here.

4.1. System Architecture

The system architecture can give a big picture of how the system works and interact without looking at the use case diagrams, use case description, sequence state diagram or the coding. The system architecture for this system can be easily understood if separated by its module.

4.1.1. Tweets Management Module

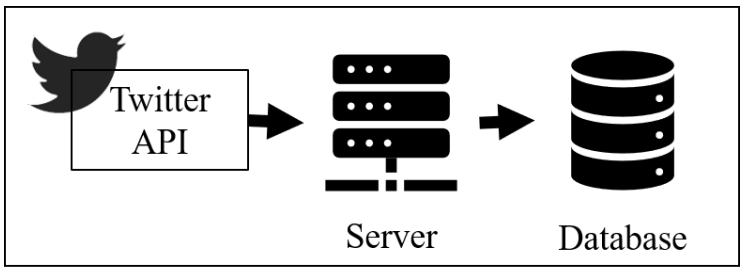


Figure 4.1.1 System Architecture of Tweets Management Module

For tweets management module, it has only one function, to store tweets in the database, MySQL. As can be seen, the interaction is very direct from Twitter API to database via WHATSON server.

4.1.2. User Management Module

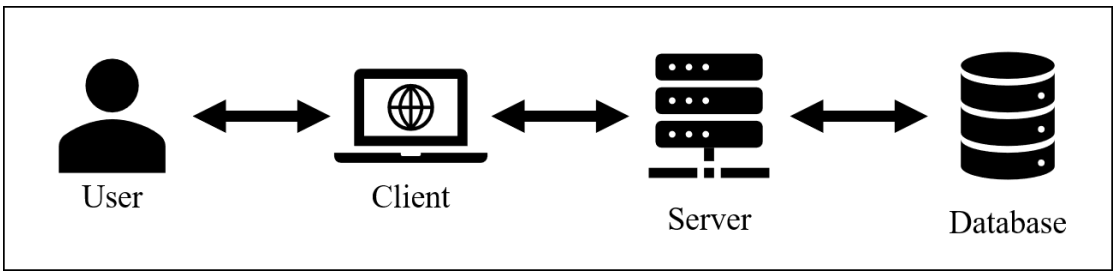


Figure 4.1.2 System Architecture of User Management Module

For this module, it does not involve Twitter API, but two more components are used in this module. The client is a website. The interaction is two-way between each component. There are no interactions between users. The reason for two-way interaction between each component is due to the module itself that has use cases of creating account and login. For creating an account, the user will give information be store in the database. Thus, it makes a very direct or one-way interaction from user to database via client and WHATSON server. While for login, the user will give the information such as username as query to the database and the database will return user's information back and verify the user's password. This makes the interaction between each component is two-way for login.

4.1.3. The Remaining Modules (Sector Identification, Sentiment Analysis & Topic Analysis)

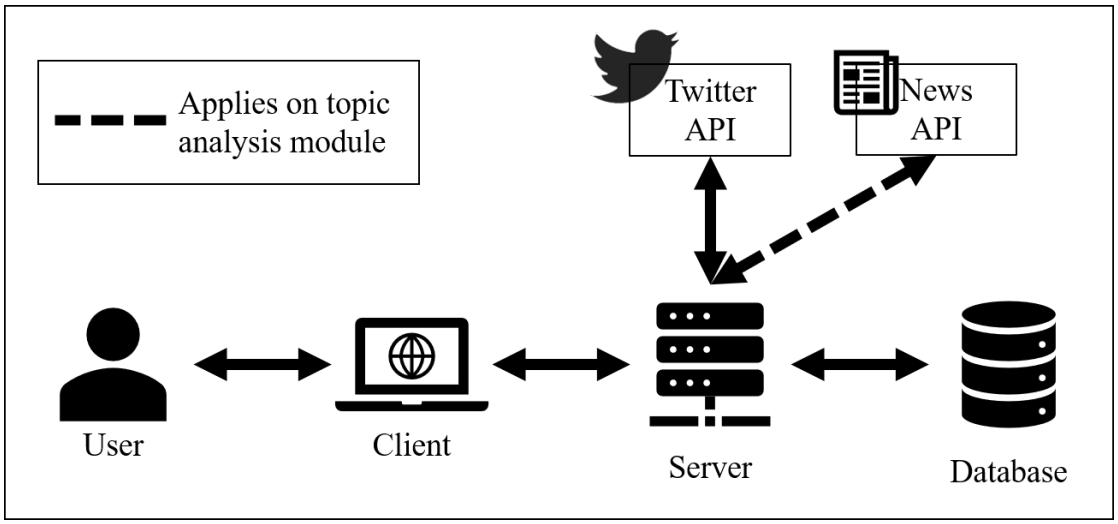


Figure 4.1.3 System Architecture of the Remaining Modules

Now, it can be seen that all components are used for the remaining modules (sector identification, sentiment analysis and topic analysis) and a new component, News API also introduced. Most processes in these modules are back end processes. Twitter API is utilized here for searching popular tweets. News API utilized here for searching news and showing news headlines by category. The interaction between user and client is two-way interaction due to the system need of user's input in some use cases such as change time series setting, save topics, search news, topic prediction and view popular tweets. Below shows a more detail diagrams for the remaining modules which describes the front end and back end processes.

Sentiment Analysis Module

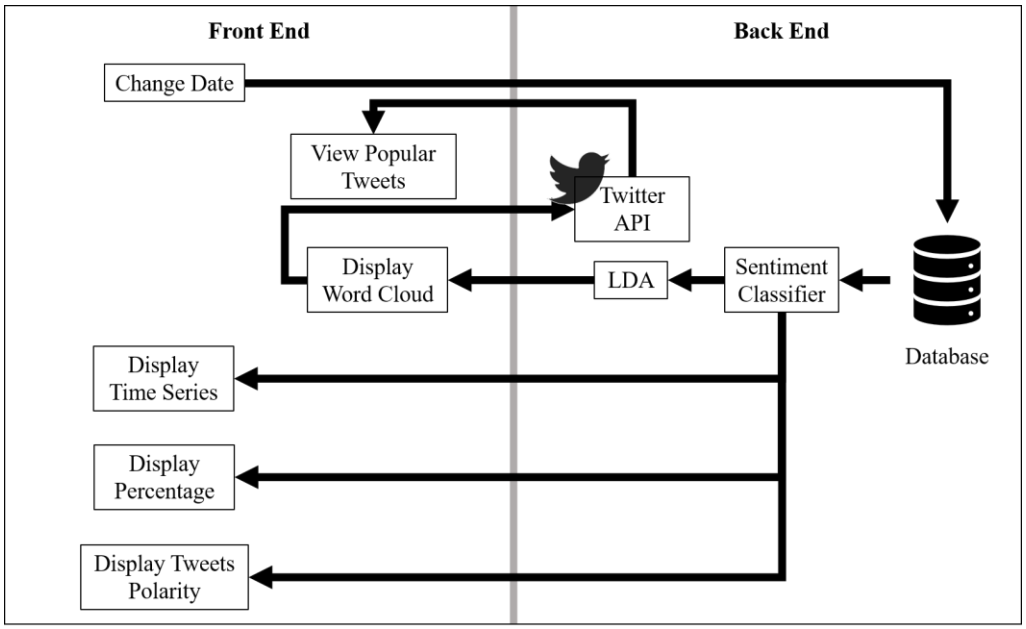


Figure 4.1.4 Flow of Processes in Sentiment Analysis Module

The Figure 4.1.4 above shows the front end and back end processes for Sentiment Analysis Module. As can be seen the processes starts from the back end process which fetch the data from database to the sentiment classifier, TextBlob. Then, from the classifier, the next processes mostly go to the front end to display time series, percentage, tweets polarity and word cloud. For displaying word cloud, the process is not direct from sentiment classifier, but it utilized a topic modeling model, Latent Dirichlet Allocation (LDA). From the word cloud, the user can click the word on the word cloud. The word clicked will be a query to Twitter API for searching tweets. From the tweets fetched, then it displays popular tweets. For change date process, the user will choose date range and send it as the query to database and fetch the tweets stored in database and goes to sentiment classifier. Then, the process will repeat from sentiment classifier to display the word cloud, time series, percentage and tweets polarity.

Sector Identification Module

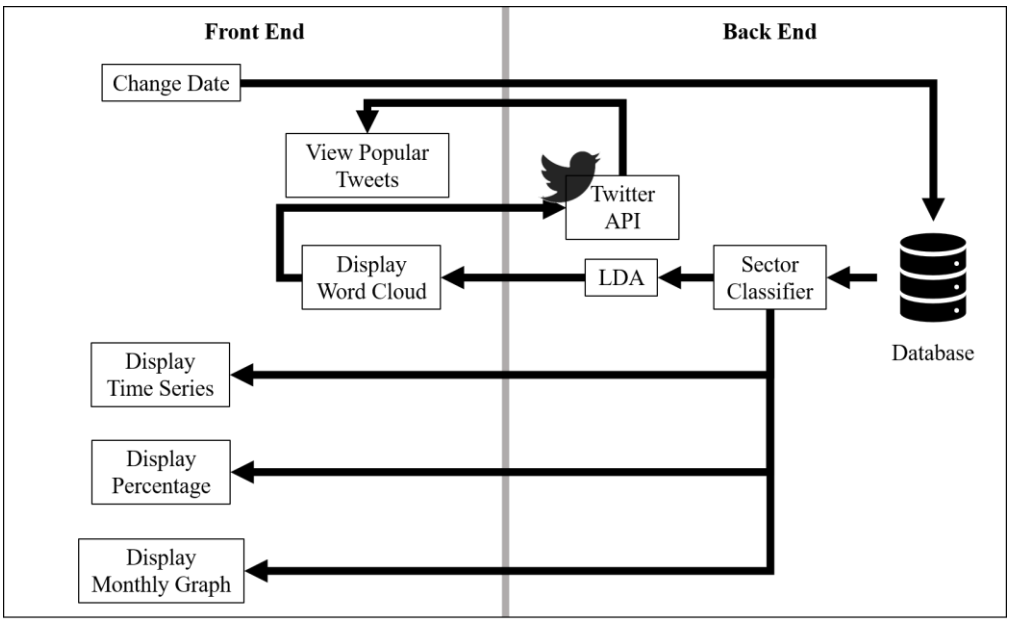


Figure 4.1.5 Flow of Processes in Sector Identification Module

The processes for Sector Identification Module are same as Sentiment Analysis Module. The display tweets polarity process is replaced with display monthly graph. The sector classifier used is Support Vector Machine (SVM).

Topic Analysis Module

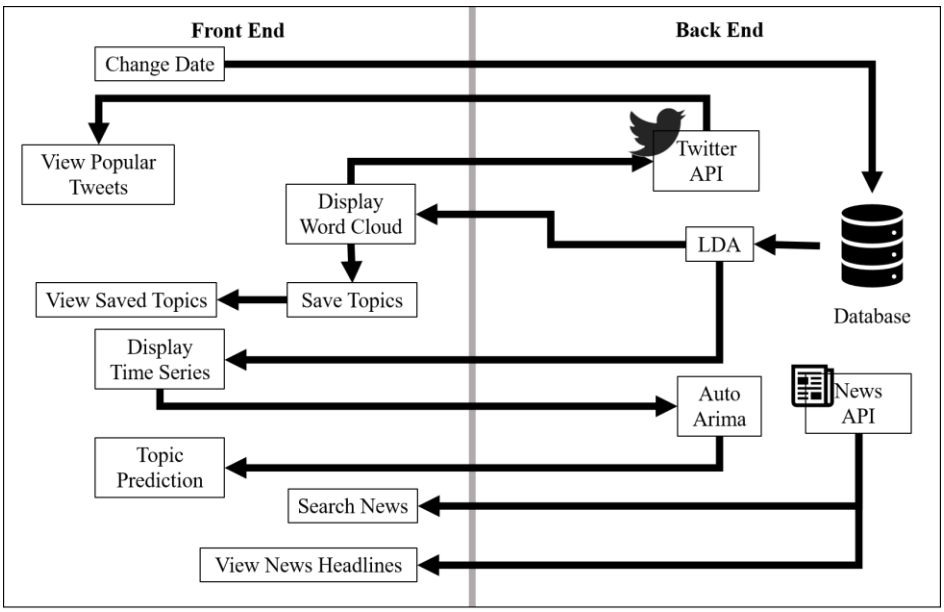


Figure 4.1.6 Flow of Processes in Topic Analysis Module

The processes for Topic Analysis Module are almost the same both modules but with some little changes. No percentage or tweets polarity or monthly chart will be displayed. The word cloud has a new function to save topics and from that process the saved topics can be viewed. From the time series, the auto ARIMA model will be used for topic prediction. From News API, the top news headlines and search news results also displayed.

4.2. Design Modeling

For design modeling, two diagrams are designed, namely, design class diagram and package diagram. The steps of utilizing machine algorithms also explained here.

4.2.1. Design Class Diagram

Figure 4.2.1 shows design class diagram which describes the classes/entities in the system and the relationship among each class. The classes consist of the members and the methods used. Since the entities in the system consist only two entities, namely, tweets and users, the relationship between these classes does not exist.

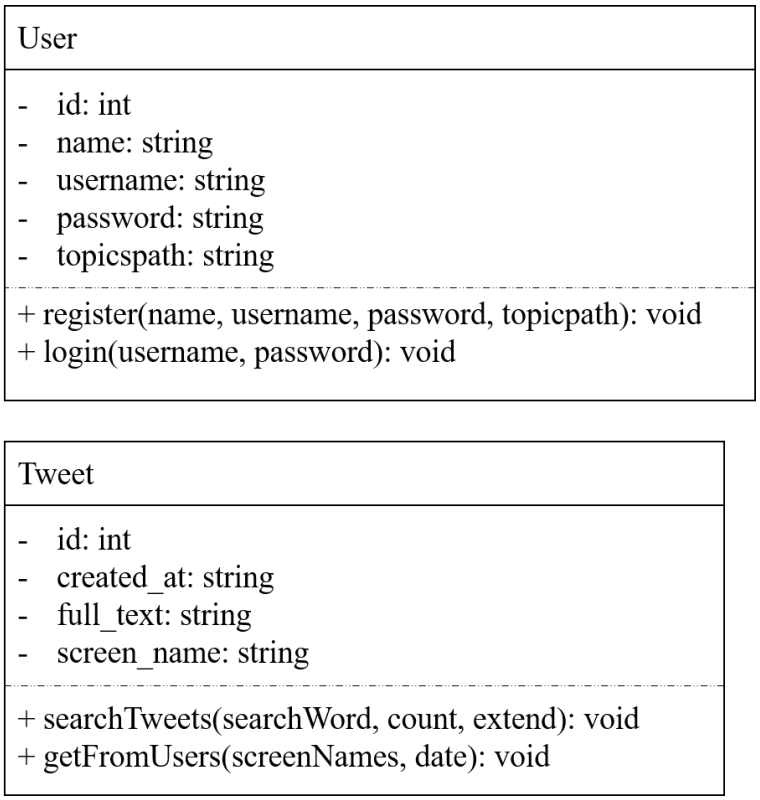


Figure 4.2.1 Design Class Diagram

4.2.2. Package Diagram

The package diagram is shown below in Figure 4.2.2. The package diagram represents all the files and folders in the system. The light green tag represents the folder's name. Some folders are empty because this diagram represents the whole files in the system without any data added. Once the system run, some files will be added especially in folder data and folder users.

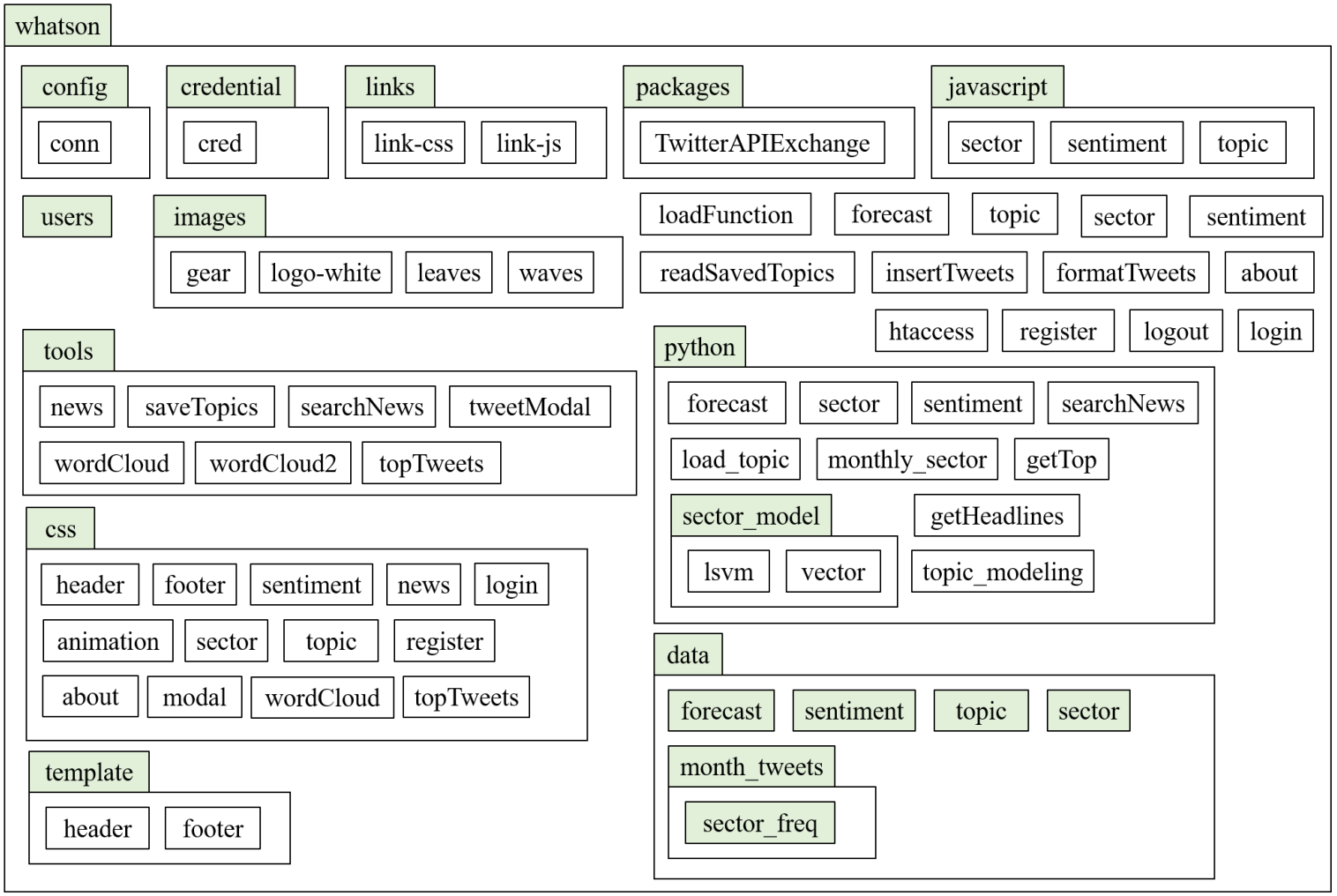


Figure 4.2.2 Package Diagram

4.2.3. Sector Identification Techniques

The sectors provided for sector identification are news, politics, sports, science and technologies, entertainment, travel, business and healthcare. The username/screen names of tweets are used to collect the tweets from different sectors. For example, if the screen name is 'YahooNews', then the sector is categorized as news. To classify the sectors, a machine learning model, Linear Support Vector Machine (SVM) is used. Below shows the steps in training and saving machine learning model.

i) importing libraries

```
import pandas as pd #to read csv files
from sklearn.model_selection import train_test_split#for training and testing data splitting
from sklearn.feature_extraction.text import TfidfVectorizer#toconvert text to vector
from sklearn.svm import LinearSVC#Linear SVM
import pickle#for saving model
import re #RegEx
from nltk import word_tokenize #to tokenize text
from nltk.corpus import stopwords#stopwords
from nltk.stem import PorterStemmer#stemmer
```

Figure 4.2.3 Importing Libraries in Sector Identification

ii) initializing variables

```
stemmer  = PorterStemmer()
stopWrd = set(stopwords.words('english'))

newsHandle = ['YahooNews', 'cnni', 'nytimes', 'FoxNews', 'NBCNews']
sportHandle = ['espn', 'SkySportsNews', 'NBCSports', 'BBCSport', 'SkySports']
scitechHandle = ['ScienceNews', 'ReutersScience', 'TechCrunch', 'newscientist', 'VentureBeat']
politicsHandle = ['CNNPolitics', 'ABCPolitics', 'BBCPolitics', 'nytpolitics', 'bpolitics']
entertainmentHandle = ['enews', 'IGN', 'nbc', 'screenrant', 'EW']
travelHandle = ['BTN_News', 'CNNTravel', 'travel_biz_news', 'USNewsTravel']
businessHandle = ['business', 'TheEconomist', 'economics', 'markets', 'FinancialTimes']
healthcareHandle = ['healthmagazine', 'NPRHealth', 'Reuters_Health', 'USNewsHealth', 'bbchealth']
```

Figure 4.2.4 Initializing Variable in Sector Identification

iii) creating functions

```
def getSector(handle): #to get sector
    if handle in newsHandle:
        return 'news'
    elif handle in sportHandle:
        return 'sports'
    elif handle in scitechHandle:
        return 'scitech'
    elif handle in politicsHandle:
        return 'politics'
    elif handle in entertainmentHandle:
        return 'entertainment'
    elif handle in travelHandle:
        return 'travel'
    elif handle in businessHandle:
        return 'business'
    elif handle in healthcareHandle:
        return 'healthcare'
```

Figure 4.2.5 Creating Function Part 1 in Sector Identification

```
def toNum(sector): #convert sector to number
    if sector == 'news':
        return 0
    elif sector == 'sports':
        return 1
    elif sector == 'scitech':
        return 2
    elif sector == 'politics':
        return 3
    elif sector == 'entertainment':
        return 4
    elif sector == 'travel':
        return 5
    elif sector == 'business':
        return 6
    elif sector == 'healthcare':
        return 7
```

Figure 4.2.6 Creating Function Part 2 in Sector Identification

```
def toSector(num):  #convert number to sector
    if num == 0:
        return 'news'
    elif num == 1:
        return 'sports'
    elif num == 2:
        return 'scitech'
    elif num == 3:
        return 'politics'
    elif num == 4:
        return 'entertainment'
    elif num == 5:
        return 'travel'
    elif num == 6:
        return 'business'
    elif num == 7:
        return 'healthcare'
```

Figure 4.2.7 Creating Functions Part 3 in Sector Identification

```
def cleanTweet(text): #clean tweets from link, reserved word, hashtag(#) and mention(@)
    text = re.sub('@\S+',' ',text) #remove mention
    text = re.sub('#',' ',text) #remove hashtag
    text = re.sub('(RT|FAV)',' ',text) #remove reserved word
    text = re.sub('(www|http|https)(\S+)',' ',text) #remove link
    text = re.sub('&\S+;',' ',text) #remove character ref (&amp; &quot; &lt; etc)
    text = re.sub('[^a-zA-Z]',' ',text) #remove other than letter. Should I keep number?

    return text

def processText(text): #preprocessing text
    text = word_tokenize(text.lower())

    text = [x for x in text if x not in stopWrd] #remove stopwords

    newWords =[]
    for word in text:
        newWords.append(stemmer.stem(word))
    return newWords #return as stemmed token
```

Figure 4.2.8 Creating Functions Part 4 in Sector Identification

iv) reading CSV files and assigning sectors to tweets

```
data = pd.read_csv('tweets.csv', encoding='utf-8')

data['sector'] = data['screen_name'].apply(getSector)
data['sector'] = data['sector'].apply(toNum)
```

Figure 4.2.9 Reading CSV Files and Assigning Sectors to Tweets in Sector Identification

v) text preprocessing

```
cleanTweets = data['full_text'].apply(cleanTweet)
tokenizedTweets = cleanTweets.apply(processText)

for i in range(len(tokenizedTweets)):
    tokenizedTweets[i] = ' '.join(tokenizedTweets[i])

data['new_text'] = tokenizedTweets
```

Figure 4.2.10 Text Preprocessing in Sector Identification

vi) splitting data to training and testing data

```
trainX, testX, trainY, testY = train_test_split(data['new_text'],data['sector'],test_size=0.1)
```

Figure 4.2.11 Splitting Data in Sector Identification

vii) converting cleaned text to vector

```
Encoder = LabelEncoder()
trainY = Encoder.fit_transform(trainY)
testY = Encoder.fit_transform(testY)

vectorizer = TfidfVectorizer(stopWrd)
vectorizer.fit_transform(data['new_text'])

trainXTfidf = vectorizer.transform(trainX)
testXTfidf = vectorizer.transform(testX)
```

Figure 4.2.12 Converting Cleaned Text in Sector Identification

viii) training model and save model and vector

```
lsvm= LinearSVC()
lsvm.fit(trainXTfidf,trainY)

with open('lsvm_76.pickle', 'wb') as f:
    pickle.dump(lsvm, f)
```

Figure 4.2.13 Training and Save Model and Vector in Sector Identification

Linear SVM is chosen because it is more accurate and faster in identifying the sector. The accuracy of the model is 76.33%. Some machine learning model also tested such as Multinomial Naïve Bayes (NB), Bernoulli Naïve Bayes (NB) and Stacking Classifier. Linear SVM produces more accurate results. Below shows the accuracy of Linear SVM, Multinomial NB, Bernoulli NB and Stacking Classifier in Figure 4.2.14.

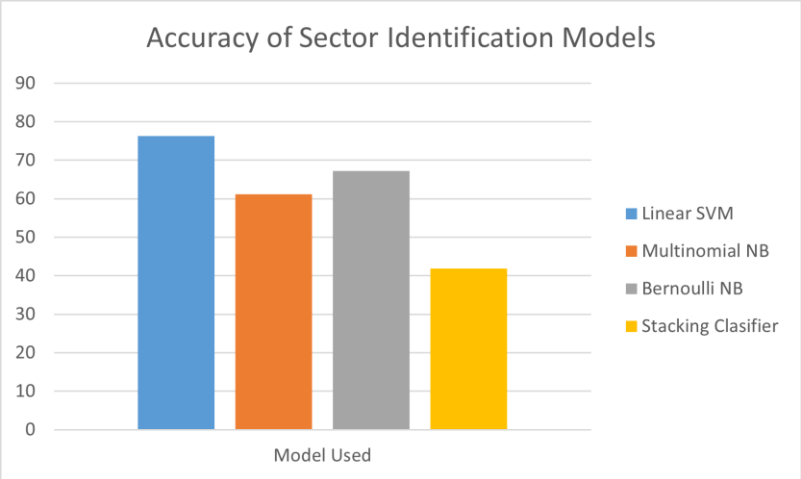


Figure 4.2.14 Accuracy of Sector Identification Models

4.2.4. Topic Modeling Technique

For topic modeling, the algorithm used is Latent Dirichlet Allocation (LDA). It is a famous algorithm for topic modeling. The result of this algorithm is a list of topics. The topic is set to 10 topics. Each topic has 15 words. Since this is a text clustering technique, the accuracy will not be calculated. The first few steps of topic modeling are same as sector identification technique until text preprocessing steps, but no sector will allocate for each tweet. There are few more libraries are added such as json, NumPy and gensim. Below shows the steps for topic modeling starting after text preprocessing steps until printing out the topics.

i) importing new libraries

```
import json
from gensim import corpora
import numpy as np
import random
import gensim.models.ldamodel as lda
```

Figure 4.2.15 Importing New Libraries in Topic Modeling

ii) initializing new variables

```
seed = 40
random.seed(seed)
num_topics = 10
dictTopics = {}
term = 15
```

Figure 4.2.16 Initializing New Variable in Topic Modeling

iii) converting text to bag of words corpus

```
id2word = corpora.Dictionary(data['new_text'])
id2word.filter_extremes(no_below=5, no_above=0.05)

if bool(id2word) == False:
    id2word = corpora.Dictionary(data['new_text'])

corpus = [id2word.doc2bow(text) for text in data['new_text']]
```

Figure 4.2.17 Converting Text to Bog of Words Corpus in Topic Modeling

iv) train model

```
lda_model = lda.LdaModel(corpus=corpus, id2word=id2word, num_topics=num_topics, random_state=seed)
```

Figure 4.2.18 Train Model in Topic Modeling

v) print output

```
for i in range(num_topics):
    dictTopics[i] = {}
    #term = random.randint(17,20)
    topic = lda_model.show_topic(i,term)
    for j in range(len(topic)):
        dictTopics[i][j] = {}
        dictTopics[i][j]['term'] = topic[j][0]
        dictTopics[i][j]['score'] = str(topic[j][1])

dictTopics = json.dumps(dictTopics)

print(dictTopics)
```

Figure 4.2.19 Print Output in Topic Modeling

4.2.5. Topic Prediction Technique

As mentioned in Chapter 2, ARIMA is the best time series forecasting technique. but it requires more parameter other than time series, namely, p is number of lags (refers to AR), d is order of differencing and q is number of lagged forecast errors (refers to MA). This is such troublesome that these values need to determine. Fortunately, there is a Python library? called *pmdarima* that can be used to get the values. This library contains method *auto_arima* which can automatically calculate the values of p, d and q and do forecasting. Below are some parts of auto ARIMA time series analysis.

i) importing libraries

```
import pandas as pd
from pmdarima import auto_arima
import sys
```

Figure 4.2.20 Importing Libraries in Topic Prediction

ii) initialize variables

```
topic = sys.argv[1]
periods = int(sys.argv[2])
```

Figure 4.2.21 Initializing Variables in Topic Prediction

iii) define functions

```
def arimamodel(timeseries):
    return auto_arima(timeseries, start_p=1, d=1, start_q=1, max_p=7, max_d=7, max_q=7, start_P=1, D=1, start_Q=1,
                      max_P=7, max_D=7, max_Q=7, seasonal=True, test="adf", trace=True, stepwise=True,  m=10, random_state=20)

def getForecast(periods, timeseries, model):
    #forecast value
    fc = model.predict(n_periods=periods)
    #hourly index
    fc_ind = pd.date_range(timeseries.index[timeseries.shape[0]-1], periods=periods, freq="1H")
    #forecast dataframe
    return pd.DataFrame(fc, columns=['forecast'],index=fc_ind)
```

Figure 4.2.22 Define Functions in Topic Prediction

iv) read CSV file

```
df = pd.read_csv('data/topic/topics.csv')
dateTime = df['dateTime'].to_list()
topic = df[topic].to_list()

new_df = pd.DataFrame(topic, columns=['topic'],index=pd.to_datetime(dateTime))
```

Figure 4.2.23 Read CSV File in Topic Prediction

v) regression and forecasting

```
model = arimamodel(new_df)
fc =getForecast(periods+1, new_df, model)
```

Figure 4.2.24 Regression and Forecasting in Topic Modeling

vi) save forecast values as CSV files

```
new_df.reset_index(level=0, inplace=True)
fc.reset_index(level=0, inplace=True)

new_df.to_csv('data/forecast/df.csv', index=False)
fc.to_csv('data/forecast/fc.csv', index=False)
```

Figure 4.2.25 Save Forecast in Topic Modeling

The forecast values produced are quite accurate. It can be seen in the graph below where the blue lines are training time series, the green one is forecast values for 24 hours and the orange one is the actual values for 24 hours.

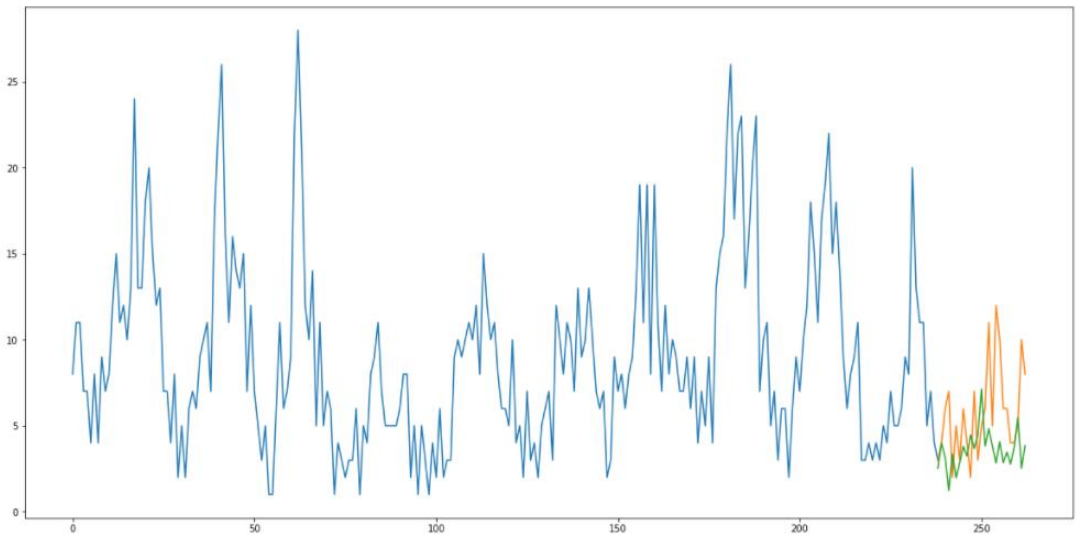


Figure 4.2.26 Time Series of Training, Forecasting and Actual Values

According to [15], there are many means of justifying prediction techniques including Mean Absolute Percentage Error (MAPE), Correlation and Min-Max Error. In here, MAPE will be used. MAPE is the mean of absolute differences between forecast and actual values over actual values. The MAPE produced for *auto_arima* is 44.72%. This means the model has accuracy of 55.28% in predicting next 24 observations.

4.3. Database Design

The database used in this project is MySQL. MySQL consists of tables. For this system, there are 2 tables, namely, user and tweet. Table user keeps the user's information such as name, username, password, and topics path (for saving and reading topics). For tweet table, it keeps the text in the tweet, the time of the tweet created, and the screen name/username. Below shows the Entity Relationship Diagram (ERD) of WHATSON.

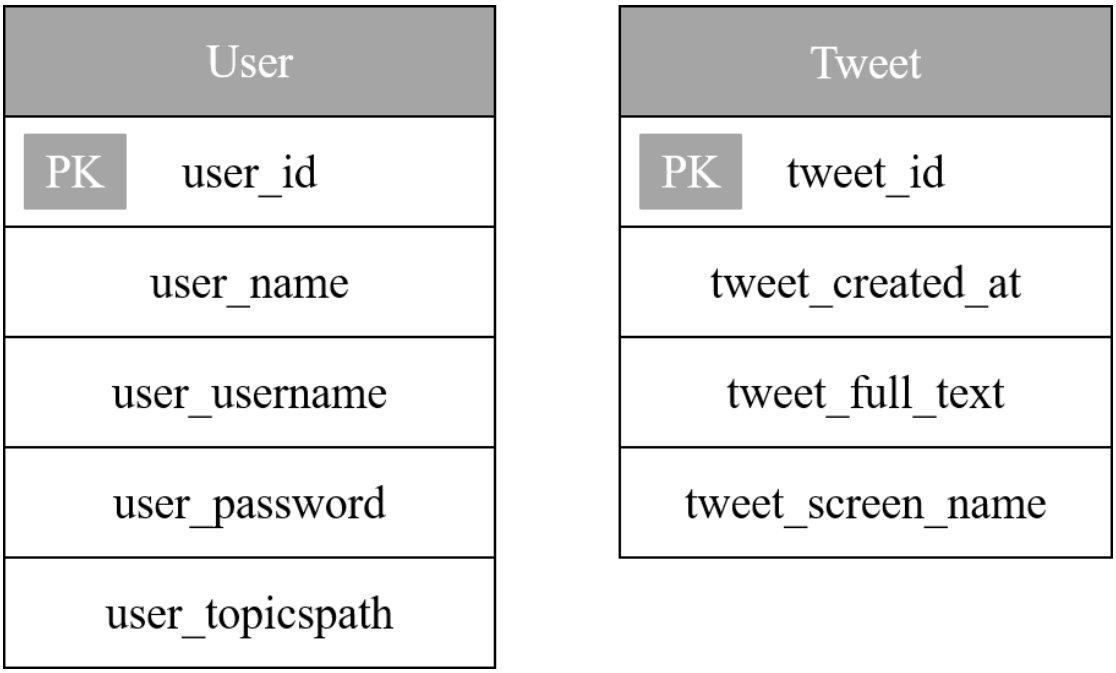


Figure 4.3.1 Entity Relationship Diagram

4.4. User Interface Design

The system delivers 3 types of views comprises of the landing page, login/register form and dashboard. The views of the website are simple, easy to understand, interactive and require less text input. Most of the inputs are provided such as date pickers, selections and buttons.

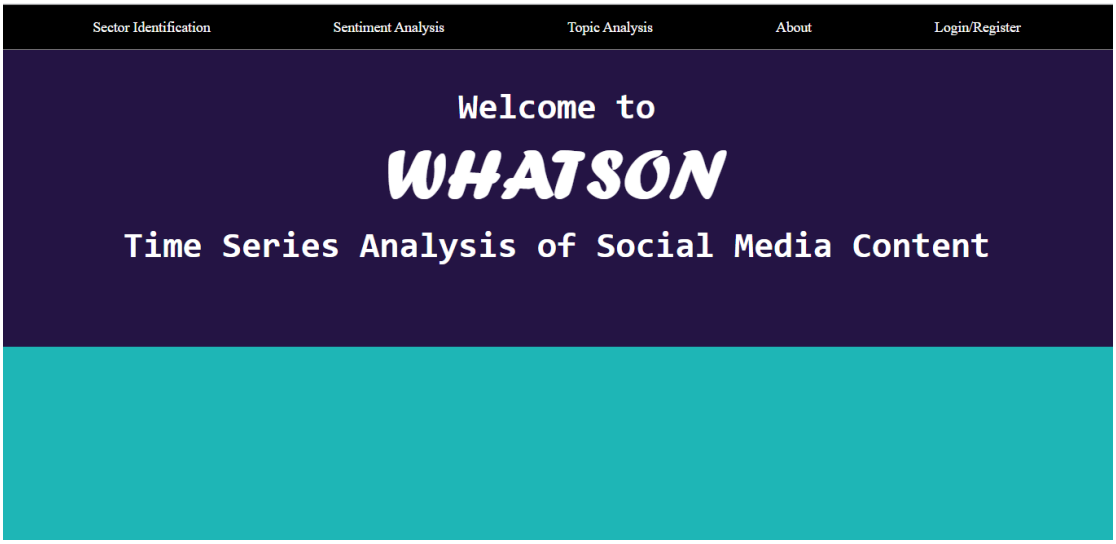


Figure 4.4.1 Landing Page

The above figure shows the landing page. It comes with a simple greetings and top navigation bar. On the navigation bar, it will navigate to 5 pages, namely, Sector Identification, Sentiment Analysis, Topic Analysis, About (this page) and Login/Register page. Sector Identification module is located at Sector Identification page and same goes to Sentiment Analysis and Topic Analysis module are located at Sentiment Analysis page and Topic Analysis page, respectively. The User Management Module is located at the Login/Register page.

Figure 4.4.2 Login Page

Figure 4.4.3 Register Page

Two figures above, Figure 4.4.1 and Figure 4.4.2 shows login and register form, respectively. It provides a little greeting, some text field to fill in information, link to register page (if the current page is login) and login page (if the current page is register) and submit button.

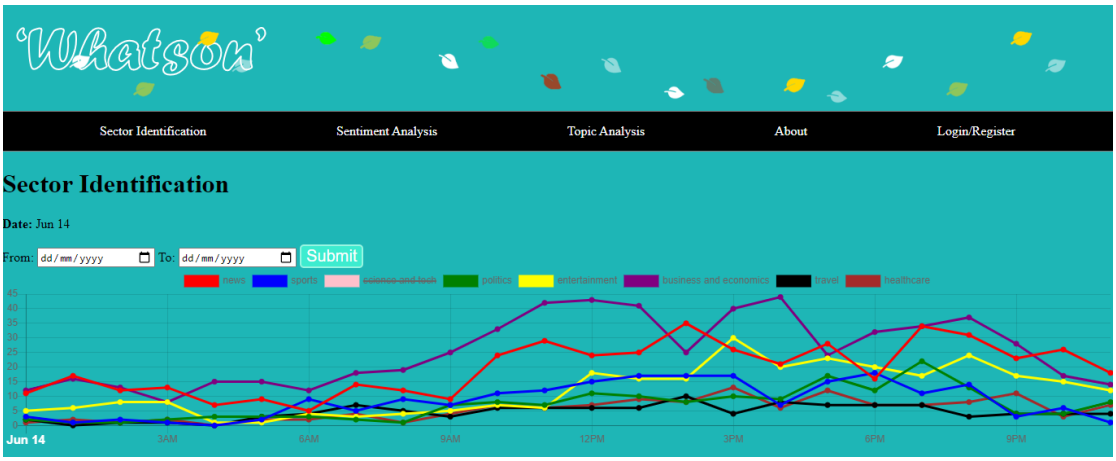


Figure 4.4.4 Dashboard

Figure 4.4.4 shows the dashboard created in the system. This is where the most interactivity of the system goes to. The figure above shows of the dashboard in the Sector Identification page. As can be seen above the navigation bar, the page displays the WHATSON logo with soothing leaves flowing animation. For every page, there is a time series displayed with its legends. Click one of the sectors in the legends, the time series that represent that sector will disappear. For example, in the figure above, the time series of science and tech is disappeared.

Other than time series, the word cloud also can be displayed on every dashboard page. Each word cloud has 15 words that represent a topic (results of topic modeling). Figure 4.4.6 shows that when a word on word cloud is hovered, the word will be floating. When a word on word cloud is clicked, a modal of popular tweets will be displayed. There are still more interactive features in the dashboard that can be explored in below and Appendix C.

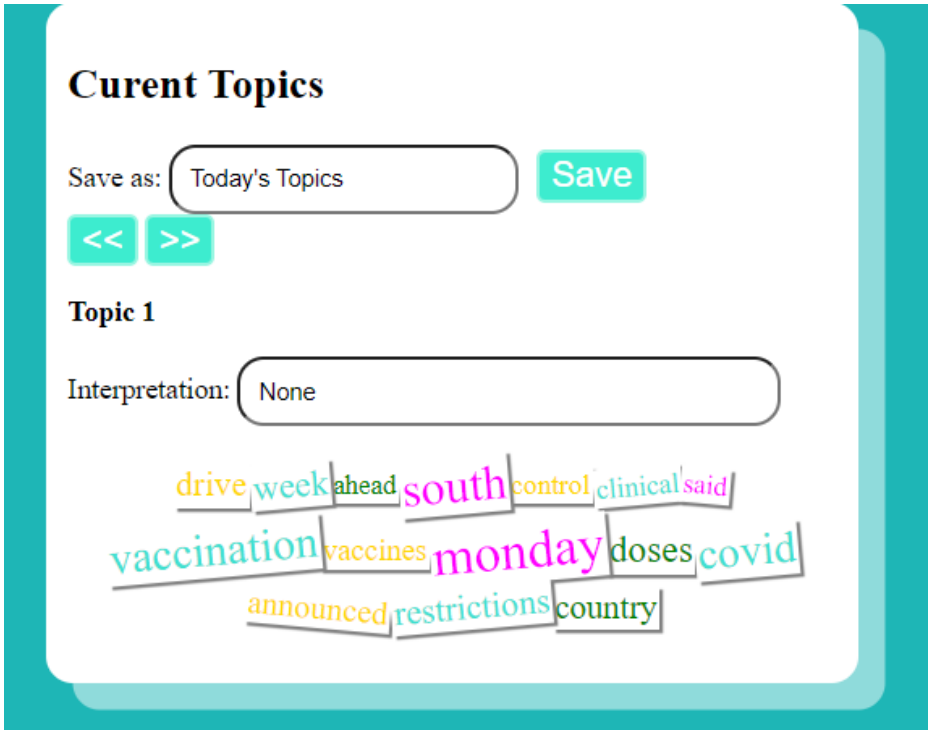


Figure 4.4.5 Word Cloud

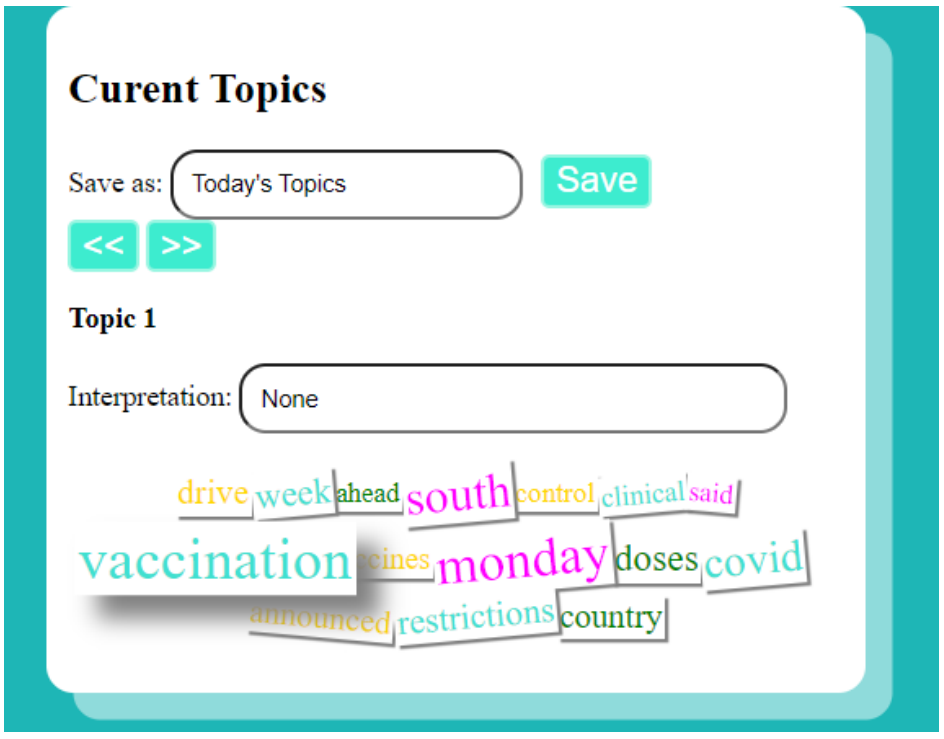


Figure 4.4.6 Word Cloud When a Word is Hovered

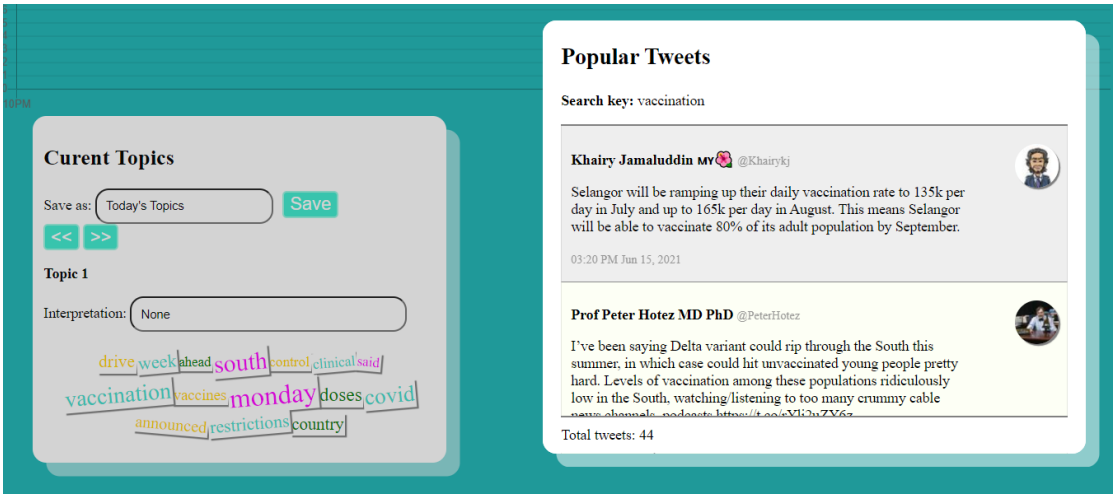


Figure 4.4.7 Modal of Popular Tweets Appear After Clicked a Word on Word Cloud

4.5. Implementation Strategy

Traditionally, there two types of implementation strategy, namely, top-down approach and bottom-up approach. Top-down approach starting from general goes to specific while bottom-up approach starting from specific to the general. Both approaches are very useful depending on the amount of information gathered or users' needs.

For this project, the approach uses is **both top-down and bottom-up approaches**. Depending on the information gathered and the users' needs, all the modules will be assigned either to implement top-down or bottom-up. The modules assigned for bottom-up approaches are the tweets management module and user management module. The rest of the modules (sector identification, sentiment analysis, topic analysis) are assigned to the top-down approach. This is due to these modules using machine learning algorithms and users' needs are still not known specifically in producing the dashboard.

4.6. Overview

So far, the system design & implementation are quite well explained. All the figures are well-drawn and the steps in utilizing machine learning are quite well explained supported with some figures that show accuracy. However, there could be some places that can be improved. Firstly, is the landing page and register/login form. The designs are too simple. It could be boring to see those interfaces. It cannot be well designed due to time constraints. Next, the entities/classes involve are just a few. This is because most processes occurred in the back end and there are few features that user can customized. Last but not least, modification in the tweet table could make a system perform better as the system takes quite a bit of time to load this could be deemed as performance problems. Fortunately, the problem is fixed, and the solution has been discussed in Chapter 5 (the last section).

5 SYSTEM TESTING & EVALUATION

This chapter covers testing strategy, test cases, test results, and test evaluation and summary. Testing is done to ensure the system meets the users' needs.

5.1. Testing Strategy

The testing is divided into unit testing, integration testing and system testing. Each testing and strategy will be explained below.

5.1.1. Unit Testing

Unit testing is also known as component testing. As its name is 'unit', this testing is carried unit by unit. This means that the testing starts by testing with a small part or function. The aim of this testing is to ensure all parts work as intended. For this project, a use case is considered as a unit. The use case will not be broken down into small components as long as there is no error(s) occurred. Tools that are used for testing are localhost, inspect console, and terminal. Localhost is good at detecting errors in PHP, inspect console at good detecting errors in JavaScript and terminal at good in identifying errors in Python.

5.1.2. Integration testing

After completing unit testing, integration testing can be conducted. Integration testing is testing if two or more units can work as intended. For this project, integration testing is done by checking if all the use cases in a module can work together. The tools using in this test are the same as in unit testing.

5.1.3. System Testing

System testing is done to check if the system meets the objectives in Chapter 1, Chapter 2 and Chapter 3. This test is done after integration testing. The WHATSON system itself will be tested as a system. All functional and non-functional capabilities were tested but more focusing on non-functional capabilities.

5.2. Unit Testing Results

Three sample of test cases are shown in Figure 5.2.1 Figure 5.2.2 and Figure 5.2.3. The rest of test cases in Appendix D.

Use Case	Add tweets
Description	The system will automatically store tweets in database
Steps Involved	1. The system will authenticate Twitter API 2. The system fetches tweets from certain users via Twitter API
Input Data	Screen Names: ['YahooNews', 'cnni', 'nytimes', 'FoxNews'], Date: June 16
Expected Result	Add tweets fail if 1) no connection with database 2) screen names are not provided 3) date is not provided 4) no internet connection
Actual Result	Add tweets fail if 1) no database connection 2) screen names are not provided 3) date is not provided 4) no internet connection
Pass/Fail	Pass

Figure 5.2.1 Test Case of Add Tweets

Use Case	View time series of sectors
Description	The system will automatically plot time series of all sectors from the tweets fetched.
Steps Involved	1. The system read the csv file of tweets 2. The system identify sector of each tweet 3. The system plot time series
Input Data	A CSV file of tweets consists columns of created_at, full_text and screen_name
Expected Result	Plot time series fail if 1) the CSV file is not exists 2) the CSV file is empty 3) no column name in CSV file 4) one of the column is not exist 5) no internet connection
Actual Result	Plot time series fail if 1) the CSV file is not exists 2) the CSV file is empty 3) no column name in CSV file 4) one of the column is not exist 5) no internet connection
Pass/Fail	Pass

Figure 5.2.2 Test Case of View Time Series of Sectors

Use Case	View time series of sentiments
Description	The system will automatically plot time series of all sentiments from the tweets fetched.
Steps Involved	1. The system read the csv file of tweets 2. The system identify sentiment of each tweet 3. The system plot time series
Input Data	A CSV file of tweets consists columns of created_at, full_text and screen_name
Expected Result	Plot time series fail if 1) the CSV file is not exists 2) the CSV file is empty 3) no column name in CSV file 4) one of the column is not exist 5) no internet connection
Actual Result	Plot time series fail if 1) the csv file is not exists 2) the csv file is empty 3) no column name in csv file 4) one of the column is not exist 5) no internet connection
Pass/Fail	Pass

Figure 5.2.3 Test Case of View Time Series of Sentiments

5.3. Integration Testing Results

Table 5.3.1 summarized the results of integration testing.

Integration	Test Scenario	Expected Results	Actual Results
Tweet Management Module	The system adds tweets automatically	The system is able to add system automatically with no problems	The system is able to add system automatically with no problems
User Management Module	The user creates account, login and logout	The user is able to create account, login and logout with no problems	The user is able to create account, login and logout with no problems
Sector Identification Module	The system plots time series, pie chart, bar graph and draws word clouds. The user can change date range of time series.	The system is able to plot time series, pie chart, bar graph and draws word clouds with no problems. The user is able to change date range of time series with no problems.	The system is able to plot time series, pie chart, bar graph and draws word clouds with no problems. The user is able to change date range of time series with no problems.
Sentiment Analysis Module	The system plots time series, pie chart, shows tweets polarity and draws word clouds. The user can change date range of time series.	The system is able to plot time series, pie chart, shows tweets polarity and draws word clouds with no problems. The user is able to change date range of time series with no problems.	The system is able to plot time series, pie chart, shows tweets polarity and draws word clouds with no problems. The user is able to change date range of time series with no problems.
Topic Analysis Module	The system plots time series, shows news headlines and draws word clouds. The user can change date range of time series, save and read topics (if login) and search news.	The system is able to plot time series, shows news headlines and draws word clouds with no problems. The user is able to change date range of time series, save and read topics (if login) and search news with no problems.	The system is able to plot time series, shows news headlines and draws word clouds with no problems. The user is able to change date range of time series, save and read topics (if login) and search news with no problems.

Table 5.3.1 Results of Integration Testing

5.4. System Testing Results

The testing is done on the system to check if it meets both functional and non-functional capabilities. Below shows the summary of the results.

5.4.1. Functional Capabilities

The testing starts with a scenario of User A going to the WHATSON website. The first thing he sees is the landing page with navigation bar and greetings. He clicks one of the sections in the navigation bar, the Login/Register. Once clicked, then the system goes to Login page. He entered his username and password and clicks submit. Then, it leads to the landing page again. This time it is a little different. The Login/Register changed to Logout and a new greeting appears. It's his name. The system greets with "Hi, User A!". User A then clicks Topic Analysis, one of the sections in navigation bar. The moment after he clicked, he sees three boxes, but mostly fills with a blinking purple text. "Loading…". Only, one box that is filled up with the title of "Saved Topics" and a refresh button. After a moment, the boxes fill with word clouds with buttons and text fields, and rows of news headlines with tabs of search news and categories. A time series with date pickers and topic prediction fields appears. Now, it becomes a dashboard. User A tries to pick a random date in the date picker. He clicks submit. After few seconds, time series and word clouds disappear. It fills with a new one. Now, he enters the interpretation of each word cloud and saves it as random name. He was aided by tweets modal when he clicked word in the word cloud and by news headlines. He also tries to do topic prediction on a certain topic.

The scenario narrated above is repeated about few times. Throughout the test, it is expected that the system worked smoothly. If errors are found, the system will be fixed and repeat the scenario a few times until no errors occurred. So far, this scenario is repeated 3 times. There was an error encountered. The error is in topic prediction. The errors are fixed, and the system worked properly.

5.4.2. Non-functional Capabilities

Table 5.4.1 shows the non-functional capabilities, testing method and the results.

Capabilities	Testing Method	Results
Usability	The users are asked to rate out of 5 for user interface, user experience and easiness to understand the system. If the average rating is 3, it is considered pass.	Pass
Performance	Since this website processes a lot of back-end works, 30 seconds to 1 minute should be enough to load all the content in the layout. It is considered pass if within time range.	Pass after some fixing
Security	Security testing is conducted on test field input. A simple and harmless script code, '<script>window.open('https://www.google.com');</script>', has been injected in the text field and see what happened after clicking submit button. If the system is not secure, then this script will lead to website Google. If nothing happens, it is considered as pass.	Pass

Table 5.4.1 Results of Non-Functional Capabilities

5.5. Test Evaluation & Summary

During the testing, the problems faced are manageable but the only problem that ruins the performance of the system is the initial page loading takes longer time especially for Sector Identification page. The problem has been detected. This is due to two factors. Firstly, it takes a month-volume of text to display monthly graph. Next, is the nature of PHP itself where it runs codes line by line. The problem has been solved according to Dr. Sukumar's advice during the project demonstration. He suggested that the layout of the page need to load first and then the content. To solve this, a JavaScript was included alongside with PHP to load the content.

6 CONCLUSION & FUTURE WORK

In conclusion, time series analysis on social media content aids users to gain valuable insights. As social media very popular nowadays, it is very important for businesses specially to utilize social media in order to understand their customers better and use it to promote the product(s). Topic prediction feature offered in this system could play an important role for organizations to plan their future campaign.

Time series analysis already exist long ago. The operations can be done are visualization, identifying trend, seasonality and cycle, and forecasting. Now, with the current technology development, time series analysis can be done more than that. For instance, fast forecasting, classification and clustering can be done with the help of machine learning. Plus, text mining is also possibly done in current technology development.

Moreover, with the help of supervisor, design and analysis of the proposed solution are finally done. Proper work planning and some analysis diagrams help to understand the proposed solution more. The feedback from examiners during Analysis and Design Presentation, and Project Demo helping in improving the system. The final outcome of this project can finally be delivered successfully.

However, positives aside, some critiques must be made and improve it in the future. Although some efficient methods/algorithms have been determined in the literature review, they still require more exploration and testing to validate the accuracy and efficiency of the algorithm/method. Now, after implementation, the accuracy of some the models such as SVM and auto ARIMA can be determined. Nevertheless, the efficiency of the models is needed to be determine. This is crucial especially for sector classifying model because Sector Identification Modules requires to classify monthly tweets. The bigger the data, the slow sector classifying process.

Last but not least, another critique can be made is on the information gathering method. Literature review is just helping developer in doing machine learning processes. Some methods are needed such as survey. From the survey, the users' needs can be determined. This is because the system is developed for users and not for developers. Unfortunately, due to time constraints, user acceptance test cannot be

done. User acceptance test also can help in understanding users' needs. Hopefully, in the future, user acceptance test can be done in order to make the system user-friendly.

REFERENCES

1. Zhu, R. (2016, May 01). Time series analysis for event detection in text data. Retrieved October 30, 2020, from http://hdl.handle.net/2142/90948

2. 15 Step Social Media Marketing Strategy for Businesses in 2020. (2020, September 09). Retrieved October 28, 2020, from https://influencermarketinghub.com/social-media-marketing-strategy/
3. Mohsin, M. (2020, October 19). 10 Social Media Statistics You Need to Know in 2020 [Infographic]. Retrieved October 28, 2020, from https://www.oberlo.com/blog/social-media-marketing-statistics
4. Alford, E. (2020, March 13). "Data complexity" is one of the biggest challenges for marketers right now, and current tools aren't helping enough. Retrieved October 28, 2020, from https://www.clickz.com/data-complexity-is-one-of-the-biggest-challenges-for-marketers-right-now-and-current-tools-arent-helping-enough/228591/

5. Newberry, C. (2020, September 09). 13 of the Best Social Media Analytics Tools (Free and Paid). Retrieved October 30, 2020, from https://blog.hootsuite.com/social-media-analytics/
6. Hootsuite. (2020, November 22). Retrieved November 27, 2020, from https://en.wikipedia.org/wiki/Hootsuite
7. Google Analytics. (2020, November 16). Retrieved November 27, 2020, from https://en.wikipedia.org/wiki/Google_Analytics
8. Singer, P. (2011, Sep 26). Time Series Analysis of Online Social Network Data and Content. Retrieve November, 28, 2020 https://www.philippsinger.info/files/masterarbeit_psinger.pdf
9. Ogayo, P. (2020, February 14). Time Series Analysis for Beginners. Retrieved November 27, 2020, from https://towardsdatascience.com/time-series-analysis-for-beginners-8a200552e332
10. Brownlee, J. (2020, August 20). 11 Classical Time Series Forecasting Methods in Python (Cheat Sheet). Retrieved November 27, 2020, from https://machinelearningmastery.com/time-series-forecasting-methods-in-python-cheat-sheet/

11. Chalaguine, L. (2020, August 03). Getting started with text analysis in Python. Retrieved November 27, 2020, from https://towardsdatascience.com/getting-started-with-text-analysis-in-python-ca13590eb4f7
12. Subramanian, D. (2020, June 11). Text Mining in Python: Steps and Examples. Retrieved November 27, 2020, from https://medium.com/towards-artificial-intelligence/text-mining-in-python-steps-and-examples-78b3f8fd913b
13. Naushan, H. (2020, October 02). Sentiment Analysis of Social Media with Python. Retrieved November 29, 2020, from https://towardsdatascience.com/sentiment-analysis-of-social-media-with-python-45268dc8f23f
14. TextBlob Sentiment: Calculating Polarity and Subjectivity. (2015, June 7). https://planspace.org/20150607-textblob_sentiment/
15. Prabhakaran, S. (2021, June 14). *ARIMA Model - Complete Guide to Time Series Forecasting in Python: ML+*. Learn applied Data Science. https://www.machinelearningplus.com/arima-model-time-series-forecasting-python/

APPENDICES

Appendix A: Use Case Description

Use Case	View time series of current topics	
Scenario	View time series of current topics in Twitter	
Triggering Event	User wants to see time series of current topics in Twitter	
Brief Description	The system will display time series of current topics in Twitter to user	
Actor(s) Involved	User	
Related Use Case	Add tweets	
Stakeholder Involved	User	
Precondition	Tweets has been added	
Postcondition	-	
Flow of Activity	Actor	System
	1. User views time series of current topics	1.1 The system fetch tweets from database 1.2 The system preprocessing text stream from tweets into data 1.3 The system clusters the data into its topics by using machine learning algorithm 1.4 The system classifies data into its topic by using trained model 1.5 The system groups data in date and hour for each topic 1.6 The system plots the time series based on the frequency of each topic
Exception Condition	User has no internet connection	

Appendix A 1 Use Case Description of View Time Series of Current Topics

Use Case	View percentage of sectors	
Scenario	View percentage of sectors	
Triggering Event	User wants to see percentage/proportion of each sector	
Brief Description	The system will display percentage of each sector	
Actor(s) Involved	User	
Related Use Case	Add tweets, view time series of sectors	
Stakeholder Involved	User	
Precondition	Tweets has been added Time series of sectors has been displayed	
Postcondition	-	
Flow of Activity	Actor	System
	1. User views percentage of each sector	1.1 From the time series, the system takes total of each sector 1.2 The system calculates the percentage of each sector 1.3 The system displays the percentage of each sector
Exception Condition	User has no internet connection	

Appendix A 2 Use Case Description of View Percentage of Sectors

Use Case	View percentage of sentiments	
Scenario	View percentage of sentiments	
Triggering Event	User wants to see percentage/proportion of each sentiment	
Brief Description	The system will display percentage of each sentiment	
Actor(s) Involved	User	
Related Use Case	Add tweets, view time series of sentiments	
Stakeholder Involved	User	
Precondition	Tweets has been added Time series of sentiments has been displayed	
Postcondition	-	
Flow of Activity	Actor	System
	1. User views percentage of each sentiment	1.1 From the time series, the system takes total of each sentiment 1.2 The system calculates the percentage of each sentiment 1.3 The system displays the percentage of each sentiment
Exception Condition	User has no internet connection	

Appendix A 3 Use Case Description of View Percentage of Sentiment

<table>
<tr><td>Use Case</td><td colspan="2">View current topics</td></tr>
<tr><td>Scenario</td><td colspan="2">View current topics in Twitter</td></tr>
<tr><td>Triggering Event</td><td colspan="2">User wants to see current topics in Twitter</td></tr>
<tr><td>Brief Description</td><td colspan="2">The system will display current topics in Twitter</td></tr>
<tr><td>Actor(s) Involved</td><td colspan="2">User</td></tr>
<tr><td>Related Use Case</td><td colspan="2">Add tweets, view time series of current topics</td></tr>
<tr><td>Stakeholder Involved</td><td colspan="2">User</td></tr>
<tr><td>Precondition</td><td colspan="2">Tweets has been added

Time series of current topics has been displayed</td></tr>
<tr><td>Postcondition</td><td colspan="2">-</td></tr>
<tr><td rowspan="2">Flow of Activity</td><td>Actor</td><td>System</td></tr>
<tr><td>1. User views current topics in Twitter</td><td>1.1 From the trained model, the system will take terms and scores for each topic

1.2 The system displays the topics</td></tr>
<tr><td>Exception Condition</td><td colspan="2">User has no internet connection</td></tr>
</table>

Appendix A 4 Use Case Description of View Current Topics

Use Case	Topic prediction	
Scenario	Predicting the trend of a topic in time series	
Triggering Event	User wants to see the trend of a topic in Twitter for the next few hours/weeks/months	
Brief Description	The system will display the trend of a topic in Twitter for the next few hours/weeks/months	
Actor(s) Involved	User	
Related Use Case	Add tweets, view time series of current topics	
Stakeholder Involved	User	
Precondition	Tweets has been added Time series of current topics has been displayed	
Postcondition	-	
Flow of Activity	Actor	System
	1. User choses a topic he/she want to predict 2. User view trend of topics	1.1 System identifies topic chosen by the user 1.2 From the time a topic chosen, the system uses regression algorithm to forecast 2.1 The system displays the initial values and forecasted values of a topic
Exception Condition	User has no internet connection	

Appendix A 5 Use Case Description of Topic Prediction

Use Case	View current topics of each sector	
Scenario	View current topics of each sector	
Triggering Event	User wants to view current topics in Twitter based on the sectors	
Brief Description	The system will display current topics in Twitter based on the sectors	
Actor(s) Involved	User	
Related Use Case	Add tweets, view time series of sectors	
Stakeholder Involved	User	
Precondition	Tweets has been added Time series of sectors has been displayed	
Postcondition	-	
Flow of Activity	Actor	System
	1. User views the list of current popular topics based on the sectors	1.1 The system reads sector classified tweets 1.2 The system preprocessing text stream from tweets into data 1.3 The system clusters the data into its topics by using machine learning algorithm 1.4 The system takes the term and scores for each topic 1.5 The system display the topics
Exception Condition	User has no internet connection	

Appendix A 6 Use Case Description of View Current Topics of Each Sector

Use Case	View current topics of each sentiment	
Scenario	View current topics of each sentiment	
Triggering Event	User wants to view current topics in Twitter based on the sentiments	
Brief Description	The system will display the current topics in Twitter based on the sentiments	
Actor(s) Involved	User	
Related Use Case	Add tweets, view time series of sentiments	
Stakeholder Involved	User	
Precondition	Tweets has been added Time series of sentiments has been displayed	
Postcondition	-	
Flow of Activity	Actor	System
	1. User views the list of current popular topics based on the sentiments	1.1 The system reads sentiment classified tweets 1.2 The system preprocessing text stream from tweets into data 1.3 The system clusters the data into its topics by using machine learning algorithm 1.4 The system takes the term and scores for each topic 1.5 The system display the topics
Exception Condition	User has no internet connection	

Appendix A 7 Use Case Description of View Current Topic of Each Sentiment

Use Case	Change time series setting	
Scenario	Change time series setting	
Triggering Event	User wants to change time series setting	
Brief Description	User will change the setting of time series by picking option date range	
Actor(s) Involved	User	
Related Use Case	View time series of sectors, view time series of sentiments, view time series of current topics	
Stakeholder Involved	User	
Precondition	Time series of sectors, sentiments or current topics has been displayed	
Postcondition	-	
Flow of Activity	Actor	System
	1. User picks the date range 2. User confirms the settings	2.1 System identifies the value of the date range 2.2 The system fetches tweets based on date range
Exception Condition	User has no internet connection	

Appendix A 8 Use Case Description of Change Time Series Setting

<table>
<tr><td>Use Case</td><td colspan="2">View popular tweets</td></tr>
<tr><td>Scenario</td><td colspan="2">View popular tweets of chosen word in topic</td></tr>
<tr><td>Triggering Event</td><td colspan="2">User wants to view popular tweets of chosen word in a topic</td></tr>
<tr><td>Brief Description</td><td colspan="2">The system will display tweets based on the word chosen</td></tr>
<tr><td>Actor(s) Involved</td><td colspan="2">User</td></tr>
<tr><td>Related Use Case</td><td colspan="2">View current popular topics of each sector, view current popular topics of each sentiment, View current topics</td></tr>
<tr><td>Stakeholder Involved</td><td colspan="2">User</td></tr>
<tr><td>Precondition</td><td colspan="2">The current topics, current topics of each sector or current topics of sentiment has been displayed</td></tr>
<tr><td>Postcondition</td><td colspan="2">-</td></tr>
<tr><td rowspan="2">Flow of Activity</td><td>Actor</td><td>System</td></tr>
<tr><td>1. User clicks on a word in a topic</td><td>1.1 The system searches tweets based on word chosen

1.1.The system displays list of tweets</td></tr>
<tr><td>Exception Condition</td><td colspan="2">User has no internet connection</td></tr>
</table>

Appendix A 9 Use Case Description of View Popular Tweets

<table>
<tr><td>Use Case</td><td colspan="2">Create account</td></tr>
<tr><td>Scenario</td><td colspan="2">Creating account for a user</td></tr>
<tr><td>Triggering Event</td><td colspan="2">User wants to create an account</td></tr>
<tr><td>Brief Description</td><td colspan="2">The system will create account of the user</td></tr>
<tr><td>Actor(s) Involved</td><td colspan="2">User</td></tr>
<tr><td>Related Use Case</td><td colspan="2">-</td></tr>
<tr><td>Stakeholder Involved</td><td colspan="2">User</td></tr>
<tr><td>Precondition</td><td colspan="2">-</td></tr>
<tr><td>Postcondition</td><td colspan="2">An account has created</td></tr>
<tr><td rowspan="2">Flow of Activity</td><td>Actor</td><td>System</td></tr>
<tr><td>1. User fills in information

2. User click submit</td><td>2.1 The system identifies user’s inputs

2.2 The system check in database if username is unique

2.3. If the username is unique, the system sends user’s information to be stored in database</td></tr>
<tr><td>Exception Condition</td><td colspan="2">User has no internet connection</td></tr>
</table>

Appendix A 10 Use Case Description of Creating Account

<table>
<tr><td>Use Case</td><td colspan="2">Login</td></tr>
<tr><td>Scenario</td><td colspan="2">Login an account</td></tr>
<tr><td>Triggering Event</td><td colspan="2">User wants to login</td></tr>
<tr><td>Brief Description</td><td colspan="2">The user will login into his/her account</td></tr>
<tr><td>Actor(s) Involved</td><td colspan="2">User</td></tr>
<tr><td>Related Use Case</td><td colspan="2">Create account</td></tr>
<tr><td>Stakeholder Involved</td><td colspan="2">User</td></tr>
<tr><td>Precondition</td><td colspan="2">User has an account</td></tr>
<tr><td>Postcondition</td><td colspan="2">User logged in</td></tr>
<tr><td rowspan="2">Flow of Activity</td><td>Actor</td><td>System</td></tr>
<tr><td>1. User fills in information

2. User click submit</td><td>2.1 The system identifies user’s inputs

2.2 The system check in database if username is exist

2.3. If the username is exist, then the system checks the password is valid

2.4 If valid, the system logged in user</td></tr>
<tr><td>Exception Condition</td><td colspan="2">User has no internet connection</td></tr>
</table>

Appendix A 11 Use Case Description of Login

Use Case	Logout	
Scenario	Logout an account	
Triggering Event	User wants to logout	
Brief Description	The user will logout from his/her account	
Actor(s) Involved	User	
Related Use Case	Create account, login	
Stakeholder Involved	User	
Precondition	User has an account, user logged in	
Postcondition	User logged out	
Flow of Activity	Actor	System
	1. User click logout	1.1 The system logged user out
Exception Condition	User has no internet connection	

Appendix A 12 Use Case Description of Logout

<table>
<tr><td>Use Case</td><td colspan="2">View monthly sectors</td></tr>
<tr><td>Scenario</td><td colspan="2">View monthly sectors</td></tr>
<tr><td>Triggering Event</td><td colspan="2">User wants to see monthly frequency each sector</td></tr>
<tr><td>Brief Description</td><td colspan="2">The system will display monthly frequency of each sector</td></tr>
<tr><td>Actor(s) Involved</td><td colspan="2">User</td></tr>
<tr><td>Related Use Case</td><td colspan="2">Add tweets</td></tr>
<tr><td>Stakeholder Involved</td><td colspan="2">User</td></tr>
<tr><td>Precondition</td><td colspan="2">Tweets has been added</td></tr>
<tr><td>Postcondition</td><td colspan="2">-</td></tr>
<tr><td rowspan="2">Flow of Activity</td><td>Actor</td><td>System</td></tr>
<tr><td>1. User views monthly frequency of each sector</td><td>1.1 The system fetches monthly tweets

1.2 The system classifies tweets into its sector after preprocessing the tweets

1.3 The system display frequency of monthly tweets for each sector</td></tr>
<tr><td>Exception Condition</td><td colspan="2">User has no internet connection</td></tr>
</table>

Appendix A 13 Use Case Description of View Monthly Sectors

<table>
<tr><td>Use Case</td><td colspan="2">View tweets polarity</td></tr>
<tr><td>Scenario</td><td colspan="2">View tweets polarity</td></tr>
<tr><td>Triggering Event</td><td colspan="2">User wants to see polarity of tweets for each sentiment</td></tr>
<tr><td>Brief Description</td><td colspan="2">The system will display polarity of tweets for each sentiment</td></tr>
<tr><td>Actor(s) Involved</td><td colspan="2">User</td></tr>
<tr><td>Related Use Case</td><td colspan="2">Add tweets, view time series of sentiments</td></tr>
<tr><td>Stakeholder Involved</td><td colspan="2">User</td></tr>
<tr><td>Precondition</td><td colspan="2">Tweets has been added, time series of sentiments has been displayed</td></tr>
<tr><td>Postcondition</td><td colspan="2">-</td></tr>
<tr><td rowspan="2">Flow of Activity</td><td>Actor</td><td>System</td></tr>
<tr><td>1. User views tweets polarity for each sentiment</td><td>1.1 The system reads sentiment classified data

1.2 The system sorts the tweets based on polarity

1.3 The system display tweets polarity for each sentiment</td></tr>
<tr><td>Exception Condition</td><td colspan="2">User has no internet connection</td></tr>
</table>

Appendix A 14 Use Case Description of View Tweets Polarity

Use Case	Save topics	
Scenario	Saving topics	
Triggering Event	User wants to save topic	
Brief Description	The user fills in interpretation of each topic and save it along with time series	
Actor(s) Involved	User	
Related Use Case	view time series of current topics, view current topics	
Stakeholder Involved	User	
Precondition	Tweets has been added, time series of current topics has been displayed	
Postcondition	Topic has been saved	
Flow of Activity	Actor	System
	1. User fills in interpretation of each topics 2. The user clicks confirm	1.1 The system identifies interpretation of each topics 2.1 The system save the topic model, interpretation and time series
Exception Condition	User has no internet connection	

Appendix A 15 Use Case Description of Save Topics

Use Case	Read saved topics	
Scenario	Read saved topics	
Triggering Event	User wants to read saved topics	
Brief Description	The user click on one of his/her saved topics and display it	
Actor(s) Involved	User	
Related Use Case	Save topics	
Stakeholder Involved	User	
Precondition	Topics have been saved	
Postcondition	-	
Flow of Activity	Actor	System
	1. User click on one of his/her topic	1.1 The system identifies the topic selected 1.2 The system load saved topic model 1.3 The system displays topics, interpretation and time series
Exception Condition	User has no internet connection	

Appendix A 16 Use Case Description of Read Saved Topics

Use Case	View news headlines	
Scenario	View news headlines	
Triggering Event	User wants to read news headlines	
Brief Description	The system display news headlines	
Actor(s) Involved	User	
Related Use Case	-	
Stakeholder Involved	User	
Precondition	-	
Postcondition	-	
Flow of Activity	Actor	System
	1. User sees news headlines	1.1 The system displays news headlines
Exception Condition	User has no internet connection	

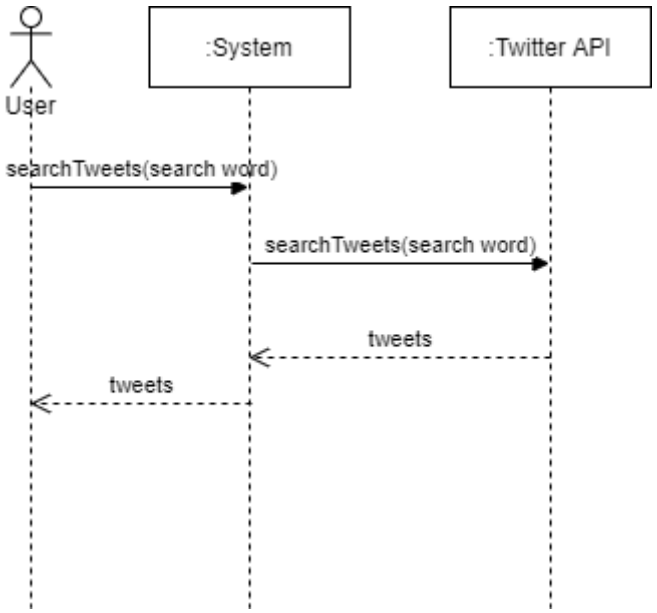
Appendix A 17 Use Case Description of View News Headlines

Use Case	Search news	
Scenario	Searching news	
Triggering Event	User wants to search news	
Brief Description	The system will display news result based on the search word	
Actor(s) Involved	User	
Related Use Case	-	
Stakeholder Involved	User	
Precondition	-	
Postcondition	-	
Flow of Activity	Actor	System
	1. User fills in sear word 2. User clicks submit	2.1 The system identifies user's input 2.2 The system displays news search results
Exception Condition	User has no internet connection	

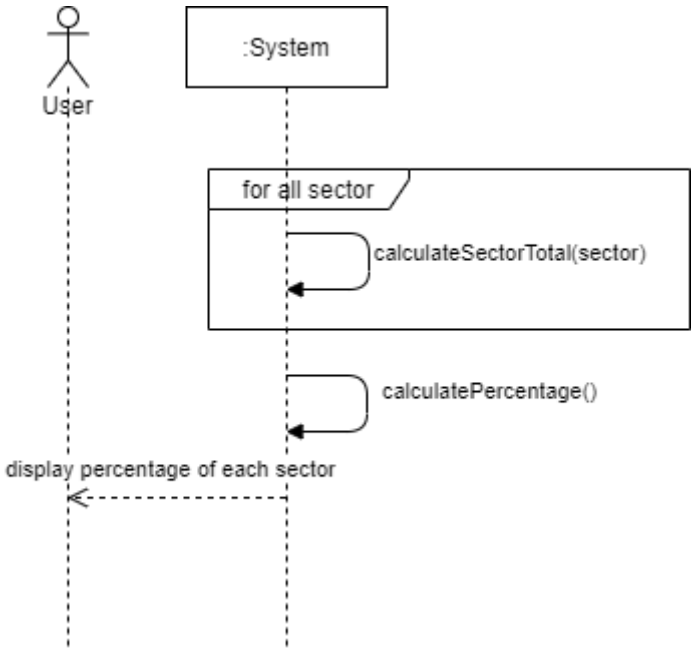
Appendix A 18 Use Case Description of Search News

Appendix B: Sequence State Diagram

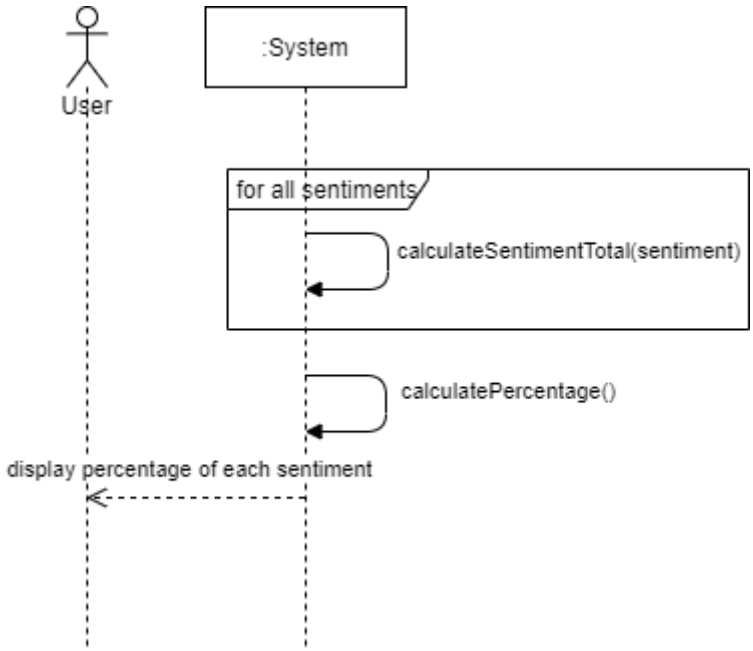
Appendix B 1SSD of View Time Series of Current Topics



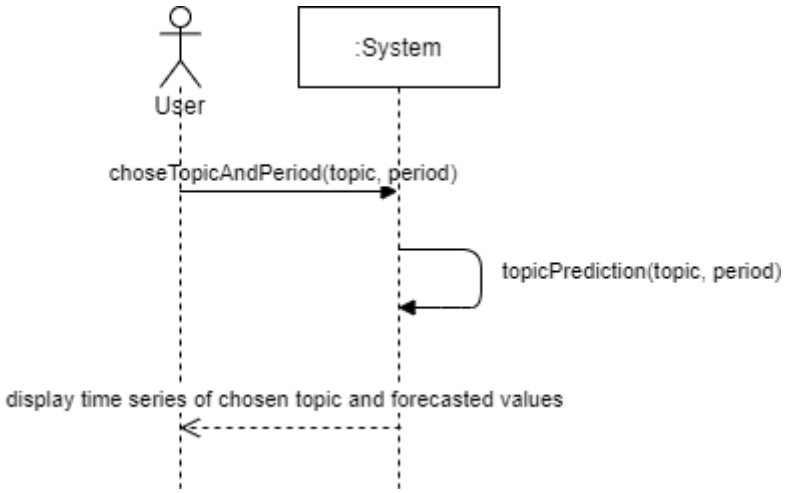
Appendix B 2 SSD of View popular Tweets



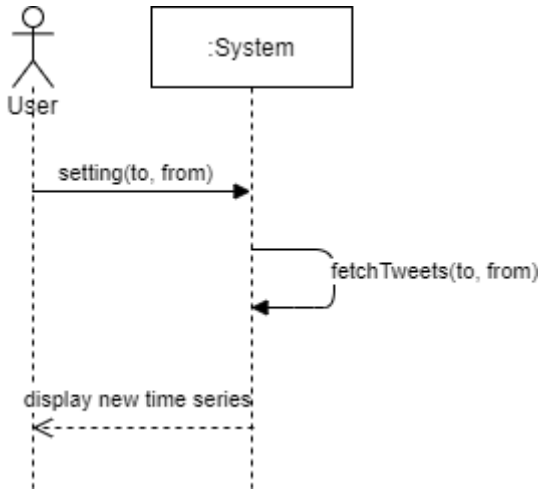
Appendix B 3 SSD of View Percentage of Sectors



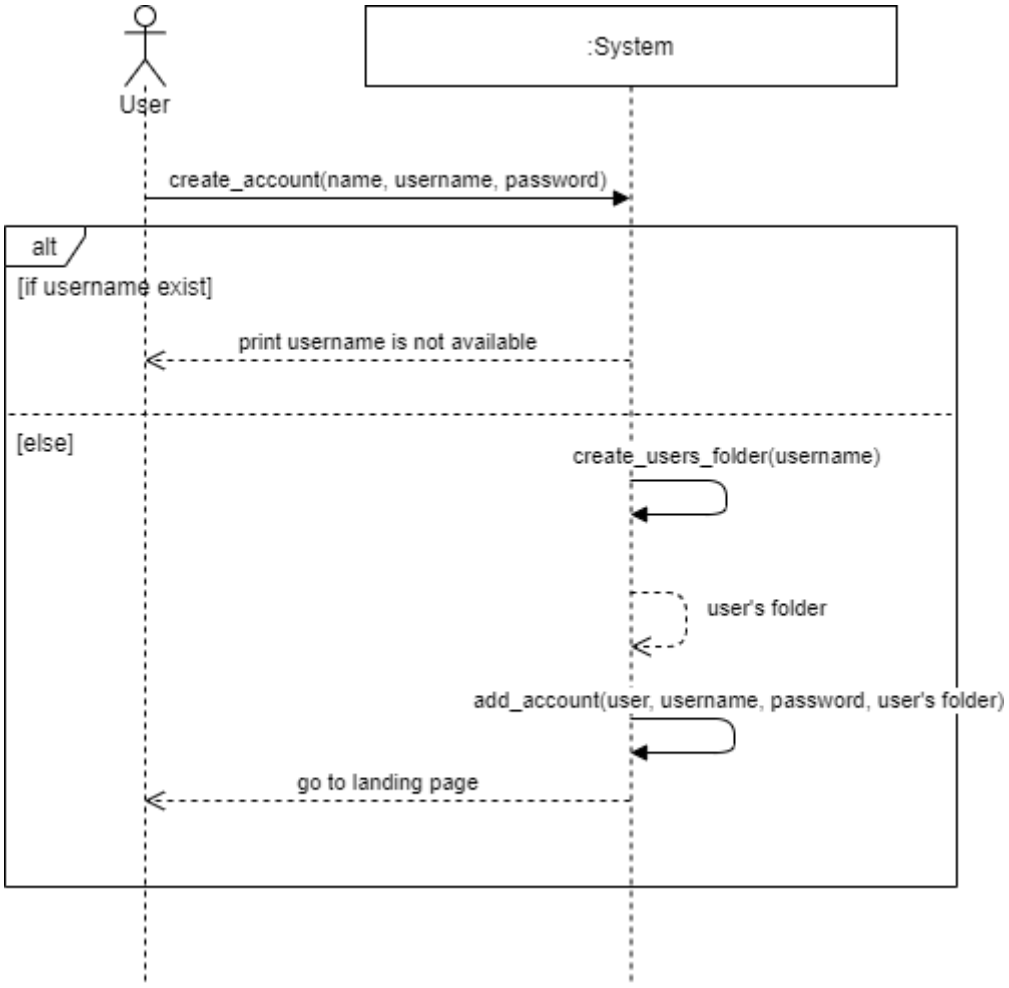
Appendix B 4 SSD of View Percentage of Sentiments



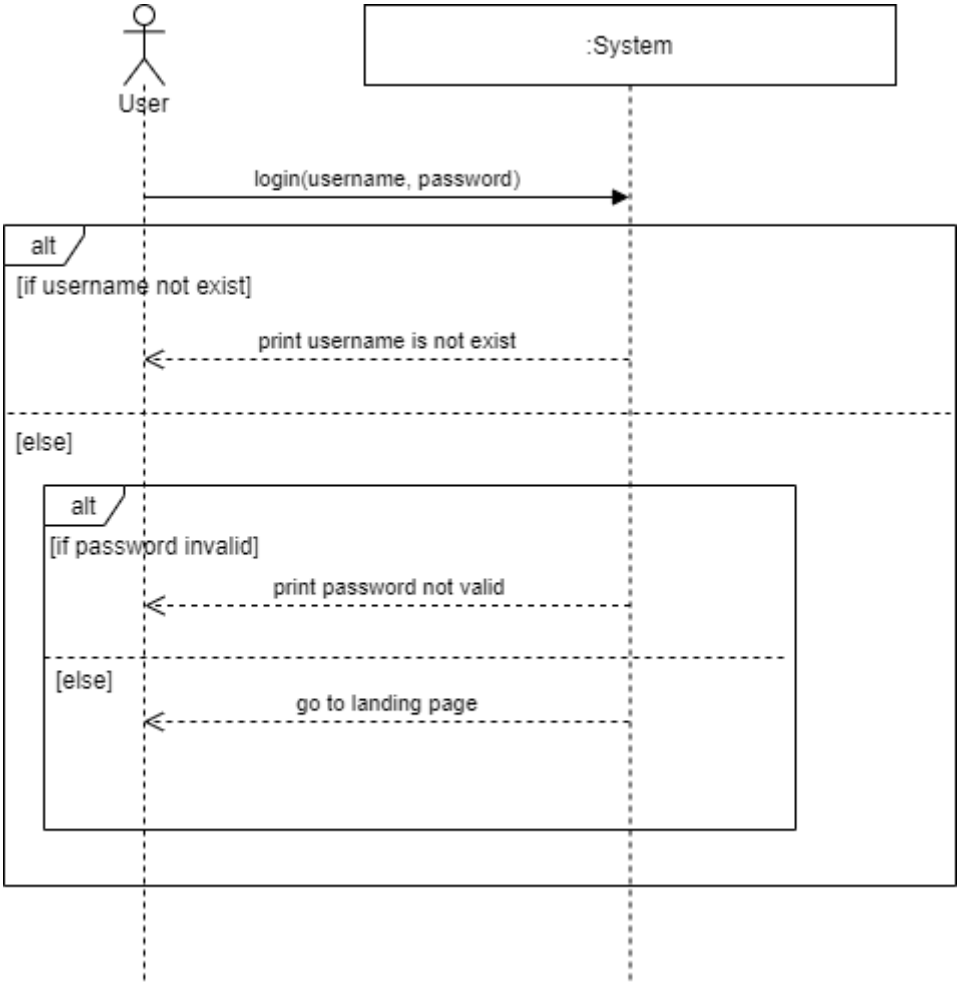
Appendix B 5 SSD of Topic Prediction



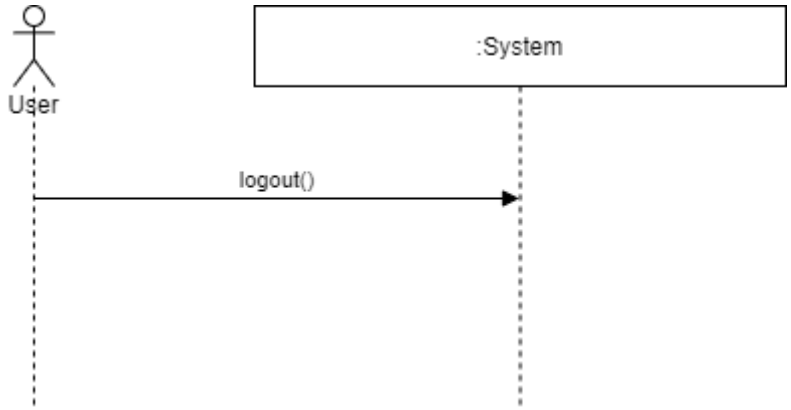
Appendix B 6 SSD of Change Time Series Setting



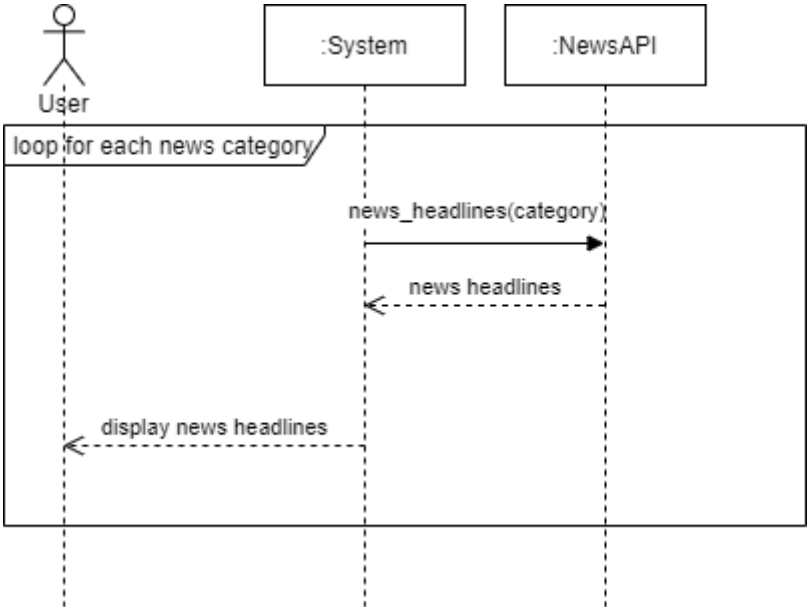
Appendix B 7 SSD of Create Account



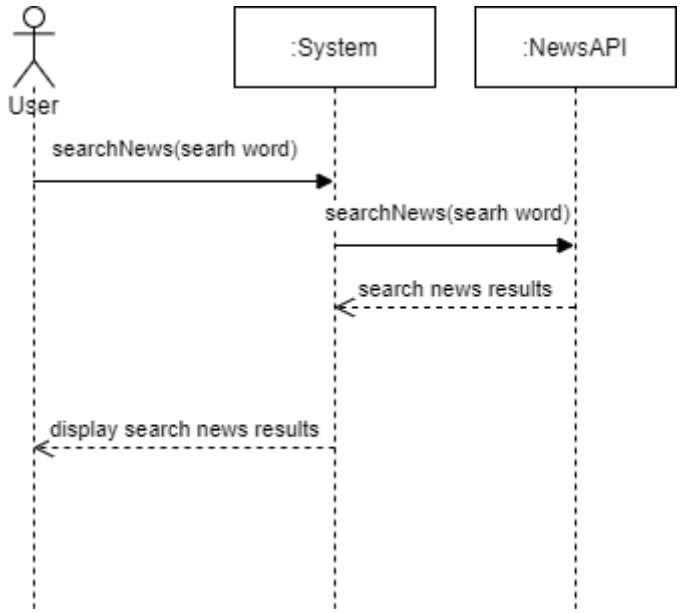
Appendix B 8 SSD of Login



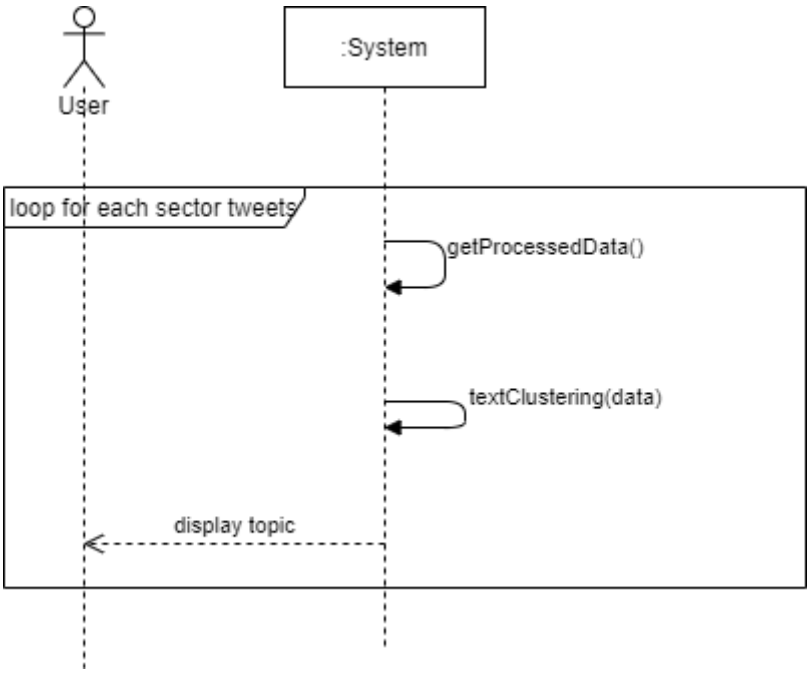
Appendix B 9 SSD of Logout



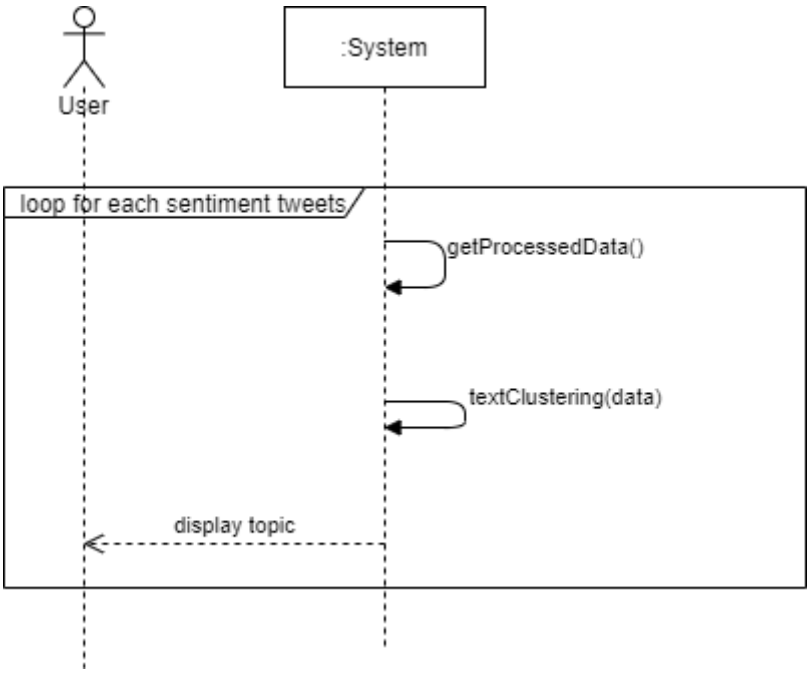
Appendix B 10 SSD of View News Headlines



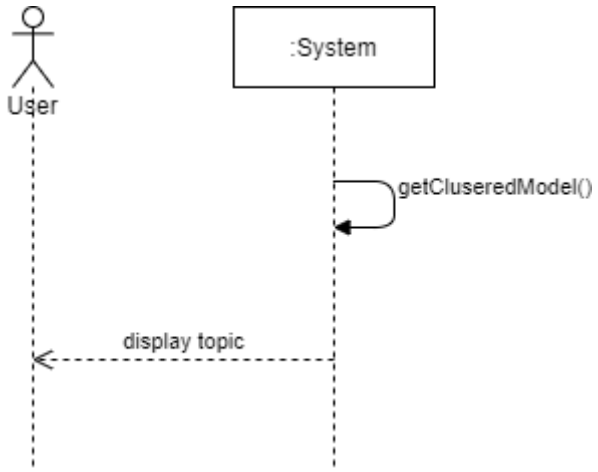
Appendix B 11 SSD of Search News



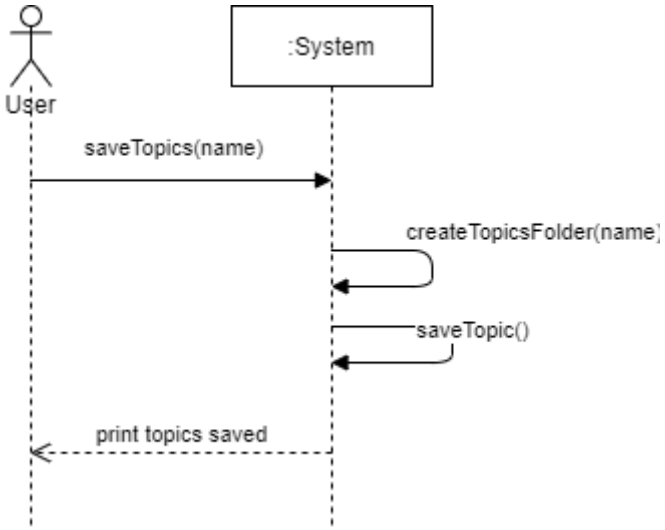
Appendix B 12 SSD of View Current Topics of Each Sector



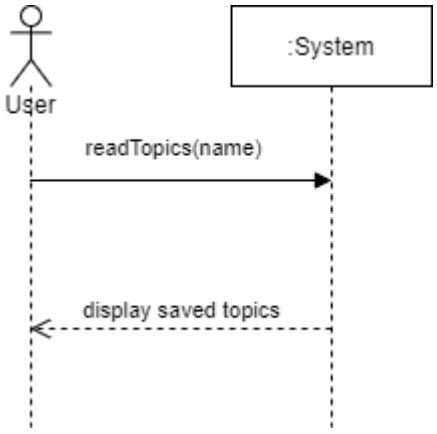
Appendix B 13 SSD of View Current Topics of Each Sentiment



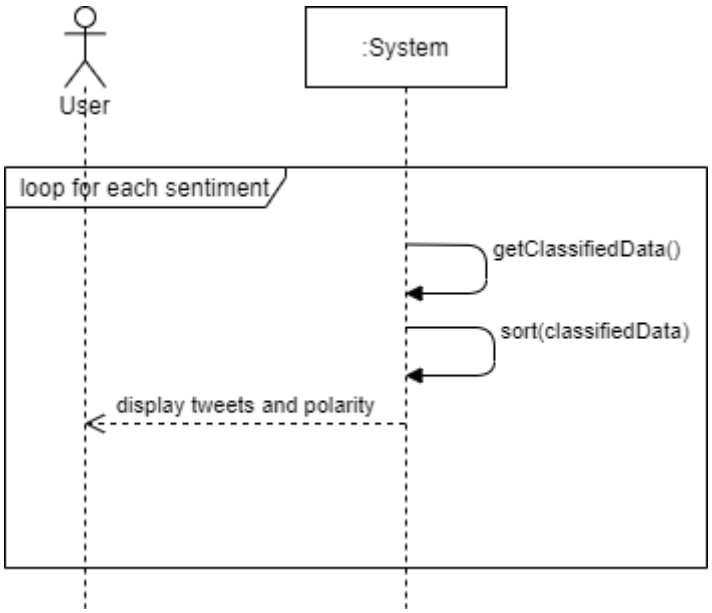
Appendix B 14 SSD of View Current Topics



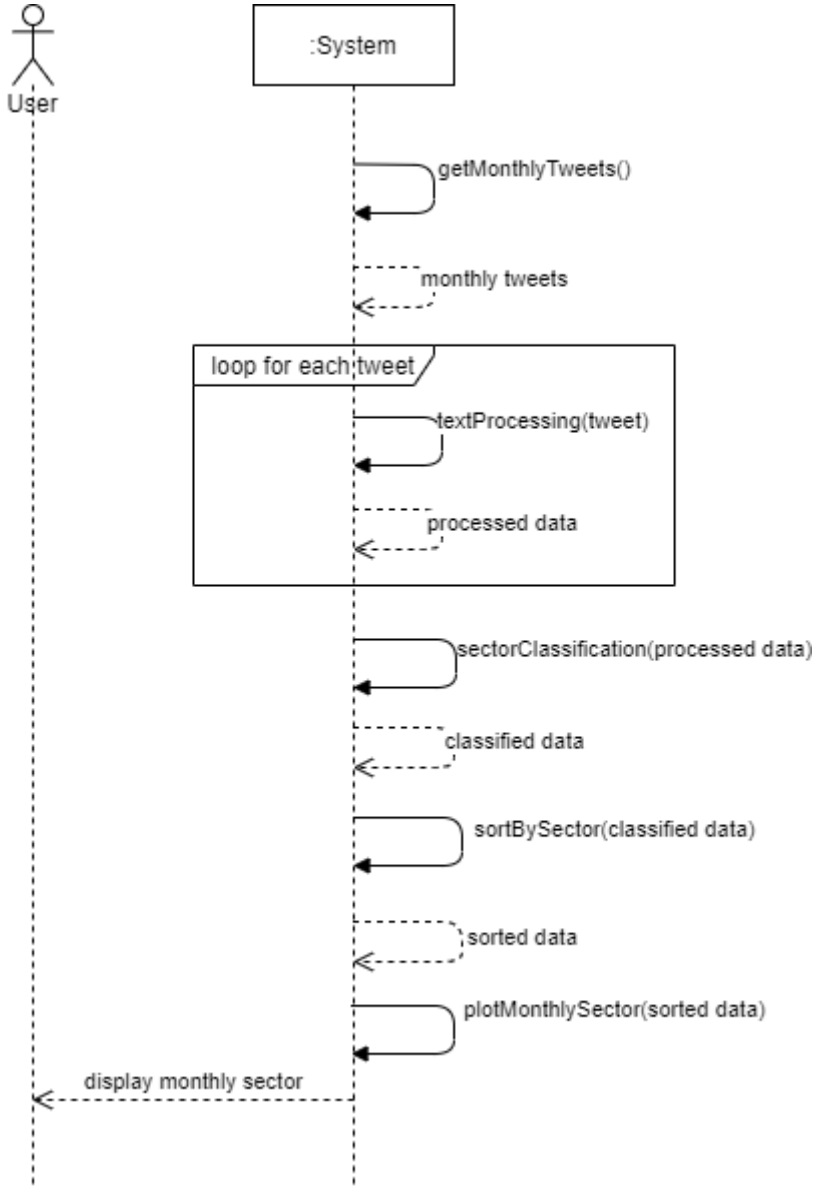
Appendix B 15 SSD of Save Topics



Appendix B 16 SSD of Read Saved Topics

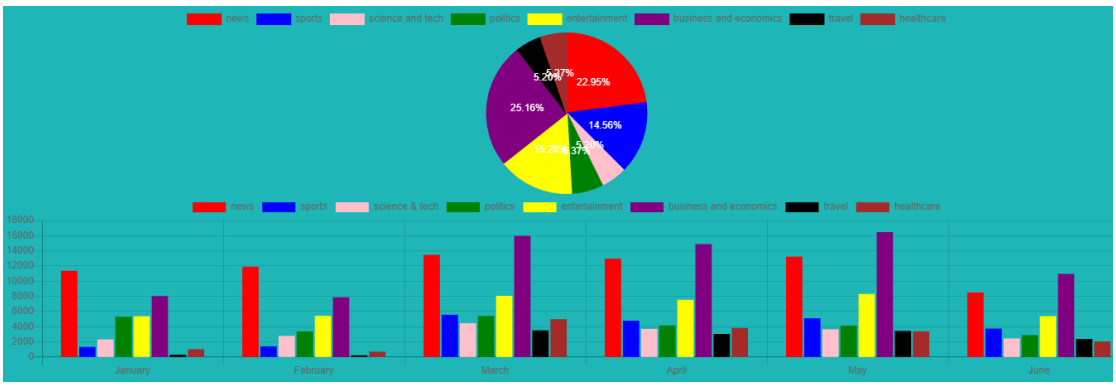


Appendix B 17 SSD of View Tweets Polarity



Appendix B 18 SSD of View Monthly Sectors

Appendix C: User Interface



Percentage of Sectors and Monthly Sectors

News Headlines

Appendix D: Test Cases

Use Case	View time series of current topics
Description	The system will automatically plot time series of current topics from the tweets fetched.
Steps Involved	1. The system read the csv file of tweets 2. The system cluster the text and identify topic for each tweet 3. The system plot time series
Input Data	A CSV file of tweets consists columns of created_at, full_text and screen_name
Expected Result	Plot time series fail if 1) the CSV file is not exists 2) the CSV file is empty 3) no column name in CSV file 4) one of the column is not exist 5) no internet connection
Actual Result	Plot time series fail if 1) the csv file is not exists 2) the csv file is empty 3) no column name in csv file 4) one of the column is not exist 5) no internet connection
Pass/Fail	Pass

Appendix D 1 Test Case of View Time Series of Current Topics

Use Case	Create account
Description	The user desire to create an account
Steps Involved	1. The user fills in information needed 2. The user clicks submit 3. The system creates user's folder to save topics 4. The save user's information in database
Input Data	Name: Adam Fikri, Username: adamfikri, Password: Adam123456, Topics Path: users/adamfikri
Expected Result	Create account fail if 1) all field empty 2) one of the fields is empty 3) no database connection 4) no internet connection
Actual Result	Create account fail if 1) all field empty 2) one of the fields is empty 3) no database connection 4) no internet connection
Pass/Fail	Pass

Appendix D 2 Test Case of Create Account

Use Case	Login
Description	The user desire to login into account
Steps Involved	1. The user fills in information needed 2. The user clicks submit 3. The system fetch user's info based on username 4. The system check if user exist or invalid password
Input Data	Username: adamfikri, Password: Adam123456
Expected Result	Login fail if 1) all field empty 2) one of the fields is empty 3) no database connection 4) no internet connection 5) username is not exist 6) invalid password
Actual Result	Login fail if 1) all field empty 2) one of the fields is empty 3) no database connection 4) no internet connection 5) username is not exist 6) invalid password
Pass/Fail	Pass

Appendix D 3 Test Case of Login

Use Case	Logout
Description	The user desire to logout from account
Steps Involved	1. The user click logout 2. The system logout user
Input Data	-
Expected Result	Logout fail if no internet connection
Actual Result	Logout fail if no internet connection
Pass/Fail	Pass

Appendix D 4 Test Case of Logout

Use Case	View Percentage of Sectors
Description	The system shows percentage of each sector
Steps Involved	1. The system read CSV file 2. The system plot pie chart and display percentage
Input Data	A CSV file with columns of sector and frequency
Expected Result	View percentage fail if 1) the CSV file is not exists 2) the CSV file is empty 3) no column name in CSV file 4) one of the column is not exist 5) no internet connection
Actual Result	View percentage fail if 1) the CSV file is not exists 2) the CSV file is empty 3) no column name in CSV file 4) one of the column is not exist 5) no internet connection
Pass/Fail	Pass

Appendix D 5 Test Case of View Percentage of Sectors

Use Case	View Percentage of Sentiments
Description	The system shows percentage of each sentiment
Steps Involved	1. The system read CSV file 2. The system plot pie chart and display percentage
Input Data	A CSV file with columns of sentiment and frequency
Expected Result	View percentage fail if 1) the CSV file is not exists 2) the CSV file is empty 3) no column name in CSV file 4) one of the column is not exist 5) no internet connection
Actual Result	View percentage fail if 1) the CSV file is not exists 2) the CSV file is empty 3) no column name in CSV file 4) one of the column is not exist 5) no internet connection
Pass/Fail	Pass

Appendix D 6 Test Case of View Percentage of Sentiments

Use Case	View Monthly Sectors
Description	The system shows monthly frequency of each sector
Steps Involved	1. The system read CSV file 2. The system plot bar chart
Input Data	A monthly CSV file with columns of sector and frequency
Expected Result	View monthly sector fail if 1) the CSV file is not exists 2) the CSV file is empty 3) no column name in CSV file 4) one of the column is not exist 5) no internet connection
Actual Result	View monthly sector fail if 1) the CSV file is not exists 2) the CSV file is empty 3) no column name in CSV file 4) one of the column is not exist 5) no internet connection
Pass/Fail	Pass

Appendix D 7 Test Case of View Monthly Sectors

Use Case	View Tweets Polarity
Description	The system shows top 10 tweets for each sentiment based on polarity
Steps Involved	1. The system read CSV file 2. The system sort tweets based on polarity 3. The system show the tweets and its polarity
Input Data	A CSV file with columns of full_text, sentiment and polarity
Expected Result	View tweets polarity fail if 1) the CSV file is not exists 2) the CSV file is empty 3) no column name in CSV file 4) one of the column is not exist 5) no internet connection
Actual Result	View tweets polarity fail if 1) the CSV file is not exists 2) the CSV file is empty 3) no column name in CSV file 4) one of the column is not exist 5) no internet connection
Pass/Fail	Pass

Appendix D 8 Test Case of View Tweets Polarity

Use Case	View Current Topics of Each Sector
Description	The system shows 10 topics for each sector
Steps Involved	1. The system read CSV file 2. The system clusters the text 3. The system print out topics
Input Data	CSV files of each sector with columns of created_at, full_text, screen_names
Expected Result	View topics fail if 1) the CSV file is not exists 2) the CSV file is empty 3) no column name in CSV file 4) one of the column is not exist 5) no internet connection
Actual Result	View topics fail if 1) the CSV file is not exists 2) the CSV file is empty 3) no column name in CSV file 4) one of the column is not exist 5) no internet connection
Pass/Fail	Pass

Appendix D 9 Test Case of View Current Topics of Each Sector

Use Case	View Current Topics of Each Sentiment
Description	The system shows 10 topics for each sentiment
Steps Involved	1. The system read CSV file 2. The system clusters the text 3. The system print out topics
Input Data	CSV files of each sentiment with columns of created_at, full_text, screen_names
Expected Result	View topics fail if 1) the CSV file is not exists 2) the CSV file is empty 3) no column name in CSV file 4) one of the column is not exist 5) no internet connection
Actual Result	View topics fail if 1) the CSV file is not exists 2) the CSV file is empty 3) no column name in CSV file 4) one of the column is not exist 5) no internet connection
Pass/Fail	Pass

Appendix D 10 Test Case of View Current Topics of Each Sentiment

Use Case	View Current Topics
Description	The system shows 10 topics
Steps Involved	1. The system read CSV file 2. The system clusters the text 3. The system print out topics
Input Data	A CSV files with columns of created_at, full_text, screen_names
Expected Result	View topics fail if 1) the CSV file is not exists 2) the CSV file is empty 3) no column name in CSV file 4) one of the column is not exist 5) no internet connection
Actual Result	View topics fail if 1) the CSV file is not exists 2) the CSV file is empty 3) no column name in CSV file 4) one of the column is not exist 5) no internet connection
Pass/Fail	Pass

Appendix D 11 Test Case of View Current Topics

Use Case	View Popular Tweets
Description	The system shows popular tweets based on search word
Steps Involved	1. The system authenticate Twitter API 2. The system fetch search tweets results from Twitter API 3. The system print out tweets
Input Data	Search word: covid, Count: 100, Extend: '&result_type=popular'
Expected Result	View tweets fail if 1) no internet connection 2) search word is undefined 3) Twitter API connection failed
Actual Result	View tweets fail if 1) no internet connection 2) search word is undefined 3) Twitter API connection failed
Pass/Fail	Pass

Appendix D 12 Test Case of View Popular Tweets

Use Case	Change Time Series Setting
Description	The system fetches the tweets based on the date range chosen by the user
Steps Involved	1. The user chose the date range 2. The user click submit 3. The system fetched tweets based on the date range 4. The system will plot time series based on sectors or sentiments or topics
Input Data	From: 12/6/2021, To: 18/6/2021
Expected Result	Change time series fail if 1) no internet connection 2) no database connection 3) one of the fields is empty
Actual Result	Change time series fail if 1) no internet connection 2) no database connection 3) one of the fields is empty
Pass/Fail	Pass

Appendix D 13 Test Case of Change Time Series Setting

Use Case	Topic Prediction
Description	The system performs topic prediction based on the topic chosen by the user
Steps Involved	1. The user chose the topic 2. The user click submit 3. The system perform topic prediction 4. The system will plot time series of topic and forecasted value
Input Data	Topic: Topic 1
Expected Result	Topic prediction fail if 1) no internet connection 2) CSV file is not exist 3) CSV file is empty 4) topic fields is empty 5) no column written in CSV file
Actual Result	Topic prediction fail if 1) no internet connection 2) CSV file is not exist 3) CSV file is empty 4) topic fields is empty 5) no column written in CSV file
Pass/Fail	Pass

Appendix D 14 Test Case of Topic Prediction

Use Case	Save Topics
Description	The user desire to save topics
Steps Involved	1. The user enter all topics interpretation 2. The user set the name to save topics as 3. The user click submit 4. The system create topic folder based on user's topic path and name 5. The system save all files needed
Input Data	Save Topics As: Today's Topic, Interpretation: [Topic 1, Vaccination, Amazon, Topic 4, Topic 5, Topic 6, Topic 7, Topic 8, Topic 9, Covid-19]
Expected Result	Save topics fail if 1) no internet connection 2) one of the field is empty
Actual Result	Save topics fail if 1) no internet connection 2) one of the field is empty
Pass/Fail	Pass

Appendix D 15 Test Case of Save Topics

Use Case	Read Saved Topics
Description	The user desire to read saved topics
Steps Involved	1. The user clicks on saved topic 2. The system display word cloud, interpretation and time series of topics saved
Input Data	Name: Today's Topic
Expected Result	Read saved topics fail if no internet connection
Actual Result	Read saved topics fail if no internet connection
Pass/Fail	Pass

Appendix D 16 Test Case of Read Saved Topics

Use Case	View News Headlines
Description	The system displays news headlines
Steps Involved	1. The system authenticate News API 2. The system fetch news headlines 3. The system display news headlines
Input Data	-
Expected Result	View news headlines fail if 1) no internet connection, 2) connection with News API failed
Actual Result	View news headlines fail if 1) no internet connection, 2) connection with News API failed
Pass/Fail	Pass

Appendix D 17 Test Case of View News Headlines

Use Case	Search News
Description	The user desire to search new
Steps Involved	1. The user typed search word and submit 2. The system authenticate News API 2. The system fetch search results 3. The system display search results
Input Data	Search Word: covid
Expected Result	Search news fail if 1) no internet connection, 2) connection with News API failed
Actual Result	Search news fail if 1) no internet connection, 2) connection with News API failed
Pass/Fail	Pass

Appendix D 18 Test Case of Search news