TEXT-BASED EMOTION TRACKING AND MONITORING SYSTEM USING BI-DIRECTIONAL LONG SHORT-TERM MEMORY

NUR SYUHADA BINTI AZHAR

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

BORANG PENGESAHAN STATUS LAPORAN

JUDUL: **TEXT-BASED EMOTION TRACKING AND MONITORING SYSTEM USING BI-DIRECTIONAL LONG SHORT-TERM MEMORY**

SESI PENGAJIAN: [2023 / 2024]

Saya: ____NUR SYUHADA BINTI AZHAR___________________

mengaku membenarkan tesis Projek Sarjana Muda ini disimpan di Perpustakaan Universiti Teknikal Malaysia Melaka dengan syarat-syarat kegunaan seperti berikut:

1. Tesis dan projek adalah hakmilik Universiti Teknikal Malaysia Melaka.
2. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat salinan unituk tujuan pengajian sahaja.
3. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. * Sila tandakan (✓)

__________	SULIT	(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)
__________	TERHAD	(Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi / badan di mana penyelidikan dijalankan)
✓	TIDAK TERHAD	

(TANDATANGAN PELAJAR)	(TANDATANGAN PENYELIA)
Alamat tetap: Lot 2043, Kampung Teluk Baru, Jalan Guchil Bayam, 15200, Kota Bharu, Kelantan.	DR. NOOR FAZILLA ABD YUSOF Nama Penyelia
Tarikh: _____15/9/2023_________	Tarikh: __15/9/2023___________

CATATAN: * Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak

TEXT-BASED EMOTION TRACKING AND MONITORING SYSTEM USING BI-DIRECTIONAL LONG SHORT-TERM MEMORY

NUR SYUHADA BINTI AZHAR

This report is submitted in partial fulfillment of the requirements for the Bachelor of Computer Science (Artificial Intelligence) with Honours.

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2023

DECLARATION

I hereby declare that this project report entitled

[TEXT-BASED EMOTION TRACKING AND MONITORING SYSTEM USING BI-DIRECTIONAL LONG SHORT-TERM MEMORY

is written by me and is my own effort and that no part has been plagiarized without citations.

STUDENT : ________________________ Date : 15/9/2023

(NUR SYUHADA BINTI AZHAR)

I hereby declare that I have read this project report and found this project report is sufficient in term of the scope and quality for the award of Bachelor of [Computer Science (Software Development)] with Honours.

SUPERVISOR : ________________________ Date : 15/9/2023

([DR. NOOR FAZILLA ABD YUSOF])

DEDICATION

Firstly, I would like to dedicate this project to my beloved parents for their support that made me able to complete this final year project.

Next, I would like to dedicate this project to my final year project supervisor, Dr. Noor Fazilla Abd Yusof, for guiding me towards the right direction and providing the suggestion and advice for me to accomplish this project.

Lastly, I would like to thank my friends and course mates who are always giving me support, suggestions, motivation and care when doing this project.

ACKNOWLEDGEMENTS

First of all, I would like to express my deepest thanks to my supervisor, Dr. Noor Fazilla Abd Yusof, for her guidance, suggestions, and advice on improvement by giving me feedback, which enabled me to complete this final year project.

Next, I would also like to thank my beloved parents who have been giving me all their support and motivation throughout my project. Motivation is important to me to finish the project.

Last but not least, I would like to thank my friends and course mate for their constant encouragement, support, and assistance in developing the project. I might not have been able to complete this project successfully without the people that I mentioned above.

ABSTRACT

In this challenging world, if we neglect our emotions and focus too much on what we have not achieved in our life yet, we will be depressed. Or maybe if we have been neglecting our emotions and just thinking that there is no need to dwell on it, it will leave a big impact our mental health soon. So this research aims to help people to take care of their emotions and also monitor it to be able to live a better life. The emotion tracker and monitoring system using text-based is a quite challenging task since it needs to learn the patterns of words and also analyses the meaning of words to be able to give the correct prediction output. The prediction techniques have become so common in our life that we can easily see it anywhere such as stock price prediction and weather forecasting. In the process of doing this research, there are some problems that need to be highlighted and improved. As a result, this project uses the Deep Learning method specifically on Bi-Directional Long Short-Term Memory to learn the patterns and tone of words from text. The parameters used in building the model needs to be further fine-tuned since it does not reach the desired accuracy yet. For the result of emotion prediction, it will give five different classes of emotions as result.

ABSTRAK

Di dunia yang mencabar ini, jika kita mengabaikan emosi dan terlalu fokus pada apa yang belum kita capai dalam hidup kita, kita akan berasa tertekan. Atau mungkin jika kita telah mengabaikan emosi kita dan hanya berfikir bahawa ianya hanya perkara remeh, ia akan meninggalkan impak besar kepada kesihatan mental kita lama-kelamaan. Jadi kajian ini bertujuan untuk membantu orang ramai untuk menjaga emosi mereka dan juga memantaunya agar dapat menjalani kehidupan dengan lebih baik. Sistem ramalan emosi dan pemantauan berasaskan teks adalah tugas yang agak mencabar kerana ia perlu mempelajari corak perkataan dan juga menganalisa makna perkataan untuk memberikan ramalan yang tepat. Teknik ramalan telah menjadi sangat biasa dalam kehidupan kita sehingga kita boleh melihatnya dengan mudah di mana-mana sahaja seperti ramalan harga saham dan ramalan cuaca. Dalam proses menjalankan kajian ini, terdapat beberapa masalah yang perlu diperhatikan dan diperbaiki. Hasilnya, projek ini menggunakan kaedah Pembelajaran Dalam khususnya pada Memori Jangka Pendek Panjang Dwi Arah untuk mempelajari corak dan maksud perkataan daripada teks. Parameter yang digunakan dalam membina model perlu diperhalusi lagi kerana ia masih belum mencapai ketepatan yang diingini. Untuk hasil ramalan emosi, ia akan memberikan lima kelas emosi yang berbeza sebagai hasilnya.

TABLE OF CONTENTS

PAGE

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

DL	**-**	**Deep Learning**
NLP	**-**	**Natural Language Processing**
ML	**-**	**Machine Learning**
LSTM	**-**	**Long Short-Term Memory**
Bi-LSTM	**-**	**Bi-Directional Long Short-Term Memory**
RNN	**-**	**Recurrent Neural Network**
CNN	**-**	**Convolutional Neural Network**
WHO	**-**	**World Health Organization**
URL	**-**	**Uniform Source Locator**
PCA	**-**	**Principal Component Analysis**
SVM	**-**	**Support Vector Machine**
BERT	**-**	**Bi-Directional Encoder Representations from Transformer**
LDA	**-**	**Latent Dirichlet Allocation**
HTML	**-**	**Hyper-Text Markup Language**
NFR	**-**	**Non-Functional Requirements**

CHAPTER 1: INTRODUCTION

1.1 Introduction

In this vast population of the world, it is estimated that roughly 3.8% from the world population have been diagnosed with depression (WHO, 2023). From the collected statistics, around 5% of adults below 60-year-old and 5.7% of adults over 60-year-old from all around the world experienced depression. In other resources that is also stated by the World Health Organization (WHO) which is, approximately 280 million from 7.9 billion people in the world have depression and it appears that depression often are diagnosed more on women than men. According to Jessica (2018), depression is considered as a common mental disorder that involves having a dejected mood or losing interest or enjoyment in doing activities that one once enjoyed for an extended periods of time. Depression can also affect negatively on the way someone thinks, feels, and acts. If one has been diagnosed with severe depression, it can lead to a number of problems concerning both mental and physical such as having unstable emotions and also decreased performance at work.

There are already numerous efforts to help people who are struggling with depression such as awareness campaigns, raising educational programs about mental health conditions, increasing access to mental health services and much more. It is stated that there is no cure for depression but there are still several options available that can help to lessen the severity of the symptoms and perhaps will help to enhance the quality of life (Herndon, 2022).

There are various application and system online to help people to cope with their depression and also monitor it as to not let it become severe over the time. The most popular application or system that will help are mood or emotion tracker app. Emotion tracker app is one of the most popular app since it acts and works much like a diary where people can write whatever they want and rant on their feelings or emotions and also help individuals to record their mood regularly. The purpose of mood tracker is to identify patterns in mood changes over time and in response to various situations and circumstances. An emotion tracker is also said to be helpful for individuals with mental health conditions especially depression and anxiety by identifying and monitoring their moods (Cherry, 2023). Currently there are many popular application or websites on emotion tracker such as MoodPanda, eMoods, MoodTracker, Daylio and many more. It is found that most of mood tracker app are build using machine learning methods such as Natural Language Processing (NLP) or Deep Learning (DL).

The background of this project is to propose a web-app emotion tracker using text emotion detection. The proposed project aims to use text emotion detection to detect the emotion on user's input text and then give the predicted emotion based on user's text. User can record their daily moods by entering their text on the web app. Besides, user can also see the visualization of their emotions that have been recorded in the web app and monitor their moods. This project will be using Deep Learning analysis to analyse the text emotion that will be entered by user when they record their daily moods. The deep learning will identify the patterns and trends in mood changes whenever user record their daily mood. The technique of deep learning chosen is bi-directional Long Short-Term Memory (LSTM).

1.2 Problem Statement

Even though mood tracker app has been proven to be useful in helping people to take care of their mood and symptoms of depression, there are still some several problems that could occur. Among the problems are, some of the tracker systems are not accurate and not reliable. It is because the systems rely heavily on data provided to give user insights on their recorded daily mood. Sometimes the techniques that is used in mood tracker systems have constraints in analyzing the language and sentiment accurately which can result in inaccuracies in mood analysis.

Another problem emotion monitoring system is user engagement with the system. The main concern for tracker system is for user to constantly engage with the system in order for the system to collect data to give insights for user. But users are merely a human, who sometimes may forget to open the system and input data by recording their mood daily. Sometimes user may also lose interest when they are feeling that the system is not helping them thus they stopped using the system.

Despite that, another problem that led to the propose of this project is, people who are battling with depression have trouble to reach out to others to confront about their conditions. They may feel shame and guilty about their condition and are afraid that others will find them annoying when they reach out to others. They also may feel that they should go through the symptoms on their own as to not trouble others and make them feel weak. Usually this kind of people prefer to let out their feelings online or by using a journal or diary online as they feel more comfortable doing so.

1.3 Objectives

This project embarks on the following objectives:

I. To explore emotion recognition techniques based on text.
II. To build a recognition model to predict text using deep learning.
III. To build a web app for monitoring and visualize emotion.

1.4 Scope

The scope that are involved in this project is divided into two categories which are modules to be developed and the involvement of target user. The scope is described as follows:

i. Modules to be developed

The modules to be developed for this project are as follows:

I. Record Daily Mood Module

This module is for user to record their daily mood by entering the text into the field. The system will keep the data provided by user and provide insights.

II. Dashboard Module

In this module, the system will show the recorded data of user's mood and insights of user's mood from a certain period of time. User can analyse their mood and track their symptoms.

ii. Target user

The target user in this project is for users who are battling with depression and users who want to keep track and monitor their emotions. In further implementations and improvements, this system can be used by professional healthcare such as doctors or psychiatrist who keeps track of the patients' record of emotions.

1.5 Project Significance

The significance of this project is to highlight the importance of depression awareness throughout the whole world. Depression has been a leading cause of suicide since a long time ago and the world have already done numerous things to prevent or at least reduce the rate of suicide that is caused by depression. By developing mood tracker system, people will have more options to help control their symptoms by tracking their daily mood and preventing them from keeping their pent up feelings all by themselves. It is better to do early invention in treating depression. When user record their daily mood with the emotion tracker system, they will be able to let out their feelings and record their mood and this will prevent them from having a worse symptom.

1.6 Expected Output

The expected output for this system is user will be able to record their daily mood into the system and keep track of their record for a certain amount of time. From the recorded data of user's daily mood, the system should be able to process the data and give insights of user's condition on how are they feeling.

1.7 Summary

In conclusion, the proposed system will let user to be able to record their daily mood into the system and the user can monitor their moods progress to help them stay positive. The system will use Deep Learning (DL) techniques which is the Bi-directional Long Short-Term Memory (Bi-LSTM) to identify the tone and pattern of the text input from user. From the recorded data of user's daily mood, the system will provide insight for user to monitor their mood and also give recommendations if user's condition worsen.

CHAPTER 2: LITERATURE REVIEW AND PROJECT METHODOLOGY

2.1 Introduction

This chapter will discuss the literature review for some of the existing system that are related to this project. The methods that are used in the researched project are Deep Learning with Recurrent Neural Network (RNN), Long-Short Term Memory (LSTM), and Convolutional Neural Network (CNN). The project methodology, project requirements, and project schedule and milestones will be stated with the details.

2.2 Facts and Findings

This part will discuss the domains regarding with this project which will help in understanding the core value of this project and the completion of this project.

2.2.1 Domain

The domain in this project is healthcare since emotion tracker and monitoring is considered a mental health disorder that affects people of all ages and gender. It is important to note that depression has become one of the major risk factor for suicide which can be fatal. Thus, this project is developed in effort to try to reduce the risk of increasing suicide cases that are caused by depression.

This project follows the development of an emotion tracker system using Deep Learning (DL) where first step is to collect mood data from the various available sources and determine how to analyze and use the data collected to provide insights for users later. After done collecting the data, it is time to gather the data and do pre-processing such as data cleaning, normalization process and tokenization and this process is followed with training the DL model. The purpose of developing and training the model is to analyze the emotion data that have been collected earlier to detect the patterns, trends, and also the tone. After that, the DL model will be used to implement the system's features such as providing personalized insights and recommendations based on mood data. This process is further consulted with a software developer. Lastly, the system will be deployed to user on various platform. After the system deployment, the system will be monitored for its performance including user engagement and user satisfaction.

2.2.2 Existing System

There are many emotion tracker and monitoring system that has been implemented using and Deep Learning (ML) and also Natural Language Processing (NLP). In this section, three papers will be studied which are related to depression monitoring and tracking system.

2.2.2.1 Deep Learning for Depression Detection from Textual Data

In this paperwork, this project uses deep learning method which includes the Long-Short Term Memory (LSTM) model, that consists of two hidden layers and large bias with Recurrent Neural Network (RNN) with two dense layers to predict depression from text (Amanat et. Al, 2022). It is stated that they trained the RNN on textual data to identify depression from text, semantics, and written content. The procedure of depression detection for this project is showed in Figure 1. The main contribution of this project are listed as follows:

- Provide a detailed discussion of depression, depressive systems, and its types. This project concentrates on processing textual data and detecting depressive traits.
- Propose a deep learning model using LSTM, with 0 LSTM units with two hidden states and bias factors, and RNN with two hidden layers for the early detection of depression by training the model with depressive and non-depressive sample data.
- Evaluate the proposed prediction model using the Tweets-scraped depression dataset and evaluate it using the evaluation matrices such as precision, accuracy, F1-Measure.
- The evaluation results show that the proposed framework improves accuracy by detecting depression from textual data.

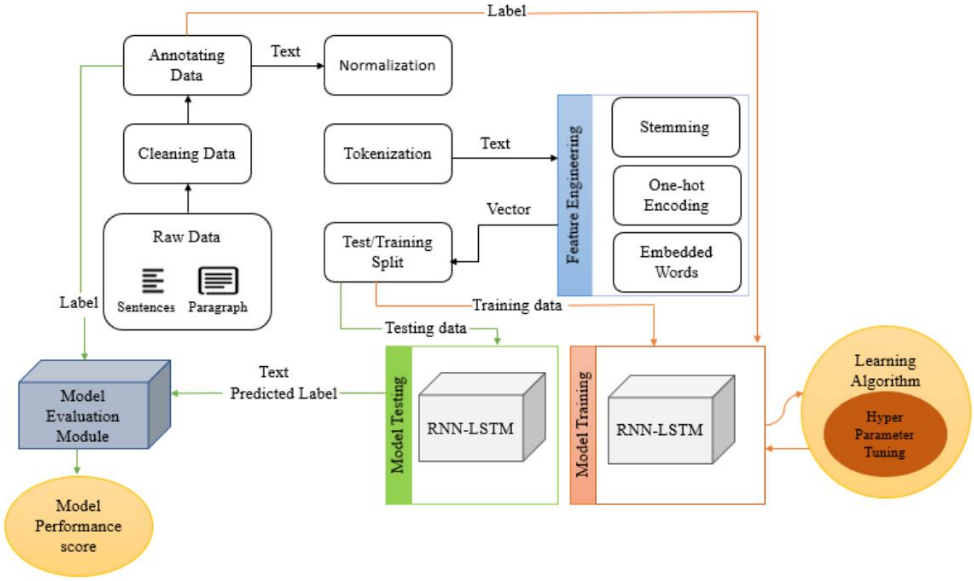

Figure 2.1: Procedure of Depression Detection (Amanat et. Al, 2022)

The team collected the datasets of Twitter from Kaggle website and did data pre-processing after cleaning the data. Then the data were divided into training and testing dataset. The next processes include transformation and normalization of the data. They removed the URLs, mentions, and stop-words and continue to tokenize the words. The methods that were used in this project are one-hot encoding and Principal Component Analysis (PCA) along with Support Vector Machine (SVM), Naïve Bayes, and TF-IDF. The evaluation results showed that their methods offer high accuracy, precision, recall, and F1-Measures compared to the naïve Bayes, SVM, CNN and Decision Trees approaches. The results for the test on each approaches are as shown in Figure 2.2 below.

Approaches	Mean Accuracy (%)
SVM	97.21
Naive bayes	97.31
One-hot + SVM	83
TF-IDf + SVM	85
One-hot + Decision Trees	82
TF-IDF + CNN	91
one-hot + DBN	89

Figure 2.2: Prediction accuracy with different approaches (Amanat et. al, 2022)

2.2.2.2 Stress detection using natural language processing and machine learning over social interactions

This paper from Nijhawan, Attigeri and Ananthakrishna (2022) used deep learning method but more precisely on model titled BERT (Bidirectional Encoder Representations from Transformers) for sentiment classification. BERT is a pre-trained language model which is developed by Google based on the Transformer architecture. The authors also mentioned on the use of LDA (Latent Dirichlet Allocation) in their project. LDA is a probabilistic generative model to identify hidden topics within a collection of documents. The main goal of LDA is to identify topics for the collection of documents and find out which topic it belongs to. LDA has been used to identify topics in large collections of documents such as news articles, scientific papers, and even social media posts and it also has many applications in NLP such as topic modelling.

Topic modelling is a statistical technique used in natural language processing and machine learning to uncover latent semantic structures or themes in a large collection of text data. It is a form of unsupervised learning, which means that it can discover hidden patterns and structures in text data without being explicitly trained on labelled examples. It is also can be defined as a method that locates a group of words i.e., the topic from a group of documents that represents the information in the group. This paper gives contributions as follows:

- Binary classification of the sentiments behind the data that are collected from web scrapping on Twitter.
- Perform topic modelling with the help of Latent Dirichlet Allocation, which is a popular probabilistic model used for topic modelling.
- Emotion classification using deep learning-based BERT model to detect stress.
- Accurately analyse and segregate the user's opinions on different topics.

In this project, it is mentioned that the team reviewed papers on mental stress detection using machine learning that used social networking sites, blogs, discussion forums, Questioner technique, clinical dataset, and real time data. The team used SVM and Naïve Bayes algorithms for predicting stress from social media sites such as Facebook, Twitter, and Live Journal. The datasets that gained are from 6773 posts from the mentioned sites which categorised to 2073 depressed posts and 4700 non-depressed posts. The team achieved the accuracy of 57% from SVM method and 63% from Naïve Bayes method.

The research of this paper aims at building models for sentiment and emotion detection which can be used for stress management, the models are also tested on primary data. The focus of the paper is identifying the sentiment or emotions of a user concerning diverse topics or domains using Latent Dirichlet Allocation (LDA). The methodology of this project is shown in Figure 2.3 below:

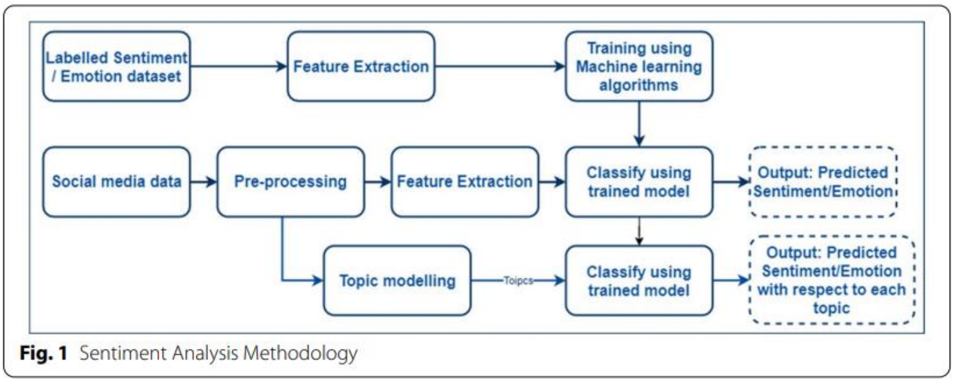

Figure 2.3: Project Methodology (Nijhawan, Attigeri and Ananthakrishna, 2022)

2.2.2.3 Depression Detection Based On Twitter Using NLP and Sentiment Analysis

In this article, Yam, Zuriani Hayati and Nabilah Filzah (2022) developed a system to detect depression by using the Recurrent Neural Network (RNN) model and Convolutional Neural Network (CNN) model in order to achieve the most optimal parameters for model building and comparing the accuracy of the prediction. They have mentioned in the paper that there are several ways to diagnose depression. The traditional way is by conducting physical interview with people which is the most used method by psychologist. But somehow this method is no longer preferable as nowadays people do not favour going to psychologist because they feel ashamed of letting someone else know that they develop depression. So nowadays, researchers tend to use a more advance method by using Natural Language Processing (NLP) and Sentiment Analysis to detect potential depression through text usage from social media.

By monitoring and tracking user’s social media activity, the method for detecting depression on someone has become a more preferable option for researchers. When user is on social media sites, they tend to be more of themselves because they can be anonymous online and they do not feel afraid to let their feelings online. From this, they will analyze the language that have been used by social media users to help in analyzing the status of user whether they are feeling depressed or not.

The authors have studied several methods that are applicable and can be used in analyzing and studying text or words and they have compared the methods for further implementation. They have chosen Twitter from many social media as the main platform for data scrapping. This is because Twitter is the most used social media for posting short updates or tweets and it is available on both mobile app and web app. Most of analysis on text classification uses Twitter for data source. In Figure 2.4 below, we can see the research methodology that is implemented by the authors.

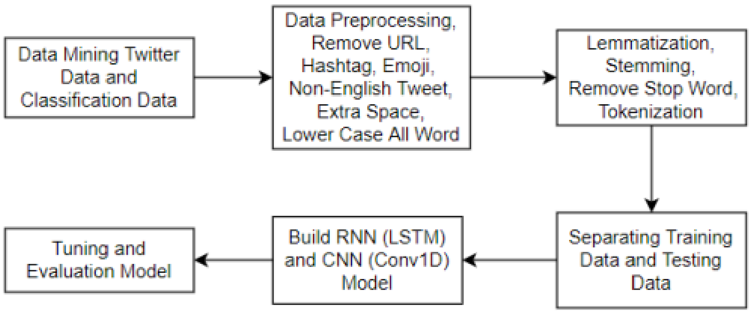

Figure 2.4: Research Methodology (Yam, Zuriani Hayati and Nabilah Filzah, 2022)

After the dataset has been collected, the authors created usernames for users that have depression and user who does not have depression. Any user that have or used words such as “depressed” or “diagnose” are considered as depression while user that do not have any of these words are considered not depressed. The authors have checked both type of users’ account to make sure that it is not a spam account. Figure 2.5 below shows the number of user with depression tweet and user with no depressed tweet.

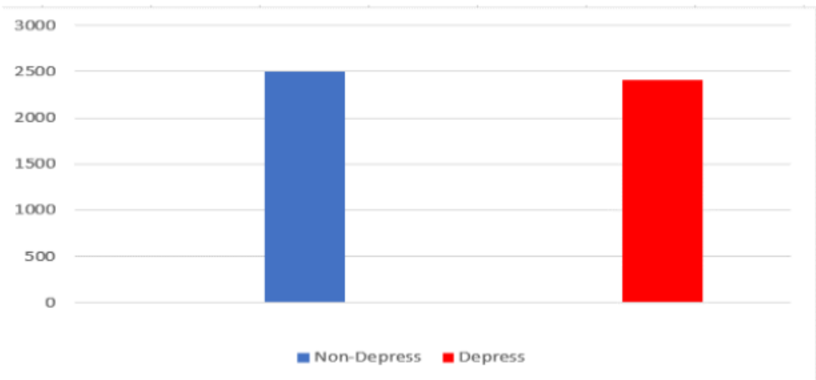

Figure 2.5: Number of user with depression tweet and number of user with no depressed tweet (Yam, Zuriani Hayati and Nabilah Filzah, 2022)

For pre-processing data, the authors clean the dataset by doing the normalization process to the dataset. Then, this process will be followed by lemmatization, stemming, tokenization, and removing stop words. After that, they separated the data into training and testing dataset to 80% and 20% respectively. When implementing the model, the authors have chosen four methods to implement which are Recurrent Neural Network (LSTM) 4 Layer, Recurrent Neural Network (LSTM) 7 Layer, CNN (Conv1D) 6 Layer, and CNN (Conv1D) 9 Layer. All of the models implemented by the authors will be hyper parameter tuning where it will find the optimal values of hypermeters for the model. The result for each model with the accuracy, loss, precision, recall, F1 measures, and training time are shown in Figure 2.6 below.

MODEL	ACCURACY	LOSS	PRECISION	RECALL	F1	TRAINING TIME (H:M:S)
RNN (LSTM) 4Layer	78.36%	45.59%	75.68%	83.66%	79.46%	1:19:04
RNN (LSTM) 7Layer	80.99%	45.01%	76.82%	83.81%	80.16%	2:19:47
CNN (Conv1D) 6Layer	76.48%	49.59%	74.69%	80.20%	77.34%	0:42:37
CNN (Conv1D) 9Layer	77.40%	47.46%	7548%	81.26%	78.26%	1:00:43

Figure 2.6: Result of each model (Yam, Zuriani Hayati and Nabilah Filzah, 2022)

From the result in Figure 2.6 shown above, we can see that Recurrent Neural Network (LSTM) 7 Layer is the most accurate model with highest accuracy, lowest loss, highest precision, highest precision, recall, and F1 measures compared to the other models that the authors have used.

2.2.3 Techniques

Deep Learning is a branch of machine learning that involves using artificial neural networks, which are algorithms that are designed to mimic the structure and function of human brain (Brownlee, 2020). There are several techniques of deep learning that are popular and are widely used. Among the techniques are Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), and Long Short Term Memory (LSTM) Network.

2.2.3.1 Convolutional Neural Network (CNN)

Convolutional Neural Networks (CNNs or ConvNets) is a type of neural network that are particularly well-suited for processing data with a grid-like structure, such as images. In an image, each pixel is represented by a binary value indicating its brightness and color, and they are arranged in a grid-like format. Hence, CNNs are commonly used in computer vision tasks, such as image recognition and object detection since it consists of layers of convolutional filters that extract features from the input data. These features are then passed through fully connected layers to produce a final output (Mishra, 2020).

2.2.3.2 Recurrent Neural Network (RNN)

Recurrent Neural Networks (RNNs) is a type of deep learning model that is used for modelling sequential data by maintaining a memory of past inputs and using it to inform future predictions. This memory is implemented using a hidden state that is updated with each input. Standard neural networks have independent inputs and outputs, which may not be suitable for certain tasks, such as predicting the next word in a sentence, where the previous words are important. RNN were developed to address this issue, using a hidden layer to keep track of past inputs. The Hidden state is a crucial component of RNN, as it retains important information about the sequence being processed (Kalita, 2022).

2.2.3.3 Long Short Term Memory (LSTM) Network

Long Short-Term Memory (LSTM) is a type of deep learning model that is designed for sequential data processing. It has a unique ability to retain information over long periods of time**.** Long Short-Term Memory (LSTM) Network is a special type of RNN that is designed to handle long-term dependencies in sequential data. They do this by using a gating mechanism that controls the flow of information through the network. LSTMs are commonly used in tasks such as speech recognition and text generation (Saxena, 2023).

2.2.3.4 Bi-directional Long Short Term Memory (LSTM) Network

Bi-directional Long Short-Term Memory (LSTM) is an upgrade over the traditional LSTM. In bidirectional LSTMs, each training sequence is analysed in both forward and backward directions using separate networks. These networks are connected to the same output layer. By employing bidirectional LSTMs, the model gains comprehensive knowledge about every point within a given sequence, including both preceding and succeeding elements (“What is LSTM?”, June 2023).

2.2.3.5 Which technique is chosen

The technique that is chosen for this proposed project is Bi-Directional Long Short-Term Memory. This reason why this method is suitable to use on emotion tracker and monitoring system is because the system is based on analyzing sequences of data over time. For example, analyzing mood changes that user has recorded in the system over a course of a day or weeks. Bi-directional LSTM can take in a sequence of mood data recorded from user as inputs and then the output of each time user recorded their mood on a new day, it can use the previous data to predict the mood for the next time user will record their mood. The hidden state of neural network can be used to remember the past mood data that can be helpful in producing a more accurate prediction in the future.

2.3 Project Methodology

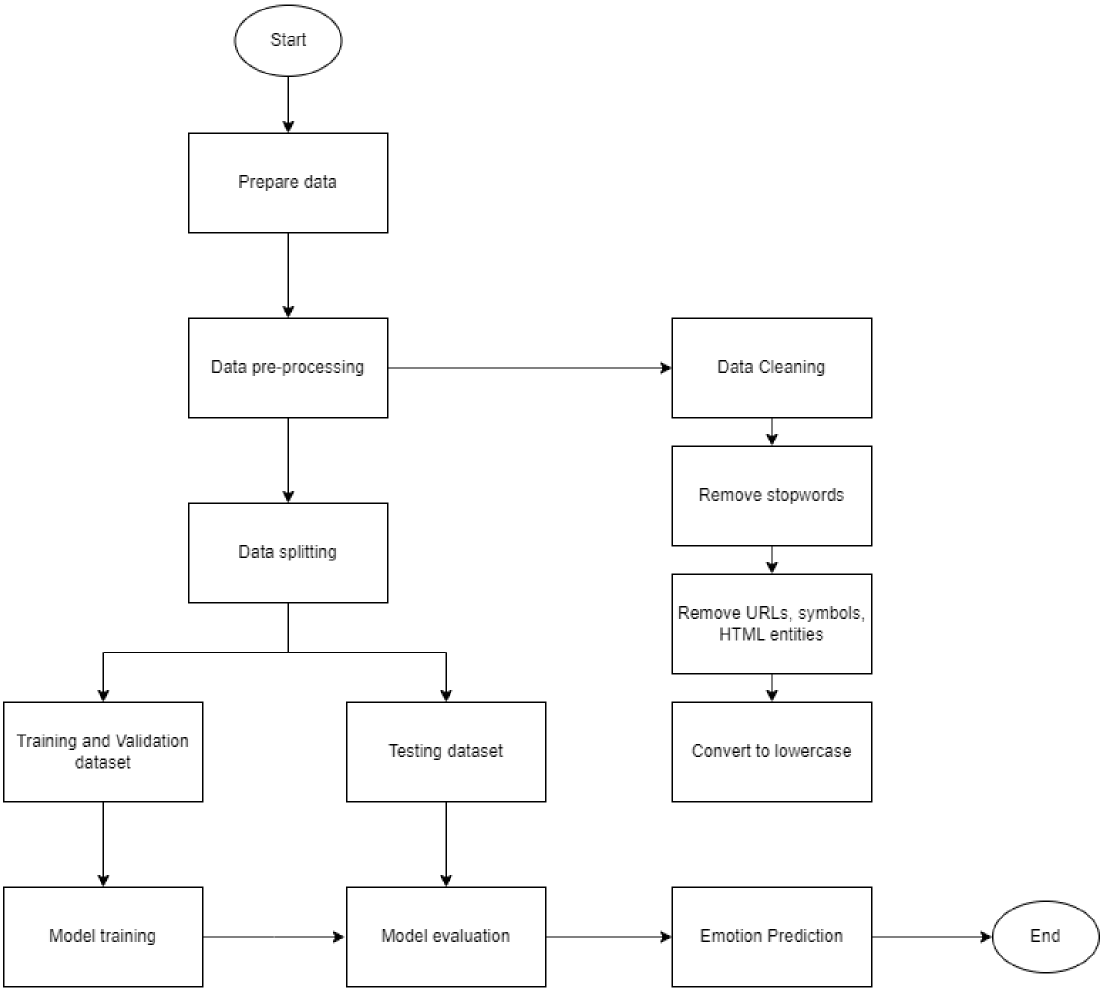

Figure 2.7: Project methodology of this project

Stage 1: Data Collection

The first stage is to collect the dataset. For this project, the dataset is obtained from GitHub which contains fifty-four thousand (45,000) data that was scrapped from Twitter website. The dataset will undergo data cleaning and pre-processing such as removing stopwords, removing URLs and HTMLs, apply lemmatization and tokenization and also converting the text to lowercase. After this process, the dataset is split into training, testing and validation set.

Stage 2: Model Development

In this stage, the model is built and tested on different architectures. There are three different architectures used in building this model which single LSTM layer, deep single LSTM layer, and also Bi-Directional LSTM layer. Along with these architectures, the parameters are adjusted and tuned to see which parameters gives the best result. After that, set up the optimizer, epoch, batch size and learning rate and the model can start training.

Stage 3: Model Evaluation

This stage will define either the model is good or not. After training with specified parameters, the result will show a line graph indicating the performance of the model. Training accuracy and validation accuracy will be monitored to prevent from overfitting. Besides that, the performance classification involving precision, recall, f-1 score and accuracy will be displayed.

Stage 4: Deploy Model

Lastly, the model will do the prediction on new data that is not from the dataset. After that, the model will be integrated into the web app using Flask platform. The web app will be tested on the performance of predicting new emotion from user’s text input.

2.4 Project Requirements

There are two project requirements that will be used to complete this proposed project which are software requirement and hardware requirements.

2.4.1 Software Requirements

A. Development Tools

1. Text Editor: Jupyter Notebook and Google Colab

   -The use of these text editors is to write the coding using python language for developing the system.

2. Microsoft Office 2013 and Microsoft 365

   -The use of these software is to write report for this project using Microsoft Word and also designing presentation slides using Microsoft Powerpoint.

B. Operating System / Server

1. Microsoft Windows 10

   - To run all the software that has been mentioned before and also web browser since it is one of the main platform for programs to let the hardware and software communicate with each either in order to develop the system.

2. Google Chrome or Microsoft Edge

   -To run, access and use the system in webpage.

2.4.2 Hardware Requirement

1. Laptop Brand: HP

   - Device Name: DESKTOP-05BHN2S

   - Processor: Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz 2.71 GHz

   - Installed RAM: 12.0 GB

2.5 Project Schedule and Milestones

Figure 2.8: PSM 1 Gantt Chart

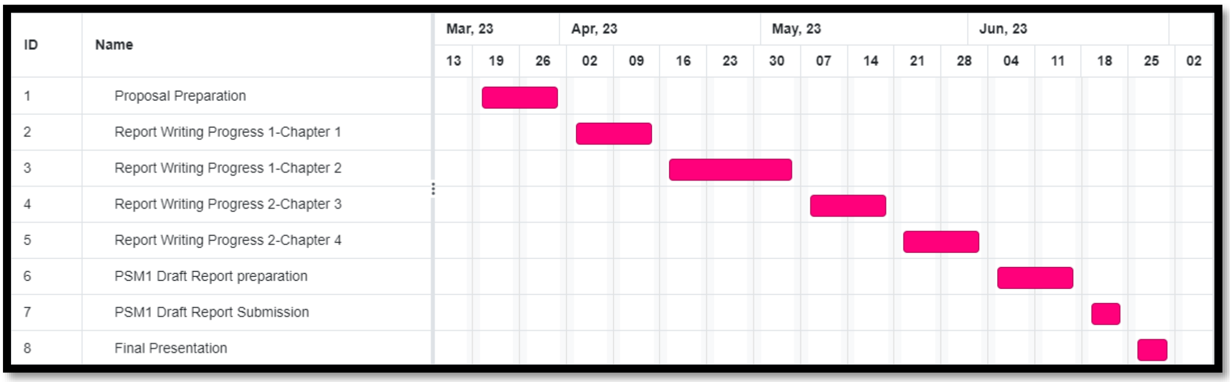

Figure 2.9: PSM 2 Gantt Chart

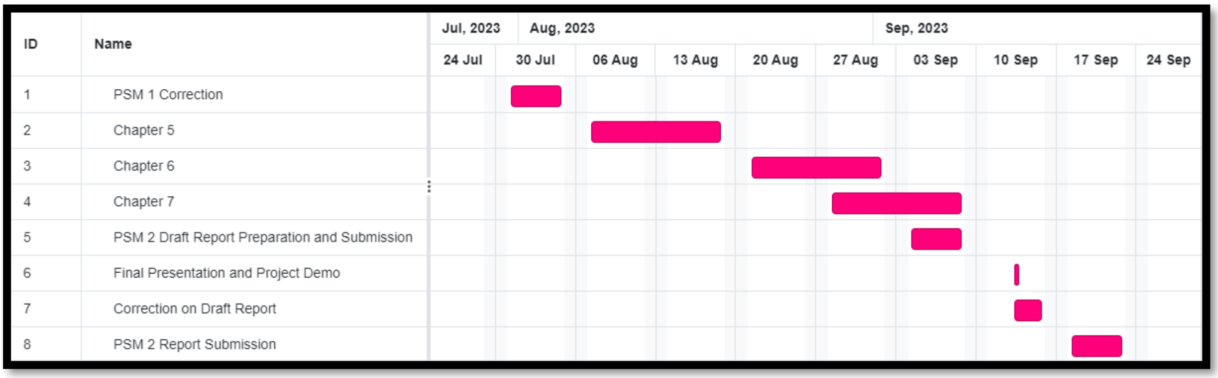

Figure 2.10: PSM 1 Milestones Week 1 - Week 7

PSM 1 MILESTONES | SEM 2 SESSION 2022/2023

WEEK	ACTIVITY	NOTE / ACTION
W1 (20/03 →24/03) (<3/10)	Select a suitable project topic and potential Supervisor	• Action - Student
W1 (20/03 →24/03) Meeting 1	List of Student vs Supervisor	• Action - Committee
	Proposal PSM: Discussion with Supervisor	• Proposal Form - Ulearn • Action - Student
	Proposal assessment & verification	• Action - Supervisor
W2 (27/03 →31/03)	Proposal Correction/Improvement Proposal submission to Committee via email	• • Deliverable - Draft Proposal Form - email PIC • Action - Student → Committee
	Proposal Approval List of Supervisors with Project's Title	• • List of Supervisor/Title - ULearn • Action - Committee → Student
W3 (03/04 → 07/04) Meeting 2	Proposal Presentation & Proposal Submission Proposal [PRJ-1]	• Proposal Presentation • Log Progress - ePSM • Deliverable – Completed Proposal Form – ePSM • Action – Student → Supervisor • Evaluate – ePSM • Action – Supervisor
	Chapter 1	• Action - Student
W4 (10/04 → 14/04)	Chapter 1 Report Writing Progress 1 [PRJ-3]	• Log Progress – ePSM • Deliverable – Chapter 1 – ePSM • Action – Student → Supervisor • Evaluate – ePSM • Action – Supervisor
W5 (17/04 →21/04)	Chapter 2	• Action - Student
W6 (24/04 →28/04)	**MID-SEMESTER BREAK**	
W7 (01/05 →05/05) Meeting 3	Chapter 2 Report Writing Progress [PRJ-3]	• Log Progress – ePSM • Deliverable – Chapter 2 – ePSM • Action – Student → Supervisor • Evaluate – ePSM • Action – Supervisor
	Project Progress 1 [PRJ-2]	• Log Progress – ePSM • Progress Presentation 1 (KP1) • Action – Student → Supervisor • Evaluate – ePSM • Action – Supervisor
	Student Status	• Warning Letter 1 to Student • Action → Supervisor, Committee

Figure 2.11: PSM 1 Milestones Week 8 - Week 15

W8 (08/05 → 12/05)	Chapter 3	• Action - Student
W9 (15/05 → 19/05)	Chapter 3 Report Writing Progress 1 [PRJ-3]	• Log Progress – ePSM • Deliverable – Chapter 3 – ePSM • Action – Student → Supervisor • Evaluate – ePSM • Action – Supervisor
W10 (22/05 → 26/05) Meeting 4	Chapter 4	• Action - Student
	Project Progress 2 [PRJ-4]	• Log Progress – ePSM • Progress Presentation 2 (KP2) • Action – Student → Supervisor • Evaluate – ePSM • Action – Supervisor
	Student Status	• Warning Letter 2 to Student • Action – Supervisor, Committee
W11 (29/05 → 02/06)	Chapter 4 Report Writing Progress 2 [PRJ-5]	• Log Progress – ePSM • Deliverable – Chapter 4 – ePSM • Action – Student → Supervisor • Evaluate – ePSM • Action – Supervisor
	PSM1 Draft Report preparation	• Action – Student
	Determination of Student Status (Continue/Withdraw)	• Submit Student status to PSM/PD Committee • Action – Supervisor → Committee
W12 & W13 (05/06 → 16/06) Meeting 5	PSM1 Draft Report preparation	• Action - Student → Supervisor
W14 (19/06 → 23/06)	PSM1 Draft Report submission to SV & Evaluator Report Evaluation [PRJ6] [PRJ-10]	• Log Progress – ePSM • Deliverable – Complete PSM1 Draft Report – ePSM • Action – Student → Supervisor, Evaluator • Evaluate – ePSM • Action – Supervisor
	Schedule the presentation	• Presentation Schedule - ULearn • Action - Committee
W15 (26/06 → 30/06) FINAL PRESENTATION	Demonstration Supervisor [PRJ-6] Evaluator [PRJ-7]	• Log Record – ePSM • Action – Student • Evaluate – ePSM • Action – Supervisor, Evaluator
	Presentation Skill [PRJ-8]	• Log Record – ePSM • Action – Student • Evaluate – ePSM • Action – Evaluator

Figure 2.12: PSM 1 Milestones Week 16 - Week 18

W16 (03/07 → 07/07) REVISION WEEK	**REVISION WEEK** Correction on the draft report based on the Supervisor and Evaluator's comments during the final presentation session. Do an EoS Survey online form.	• Deliverable – PSM1 Report – ULearn • Action – Student → Committee • Deliverable – EoS Survey – Online Form • Action – Student
	Complete overall marks to Committee	• Deliverable: Overall Evaluation PSM 1 – e-PSM • Action – Supervisor, Evaluator → Committee
W17 & W18 (10/07 → 21/07)	**FINAL EXAMINATION WEEKS**	

Figure 2.13: PSM 2 Milestone Week 1 - Week 6

PSM 2 MILESTONES | SEM 3 SESSION 2022/2023

WEEK	ACTIVITY	NOTE / ACTION
W1 (31/7 → 4/8) Meeting 1	Chapter 4	• Discussion with supervisor - PSM 1 correction, PSM 2 planning • Action - Student
W2 (7/8 → 11/8) Meeting 2	Chapter 5 Project Progress 1 [PRJ-1]	• Log Record - ePSM • Progress Presentation 1 (KP1) • Action - Student → Supervisor
		• Evaluate - ePSM • Action - Supervisor
W3 (14/8 → 18/8)	Chapter 5	• Action - Student
	Report Writing Progress [PRJ-3]	• Log Record - ePSM • Deliverable - Chapter 5 -SV email • Action - Student → Supervisor
		• Evaluate - ePSM • Action - Supervisor
	Student Status	• Warning Letter 1 • Action - Supervisor, Committee → Student
W4 (21/8 → 25/8) Meeting 3	Chapter 6	• Action - Student
	Project Progress 2 [PRJ-2]	• Log Record - ePSM • Progress Presentation 1 (KP2) • Action - Student → Supervisor
		• Evaluate - ePSM • Action - Supervisor
W5 (28/8 → 1/9) Meeting 4	Chapter 6	• Action - Student
	Report Writing Progress [PRJ-3]	• Log Record - ePSM • Deliverable - Chapter 6 -SV email • Action - Student → Supervisor
		• Evaluate - ePSM • Action - Supervisor
	Chapter 7	• Action - Student
	Student Status	• Warning Letter 2 • Action - Supervisor, Committee → Student
	Schedule the presentation	• Presentation Schedule - ULearn • Action - Committee
W6 (4/9 → 8/9) Meeting 5	Chapter 7 Report Writing Progress [PRJ-3]	• Log Record - ePSM • Deliverable - Chapter 7 -SV email • Action - Student → Supervisor
		• Evaluate - ePSM • Action - Supervisor
	Determination of student status (Continue/Withdraw)	• Submit student status to Committee • Action - Supervisor → Committee
	PSM2 Draft Report preparation PSM2 Draft Report submission to SV & Evaluator	• Deliverable - PSM2 Draft Report -ePSM • Action - Student → Supervisor, Evaluator

Figure 2.14: PSM 2 Milestone Week 7 - Week 9

W7 (11/9 → 15/9) FINAL PRESENTATION	Report Evaluation [PRJ6] [PRJ-10] DEMONSTRATION Supervisor [PRJ-4] [PRJ-5] DEMONSTRATION Evaluator [PRJ-9]	• Log Record - ePSM • Action - Student
		• Evaluate - ePSM • Action - Supervisor, Evaluator
	English Proficiency [PRJ-7]	• Log Record - ePSM • Action - Student
		• Evaluate - ePSM • Action - Supervisor
	Presentation Skill [PRJ-8]	• Log Record - ePSM • Action - Student
		• Evaluate - ePSM • Action - Evaluator
W8 (18/9 → 22/9) FINAL EXAMINATION WEEKS	Correction on the draft report based on the Supervisor and Evaluator's comments during the final presentation session. Do an online EoS Survey form.	• Deliverable -EoS Survey -Online Form • Action - Student
	Complete of overall marks to Committee	• Deliverable: Overall Evaluation PSM2 -ePSM • Action - Supervisor, Evaluator → Committee
	Submission of the final complete report, which is the updated & corrected PSM2 report	• Deliverable - Complete Final PSM Report - ULearn • Action - Student → Committee
W9 (25/9 → 29/9) INTER-SEMESTER BREAK	Submission of the final complete report, which is the updated & corrected PSM2 report and Plagiarism Report etc. onto the OneDrive	• Deliverable - Complete Final PSM Report, Plagiarism Report, etc. - OneDrive (SV) • Action - Student → Supervisor

2.6 Conclusion

As a conclusion, the literature review is very important in developing this project since it acts as a valuable resource for identifying relevant features and requirements for the proposed system and defining this project's overall requirements. Furthermore, literature review can serve as a basis for new research that seeks to enhance the current system, as well as drawing insights from previous research published by other authors. Additionally, a well-defined methodology is crucial for providing guidance to individuals as they navigate the project's domain scope and field. The next chapter of this project will focus on analyzing the proposed system for this project.

CHAPTER 3: ANALYSIS

3.1 Introduction

This chapter will discuss on the analysis of this system. This chapter is important in developing the system since it can serve as the requirements guide in completing the project. In this chapter, analysis will be divided into two main parts which are the problem analysis and requirement analysis. Requirement analysis will be separated into several parts which are data requirement, functional requirement, non-functional requirement, and other requirements related to this project. For this project, the method used deep learning method which is Recurrent Neural Network (RNN) focusing on Bi-Directional Long Short Term Memory (LSTM) Network. This is because Bi-Directional LSTM has the ability to retain information over long periods of time. The details of the network in this project will be further discussed.

3.2 Problem Analysis

This project aims to help someone who is suffering with depression to help them track and monitor their emotion. By recording their daily mood, the user will be able to monitor and see the trend of their emotion. This proposed project is developed by using the Bi-Directional Long Short Term Memory (LSTM) Network under Recurrent Neural Network (RNN) due to the nature of the network that is sequential. In this case, training a machine learning model for text emotion detection using Bi-Directional LSTM is the best choice because it has given the ability for Recurrent Neural Networks (RNNs) to retain information from previous inputs for an extended period (Jayawardhana, 2020).

In Figure 3.1 below shows the flowchart of this system. The first step is preparing the data that is obtained from GitHub website and do data pre-processing that includes removing stopwords, removing URLs, symbols, HTML entities, and also converting the text to lowercase. After that, the dataset will be split into training, validation and testing set. The training and validation set will be trained to get the evaluation of the model which will later be used by test set. Lastly, after achieving a significant accuracy, emotion detection is done on a new text that is not in the dataset.

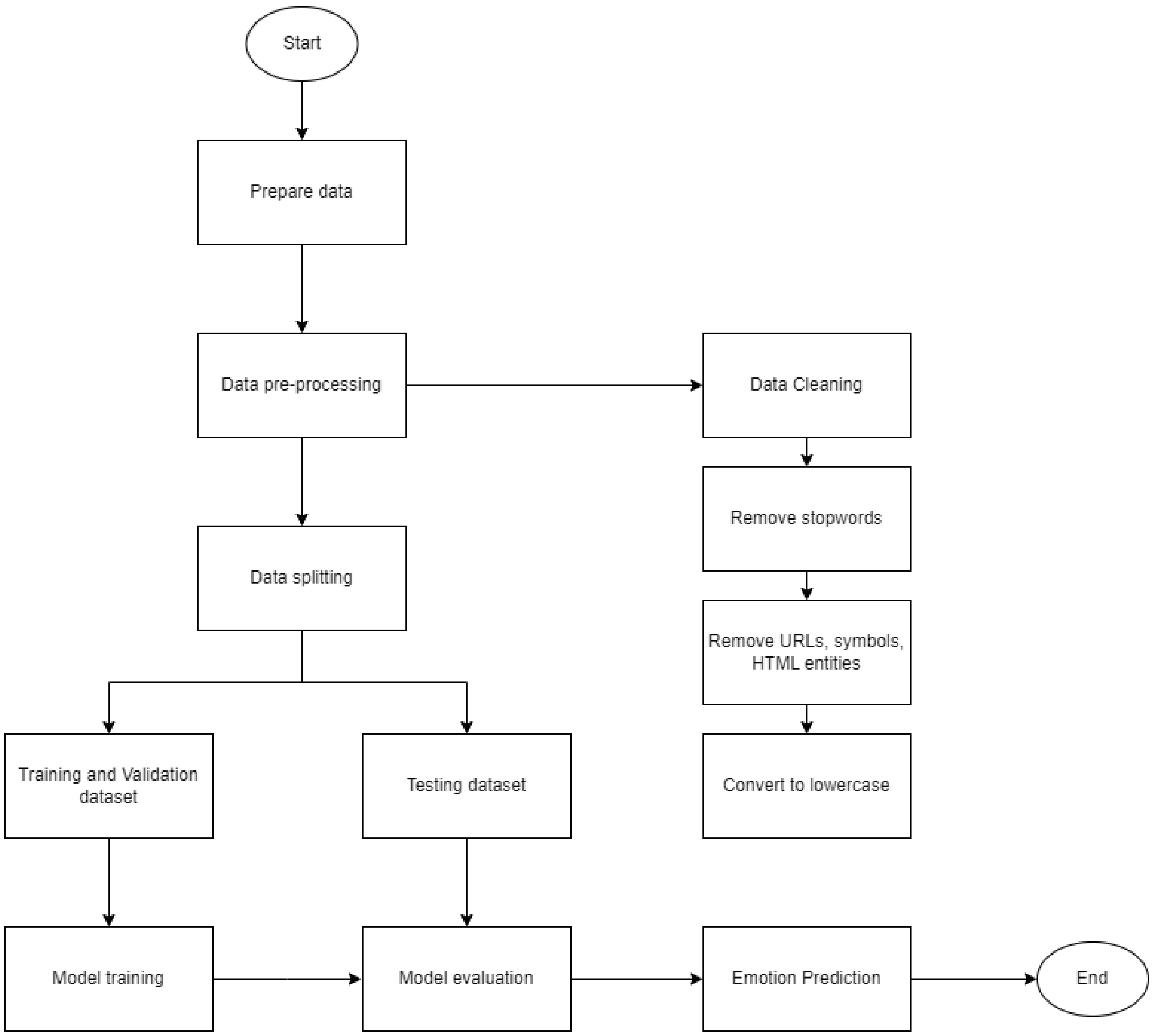

Figure 3.1: System Architecture

3.3 Requirement Analysis

3.3.1 Data Requirement

The dataset that is used for this system is obtained from GitHub website. This dataset contains 55,774 data of tweets from the social media Twitter. Originally, the dataset has thirteen different classes of emotion which are ‘empty’, ‘sadness’, ‘enthusiasm’, ‘neutral’, ‘worry’, ‘’surprise’, ‘’love’, ‘fun’, ‘hate’, ‘happy’, ‘boredom’, ‘relief’, and lastly ‘angry’. But after consideration and for the ease of prediction part later, it is decided that the thirteen classes of emotion are simplified to only five classes which are ‘neutral’, ‘happy’, ‘sad’, ‘anger’, and ‘love’. Figure 3.2 below shows the dataset that is used for this project.

1	text	label by number	label
2	i m looking forward to going home tomorrow but i really wish it was for a different reason	2	sad
3	just got to kansas city and excited for a fun weekend with my family my sis parker and josh	3	love
4	hey adt guess what my princelple s number plate is adt 000 well its not 000 i just dunno the numbers	1	happy
5	b gt not the best song for her	0	neutral
6	the wind tried to hate on us today lol hello floral s rose gold w gold because goldi s best that s why	4	anger
7	i need to change my ways instead of just being weak i love she s a great role model	3	love
8	i am so excited rob thomas is back	1	happy
9	feeling the pain from the accidente not feeling too good	2	sad
10	it rains and it sucks so much because it s the second day in a row	2	sad
11	have you heard our podcast review of the boyfriend project by and i explore all the bea	3	love
12	yes nice i missed a lot of fun damn exams you are looking good hair	2	sad
13	rain one more reason to stay snuggled beneath the duvet	0	neutral
14	is sad his new sb6 cd s got nicked	2	sad
15	without whipped topping there is no shortcake shortcake fail	2	sad
16	meet yourself with absolute love quote panache desai monday motivation iq g monday thoughts quotes monda	3	love
17	i have been called by my friend the bed it is time to acknowledge the inevitable goodnight all hello sleep	0	neutral
18	restoring my ipod touch seemed like euthanasia to me i was willingly killing it	2	sad
19	do the right thing ann otti music wall of fame a spike lee joint blm love over hate	4	anger
20	just got home from dinner with my mommy and my new grandma i d rather be in hollywood right now	1	happy
21	everything heals your body heals your heart heals the mind heals wounds heal your happiness is always going to come ba	1	happy
22	you might have missed my latest experiment i know blogs have too many words these days	0	neutral

Figure 3.2: Emotion_data.csv dataset

Originally, the dataset only had ‘text’ and ‘label’ that represents the emotion in numbers. But for the ease of recognizing each label, the ‘label’ column is changed to ‘label by number’ and a new column is added to represent the emotion based from the label number.

3.3.2 Functional Requirement

Functional requirement is crucial for a project since it specifies what a system should do and also describing the functionalities of the system. As stated in previous chapter, this project will be using deep learning model which is Recurrent Neural Network (RNN) architecture, more specifically using Bi-Directional Long Short Term Memory (LSTM) architecture. This Bi-Directional LSTM will be trained on Python using the dataset that has been prepared earlier. The performance and evaluation of the model will be monitored.

When user enter a text input in the system, the text will be pre-processed and will be using the trained Python model to get the prediction of user's text input on how are they feeling. The predicted emotion will depend on the accuracy of the trained model. The trained model will later be fine-tuned to achieve higher accuracy for better prediction result.

3.3.3 Non-functional Requirement

The non-functional requirement for this project is accuracy and performance. In most Machine Learning (ML) project, accuracy and performance is considered the mostly chosen NFR (non-functional requirement). In this project, currently the accuracy is a bit low but in the future, the model will be fine-tuned with more detail to achieve a better result.

3.3.4 Others Requirement

This section will discuss about the analysis of software and hardware requirements which are involved in this project to achieve the deep learning model as well as the structure of the system. This information is displayed in Table 3.1 and Table 3.2 below.

Table 3.1: Software Requirements

Software Requirements	
Software Name	**Description**
Google Colab	Google Colab is a cloud-based platform that provides a free environment for writing and executing Python code, collaborating with others, and accessing powerful computational resources.
Jupyter Notebook	Jupyter Notebook is a web application that allows users to create and share computational documents, providing a user-friendly and focused experience centered around documents.
Flask	Flask is a Python web framework that has a compact core and a philosophy focused on simplicity and extensibility.
Sublime Text	Sublime Text is a versatile text and source code editor available for Windows, macOS, and Linux. It offers native support for a wide range of programming languages and markup languages, making it suitable for various coding and text editing tasks.

Table 3.2: Hardware Requirements

Hardware Requirements	
Hardware Name	**Description**
HP portable laptop	- Device Name: DESKTOP-05BHN2S - Processor: Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz 2.71 GHz - Installed RAM: 12.0 GB

3.4 Conclusion

In conclusion, analysis involves breaking down a complex relational process into its individual components to gain a deeper understanding of the project. Throughout this chapter, the problem and requirements were thoroughly analyzed and documented. Moving forward to the next chapter, we will delve into the design phase of the proposed system.

CHAPTER 4: DESIGN

4.1 Introduction

In this chapter, the system design for emotion tracker and monitoring system will be explained in detail. This chapter will include the design flow of the system, the development, and also deployment of the system website. The explanation for every process will be further discussed in this section in the high-level and detail design sections.

4.2 High-Level Design

The aim of developing this project is to help someone who is suffering with depression to help them track and keep monitoring their mood using Bi-Directional Long Short Term Memory (LSTM) model that is trained in Python. The model will be deployed in the system by using Flask web app platform where user later can enter their input for mood tracking and monitoring. The model is trained to predict user's emotion from the text input and it is evaluated by the model's accuracy.

4.2.1 System Architecture for expert system / DSS / simulation

The first step in developing the model is to do data pre-processing. The dataset that is used for this project is obtained from GitHub. The dataset is uploaded into Jupyter Notebook to do pre-processing such as converting texts to lowercase, removing stopwords, removing symbols, URLs, hashtags, numbers and also performing lemmatization on the texts to ensure that the model will be able to process the text. After that, the texts will be split into training, testing and validation set. The process follows with tokenization and label encoding to be feed into the model to be trained.

As mentioned before, the deep learning method that is chosen for developing this system is Bi-Directional LSTM (Long Short Term Memory). In Figure 4.1 below shows the model building for this system. Since it can only remember sequences of 10s and 100s but not 1000s or more, Bidirectional network is applied here. Bidirectional network is effective when dealing with long sequences of data where the model needs to learn the relationship between the past word and future word (Mungalpara, 2021).

```
# Build the model
from keras.layers import Bidirectional, Dropout

embedding_dim=100

model = Sequential()
model.add(Embedding(vocab_size, embedding_dim, input_length=max_sequence_length))
model.add(Bidirectional(LSTM(32, activation='relu')))
model.add(Dropout(0.5))
model.add(Dense(512, activation='relu'))
model.add(Dense(num_classes, activation='softmax'))
```

Figure 4.1: Building the model with Bi-Directional LSTM

Embedding dim is set to 100 so that each word in the input sequences will be represented by a dense vector of size 100. The embedding layer is then added to the model which will later convert the input sequences that was represented in integers, into dense vectors. The embedding layer also has vocab size which is 26838, and also input length which represents the maximum sequence length on input sequences. The Bi-directional LSTM layer uses 32 units and will process the input sequences in both forward and backward directions. The activation function used for Bidirectional layer is ReLu (Rectified Linear Unit) since it introduces non-linearity and allows the model to learn complex relationships between the texts and labels.

Then, a dropout layer of 0.5 is added. Dropout is a regularization technique to prevent the model from overfitting. A fully connected dense layer with 512 units and ReLu activation function is also added to the model to help the model in learning non-linear relationships between the features learned by the preceding LSTM layer. Lastly, the output layer is added which has num_classes unit which contains the number of classes in this dataset. The activation function used for this layer is softmax since there is five (5) classes of emotion.

Figure 4.2 below shows the code for compiling the model. The optimizer used is 'Adam' and the learning rate is set to 0.0001 since starting with lower learning rate seems to give a better result. The loss function used here is sparse categorical cross entropy since in this dataset there are five (5) classes of emotions. After the model has been compiled, the model will start training the data in model.fit with 10 epochs and batch size of 128.

```
# Compile the model with optimizer Adam with the desired learning rate
optimizer = Adam(learning_rate=0.0001)
model.compile(loss='sparse_categorical_crossentropy', optimizer=optimizer, metrics=['accuracy'])

history=model.fit(train_data, train_labels, validation_data=(val_data, val_labels), epochs=10, batch_size=128, shuffle=True)
```

Figure 4.2: Compiling the model and training the model

When training the model, there are several other architectures used for building the model. The first architecture is using single LSTM layer, the second method is using single deep LSTM layer, and the third architecture is using bi-directional LSTM layer with additional one dense layer. All of these mentioned architectures used dropout regularization to help prevent overfitting. After the training for all three architectures has been performed, it is obvious that the architecture using Bi-Directional LSTM has the best performance and thus it is chosen as the main architecture to use for training the model.

```
    {
        'layers': [
            Embedding(vocab_size, 100, input_length=max_sequence_length),
            LSTM(64, kernel_regularizer=l2(0.01)),
            Dropout(0.5),
            Dense(num_classes, activation='softmax')
        ],
        'name': 'LSTM (with Dropout and L2 Regularization)'
    },
```

Figure 4.3: Architecture for single LSTM layer

In Figure 4.3, it shows the architecture for single LSTM layer and the output of the model training for this architecture is shown is Figure 4.4 below. The training accuracy is 86% percent while the validation accuracy is 54%. Next, for deep LSTM layer, this architecture gives training accuracy of 79% and validation accuracy of 54% as can be seen Figure 4.5 and Figure 4.6. And lastly for the Bi-Directional LSTM layer in Figure 4.7 and 4.8 below, the training accuracy is 71% and the validation accuracy is 58%.

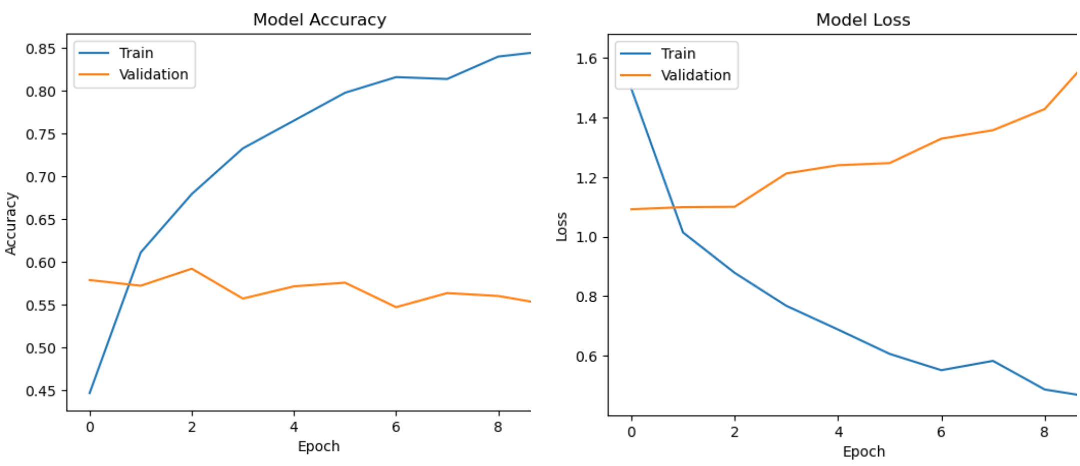

Figure 4.4: Result of single LSTM layer

```
{
    'layers': [
        Embedding(vocab_size, 100, input_length=max_sequence_length),
        LSTM(128, return_sequences=True, kernel_regularizer=l2(0.01)),
        LSTM(64, kernel_regularizer=l2(0.01)),
        Dropout(0.5),
        Dense(num_classes, activation='softmax')
    ],
    'name': 'Deep LSTM (with Dropout and L2 Regularization)'
},
```

Figure 4.5: Architecture for single deep LSTM layer

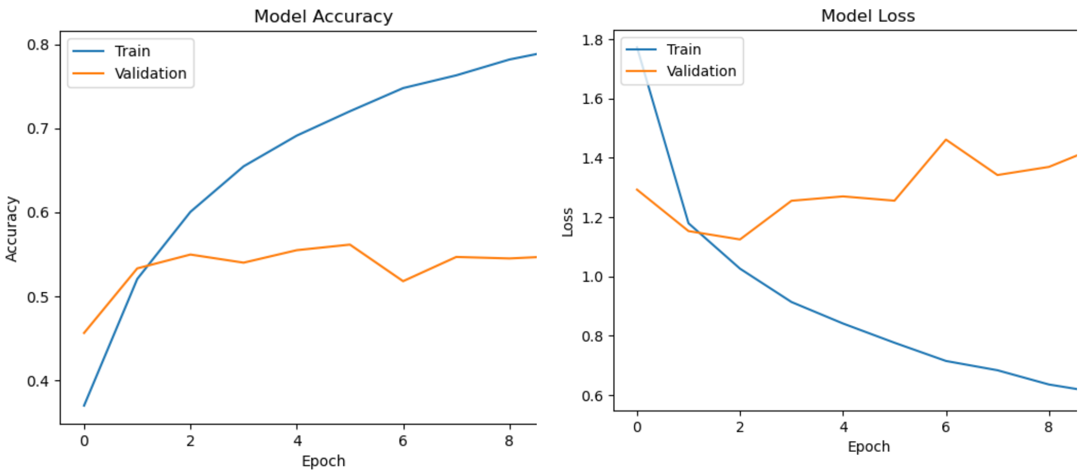

Figure 4.6: Result of deep single LSTM layer

```
embedding_dim=100

model = Sequential()
model.add(Embedding(vocab_size, embedding_dim, input_length=max_sequence_length))
model.add(Bidirectional(LSTM(32, activation='relu')))
model.add(Dropout(0.5))
model.add(Dense(512, activation='relu'))
model.add(Dense(num_classes, activation='softmax'))
```

Figure 4.7: Architecture for bi-directional with additional dense layer

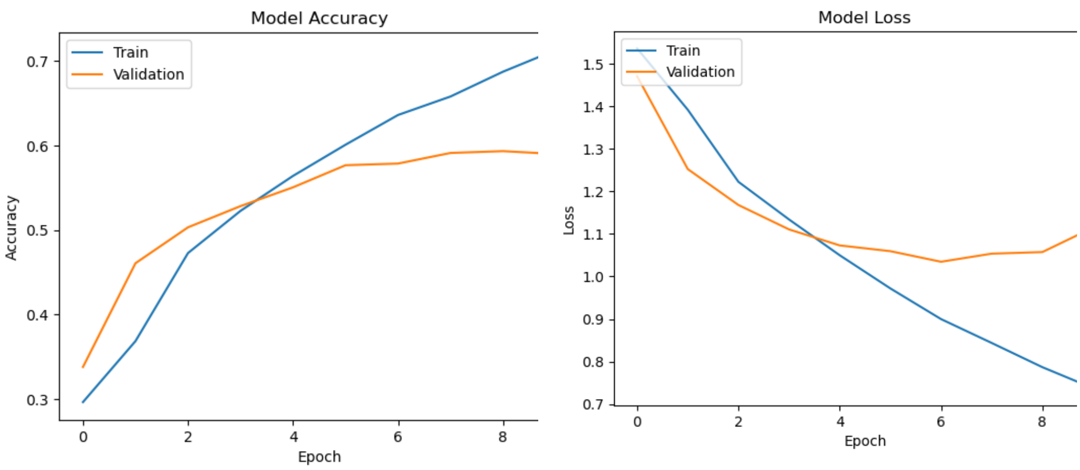

Figure 4.8: Result of Bi-Directional LSTM layer

As soon as the model is done with training the data, the model proceeds to do the testing set and the result on classification report. Classification report includes precision, f-1, and also recall value. The model achieved an accuracy of 59% which is quite low for text emotion detection but during further implementation and improvement, the model will be fine-tuned to get a better accuracy and better prediction. The summary of the model has the output shape for every layer and also the number of parameters in each layer as shown in Figure 4.9 below.

```
Model: "sequential_19"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_19 (Embedding)    (None, 36, 100)           2683800

 bidirectional_18 (Bidirecti  (None, 64)               34048
 onal)

 dropout_17 (Dropout)        (None, 64)                0

 dense_20 (Dense)            (None, 512)               33280

 dense_21 (Dense)            (None, 5)                 2565

=================================================================
Total params: 2,753,693
Trainable params: 2,753,693
Non-trainable params: 0
```

Figure 4.9: Model summary

4.2.2 User Interface Design for expert system / DSS / simulation

The user interface design of this project is in the website that is developed using Flask platform. On the website, user will enter their text input on how are they feeling and also choose an intensity level of the feelings. After user click on the submit button, the system will predict the emotion and display the result. In future implementation, a graph will be displayed for user to monitor their feelings on a period of time.

4.2.2.1 Navigation Design

Figure 4.10 shows the navigation design of the system website. Navigation design is the implementation of flow of the website for user to navigate when accessing the website. Users need to record their daily moods by entering the text into the text field and also choose an intensity of the feelings. After user click the button, the system will give an output on the prediction of user's emotion based from the text.

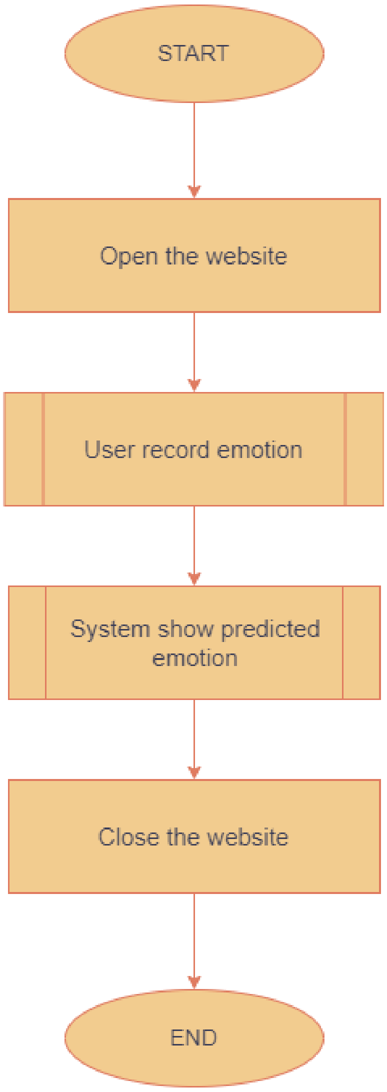

Figure 4.10: Navigation Design of system website

4.2.2.2 Input Design for expert system / DSS / simulation

The input for this website is from user where they will enter text on how are they feeling today in the input field as can be seen in Figure 4.11. After user has entered their text, the system will give an option to choose an intensity on how strong they relate to what they have written. This can be referred in Figure 4.12.

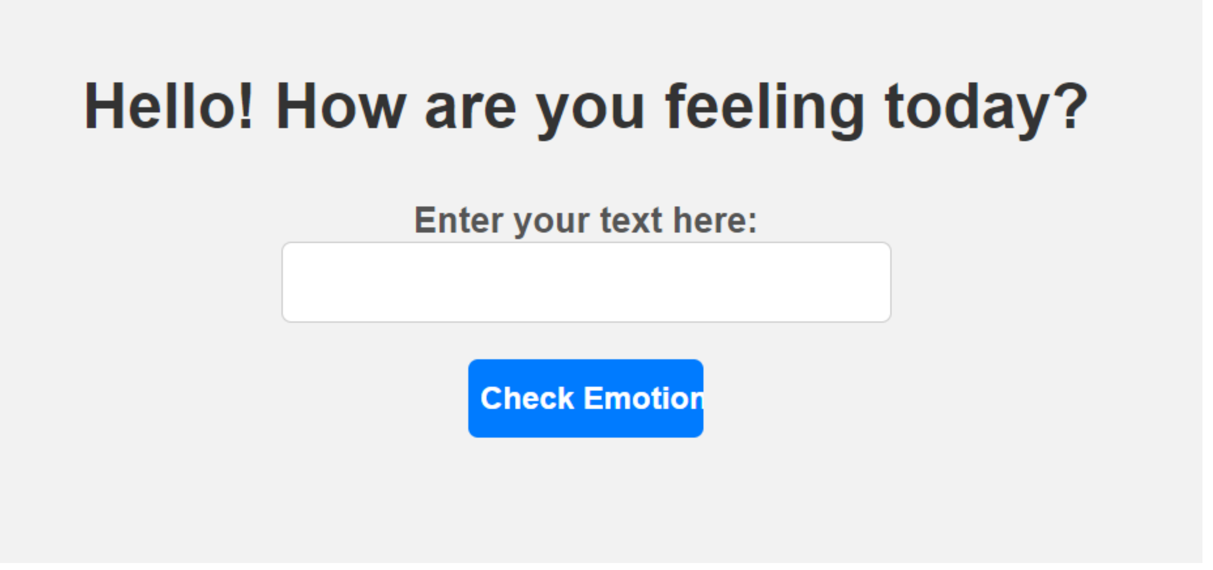

Figure 4.11: User text input

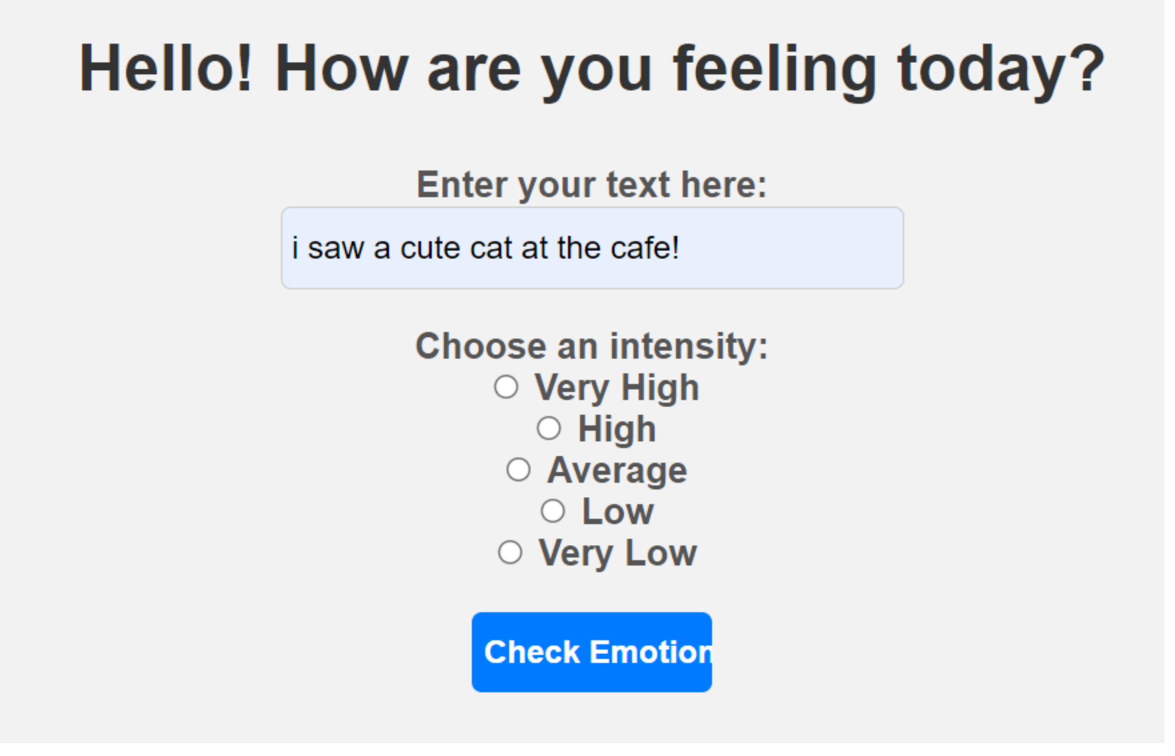

Figure 4.12: User can choose an intensity

The input from user will be processed by the model. The model will process the new input text from user and apply some pre-processing to the new input such as tokenization. Tokenization is the most crucial part because the system needs to tokenize the words first to be able to detect the words that represents the emotions. After that, the model will load the training data that has been used when building the model and apply it onto the new input text and follows by label encoding. Label encoding is used to label the categorical emotions to convert them into integer so that the model because machine learning can only process numerical form. Figure 4.13 shows the back end process of the system website.

```
# Load the model
model = load_model('trained_model_latest.h5')

# Function to load the training texts
def load_training_texts():
    training_texts = []
    training_labels = []
    with open('training_data.csv', 'r') as file:
        reader = csv.reader(file)
        next(reader, None)  # Skip the header row if it exists
        for row in reader:
            text = row[0]  # Adjust the column index if needed
            label = row[1]  # Adjust the column index if needed
            training_texts.append(text)
            training_labels.append(label)
    return training_texts, training_labels

# Load the training texts from a data source
train_texts, train_labels = load_training_texts()

# Initialize the tokenizer
tokenizer = Tokenizer()
tokenizer.fit_on_texts(train_texts)

# Load the tokenizer and max_sequence_length
with open('tokenizer.pkl', 'rb') as tokenizer_file:
    loaded_tokenizer = pickle.load(tokenizer_file)
    max_sequence_length = max(len(sequence) for sequence in loaded_tokenizer.texts_to_sequences(train_texts))
```

Figure 4.13: Back end code to process the new text input

4.2.2.3 Technical Design

LSTM neural network that is used in building the model has several layers such as embedding layer, LSTM layer, bi-directional LSTM layer, dropout layer and dense layer and each layer has different functionalities. Let’s take a look at what each layer means in text classification.

A. Embedding Layer

The first layer in building the model is embedding layer. This layer uses dense vector representation as an approach to represent words and documents. In dense vector, it projects the words into a continuous vector space and it also take counts of the position of the words. The vector space will learn the position of the word in the text and will observe the surrounding words in that sentence. This process is called embedding (Brownlee, 2017). Additionally, there are pre-trained word embedding that are available which are GloVe, Word2Vec, and fastText.

B. LSTM Layer

Next layer is LSTM layer. This layer is responsible for capturing and processing the text in a sequential manner by taking into account of the order and context of the words. By doing so, LSTM layer can capture long-term dependencies and keep the important information (Intelliipaat, 2023). LSTM layer achieves this by utilizing the memory cells and gates. The memory cells store and update information, allowing the layer to remember relevant information from earlier words. While the gates control the flow of information within the layer and also deciding what to keep and what to discard based on the needs of the model. In Figure 4.14 below, the gates are visualized as water filters where the gates will selectively remove and irrelevant information. This process is just like water filters when filtering the water. The water filter will not let any impurities pass the gate and will only let purified and clean water with nutrients to pass through the gates (Loye, 2019).

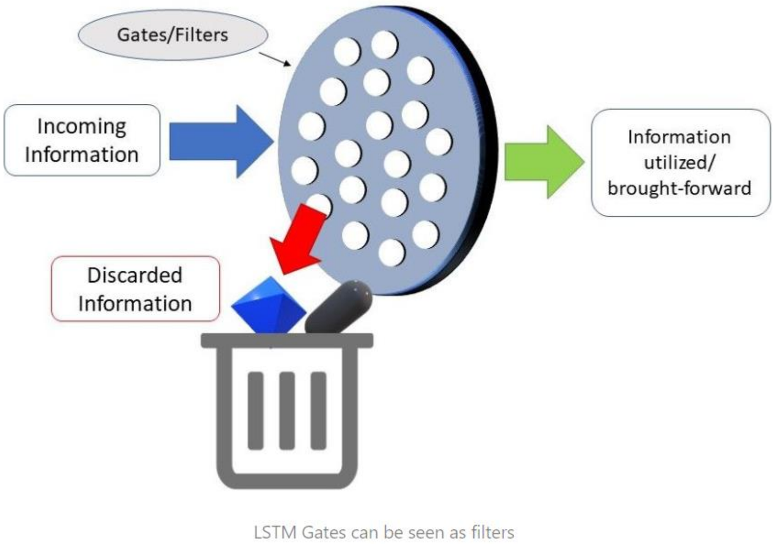

Figure 4.14: Visualization of gates as water filters (Floydhub, 2019)

C. Bi-directional LSTM Layer

Bi-directional LSTM layer is applied to a situation where the sequence of the data is very long and the model needs to adapt and understand the relationship between words that come before and after (Mungalpara, 2021). Bi-directional LSTM will analyse the input in two directions, forward and backward. This means that the model processes the text not only in the conventional forward direction, but also in reverse direction. By doing so it considers both the words that come before and after and capture the meaning from both directions. This can be seen in Figure 4.15 below.

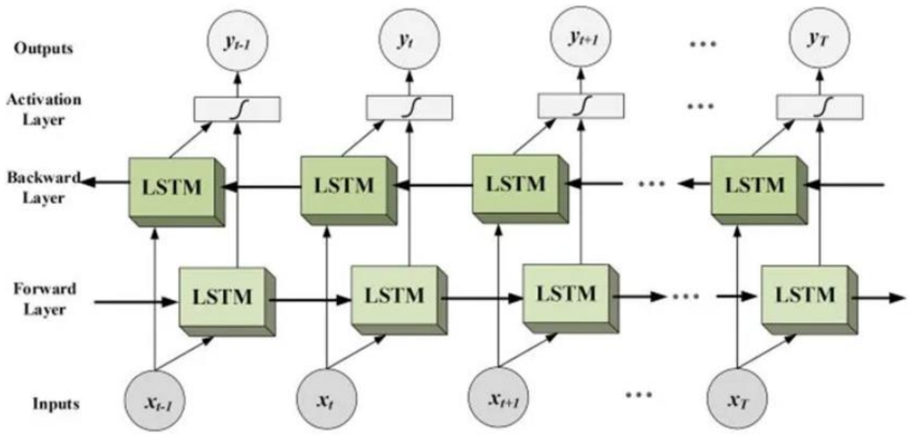

Figure 4.15: Bi-directional LSTM (Mungalpara, 2021)

D. Dropout Layer

After that, there is dropout layer. The concept of dropout involves temporarily removing nodes from both input and hidden layers within a neural network. The removal includes all the connections associated with the dropped nodes which results in the creation of a modified network architecture that is derived from the original network (Yadav, 2022). In Figure 4.16 shows the neural network that has been applied with dropout layer. The main purpose of adding a dropout layer is to reduce overfitting and improve generalization. Overfitting occurs when the model becomes too complex and excessively specialized to the training data. So in order to prevent overfitting, a dropout layer is added.

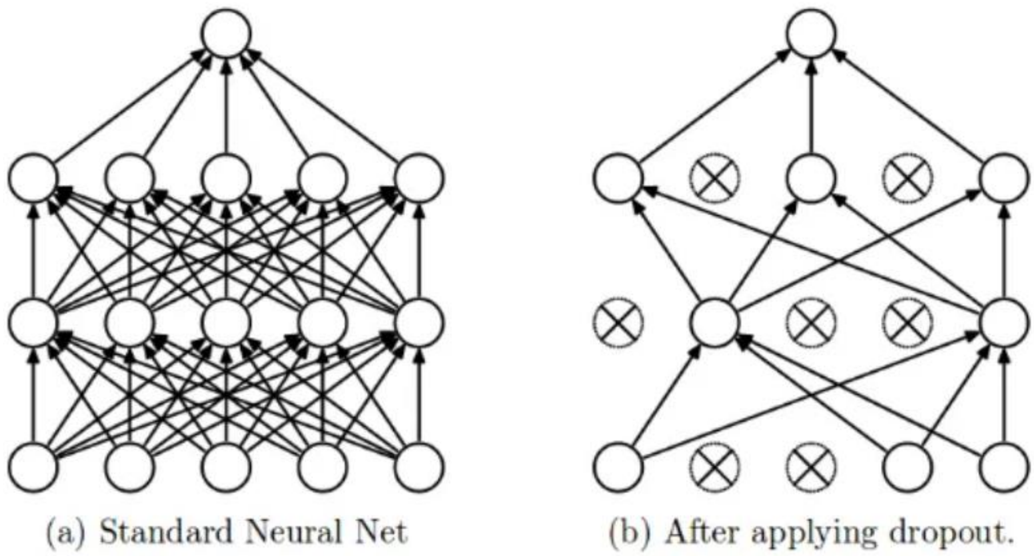

Figure 4.16: Dropout layer in neural network (Yadav, 2022)

E. Dense Layer

Lastly there is dense layer. Dense layer is a fully connected layer type that is deeply connected with each neuron from the previous layer to every neuron in the current layer and it is used in the final stages of neural network (Verma, 2021). The main purpose of dense layer is to adjust the dimensionality of the output from the previous layer which allows the model to learn effectively on the relationships between values of data that it is processing.

4.2.2.4 Output Design

The output of the system is to be able to display the predicted emotion on the text that has been input by user. After user enter their text in the input field, there is an option for user to choose the intensity level of their feelings. After that, when user can click on the "check emotion" button, the system will display the result of predicted emotion and also display a message for user based on the predicted emotion. This is shown in Figure 4.17.

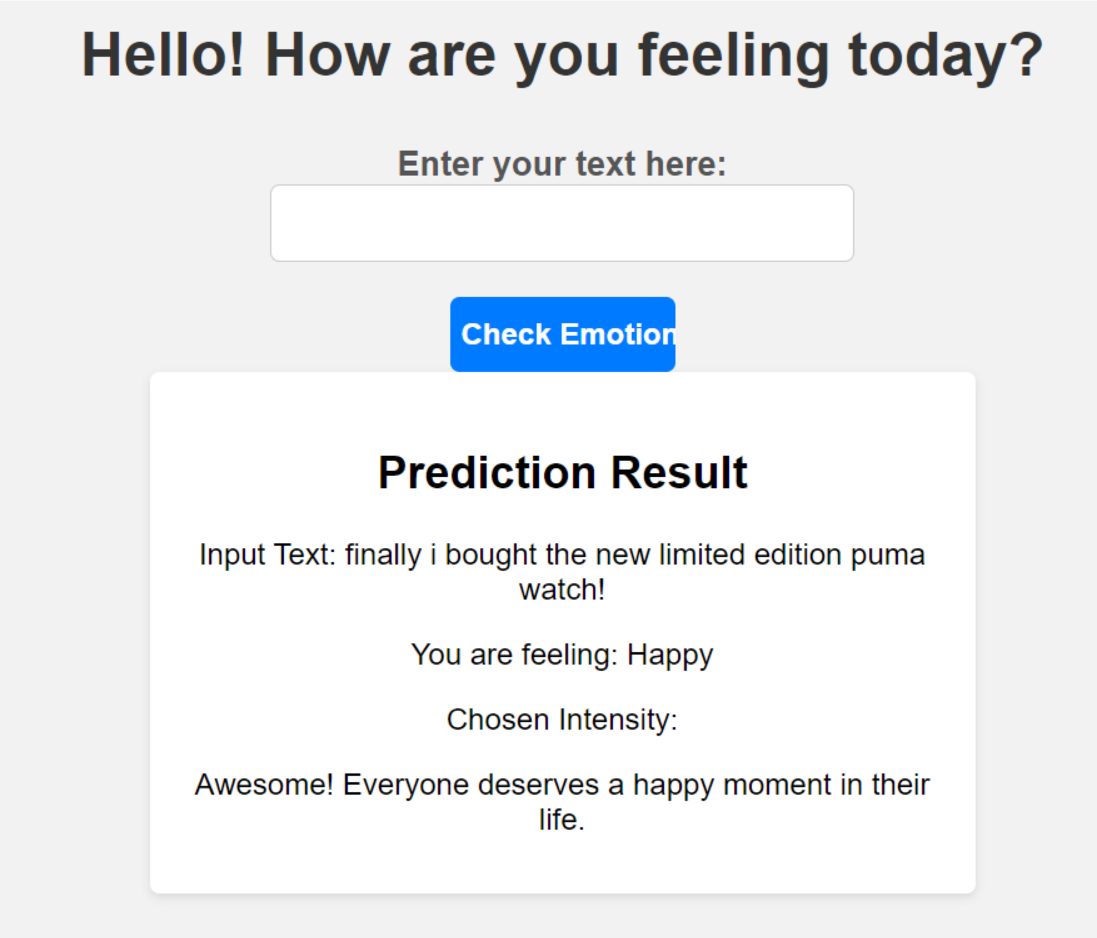

Figure 4.17: Output of predicted emotion

Based from the user input, user can generate a chart for the predicted emotion. This chart will display the user's mood throughout every time they use the website to record their emotions as can be seen in Figure 4.18. From this chart, user can see which emotions that they had the most and which emotions that they experience less.

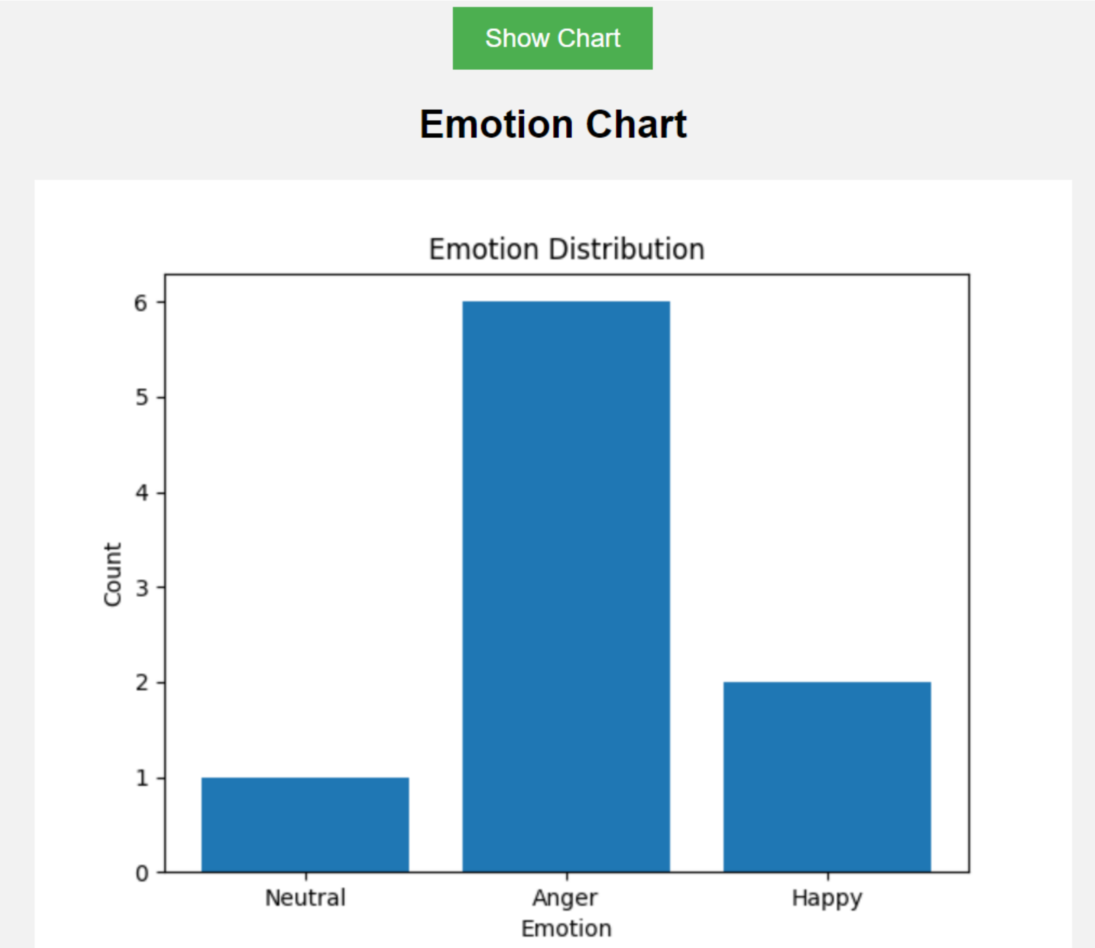

Figure 4.18: Show a chart of user's emotion

4.2.3 Database Design

4.2.3.1 Conceptual and Logical Database Design

The database for this project is the user's text input and the predicted emotion. The text input data and the predicted emotion are stored in a database file (.db) using SQLite and saved in the system web folder for future use to implement more advanced visualization. The file is saved as "mood_data.db". Figure 4.19 shows the location of the stored database in the specified folder.

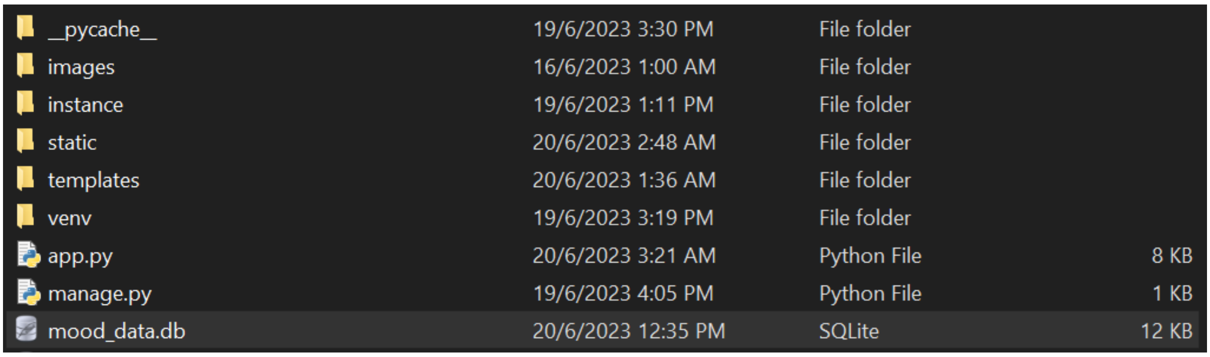

Figure 4.19: The database is saved in the website system folder

The image for displaying the chart of user's recorded emotion is saved in the "static" folder in the website system folder. In the "static" folder, there are other folders such as "css" and "js" that are involved in creating the website of the system. Figure 4.20 shows the location of the chart image that is saved as "emotion_chart" in PNG (.png) file.

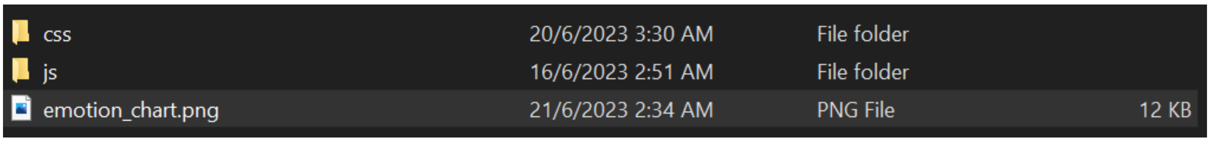

Figure 4.20: The location of the chart image in 'static' folder

4.3 Detail Design

This system uses deep learning model which is LSTM to develop and build the model. Thus, there are several software libraries and packages that need to be downloaded and installed so that the model can be developed properly. Some of them are pip, numpy, pandas, matplotlib, nltk, keras, tensorflow, sklearn, pickle and flask.

4.3.1 Software or Hardware Design

For this project, the chosen type of methodology is Agile methodology. The reason why Agile methodology is chosen because it is a collaborative process that involves ongoing collaboration and iterative work ("Agile project management", n.d).

This methodology operates on the principle that a project can be consistently enhanced throughout its lifecycle, allowing for rapid and responsive changes to be made. So Agile methodology is suitable for this project. Figure 4.21 below shows the phases that are involved in this methodology.

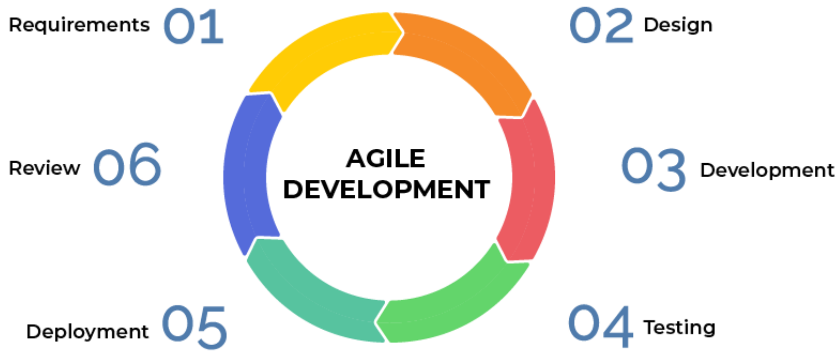

Figure 4.21: Phases of Agile methodology (Ta, 2020)

1) Requirements

In the first phase, the requirements for the proposed project are gathered. The requirements include identifying what type of emotions to detect, who is the target user, determining the desired accuracy levels, and what platform to use for developing the web app.

2) Design

After the requirements has been defined, the next step is to proceed with designing the architecture and components of the model. In this process, Bi-directional LSTM has been chosen as the architecture to use for building the model. After that, starts to plan the data pre-processing and other important process such as label encoding.

3) Development

During development phase, starts implementing the model with the desired architecture and start building the model. The model is separated into training, testing, and validation set. After performing label encoding on the 'label', starts training the model with specific parameters which are later fine-tuned to optimize the performance.

4) Testing

After done with training, the model is evaluated on the testing set to see the effectiveness of the text emotion detection. Tested the trained model on new unseen data and get the classification report on evaluation metrics such as precision, recall, and f-1 score.

5) Deployment

The next phase is deployment of the model to production environment. Starts developing a web app using Flask platform and integrate the model with the website. The performance of the model and website is observed and monitored.

6) Review

The last phase is reviewing. After the web app has been deployed to use, gather user feedback and continue monitoring the model's performance. Feedback gained from user will be used to implement improvement and updates.

4.4 Conclusion

As a conclusion, the detailed description of the system design process flow ensures smooth system operation and the achievement of objectives. The design phase provides an understanding of the system's development and deployment. The following chapter will dive into the implementation phase.

CHAPTER 5: IMPLEMENTATION

5.1 Introduction

This chapter will discuss on the implementation phase of this project that includes software development environment setup, software configuration management, and also implementation status. Jupyter Notebook computing environment was used as a platform for data pre-processing, training the deep learning model and evaluating the prediction of the model. Meanwhile, Sublime Text is used as a platform for writing python codes to create a Flask-based web app and also launch the web app in the browser. For the training and validation process, the dataset is trained by using pre-loaded model which is Word2Vec from Gensim library which is a popular open-source Python library designed for natural language processing (NLP) tasks. The validation process is used to monitor the performance of the model during training and also avoids overfitting. The configuration environment setup and version control procedure will be stated and explained in the next section.

5.2 Software Development Environment setup

This project is developed using Python programming language. Python is an easily interpreted and high level programming language and it can be installed free on Windows, MacOS and Linux. Python is chosen for this project because it is easy to learn and write code on it. Since Python executes the code line by line, there is no need to compile the code. In this, the version of Python used is 3.10.8. Figure 5.1 below shows the official logo of the Python.

Figure 5.1: The logo of Python

After Python installation, the code will be written on a Jupyter Notebook. To use Jupyter Notebook, Anaconda Navigator is required to be installed first. It is an open-source distribution for Python programming language designed for scientific computing. It is very suitable for data analysis and data science tasks. Figure 5.2 below shows the logo of Anaconda Navigator. In addition, a text editor is used to write code for creating a Flask web app for this project which Sublime Text. Sublime Text is a free and highly customizable text editor that support multi language such as HTML, CSS, Javascript, Pyhton, and many more. Sublime Text is chosen because it is very lightweight and fast even when working with large files. Figure 5.3 below shows the logo of Sublime Text.

Figure 5.2: The logo of Anaconda Navigator

Figure 5.3: The logo of Sublime Text

5.3 Software Configuration Management

5.3.1 Configuration Environment Setup

This section will discuss the main software that are involved in developing the deep learning model and explain the configuration settings for the environment. First, for the Python software, it can be installed directly from their official website which can be seen in Figure 5.4. After installing the Python, simply launch the Python and after it is done setting up, the app is ready to use. Figure 5.5 shows the interface of installed Python on a device and its version.

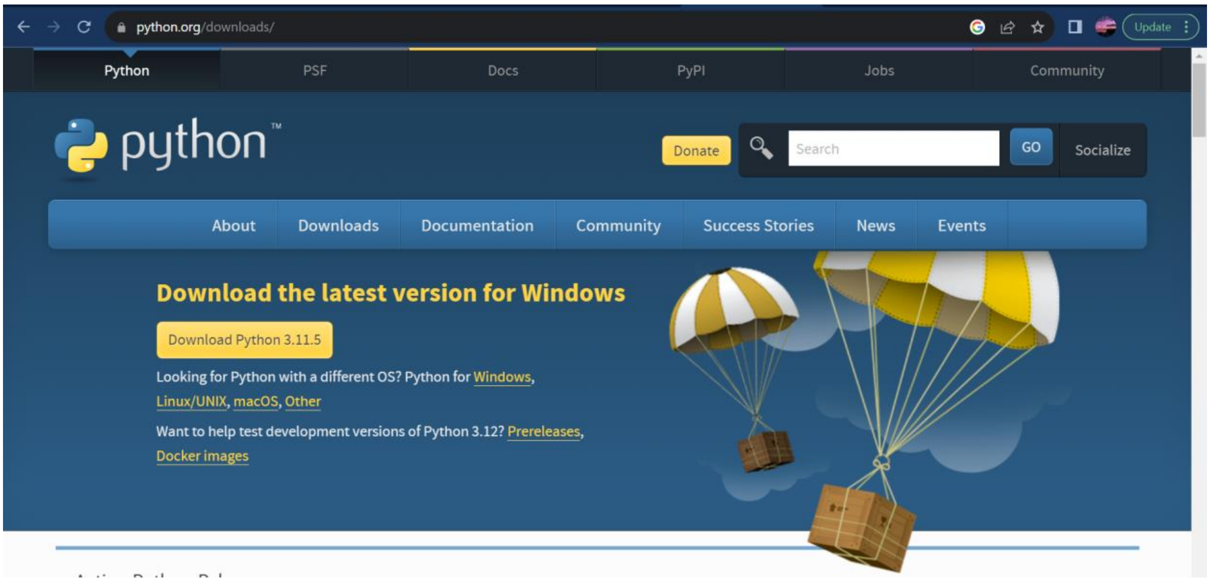

Figure 5.4: Python's official website

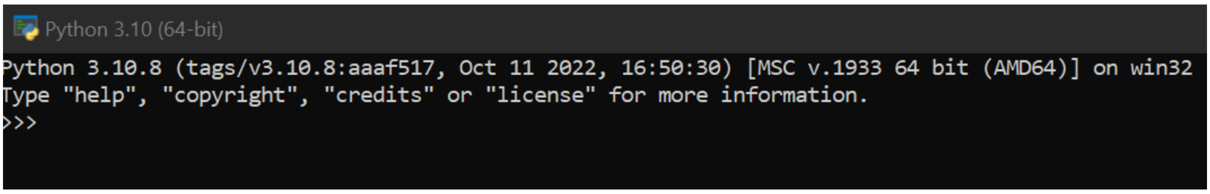

Figure 5.5: Interface of Python and its current version

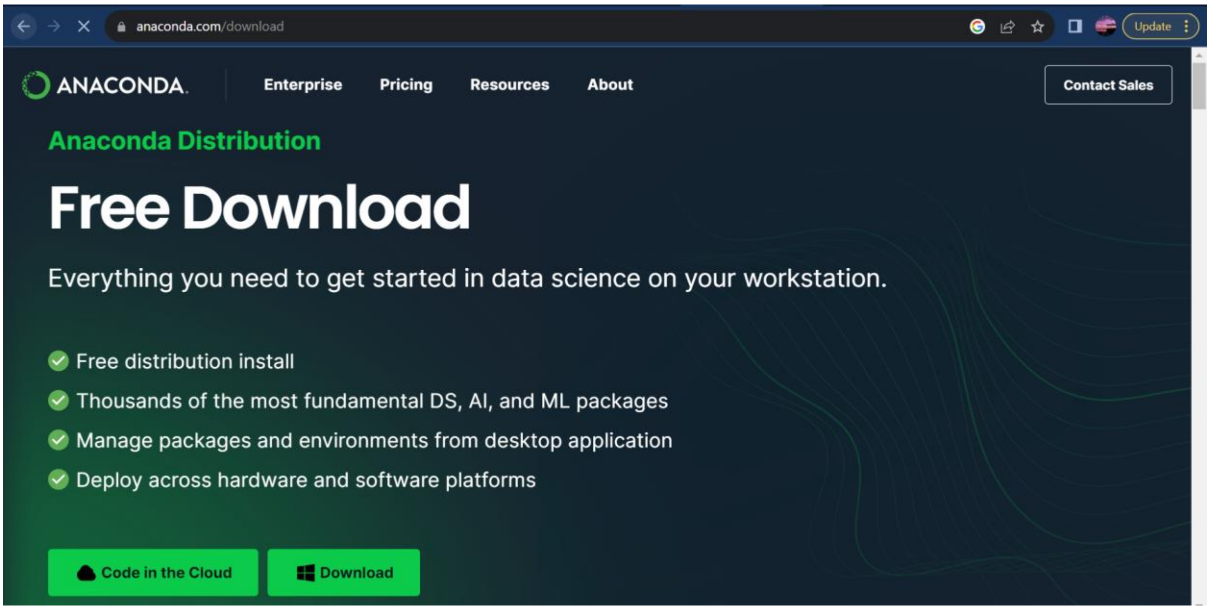

Figure 5.6: Anaconda Navigator's official website

To launch Jupyter Notebook for writing Python code, user needs to install Anaconda Navigator as shown in Figure 5.6 above. After installing, user can finish the configuration settings and simply launch the app. User can choose to either open Jupyter Notebook from the Anaconda Navigator menu as can be seen in Figure 5.7 or want to straight away open it directly from the windows menu as can be seen in Figure 5.8. Nonetheless, it is very easy to access.

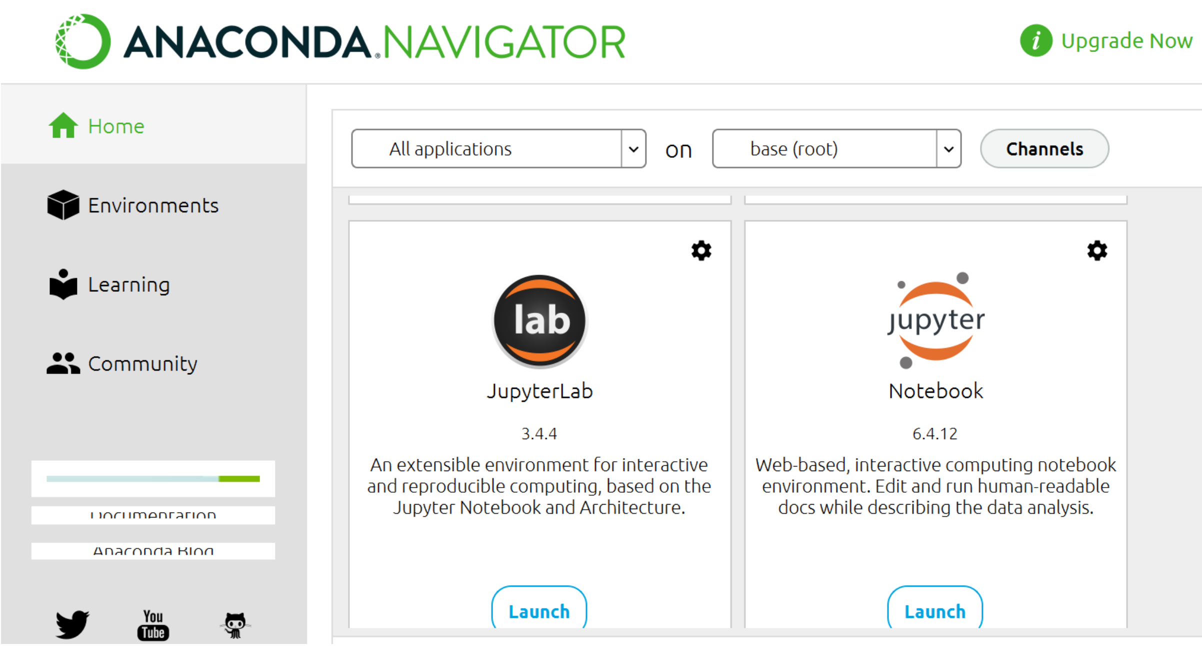

Figure 5.7: Access Jupyter Notebook from Anaconda Navigator

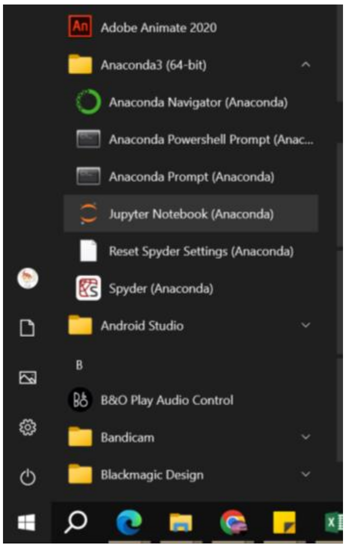

Figure 5.8: Access directly from the windows menu

After that, user can open up Jupyter Notebook to start writing Python codes. Jupyter Notebook will allow user to write code at any location or folder they want. It can easily access the available data as long as the data is in the same destination as the notebook. Jupyter Notebook will automatically save user's notebook if it is leave open for a long time so user do not have to worry about losing the codes. Figure 5.9 below shows the interface of Jupyter Notebook.

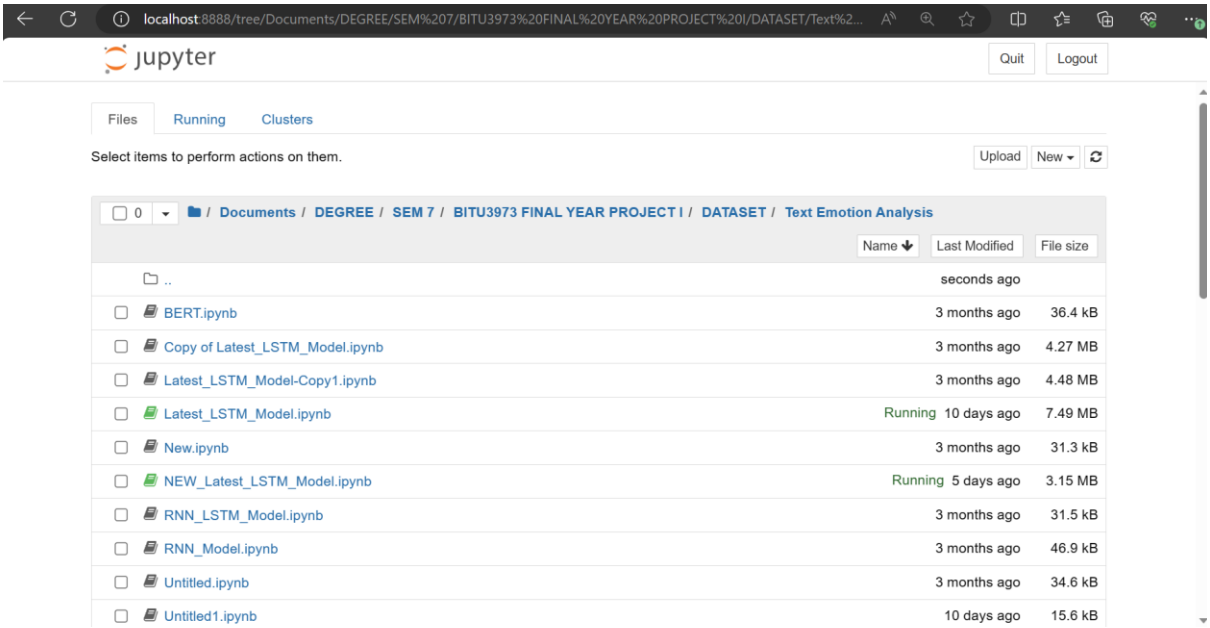

Figure 5.9: Environment of Jupyter Notebook

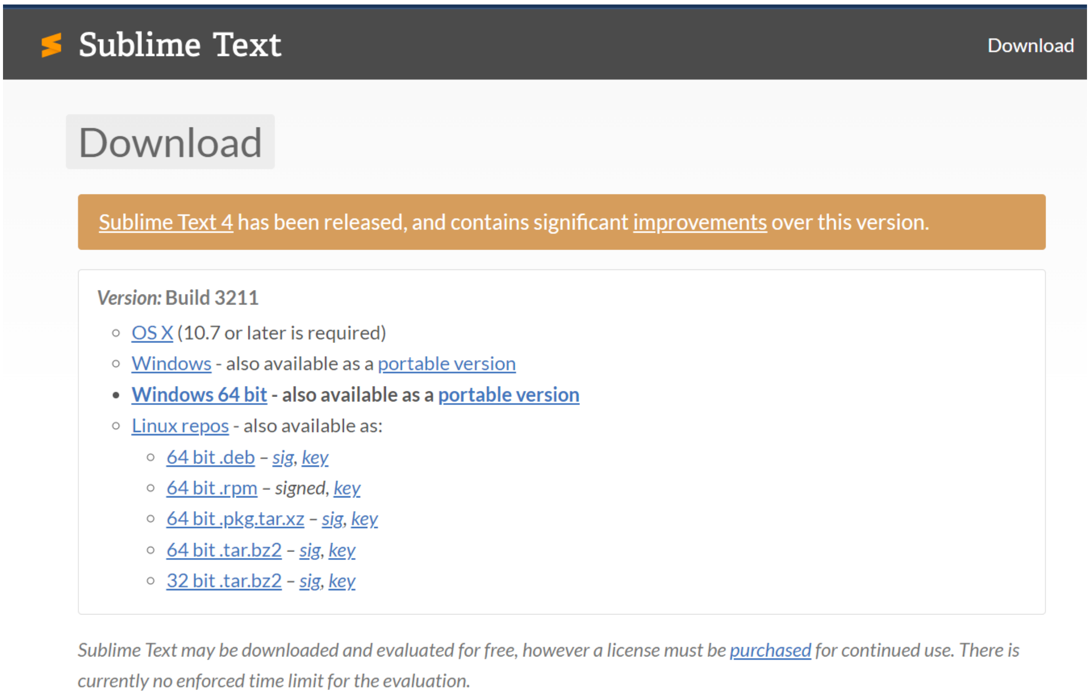

Figure 5.10: Sublime Text official website

Finally, for Sublime Text, it is available to be installed for free on their official website. Figure 5.10 above shows the official website of Sublime Text. To configure the environment of Sublime Text, user just needs to open up Sublime Text after done installing and go to View and then click on Show Console and it will install the package control as shown in Figure 5.11 below.

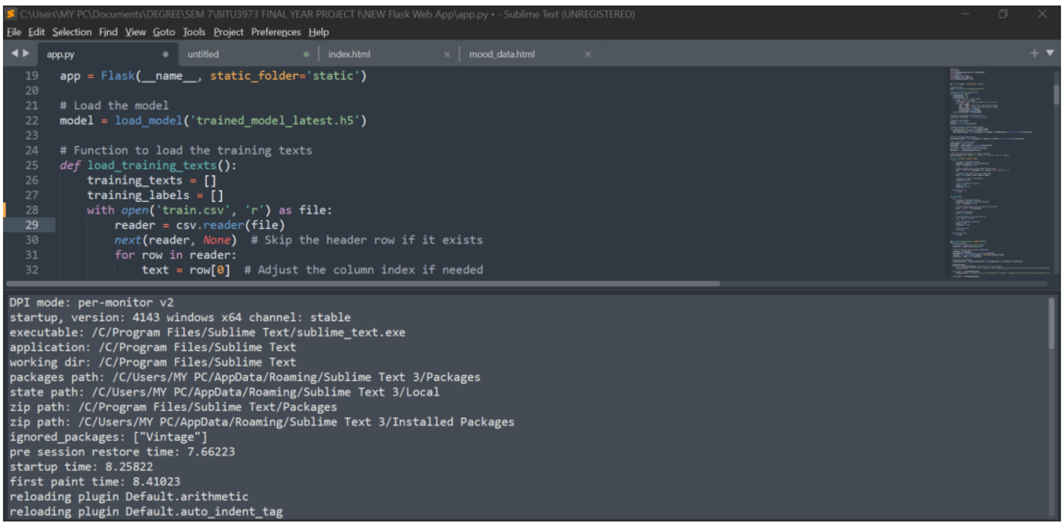

Figure 5.11: Configure environment in Sublime Text

5.3.2 Version Control Procedure

Table 5.1: List of installed packages

Package Name	Version
python	3.8.10
pip	23.1.2
sklearn	1.0.2
keras	2.12.0
tensorflow	2.12.0
pandas	1.4.4
numpy	1.23.5
matplotlib	3.5.2
flask	1.1.4
gensim	4.1.2

5.4 Implementation Status

This section will describe the implementation status and their duration for each progress in this project. Additionally, there are two categories of implementation status which can be seen in Table 5.2 and Table 5.3 below. Table 5.2 will describe about the development implementation status while Table 5.3 will describe the module analysis implementation status. The module name, module description, the duration to complete, the date will be detailed in the following table below.

Table 5.2: Development Implementation Status

Task	Duration completed (days)	Date completed
Research and Data collection	30	21 April 2023
Proposed Technique	14	5 May 2023

Literature Review	21	26 May 2023
Design Model	21	16 June 2023
Others	14	30 June 2023

Table 5.3: Module Analysis Implementation Status

Module name	**Description**	**Duration completed (days)**	**Date completed**
Artificial Intelligence Module:			
Pre-processing Train the data Performance evaluation Recommendation Development	The processes are data cleaning and tokenization Train the data using Word2Vec from Gensim library Quantifiable measures metrics: Accuracy, Precision, Recall, F-1 score Improve the quality of prediction and web app	10 21 15 15	11 August 2023 25 August 2023 25 August 2023 1 September 2023
System Module:			
Design website	Create a simple web app using Flask framework	7	8 September 2023

5.5 Conclusion

As a conclusion, the implementation phase serves as a crucial roadmap for the project, to ensure the successful execution of the system. This is important as certain software and packages are compatible only with specific versions, and it provides clear guidance for subsequent steps. The next chapter will discuss the testing phase of this project.

CHAPTER 6: TESTING

6.1 Introduction

In this chapter, the trained model will be tested and evaluated in order to make sure the model achieved desired accuracy to give the best prediction on detecting emotion from text. Performance of the model will be evaluated using quantifiable metrics such as Accuracy, Precision, Recall and F-1 score. It is important to note that testing phase is the last phase of the project before the improvement and recommendation, that is why this phase need to be properly done to make sure that there will be no errors and the system can function properly. This chapter will include the test plan, test strategy and test result and analysis.

6.2 Test Plan

A test plan is a comprehensive document that outlines the approach, scope, resources, schedule, and deliverables for testing project. It serves as a guide for the testing process and provides a clear framework for how testing activities will be conducted. Test plan is crucial since it can serve as a guide for someone who is not from the development team to understand easily on how the system works and how it is tested. Additionally, when all the information is put together in one document, it makes it easy for the managers to look at it or use it again for other projects.

6.2.1 Test Organization

In order to do testing phase, there will be several people involved and each people will have their own role. This can be referred in Table 6.1 below where the people involved are each given their role and their own responsibility in implementing this testing phase. They will contribute in checking the system to catch any errors and also see if the requirements of the system are met.

Table 6.1: Roles and Responsibilities involved in testing phase

Role	Responsibility
Developer	The person who developed the project and system. Developer must make sure the project meet all the requirements and check the testing phase in detail so the system will run smoothly.
Course mate	The person who acts as a user or tester of the Flask web app. They will try to input the text and check if it is able to detect correctly.
Supervisor:Dr Noor Fazilla Abd Yusof	The person who supervised the project and guided in doing the project. Supervisor will test the system to see if it is functioning correctly.

6.2.2 Test Environment

Testing the system is done in a working or studying place with internet access. There are two types of setup in this system which are physical setup and logical setup. Physical setup is involved in hardware components while the logical setup is involved in software components. Listed in the Table 6.2 below are the details of the test environment along with the descriptions of the two components.

Table 6.2: Test Environment

Component	Tool	Description
Hardware	Operating system	Microsoft Windows 10, 64 bits
	Processor	Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz 2.71 GHz
	Memory / RAM	12 GB
	Storage or Hard Disk	Local Disk (C:): 480 GB DATA (D:) : 449 GB
	Input Devices	Keyboard Mouse
Software	Jupyter Notebook	Platform to write and execute Python code for building the model and evaluate the model performance
	Sublime Text	Platform to write Python code to create a Flask web app and to run the web app
	Browser	Display the web app in Google Chrome or Microsoft Edge

6.2.3 Test Schedule

Test schedule is created in order to make sure the test phase is working smoothly. The test schedule includes testing task, its responsibilities and the target start dates and end dates, which can be seen in Table 6.3. In every testing task, there should be a remark section to check whether every testing task is properly functioned or not. Test schedule will help in estimating the overall time required to finish the testing phase. The tester needs to follow the instructions on every task in order to achieve a good outcome.

Table 6.3: Test Schedule

Testing Task	Description	Duration (days)	Remark
Unit Testing	To test if every part of the system is functioning properly	3	OK
Model Testing	To test the model functionality	3	OK
User Input Function	To test if the user is able to get correct prediction from their input text	1	OK
Prediction by Bi-LSTM RNN	To test if the model can predict the correct emotion based from text input	3	OK

6.3 Test Strategy

This project applied white-box test strategy which involves in examining the internal logic, code structure, and implementation details of the software. Testers who will conduct white-box testing are the software developers who have knowledge of the internal workings and use this to design the tests, including the source code. White-box testing will focus on testing the program code such as code structure, training the model, evaluating the performance of the model and giving the correct output of emotion detection.

6.4 Test Implementation

6.4.1 Experimental / Test Description

In this section, the process starts with finding all the test cases involved. A typical test case usually includes user input, display prediction emotion, expected result and actual result. The expected results are recorded to categorize the emotions.

The data then will be utilized to calculate the accuracy of the model and will generate the predicted emotions. The details can be referred in Table 6.4 and 6.5 below.

Table 6.4: Deep Learning Model Testing Description

Test Case Number	Action	Expected Result	Actual Result
T01	Load text data	Text data are loaded	OK
T02	Pre-processing	Data cleaning, tokenization and split into training, validation and testing	OK
T03	Train model	Train the deep learning model using Bi-LSTM RNN	OK
T04	Evaluate model performance	Evaluate the model performance with quantifiable metrics	OK

Table 6.5: Flask Web App for detecting emotion system

Test Case Number	Action	Expected Result	Actual Result
T01	User input text	Text input is read and stored in database	OK
T02	Emotion prediction	System give the prediction for the input text	OK
T03	Display emotion chart	User click on show emotion button and the system will show a chart of emotions	OK

6.4.2 Test Data

The dataset for this project is split into training, validation and testing with 80%, 10% and 10% distribution respectively. The test data contains 2000 data from the total of 20,000 data and is used for testing the emotion prediction after training the model. After the evaluation performance, the model is also tested on random data which is not taken from the dataset.

6.5 Test Results and Analysis

This section will explain the test results for the developed model and compare it to training and validation results to see if the model performance is good enough. During early implementation of building the model, the performance of the model was very poor. The result can be seen in Figure 6.1 below. In Figure 6.1, the model is built with a Bi-Directional LSTM layer with an additional dense layer and achieved 71% on the training accuracy and 58% on validation accuracy. In this case, it is clear that the model was overfitting since the training accuracy is high but the validation accuracy is very low and is not increasing.

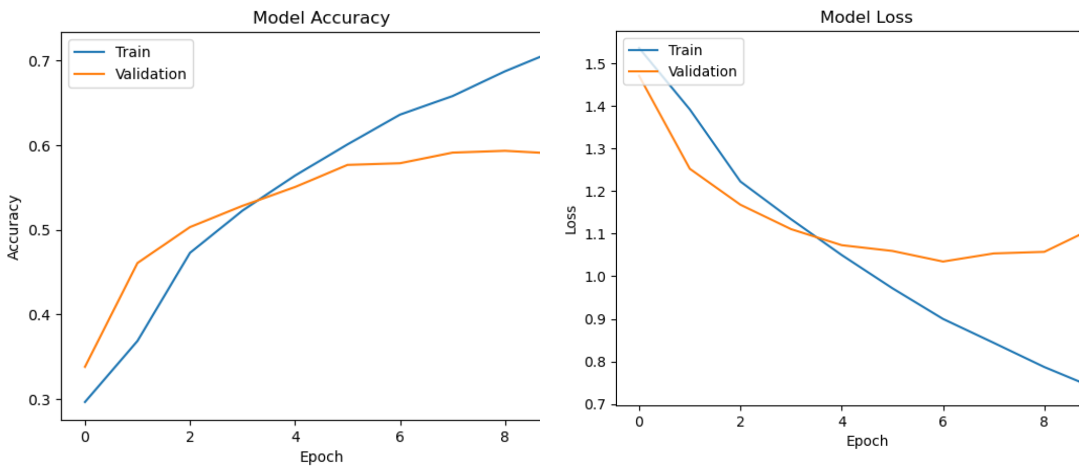

Figure 6.1: Training and validation accuracy with one Bi-LSTM layer

After doing thorough checking, it is found that the model has problems with the dataset that was used. The dataset used was unreliable because the labelling for emotions for some of the data was not correct. This has led to a problem where the validation accuracy was very low because the accuracy of a model is calculated based on the number of correctly predicted instances divided by the total number of instances. So if the labels are not correctly labelled, the model will not make accurate predictions based on the provided labels, thus resulting in a low accuracy.

Therefore, the dataset is replaced with another one. The new dataset is checked thoroughly to make sure that the labelling of emotions for the text data is correct. Additionally, an algorithm is added into the model to do word embedding. The chosen algorithm is Word2Vec which is loaded from Gensim library. Figure 6.2 below shows the implementation of Word2Vec in the model. For model building, another two Bi-LSTM layers were added, which can be seen as in Figure 6.3 below. The reason for adding another two Bi-LSTM layers is to give the model more capacity to learn from the data. Similarly, there are also dropout layers to prevent the model from overfitting.

```
from gensim.models import Word2Vec

model_w2v = Word2Vec(x_train + x_test + x_validation,
                     vector_size = 300,
                     min_count = 2).wv
```

Figure 6.2: Implementation of Word2Vec

```
model.add(Bidirectional(LSTM(128, return_sequences=True)))
model.add(Dropout(0.2))
model.add(Bidirectional(LSTM(256, return_sequences=True)))
model.add(Dropout(0.2))
model.add(Bidirectional(LSTM(128, return_sequences=False)))
model.add(Dense(6, activation = 'softmax'))
model.compile(loss='sparse_categorical_crossentropy', optimizer='adam', metrics='accuracy')
```

Figure 6.3: Bi-LSTM layers for building model

After implementing Word2Vec into the model, the training and validation accuracy are observed and recorded. It is noticeable that after replacing the database with another data and also adding an algorithm into the model, the training accuracy and validation accuracy of the model has significantly improved. The result can be seen in Figure 6.4 below where both the training and validation accuracy better than before. Meanwhile the training and validation loss also not greatly varied.

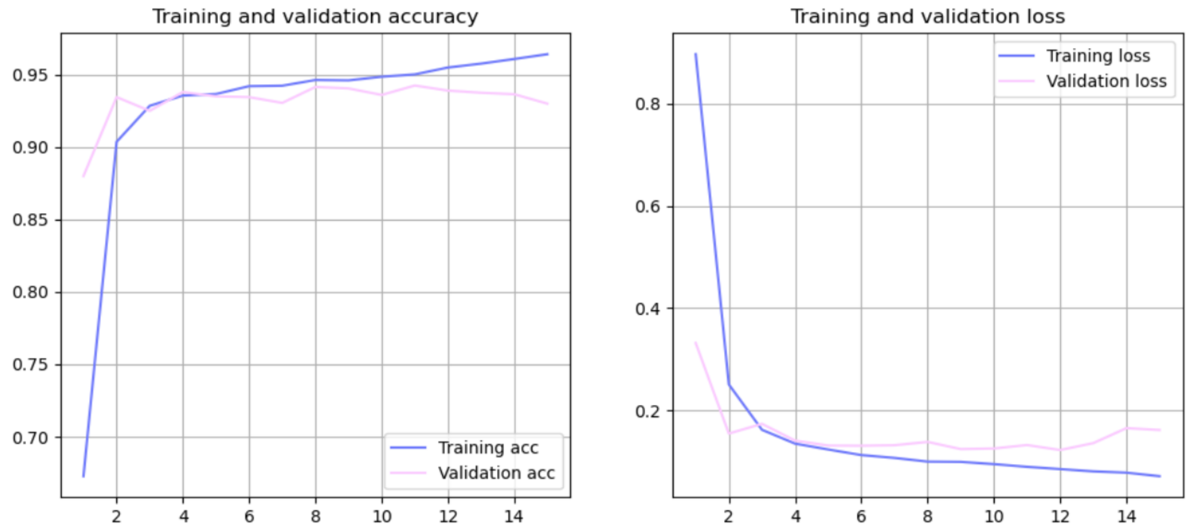

Figure 6.4: Plot graph after replaced with new dataset

Besides accuracy, the model is also evaluated by using precision, recall and F-1 score. In Figure 6.5 below, for each of the class labels, overall the precision score does well except on class number 1 where it might be a bit challenging for the model to identify. Meanwhile class number 2 and 6 are a bit low in value for recall. F-1 score indicates the weighted average between precision and recall and majority of the score are almost equal to which is okay.

	precision	recall	f1-score	support
0	0.91	0.94	0.93	275
1	0.78	0.91	0.84	159
2	0.93	0.84	0.88	224
3	0.97	0.92	0.95	695
4	0.96	0.97	0.97	581
5	0.70	0.86	0.77	66
accuracy			0.93	2000
macro avg	0.87	0.91	0.89	2000
weighted avg	0.93	0.93	0.93	2000

Figure 6.5: Table of precision, recall and f-1 score for the model

6.6 Conclusion

As a conclusion, testing phase is crucial as it ensures that all components of the system will operate effectively and obtain accurate results. This chapter has described and explain various types of tests that need to be carried. The next chapter will be discussing about the conclusion of the project, highlighting the strengths and weaknesses, offering suggestions for improvement and also discuss the project's contribution.

CHAPTER 7: CONCLUSION

7.1 Observation on Weaknesses and Strengths

When developing this project, the most notable weakness is the problem with the dataset. Many datasets that are collected for this project are not reliable at labelling the emotions of the text. Finding the suitable and reliable dataset quite a challenge as the dataset itself can affect the performance of the model. On the other hand, the strength of this project is the model has finally achieved high accuracy after replacing with new dataset and the prediction becomes more accurate.

7.2 Propositions for Improvement

As the weakness have been point out, then it must be fixed in order to make improvement in future. Among the propositions for improvement with the problem of unreliable dataset is by manually labelling the text data and check if the label is correct or not. But the downside for this solution is it requires more than one person to work on the labelling and it also requires knowledge from experts to correctly labelled the text.

7.3 Project Contribution

The main contribution of Text Emotion Detection using Bi-Directional Long Short-Term Memory is to be able to detect emotion from user's text input accurately and help monitor user's input emotion.

7.4 Conclusion

As a conclusion, this project thoroughly explores both the strengths and areas for improvement. Enhancements for this project could involve integrating more advanced deep learning models and providing users with more accurate text emotion detection. The potential for significant advancement in the future is evident in this system.

REFERENCES

Depressive disorder (depression). (2023, March 31). Depressive Disorder (Depression). https://www.who.int/news-room/fact-sheets/detail/depression#:~:text=An%20estimated%203.8%25%20of%20the,world%20have%20depression%20(1).

GBD Results. (n.d.). Institute for Health Metrics and Evaluation. https://vizhub.healthdata.org/gbd-results

How to Use a Mood Tracker. (2023, April 13). Verywell Mind. https://www.verywellmind.com/what-is-a-mood-tracker-5119337

Kalita, D. (2022, March 11). *A Brief Overview of Recurrent Neural Networks (RNN)*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2022/03/a-brief-overview-of-recurrent-neural-networks-rnn/

Saxena, S. (2021, March 16). *Learn About Long Short-Term Memory (LSTM) Algorithms*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/

Jayawardhana, S. (2020, July 30). *Sequence Models & Recurrent Neural Networks (RNNs)*. Medium. https://towardsdatascience.com/sequence-models-and-recurrent-neural-networks-rnns-62cadeb4f1e1

Mungalpara, J. (2021, March 2). *What does it mean by Bidirectional LSTM?* Medium. https://medium.com/analytics-vidhya/what-does-it-mean-by-bidirectional-lstm-63d6838e34d9

Shekhar, S. (2021, June 14). *What is LSTM for Text Classification?* Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/06/lstm-for-text-classification/

What is LSTM - Introduction to Long Short Term Memory. (2020, May 28). Intellipaat Blog. https://intellipaat.com/blog/what-is-lstm/

Yadav, H. (2023, May 31). *Dropout in Neural Networks*. Medium. https://towardsdatascience.com/dropout-in-neural-networks-47a162d621d9

Verma, Y. (2021, September 19). *A Complete Understanding of Dense Layers in Neural Networks*. Analytics India Magazine. https://analyticsindiamag.com/a-complete-understanding-of-dense-layers-in-neural-networks/

What Is Agile Methodology in Project Management? (n.d.). What Is Agile Methodology in Project Management? https://www.wrike.com/project-management-guide/faq/what-is-agile-methodology-in-project-management/