MENTAL HEALTHCARE CHATBOT

BY

OON XIN YI

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONOURS)

Faculty of Information and Communication Technology

(Kampar Campus)

JAN 2021

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: _____MENTAL HEALTHCARE CHATBOT_____

Academic Session: ___ Jan 2021____

I _____OON XIN YI _____

(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

_____ _____

(Author's signature) (Supervisor's signature)

Address:

_53,Jalan Tasek 51_____

_ Bandar Seri Alam _____

_81750 Masai,Johor,Malaysia _

Jasmina Khaw Yen Min

_____

Supervisor's name

Date: 16 April 2021 **Date**: 16 April 2021

MENTAL HEALTHCARE CHATBOT

BY

OON XIN YI

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION TECHNOLOGY (HONOURS)

COMPUTER ENGINEERING

Faculty of Information and Communication Technology

(Kampar Campus)

JAN 2021

DECLARATION OF ORIGINALITY

I declare that this report entitled "**MENTAL HEALTHCARE CHATBOT**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : 

Name : OON XIN YI

Date : 16/04/2021

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisors, Dr. Jasmina Khaw Wen Min who has given me this bright opportunity to engage in speech and voice recognition project. A million thanks to you.

I will be not able to complete my course without the support and help from my family and friends. Therefore, I must express my gratitude to my family and friends for their unconditional love, support, and continuous encouragement throughout the course.

Abstract

More and more mental health issues such as depression are getting known and recognized by our society today. However, not all of them can receive appropriate treatment. There are many of us still facing the problem of getting the appropriate mental health services every day. We cannot deny the fact that not everyone can get mental healthcare services as they might face some difficulties such as financial problems. Therefore, we may look for new solutions to fix this mental health issue. This demand for solving this issue has led to the proposal of technology as a solution. Chatbot, also known as a conversational agent which can participate in the conversation might be considered one of the solutions too. By mimicking the conversation between human counselor and patient, it can provide counselor service to the patient at some point. However, to further improve the quality of the counselor service, the improvement of the chatbot has to be carried out. By using deep learning, this proposed chatbot can recognize the meaning of the conversation and give a relevant response. Moreover, by using speech reorganization and speech synthesis, the chatbot can serve people in more ways. For example, those who are blind can use this chatbot by speaking to the chatbot and receive the output by listening to the sound. Moreover, this chatbot also able to serve those has a listening problem since they can just read the sentences that output by the chatbot.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

1 D	One Dimension
ASR	Automatic Speech Recognition
API	Application Program Interface.
CBT	Cognitive Behavior Therapy
CMC	Computer Mediated Communication
CNN	Convolutional Neural Network
DBT	Dialectical behavior therapy
DNN	Deep Neural Networks
FYP	Final Year Project
JSON	JavaScript Object Notation
NLP	Natural Language Processing
MFCC	Mel-frequency spectrum
ML	Machine Learning
QR code	Quick Response code
RAVDESS	Ryerson Audio-Visual Database of Emotional Speech and Song
Siri	Speech Interpretation and Recognition Interface
STT	Speech-to-Text
TTS	Text-to-Speech

TESS	Toronto emotional speech set
UI	User Interface
WHO	World Health Organization

Chapter 1: Introduction

1.1 Introduction

According to World Health Organization (WHO), mental health can be defined as "Health is a state of complete physical, mental and social well-being and not merely the absence of disease or infirmity." (WHO, 2018) As we know, life does not always go well. There are always up and down in our life. However, not everyone able to bear the stress well when the problem comes to them. At this moment, they are suggested to find help for their mental illness.

However, not everyone is lucky enough to have the right to access mental healthcare services. Community-based mental health care is also rare in low-income countries; about 52% of low-income countries offer community-based mental health care programs, compared to about 97% of high-income countries (Saxena et al, 2007). Having poor financial issues is not the only obstacle they need to face when they are seeking medical help. The limited availability of medication and health professionals in the mental healthcare field in their country even makes the scenario worst.

The purpose of building this chatbot is to offer some mental healthcare services to people without charging any cost. The service will able to deliver to them wherever and whenever they are. All they need is a device that able to connect to the internet and then the people at least will have the more easy option to relieve their stress and anxiety.

1.2 Problem Statement and Motivation

Problem Statement

Although there are many existing chatbots, there are also several problems that need to be solved to give better mental healthcare services.

i. Lack of voice-based communication applied in mental healthcare chatbot

Most of the existing chatbots that giving mental healthcare services in the market are text-based communication such as Woebot and Wysa. The user only can communicate with the chatbot by just texting or selecting the options that are provided by the chatbot. The user needs to do extra work such as import the library to the system if they want the chatbot able to speak to them. This will extremely hard for those who do not have relevant knowledge. For the proposed chatbot, there is no need to trouble the user since voice-based communication is done by building the chatbot. All the user needs to do is choose either input in voice or audio form only. We truly hope that this chatbot can reach out to help different kinds of people especially those who have difficulty in hearing and vision since the chatbot able to provide both voice and text-based communication. Besides, it can be an advantage if the chatbot possesses more ways to carry out the communication so that it can provide a better service to the people.

ii. Lack of diverse content of response for mental healthcare chatbot

The disadvantages of most chatbots are the lack of generality and variation in style. (Woudenberg, A.V, 2014). If the chatbot is built by using some API such as Facebook Message Bot, the user might get the same response over and over again if they are kept asking the same things to the chatbot. Not to say that, if the user inputs something that chatbot never know before, the chatbot would not know the meaning of the word and unable to respond back correctly. Chatbot that replies to the same response maybe can use for other fields but it is might not suitable for providing counseling mental healthcare service to humans. To providing more diverse content of the response from the chatbot, we are using deep learning to train the chatbot so that the chatbot can recognize the meaning of the words more accurately and able to give a relative response.

Motivation

The motivation for creating this chatbot is to provide an easier and convenient alternative way of giving mental healthcare services. This chatbot is introduced to focus on giving service to all people to ease their suffering and make their life become better. Everyone no matter who they are should have the right of getting medical help equally. Hence, it is meaningful for me to make use of the knowledge that I learned from university and contribute value to the community and make this world become better.

1.3 Project Scope

The purpose of this project is to develop a chatbot that can provide a counseling service that is able to give different responses to humans by using machine learning to recognize different text inputs and give related responses to the user. This chatbot also can carry out voice-based or text-based communication with a human by using Automatic Speech Recognition (ASR).

i. This chatbot only supports oral and text conversations in English. Another language is not supported currently.

ii. This chatbot only focuses on giving the content of the response which is related to the mental healthcare support service only. The conversation which involved other unrelated field is not supported.

iii. This chatbot will able to recognize the meaning of the text input from the user whether in voice or text form and respond back to them accordingly. The response of the chatbot will be in voice form and text form together.

iv. This chatbot will try to avoid giving the repeating response although it keeps getting the same text input from the user.

v. This chatbot is only focused on providing counseling services to humans and it will not be giving any diagnosis to the user.

1.4 Project Objectives

The objectives of this project are:

i. To implement oral and text communication between humans and chatbot by using speech recognition and speech synthesis systems.

    a. Speech recognition will be implemented in this chatbot to recognize the voice input from humans and convert it to text. Meanwhile, the chatbot should be able to convert the text response to voice by using speech synthesis.

ii. To develop a mental healthcare chatbot that able to give matching responses according to what human input whether in text or voice from.

    a. This chatbot would recognize the meaning of the words that input from humans and output the related responses accordingly. It would avoid giving repetitive responses to humans although it had detected the same meaning of the words from the conversation. Besides, this chatbot also able to recognize the emotion of the voice input from the user and make a relative response.

1.5 Impact, Significance, and Contribution

By the end of this project, we hope that the proposed chatbot is able to carry out a counselor service to the user at some points. So that this chatbot can give help to people who are having problems with access to counselor service. In another word, it allows the user to access self-help information without spending any cost. We hope that the chatbot counselor service can deliver to those in need of help more easily and conveniently.

Besides, the user can experience the counselor service before going to a human counselor. It helps the user to think and reconsider whether they really need the help of a human counselor or they just want a way to relieve their stress and anxiety. Hence, this makes the human counselor service deliver to someone that needs the help and prevent waste of human resources in the mental healthcare field.

Moreover, this chatbot could be useful to those who are handicap or unable to type the words by themselves. Since this chatbot accepts input in whether text or voice form, this makes the chatbot can serve more kinds of people including those who may have difficulty with typing or speaking.

Nevertheless, this chatbot can provide the counselor service to the user whenever and wherever they are. This chatbot can act as a "first aid" to the user's emotional problem. It can help the user either when his bad mood comes to him anytime or there is no psychology clinic nearby his staying place ·

1.6 Background information

According to the World Health Organization (WHO)'s estimates, more than 300 million people are suffering from depression and it is increasing about 20% every 10 years (Chaib F, 2017). Depression is a state of low mood and an aversion to activity. It can affect a person's thoughts, behavior, motivation, feelings, and sense of well-being (de Zwart, P L et al.,2019). Therefore, we can see that depression is endangering our society and it should not be ignored by anyone.

To rescue the situation, chatbot have been deployed in the mental health field for a long time. Chatbots are ''machine conversation system[s][that] interact with human users via natural conversational language''(Shawar & Atwell, 2005, p. 489). This human-computer interaction technology was established academically half a century ago. In 1964, the programmable natural language processing program ELIZA was developed at the MIT Artificial Intelligence Laboratory by Joseph Weizenbaum. (Vaidyam, A. N., Wisniewski, H., Halamka, J. D., Kashavan, M. S., & Torous, J. B. , 2019)

Some studies have shown that the chatbot can help people to improve their mental health. Studies using internet-based, one-to-one text-based chat interventions for psychological support have shown feasibility and improvement in mental health outcomes when compared with wait-list conditions (Hoermann S, McCabe KL, Milne DN, Calvo RA., 2017).

However, we cannot deny that the chatbot still has some limitations when we do the comparison between human-human conversations andhuman–chatbot conversations. Chatbots are limited in their ability to have an extended goal-directed discussion and can offer little in the way of common history or shared experience.(Jennifer & W. Randolph,2015). Some chatbot cannot deal with mistakes like what humans will do. It examined a chatbot's robustness when faced with unconventional linguistic features from non-native ESL speakers (such as misspellings and incorrect word order) (Mak, B., & Coniam, D, 2008)

This may reduce the people's confidence in the chatbot. People were actually inclined to send more than twice as many messages to chatbots compared to other people, people feel less confident or comfortable communicating with chatbots(Hill, Randolph Ford and Farreras, 2015). Hence, we can say that the chatbot still has a lot of room to be improved in the future.

Chapter 2: Literature Review

2.1 Overview

The purpose of doing review on similar chatbots is to compare the chatbots that in the market and find out what are their strengths and weaknesses.

2.2 Review on Similar System

2.2.1 Woebot

Figure 2.2.1.1 Logo of Woebot

Introduction

Woebot is an artificially intelligent chatbot that uses the principles of cognitive-behavioral therapy (CBT) which is a researched clinical approach to treating depression. Woebot is a “fully automated conversational agent developed by Woebot Labs in San Francisco. It is designed to help people cope with feelings of depression and anxiety.It also can act as a tool to improve mental health.

Strengths

The conversation makes us of emoticons. Emoticons can help Woebot's response look more emotional and human-like. It also set the tone of the conversation and make the conversation feel easier and relax. This help the users to easier to express their feelings to the chatbot.

Woebot do not randomly reply the users by searching answer on the internet. In fact, woebot answer the users by using the method called Cognitive Behavioral Therapy (CBT) which is an evidence-based approach to treatment.

Weakness

There are few limitations existed in Woebot .Woebot accept input by user typing words into the content box or select the option that they provided. In other words, Woebot do not provide oral communication with the user. This is inconvenient for those who have visual impairment.

2.2.2 Wysa

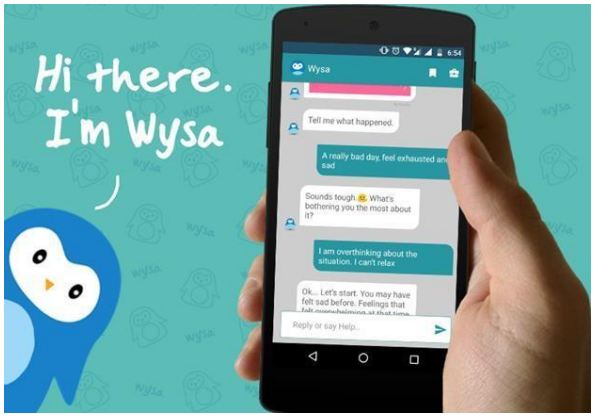


Figure 2.2.2.1 Logo of Wysa

Introduction

Wysa, is an artificial intelligence based chatbot produced by an Indian startup called Touchkin. It uses evidence based therapies like Cognitive Behavior Therapy. Wysa helped many users to deal with depression, anxiety and suicidal thoughts. The conversations in the Wysa were private and encrypted.

Strengths

Wysa also use CBT method to help the user to restructuring negative thoughts. Wysa use not only one but two evidence based self-help techniques which is Cognitive Behavioral Therapy (CBT) and Dialectical behavior therapy (DBT) to life coaching the users.

Wysa will continue the conversation based on previous question. This make the conversation become more close to real life counseling example because all the conversation will focus on the feeling of the users. The conversation will not suddenly jump to other topic. In fact, it will stay in topic and direct the user on how to get over their difficulties.

Weaknesses

Wysa also has same limitations like Woebot which is accept input in the form of words or select the option provided. The interaction would be limited due to lack of sophisticated entry. With prefilled answers and guided journeys, Wysa felt more like an interactive quiz or game than a chatbot.

2.2.3 Apple Siri

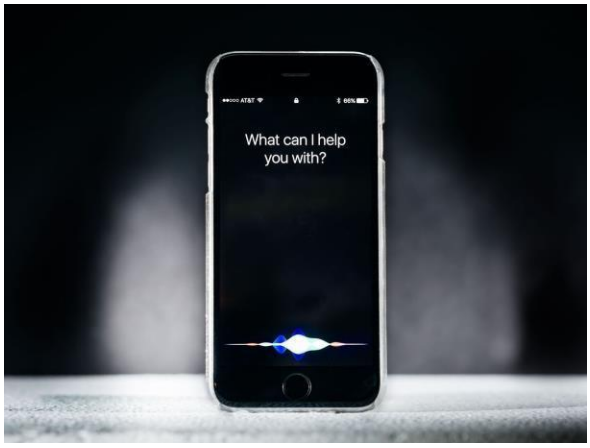


Figure 2.2.3.1 Siri running on iPhone

Introduction

Siri (Speech Interpretation and Recognition Interface) is an artificial intelligent software built onto Apple's iOS and MacOS system. It uses Natural Language Processing (NLP) technology, allowing users to interact with their mobile phones to perform some function such as completing search, querying the weather, setting up a mobile calendar, setting alarms, and more (Apple,2018)

Strengths

Apple Siri able to answer all kinds of questions. If it encounter some questions that it unable to answer, it will perform searching on the internet. Apple Siri do not reply on internet to perform its task. It can still carry out some task such as make a phone call without internet.

Weaknesses

There are few weakness existed in Apple Siri. It required run on device which install iOS and MacOS only. In other words, other product which does not belong to Apple cannot use Siri.

It does not build to deal with the emotional problem of the user.When it encounter any problem that out of its range,it only can reply the user by searching on the internet.The user can get the result that searching on the internet but not the comfort they want.

2.2.4 Facebook Messenger bot

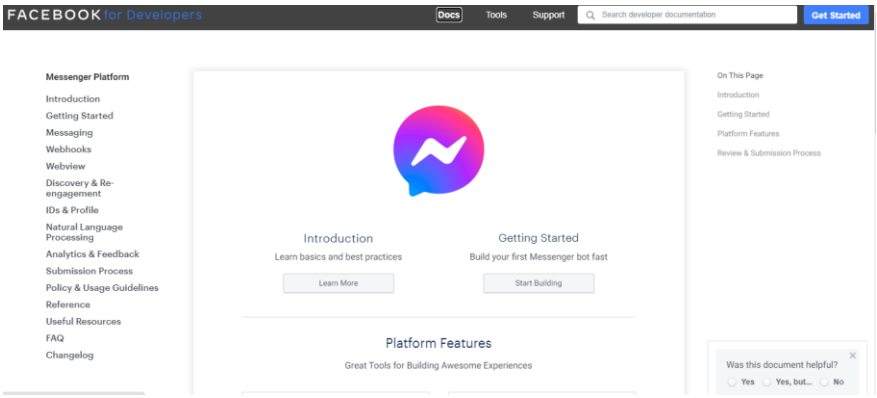


Figure 2.2.4.1 Facebook Messenger bot developer webpage

Introduction

Facebook Messenger is an messaging app that owned by Facebook, Inc. Facebook released a bot platform for Messenger in April 2016.This included an API that enable the user to build the chatbots to interact with others. Facebook further announced few enhancements for bots in 2017 such as the bot is allowed to participate in group chats or direct to the bot by just scanning the QR code.(Quick Response code)

Strengths

The non IT background user can customized the content of the chatbot to serve different purposes such as selling items or answer questions since it is simple to use. Apart from that, it can work with the chat extensions that enable users to interact with dedicated apps such as including play games or purchase items.

Weaknesses

Currently, the Facebook Messenger Bot hasn’t come with Speech Recognition yet. People might need to install extra library themselves if they want to do so. Besides, the Facebook Messenger Bot is meant to serve multi-purpose and it do not specialized and going deep to any field such as giving mental healthcare service.

2.3 Critical Remarks of previous works

The following table shown that the comparison between similar chatbots which is Woebot, Wysa ,Apple Siri and Facebook Messenger bot with the proposed chabot.

2.3.1 Summary of Comparison Similar Application – Chatbot

Features/Name	Woebot	Wysa	Apple Siri
Conversational	Text-based	Text-based	Voice-based
Platform Supported	Cross Platform Apps	Cross Platform Apps	iOS & MacOS only
Cognitive behavioral therapy	Implemented	Implemented	Not Implemented
Memory	Reply based on previous conversation.	Reply based on previous conversation.	Reply not based on previous conversation.
Answer Specific pre-defined Question	Yes	Yes	No
Completely Free	No	No	Yes

Features/Name	Proposed App	Facebook Messenger bot
Conversational	Voice and Text-based	Text-based
Platform Supported	Website	Cross Platform Apps and Website
Cognitive behavioral therapy	Implemented	Not
Memory	Reply not based on previous conversation.	Reply not based on previous conversation.
Answer Specific pre-defined Question	Yes	Yes
Completely Free	Yes	Yes

Table 2.3.1.1 Comparison between similar systems (Chatbot)

Chapter 3: System Design

3.1 Overview of the Chatbot

Based on the project scope that we have mentioned, the module in the proposed app in Figure 3.1.1.

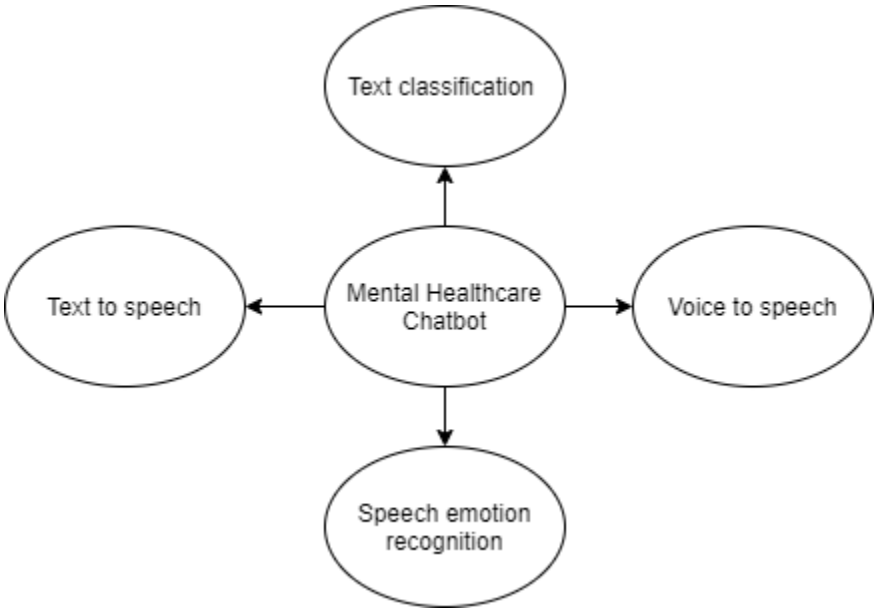


Figure 3.1.1 Modules in the Mental Healthcare Chatbot

3.2 Activity diagram

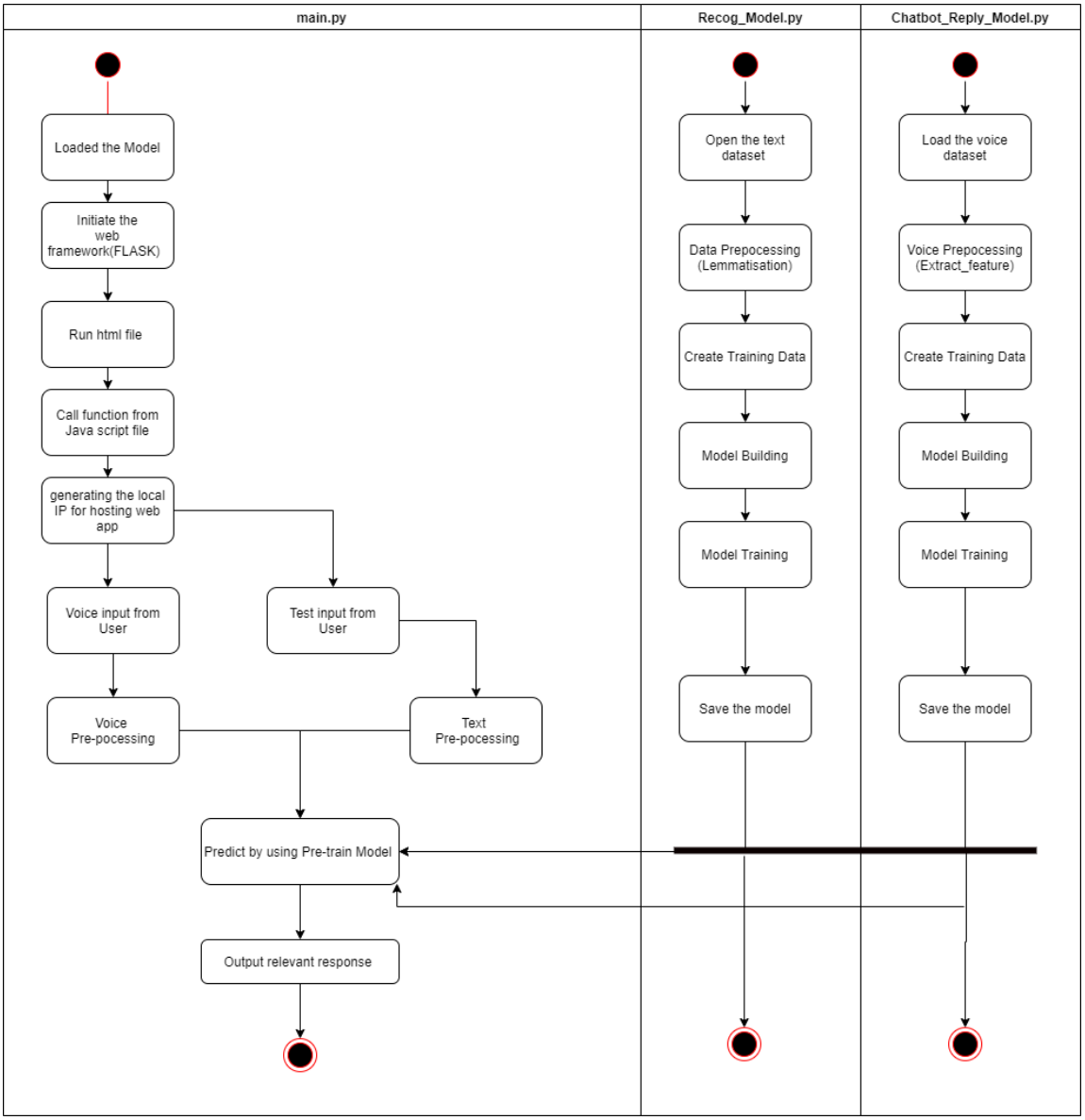


Figure 3.2.1 Activity diagram for Mental Healthcare Chatbot

3.3 Flowchart

The proposed chatbot will receive the input from the user whether in text or voice form. If the input is in voice form, it have to convert voice input into text by using Speech-to-Text (STT) technology before doing any process. After that, the text input will first undergo some Natural Language Processing (NLP) such as stemming and further processed by the pre-train model. After that, the response text will be output based on the predict result. The output text will be converted into voice form by using Text- to-Speech (TTS) technology. The chatbot will reply to the user in both text and voice form. Meanwhile, the chatbot will also record the sound of the user and further process it. If the result is belong to negative emotions such as sad and angry, the chatbot will also output the relevant response.

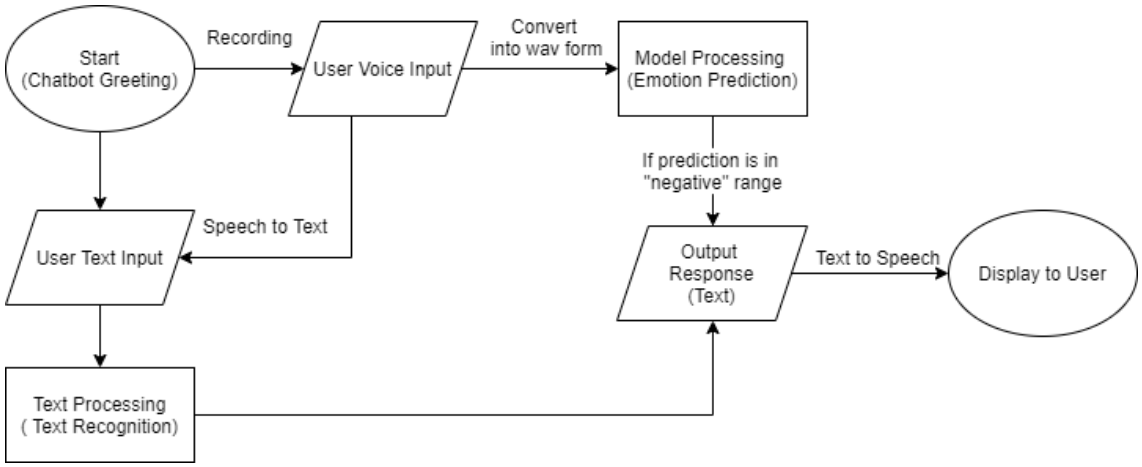


Figure 3.3.1: Flowchart of the proposed voice based mental healthcare chatbot

3.4 Timeline

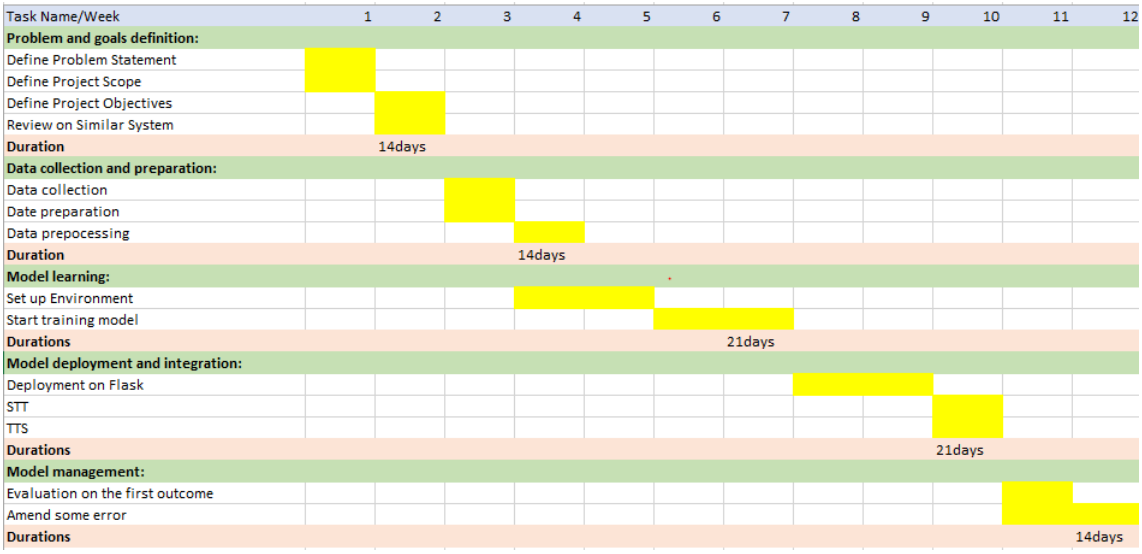


Figure 3.4.1: Timeline for FYP1

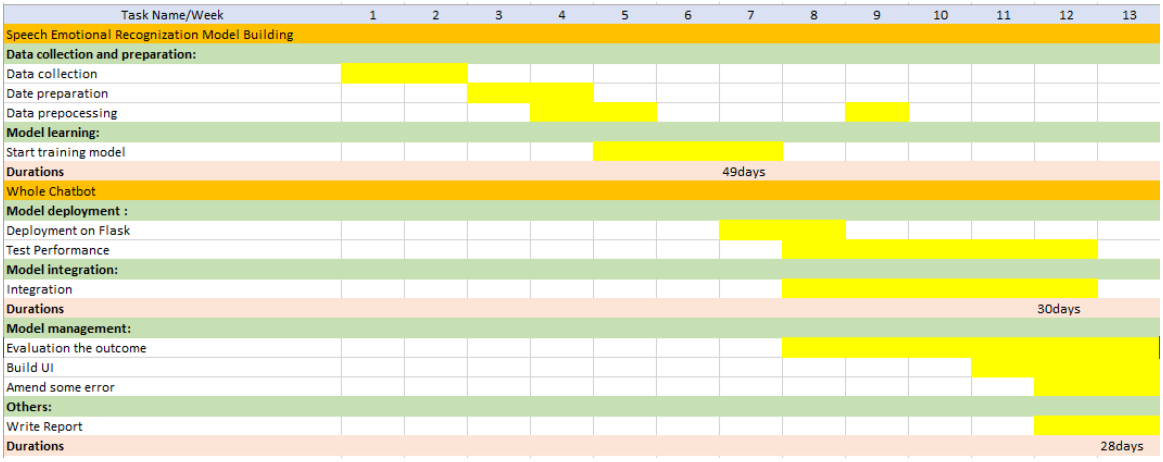


Figure 3.4.2: Timeline for FYP2

Chapter 4: System Implementation

4.1 Overview

The proposed chatbot is developed by following the Machine Learning Model Development Process. In this chapter, the methodology and tools that are applied to the modules of the chatbot will be discussed here. Modules kindly refer to Figure 3.1.1.

4.2 Methodologies

Machine Learning Model Development Process

The proposed chatbot is developed by following the procedures of the machine learning model.

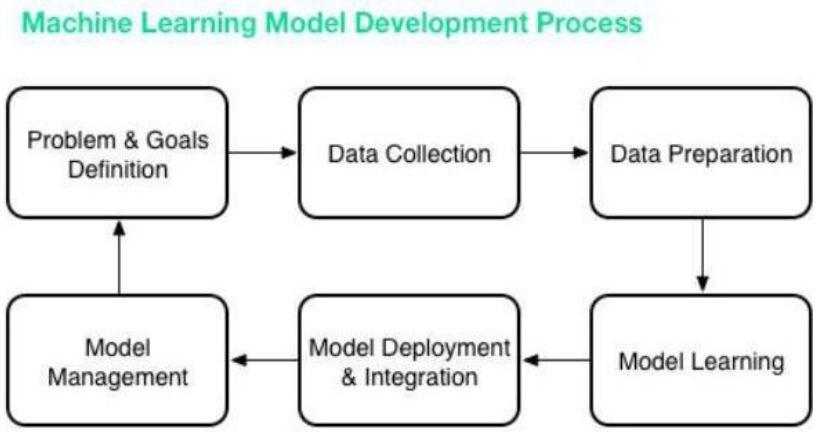


Figure 4.2.1: Flowchart of the machine learning model development process

a. Problem and goals definition:

The proposed chatbot is developed by using Python language with an open source software library, Keras from Tensorflow. There are few procedures when building the ML model. First of all, we need specified the goal of building the model such as either classify the text input or recognize the emotion of the voice input.

b. Data collection and preparation:

Next, we had to prepare the data for training our model. The datasets should be large enough to increase the accuracy of the result. Besides, the data also need to undergo some pre-processing so that the data able to fed into the model training.

c. Model learning:

In this stage, we will start to train our model by using Tensorflow. Tensorflow is an open-source software library developed by Google Brain to allow people to use it for machine learning. All the training process is done on the laptop by using the Nvidia GeForce 940MX GPU. After training, the chatbot was expected to recognize the categories of the input according to the trained model and reply to the user.

d. Model deployment and integration:

After the model is done training, the proposed chatbot would be deployed on the website by using FLASK which is a micro web framework that is written in Python. This chatbot web application allows people to get the counseling service by just entering the website address to the browser.

e. Model management:

After that, we had to do some testing in order to see how the chatbot would reply to us after training. The proposed chatbot will collect the voice input from the user and convert it to text by using Speech-to-Text (STT) technology. The chatbot will process the input and produce the output by using the trained model. The output text will be converted into voice output to user by using Text- to-Speech technology. The chatbot will reply to the user in text form and voice form together. We also need to check whether the chatbot has achieved the goal that we expected by starting a conversation with the chatbot. If the output is not correct, then we need to adjust some part of the processing module until we get the expected outcome.

4.3 Speech-to-Text Module

i. WebkitSpeechRecognition is a new JavaScript Web Speech API which allows fine control and flexibility over the speech recognition capabilities in Chrome version 25 and later. There is no need to import any library for using this Web Speech API.

ii. In this chatbot, we only allow it to accept the input in English.

iii. After recognizing the voice, it will convert it to text and pass it to the processing output module.

```
function record() {
    var recognition = new webkitSpeechRecognition();
    recognition.lang = "en-GB";
    recognition.onresult = function(event) {
        // console.log(event);
        var rawText2 = event.results[0][0].transcript;

        var userHtml2 = '<p class="userText"><span>' + rawText2 + '</span></p>';

         $("#textInput").val("");
         $("#chatbox").append(userHtml2);

        document.getElementById('userInput').scrollIntoView({block: 'start', behavior: 'smooth'});

        $.get("/get", { msg: rawText2 }).done(function(data) {
        var botHtml = '<p class="botText"><span>' + data + '</span></p>';
         $("#chatbox").append(botHtml);
       document.getElementById('userInput').scrollIntoView({block: 'start', behavior: 'smooth'});
  });
```

Figure 4.3.1: WebkitSpeechRecognition in JavaScript

4.4 Deep learning Model for Text Input

Since the proposed chatbot is a chatbot that is involved with mental healthcare services. It is better to let the content stay on topic. Hence, we are building a retrieval-based chatbot that uses some type of heuristic approach to select the appropriate response. So that we can control the flow of the chatbot to achieve the purpose of giving mental health care service.

i. Natural Language Processing (NLP)

- Import NLP is necessary to allow the program to understand the input text.
- We import Nltk which is a toolkit for NLP libraries that is used to understand human language.
- Punkt also downloaded in order to tokenize the sentences into individual words.
- Lancaster stemmer is a tool to stammer words based on the Lancaster stemming algorithm.
- Stemming is a process that is used to reduce the words in order to make it into root form.

```
import nltk
nltk.download('punkt')
from nltk.stem.lancaster import LancasterStemmer
stemmer = LancasterStemmer()
```

Figure 4.4.1: Libraries of the needed tool kit of NLP

ii. Data preparation for training model with NLP

- The training data is saved in a Json file format that is easily readable by machine learning models.
- This document is used to train the model for learning how to classify the input and output the response.
- The intents document included pattern, tags, response.

```
{"intents": [
    {"tag": "greeting",
     "patterns": ["Hi", "How are you", "Is anyone there?", "Hello", "Good day"],
     "responses": ["Hello, what do you feel today", "Good to see you again", "Hi there, how can I help?"],
     "context_set": ""
    },
    {"tag": "goodbye",
     "patterns": ["Bye", "See you later", "Goodbye"],
     "responses": ["See you later", "Have a nice day", "Bye! Come back again soon."]
    },
    {"tag": "thanks",
     "patterns": ["Thanks", "Thank you", "That's helpful"],
     "responses": ["Happy to help!", "Any time!", "My pleasure"]
    },
```

Figure 4.4.2: JSON file that used for training the model

- Doing some "cleaning" to the training data is important. Some tools of the NLP are used in this step.
- First, we have to create a few empty lists for storing the individual words that used to do processing later.
- For example, a list named ignore is created to ignore all the punctuation marks and exclude the unwanted character in order to collect the actual content.
- The loop is used for doing the tokenization and ignores the unwanted words for the pattern words.
- Tokenization is to split the input data into individual units that have a value associated with it.
- After Tokenization, we save the pattern words into a list named words and the tags are saved into a list named classes.
- Then, we continue to do the stemming for the words list and sorted both lists.
- Stemming is a process that is used to reduce the words in order to make it into root form.

```
words = []
classes = []
documents = []
ignore = ['?']
# loop through each sentence in the intent's patterns
for intent in intents['intents']:
    for pattern in intent['patterns']:
        # tokenize each and every word in the sentence
        w = nltk.word_tokenize(pattern)
        # add word to the words list
        words.extend(w)
        # add word(s) to documents
        documents.append((w, intent['tag']))
        # add tags to our classes list
        if intent['tag'] not in classes:
            classes.append(intent['tag'])
words = [stemmer.stem(w.lower()) for w in words if w not in ignore]
words = sorted(list(set(words)))
# remove duplicate classes
classes = sorted(list(set(classes)))
```

Figure 4.4.3: Preprocessing data in the JSON file by using NLP

iii. Build the neural network model by using deep learning

- An artificial neural network is a series of algorithms that mimics the way the human brain operates in order to recognize the underlying relationships in a set of data through a process.
- To build the neural network from scratch, we need to reset the underlying graph.
- We will build a deep neural network that has 3 layers by using the Keras which is a deep learning library.
- First layer will have 128 neurons, and then the second layer will have 64 neurons and the 3rd output layer contains a number of neurons that equal the number of intents in order to predict the output intent with the softmax.
- Then, we start building the neutral network by setting the value of the network layers respectively.
- We are using Stochastic gradient descent with Nesterov as our optimizer as it able to accelerated gradient gives good results for this model.

- Finally, we can start the training of the model by fitting the model with the training data and save it after training.
- The model will further save it as a JSON file so that we can just load it later without repeating the training process.

```
tf.compat.v1.reset_default_graph()

from keras.models import Sequential
from keras.layers import Dense
# Create model - 3 layers. First layer 128 neurons, second layer 64 neurons and 3rd outp
# equal to number of intents to predict output intent with softmax
model = Sequential()
model.add(Dense(128, input_shape=(len(train_x[0]),), activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(64, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(len(train_y[0]), activation='softmax'))
# Compile model. Stochastic gradient descent with Nesterov accelerated gradient gives go
sgd = SGD(lr=0.01, decay=1e-6, momentum=0.9, nesterov=True)
model.compile(loss='categorical_crossentropy', optimizer=sgd, metrics=['accuracy'])
model.summary()

import datetime
log_dir = "logs/fit/" + datetime.datetime.now().strftime("%Y%m%d-%H%M%S")

tb_callback = tf.keras.callbacks.TensorBoard(log_dir=log_dir, update_freq=1)
```

Figure 4.4.4: Training of the neural network model

iv. Classification and output the response after prediction by the model.

- Again, we need to do tokenization and stemming for the input that we get from the user by using NLP.

```
def clean_up_sentence(sentence):
    # tokenizing the pattern
    sentence_words = nltk.word_tokenize(sentence)
    # stemming each word
    sentence_words = [stemmer.stem(word.lower()) for word in sentence_words]
    return sentence_words
```

Figure 4.4.5: Preprocessing the input that get from the user

- After that, we can start classifying the input that get from the user.
- The input sentence is used as an argument to perform classification by using the trained model.
- We had set the error threshold as 0.30 to guarantee the correctness of the output by filtering out all the error predictions which were below threshold.

- We also need to do the sorting for all the percentage of the predictions in order to get the highest one and append it to the list.

```
ERROR_THRESHOLD = 0.30
def classify(sentence):
    # generate probabilities from the model
    results = model.predict([bow(sentence, words)])[0]
    # filter out predictions below a threshold
    results = [[i,r] for i,r in enumerate(results) if r>ERROR_THRESHOLD]
    # sort by strength of probability
    results.sort(key=lambda x: x[1], reverse=True)
    return_list = []
    for r in results:
        return_list.append((classes[r[0]], r[1]))
    # return tuple of intent and probability
    return return_list
```

Figure 4.4.6: Classification of the input that get from the user

- By looping through all the classes in the intents file the corresponding response from the intents JSON file will be returned if the matching is found.
- The response will pop out as stack and pass it to frontend part when the function is called.

```
def response(sentence):
    results = classify(sentence)
    # if we have a classification then find the matching intent tag
    if results:
        # loop as long as there are matches to process
        while results:
            for i in intents['intents']:
                # find a tag matching the first result
                if i['tag'] == results[0][0]:
                    # a random response from the intent
                    return random.choice(i['responses'])

            results.pop(0)
```

Figure 4.4.7: Production of the output sentences

4.5 Deep learning Model for Voice Input

This model enable chatbot to recognize whether the person is under negative emotions by detecting the feature of the input voice.

i. Data Collection

- In order to boost the accuracy of a Model, we need to a large number of datasets to train the model.
- The dataset have to be in the .wav form so that it can be processed later.
- It contains the human voice that conveys different kinds of emotions such as happiness and sad.
- Both files totally contain 4,240 sound files voiced by different actresses.
- These two datasets that will be used at this stage are:

1. Toronto emotional speech set (TESS) (Kate, M. Kathleen, 2010.)

2. The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) (Livingstone SR, Russo FA,2018)

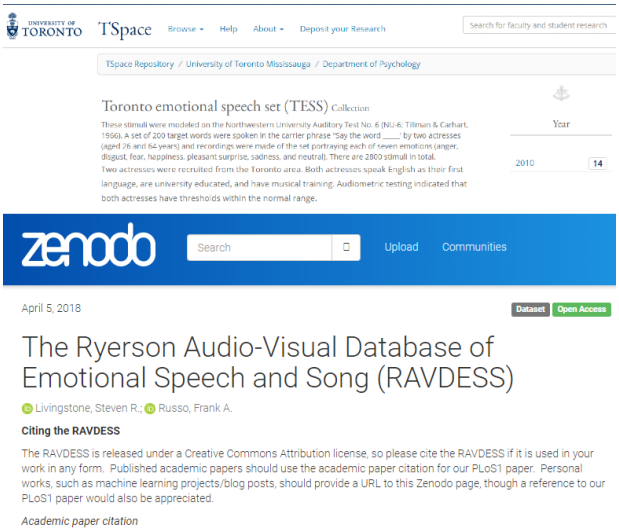


Figure 4.5.1: Training datasets of the voice emotional recognition model

ii. Data Preparation

- The emotion of the data will append to a list according to the file naming formats for training and testing purpose.

```
#DataFlair - Load the data and extract features for each sound file
def load_data(test_size=0.20):
    x, y = [], []
    for file in glob.glob("C:\\Users\\Oon\\Documents\\CS\\Y4S2\\FYP2\\DataFlair\\ravdess data\\Actor_*\\*.wav"):
        file_name=os.path.basename(file)
        emotion=emotions[file_name.split("-")[2]]
        if emotion in unobserved_emotions:
            continue
        feature=extract_feature(file, mfcc=True, mel=True)
        x.append(feature)
        y.append(emotion)
        print(file_name)
    return train_test_split(np.array(x), y, test_size=test_size, random_state=9)
```

Figure 4.5.2: Function that load the training data

- Then, we will read the data using Sound File library as "float32" format.
- We had added some noise with our noise function in order to further improve the accuracy of the results.
- After that, we will extracted the features of the voice data by using Mel-frequency spectrum, (MFCC) with 40 vectors.
- We are taking the mean of these vectors and storing in an array. The array will return when it called by other functions.

```python
def noise(data):
    noise_amp = 0.05*np.random.uniform()*np.amax(data)   # more noise reduce the value to 0.5
    data = data.astype('float64') + noise_amp * np.random.normal(size=data.shape[0])
    return data

#Extract features (mfcc, chroma, mel) from a sound file
def extract_feature(file_name, mfcc, mel):
    sound = AudioSegment.from_wav(file_name)
    sound = sound.set_channels(1)
    sound.export(file_name, format="wav")
    print(file_name)
    with soundfile.SoundFile(file_name) as sound_file:
        X = sound_file.read(dtype="float32")
        X = noise(X)
        sample_rate=sound_file.samplerate
        result=np.array([])
        if mfcc:
            mfccs=np.mean(librosa.feature.mfcc(y=X, sr=sample_rate, n_mfcc=40).T, axis=0)
            result=np.hstack((result, mfccs))
        if mel:
            mel=np.mean(librosa.feature.melspectrogram(X, sr=sample_rate).T,axis=0)
            result=np.hstack((result, mel))
        return result
```

Figure 4.5.3: Function that extracted the features of the voice data

- Before training the model, we need to reshaped the data (convert it into np array), so that it can be feed into CNN model training.
- The y_train and y_test training that store the emotion classes like “happy” or “sad” will also be saved by using pickle. It is used to transform the prediction into a readable form later.

```
#Split the dataset
x_train,x_test,y_train,y_test=load_data(test_size=0.20)
hx_test = load_data_test()
print(hx_test.shape)
print(hx_test.ndim)

#Get the shape of the training and testing datasets
print((x_train.shape[0], x_test.shape[0]))
print(f'Features extracted: {x_train.shape[1]}')

#preparation steps to get it into the correct format for Keras
X_train = np.array(x_train)
y_train = np.array(y_train)
X_test = np.array(x_test)
y_test = np.array(y_test)

a=y_train
b=y_test

#to load it
with open("train.pkl", "rb") as f:
    a, b = pkl.load(f)
```

Figure 4.5.4: Transform the shape of the training data

iii. Model Learning

- We will develop a one-dimensional convolutional neural network model (1D CNN) for speech emotion recognition module.
- The reason why we are using 1D CNN model for the chatbot is that the kernel of 1D CNN model can only move in one dimension along the axis of time.
- Hence, it is consider to be used on Time-Series data such as the sound and texts.

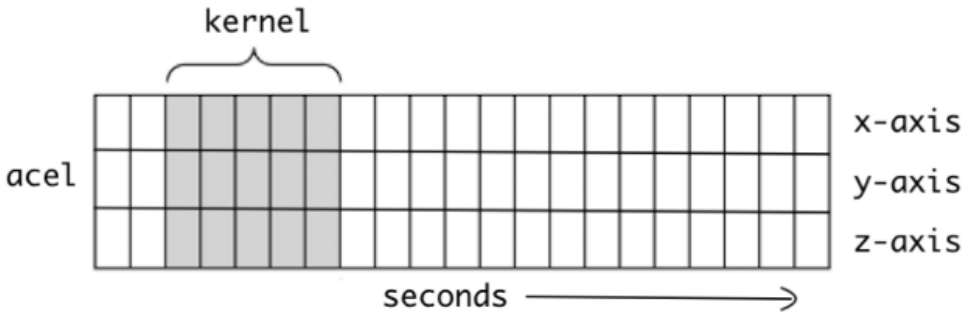


Figure 4.5.5: Kernel sliding over accelerometer data

- After we have done preparing the training data for the model, we can start to define, fit, and evaluate a 1D CNN model.
- The model is building by using the Keras deep learning library.
- We will define the model as having two 1D CNN layers, followed by a dropout layer for regularization, then a pooling layer.
- The purpose of adding the dropout layer is to slow down the learning process in order to have a much better final model.
- Besides, the pooling layer is used to compress feature representations. So that it can help to reduce the overfitting of the training data by the model.
- Other layers such as batch normalization also been added in order to help to get a much better result by reducing generalization error.

```
#Dimension preps for 1D CNN
X_train = np.expand_dims(X_train, axis=2)
X_test = np.expand_dims(X_test, axis=2)
print(X_train.shape)
print(X_train.ndim)


model = Sequential()
model.add(Conv1D(256, 8, padding='same',input_shape=(X_train.shape[1],1)))  # X_train.shape[1] = No. of Columns
model.add(Activation('relu'))
model.add(Conv1D(256, 8, padding='same'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(Dropout(0.25))
model.add(MaxPooling1D(pool_size=(6)))
model.add(Conv1D(128, 8, padding='same'))
model.add(Activation('relu'))
model.add(Conv1D(128, 8, padding='same'))
model.add(Activation('relu'))
model.add(Conv1D(128, 8, padding='same'))
model.add(Activation('relu'))
model.add(Conv1D(128, 8, padding='same'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(Dropout(0.25))
model.add(MaxPooling1D(pool_size=(5)))
model.add(Conv1D(64, 8, padding='same'))
```

Figure 4.5.6: 1D CNN model building

- RMSProp will be used as optimizer for this chatbot since it is applicable to mini batch learning.
- Once the model is fit, it is evaluated on the test dataset and the accuracy of the fit model on the test dataset is returned.
- The model will be save in .h5 form and save it to disk as JSON file.

```
model.add(Conv1D(128, 8, padding='same'))
model.add(Activation('relu'))
model.add(Conv1D(128, 8, padding='same'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(Dropout(0.25))
model.add(MaxPooling1D(pool_size=(5)))
model.add(Conv1D(64, 8, padding='same'))
model.add(Activation('relu'))
model.add(Conv1D(64, 8, padding='same'))
model.add(Activation('relu'))
model.add(Flatten())
model.add(Dense(6))# Target class number
model.add(Activation('softmax'))
opt = RMSprop(lr=0.00001, decay=1e-6)
model.summary()

model.compile(loss='categorical_crossentropy', optimizer=opt,metrics=['accuracy'])
model_history=model.fit(X_train, y_train, batch_size=128, epochs=100, validation_data=(X_test, y_test))
score = model.evaluate(X_test, y_test, verbose=0)
print("%s: %.2f%%" % (model.metrics_names[1], score[1]*100))
model.save('model6.tflearn.h5')

# Save the model to disk
model_json = model.to_json()
with open("model_json.json", "w") as json_file:
    json_file.write(model_json)
```

Figure 4.5.7: Save Model After Building

iv. Output the response after predicted the input voice by the model.

- After receive the voice input (done recording) , it will post back to the flask as blob file in webm form.
- Then, we need to convert it into the wav form by using FFmpeg library to prevent loss of the voice details such as bit rate.

```python
def convert_to_wav(b):
    c = b + "\\*.webm"
    d = b + "\\sample.wav"
    for file in glob.glob(c):
        command = f"ffmpeg -i {file} -vn -ar 44100 -ac 2 -b:a 192k {d}"
        os.system(command)
        os.remove(file)


def load_data_test(b):
    x = []
    convert_to_wav(b)
    c = b + "\\*.wav"
    d = b + "\\sample.wav"
    print(c)
    for file in glob.glob(c):
        print(file)
        feature = extract_feature(file, mfcc=True, mel=True)
        x.append(feature)
    os.remove(d)
    return np.array(x)
```

Figure 4.5.8: Load and process the voice input from user

- After done converting into wav form, it will go through the same previous data preparation process in order to get the features of the input audio and also reshape it into the form that can predicted by the loaded model.
- After let the model to predict the input data, we also need to let the outcome convert into the readable form like “sad” or “happy”.
- We are open the previous training material that saved by using pickle so that we can convert array of labeled data to one-hot vector.
- After that, we can transform the predicted result into the understandable words like “sad” or “happy”.

```
with open("data/train.pkl", "rb") as f:
    a, b = pkl.load(f)

# # one hot encode the target
lb = LabelEncoder()
y_train = np_utils.to_categorical(lb.fit_transform(a))
y_test = np_utils.to_categorical(lb.fit_transform(b))

from keras.models import model_from_json

json_file = open('data/model_json.json', 'r')
loaded_model2_json = json_file.read()
json_file.close()
loaded2_model = model_from_json(loaded_model2_json)
# load weights into new model
loaded2_model.load_weights("data/model6.tflearn.h5")
print("Loaded model from disk")


def predict(hx_test):
    hx_test1 = np.expand_dims(hx_test, axis=2)
    hy_preds = loaded2_model.predict(hx_test1)
    hy_preds = hy_preds.argmax(axis=1)
    hy_preds = hy_preds.astype(int).flatten()
    hy_preds = (lb.inverse_transform((hy_preds)))
    return hy_preds
```

Figure 4.5.9: Predict the voice input from user

- If the result is in to the negative emotion list, it will output the corresponding sentence to the user.

```
Negative = ['sad', 'angry', 'fearful', 'disgust', 'surprised']

@app.route("/record_sound", methods=["POST", "GET"])
def record_sound():
        a = request.data
        with open('audio/audio.webm', 'wb') as f:
            f.write(a)
        b = os.path.dirname(os.path.abspath('audio/audio.webm'))
        hx_test = load_data_test(b)
        hy_test = predict(hx_test)
        if hy_test in Negative:
            message = {'words': 'I know you are having such a hard time.Try to take a slow and deep breathing first!
        else:
            message = {'words': ':)'}
        return jsonify(words)
```

Figure 4.5.10: Output the response to user based on model prediction

4.6 Text to Speech (TTS)

i. TTS is a natural language modelling process that converts text into spoken voice output.

ii. The Voice output is speak by using the computer- generated spoken voice.

iii. The Google Cloud Text-to Speech API and Playsound API is used in order to convert the text into voice and play the voice file.

```
#Import Google Text to Speech
from gtts import gTTS
# play the mp3 file
import playsound
```

Figure 4.6.1: Google Cloud Text-to Speech and Playsound libraries

iv. Google Cloud Text-to Speech API is used to convert the output text into an mp3 file and save it.

v. The Playsound API is used to play the mp3 file to the user.

```
def text_to_voice(text2voice):
    tts = gTTS(text=text2voice)
    sound_file = 'Voice.mp3'
    tts.save(sound_file)
    playsound.playsound(sound_file)
    os.remove(sound_file)
```

Figure 4.6.2: A function that is used to convert the text into voice by using Google Cloud Text-to Speech API and Playsound API

4.7 Tools to use.

The following table shows the software tools needed for the development of the proposed Chatbot:

Particulars	Software Tools
Operating System	Microsoft Windows 10
Integrated Development Environment (IDE)	PyCharm 2020
Programming Languages	Python (for model training)
	Javascript (for webpage)
	HTML,CSS (for webpage UI)
Platform	Flask (for model deployment and Integration)
Libraries, API and other tools	Google TTS API
	librosa
	FFmpeg
	Audiosegment 0.23.0
	Google Tensorflow
	Google WebkitSpeechRecognition API

Table 4.7.1 : Software requirement for Chatbot

4.8 Implementation Issues and Challenges

a. The environment setup procedures is complex

There are many libraries need to be install one by one. Not to say that some of the libraries is not compatible with the PC or some of the libraries' version is not suitable for carry out the training process. Hence, this force me to convert the tflearn model (FYP1) into keras model (
FYP2) since tflearn is hard to compatible with TensorFlow v2.0 and over. The action to uninstall some of the libraries and reinstall the correct version of the library is necessary.

Solution: Used the library that able to compatible with each other.

b. Difficulty to pass the data between backend and backend

Passing of the data between backend and backend is very troublesome. Since we are not only transfer the text data only, but also transfer the audio data from frontend to backend. It take time to find a way to pass the data and fix some problem such as missing details of the audio like bit rate or audio sample rate.

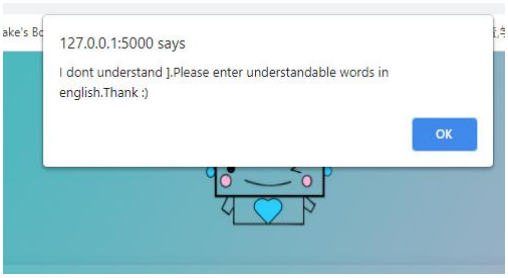
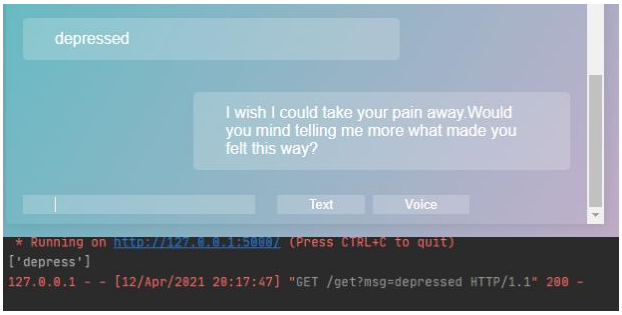
Solution: The audio file will first saved as webm form and post to FLASK as blob file. After fetching the data , then convert it to wav form for further processing of the audio input.

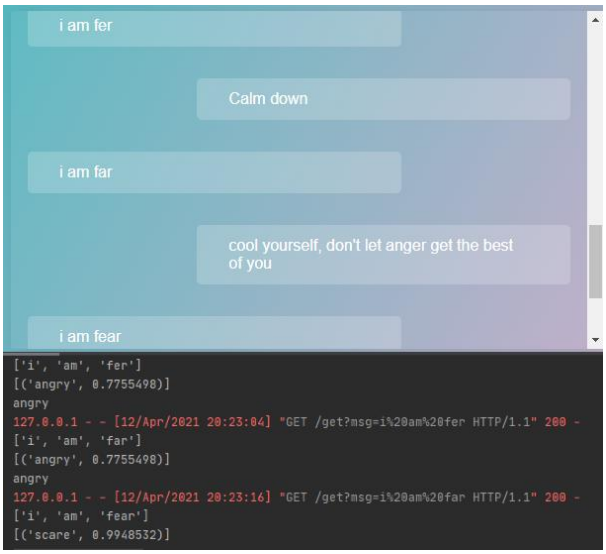
Chapter 5: Verification Plan

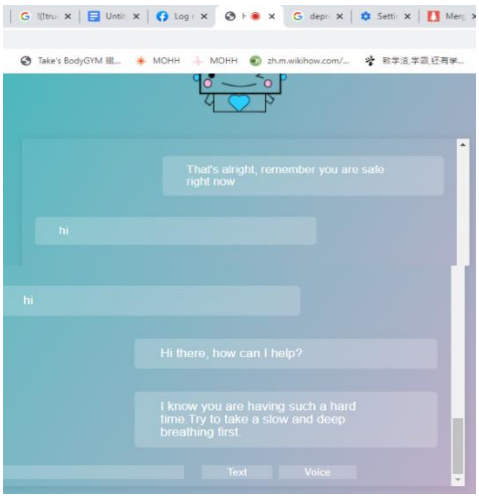
5.1 Usability Testing

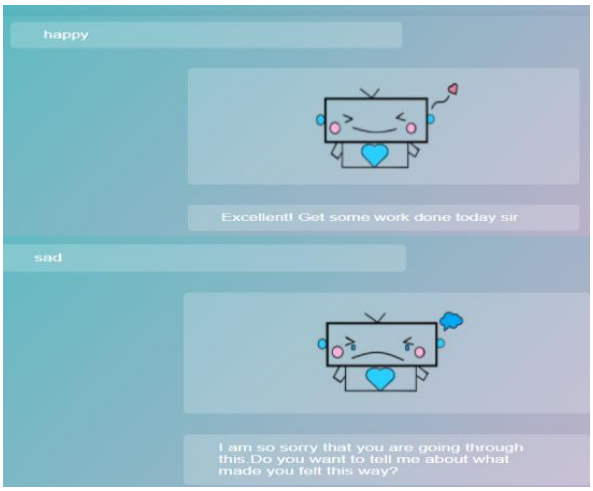
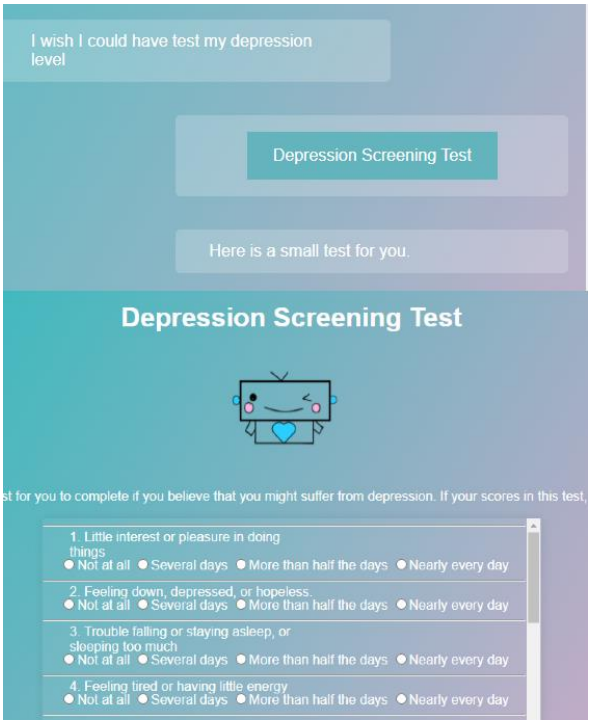
1. Visibility of the Chatbot Page			
Purpose: The proposed chatbot's UI should keep the user easy to read order to ensure quick access to content that the user is looking for.			
No.	Review Checklist	Yes / No / Partially	Comments
1.1	Are all the font design and stylist remain consistent all time?	Yes	All fonts that are on the page are kept consistent.
1.2	Are all the fonts on the page displayed clearly without overwrapping the screen?	Yes	All the fonts is big enough, easily to be seen and read without any difficulty.

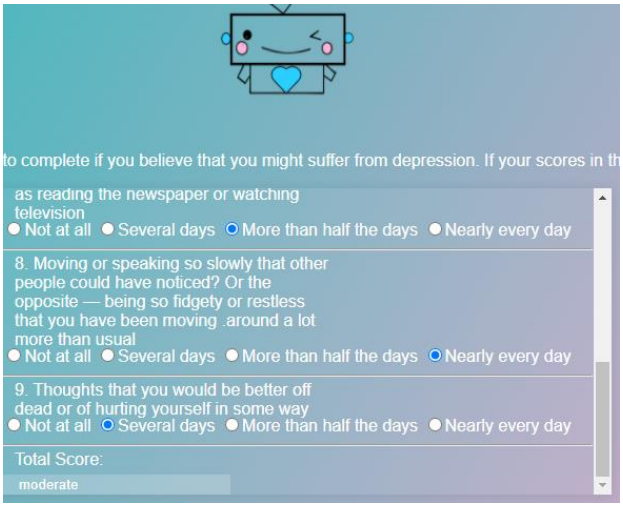
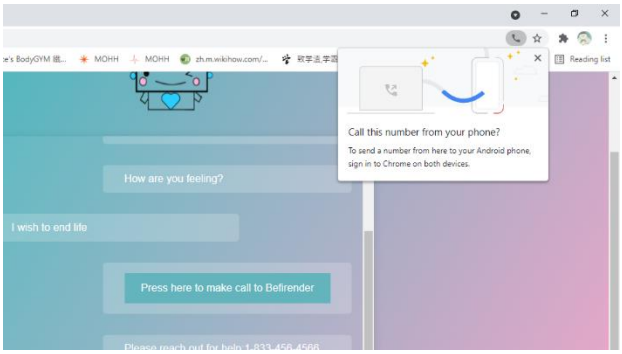
2. Usability of the Chatbot Page			
Purpose: The proposed chatbot should allow users to interact with it and perform relative tasks.			
No.	Review Checklist	Yes / No / Partially	Comments
2.1	Are Chatbot should get a load with the webpage?	Yes	It will first go to the landing page then followed by the main chatbot page. If the user requests to do a quiz, it also will direct to the relevant page.
2.2	Did the chatbot showing greeting if programmed as per the Time zone?	Yes	Yes. The chatbot will greet the user along with the audio.
2.3	Did the chatbot ask polite question e.g.? (“How may I assist you?”)	Yes	Yes. It will start greeting and ask polite question every time.
2.4	Did the chatbot allow the input text need to be	Yes	Yes. It will validate the unwanted input characters like “[,]@#$%^”.

	follow the proper validation?		 Figure 5.1.1: Input validation of chatbot
2.5	Did the chatbot sharp enough to understand the words with the mistakes like (past tense or wrong spelling)	Partially	The chatbot will do the stemming to the input text if it is in past tense form.  Figure 5.1.2: Chatbot is stemming word However, the chatbot needs to be further improved since it will get confused with the similar spelling words. A larger dataset is needed for training the model to increase its accuracy.

			 Figure 5.1.3: Result of the chatbot accuracy
2.6	Chatbot should not crash if some text pastes by the user	Yes	Yes. It will still be working fine.
2.7	Did the chatbot able to capture the audio input?	Partially	The chatbot currently support the recording function for Chrome browser only.
2.8	Did the chatbot able to convert speech to text?	Partially	The chatbot sometimes will hardly convert the speech into text if the English pronunciation of the user is not accurate enough.
2.9	Did the chatbot able to recognize and respond to the audio input?	Yes	It able to give corresponding response if it capture the audio input successfully.

			 Figure 5.1.4: Chatbot capture audio
2.10	Did the chatbot able to convert text to speech?	Yes.	It will show the response sentences along with the speech.
2.11	Did the buttons able to perform their functions when they clicked?	Yes.	The buttons are working well.
2.12	Did the chatbot able to give emojis when reply?	Yes	The chatbot will give different emojis depend on the response sentences.

			 Figure 5.1.5: Chatbot Emoji
2.13	Did the chatbot able to link to a depression screening test page when the user request?	Yes	It will direct user to the related page when it detect the user's request.  Figure 5.1.6: Chatbot test

2.14	Did the depression screening test able to give user the result?	Yes.	 Figure 5.1.7: Chatbot test result
2.15	Did the chatbot able to make phone call?	Partially	The chatbot able to trigger a phone call when it detects something like suicidal though. However, the device of the user needs to install related apps to make a successful phone call.  Figure 5.1.8: Chatbot make a call

5.2 Testing on the conversation with the proposed chatbot

The figures below show a conversation between the user and the chatbot.

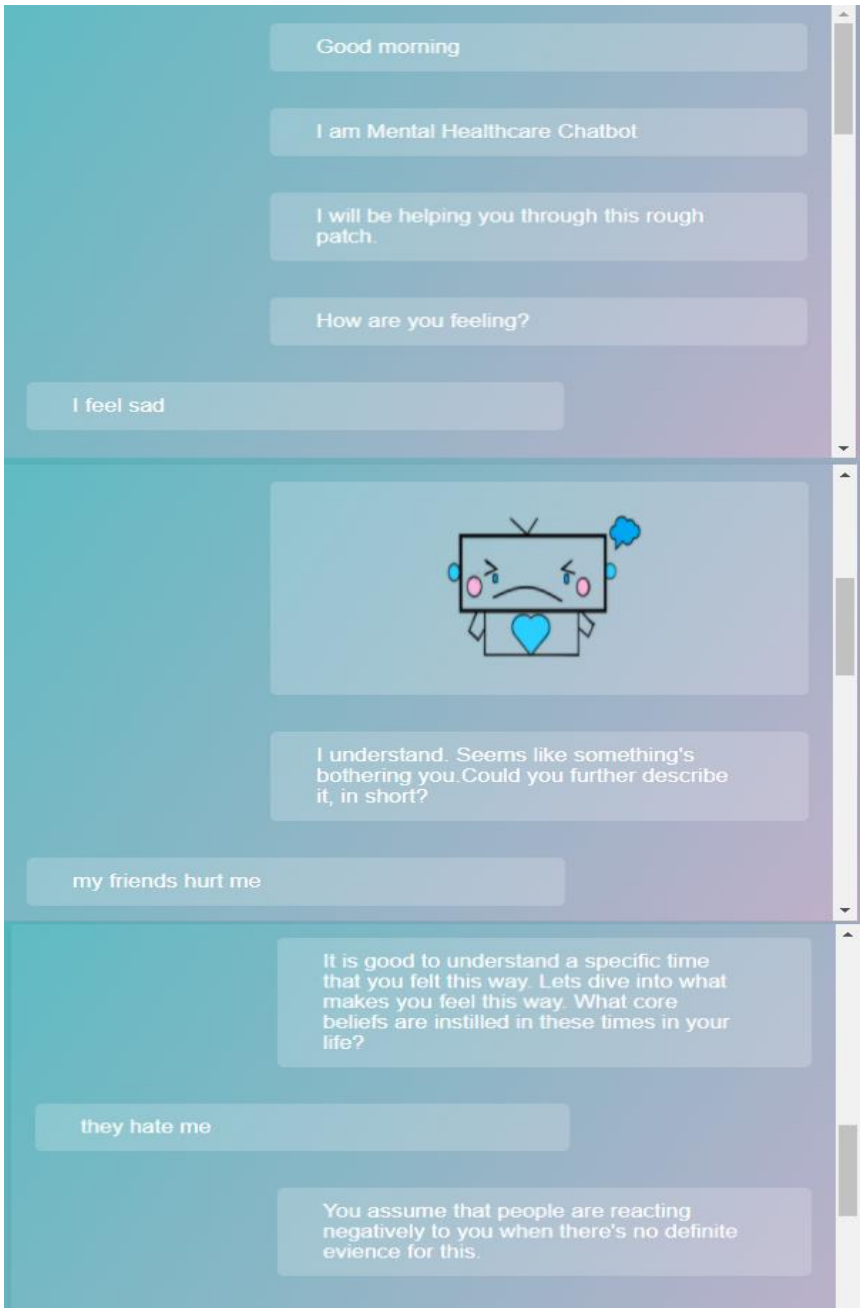


Figure 5.2.1: Chatbot conversation sample 1

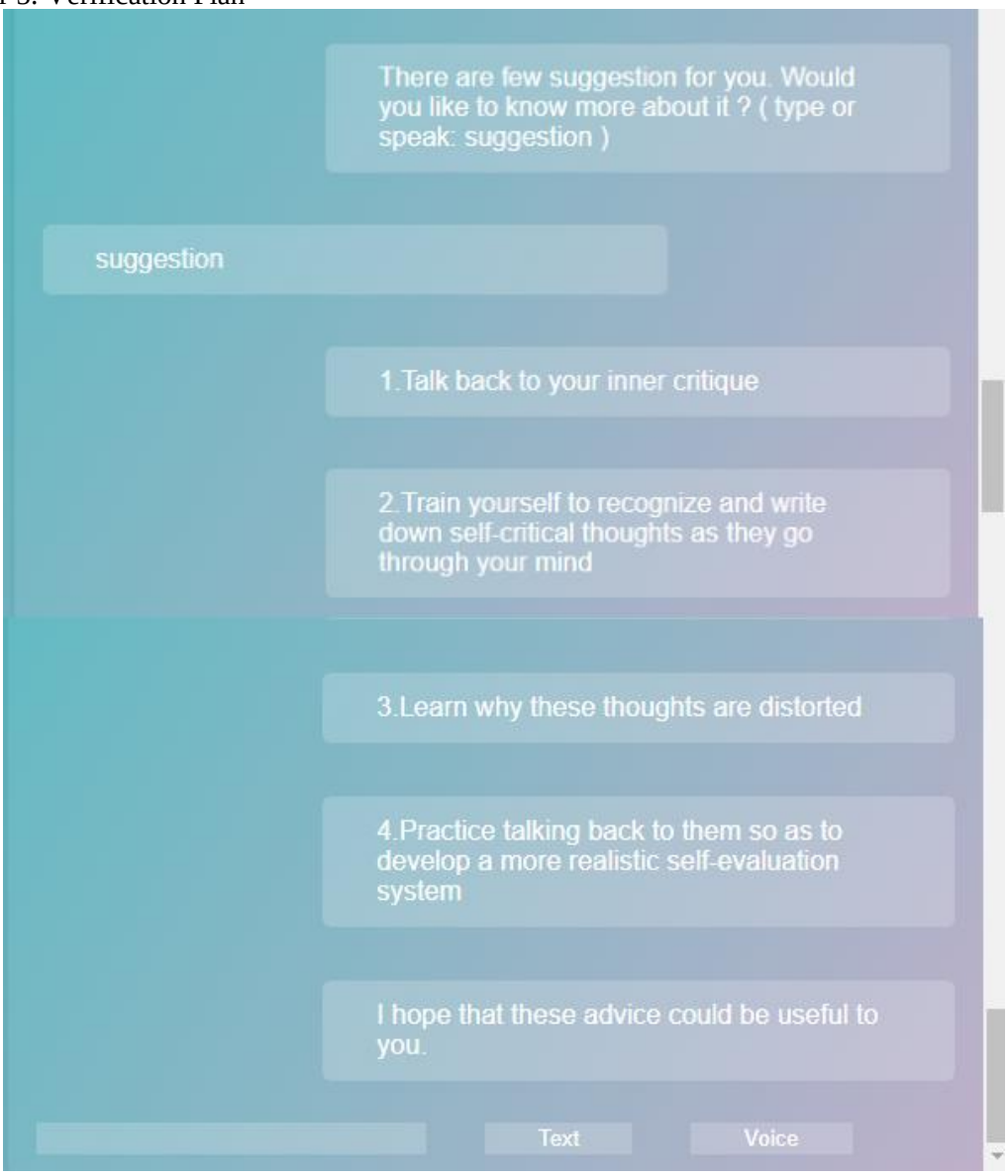


Figure 5.2.2: Chatbot conversation sample 2

Chapter 6: Conclusion

6.1 Overview of the chatbot

Figures below show chatbot UI.

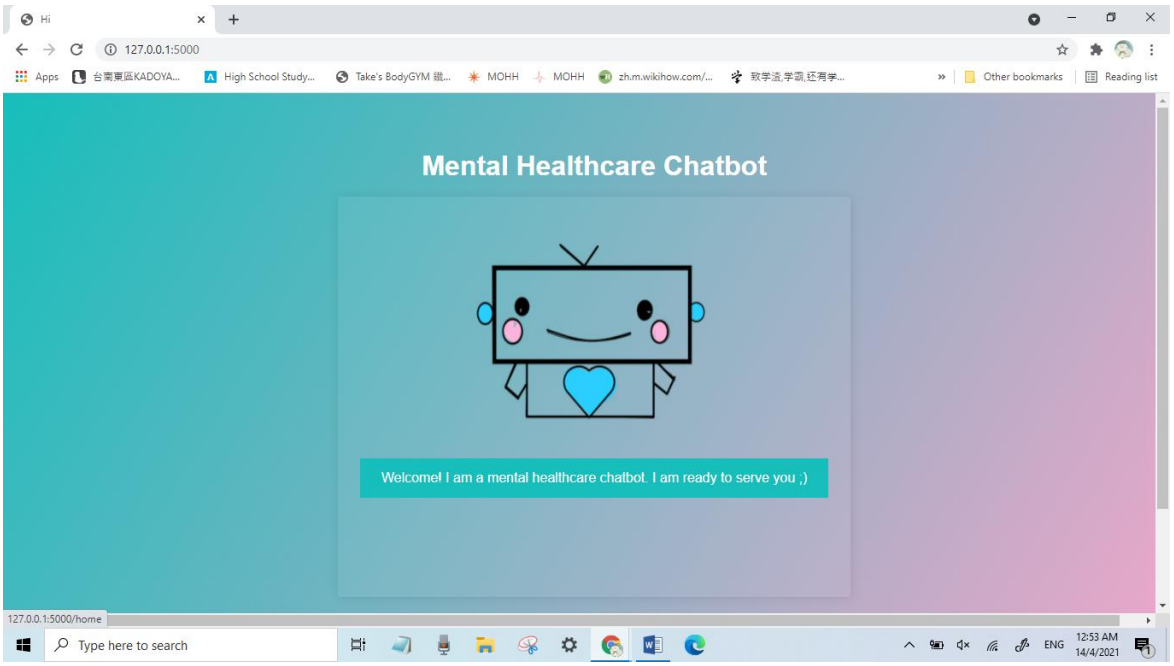


Figure 6.1.1: Landing page of the chatbot

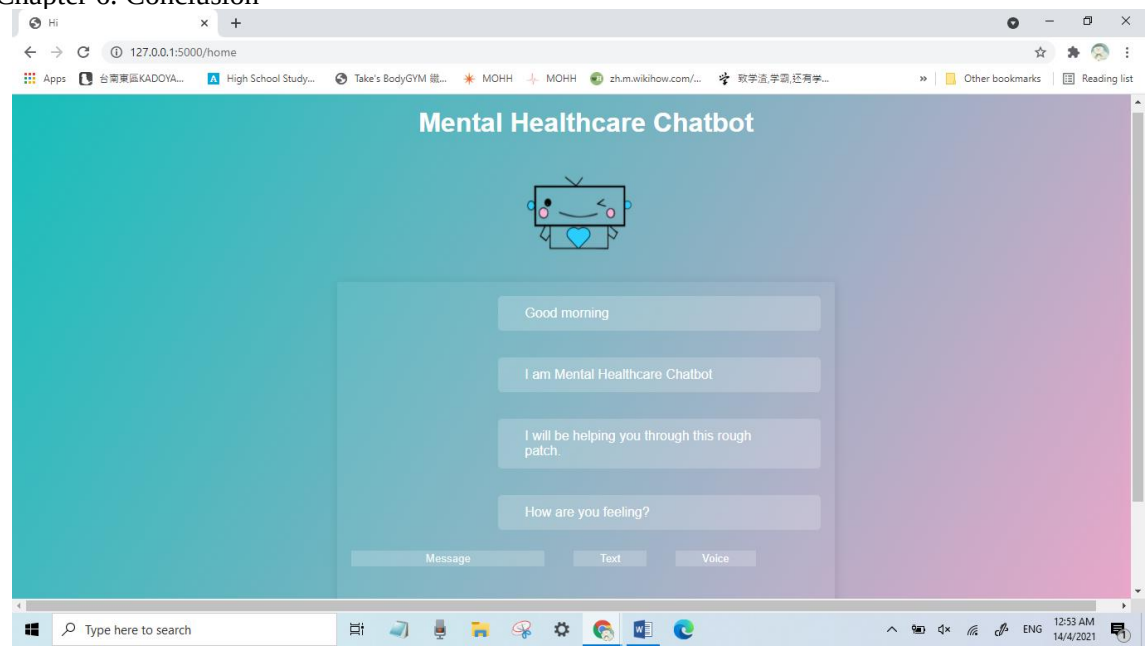


Figure 6.1.2: Home page of the chatbot

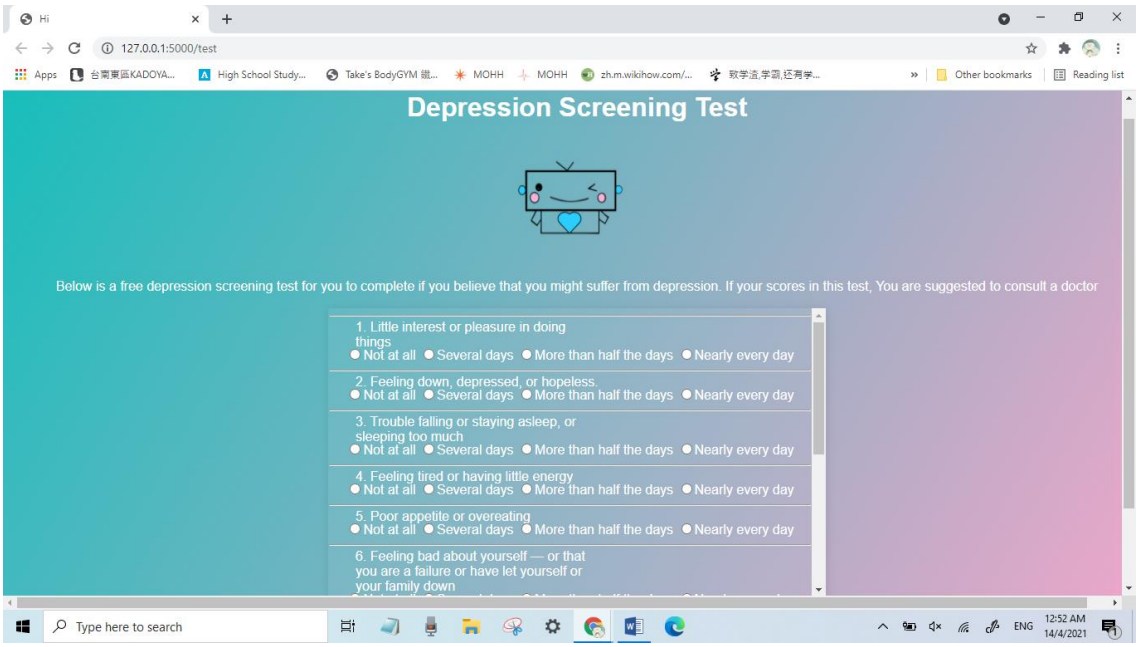


Figure 6.1.3: Test page of the chatbot

6.2 Conclusion

The main purpose of this proposed chatbot is to help people by providing them not just text-based but also voice-based counselor service. Not every people can easily access mental healthcare services. By using this chatbot, people can get accompany for a 24/7 whole day and also not spending any cost.

In this project, although the proposed chatbot is developed it is regrettable to say that this chatbot still considers apart from giving the diverse response every time. The accuracy of the model prediction can be further improved by training with larger datasets.

Besides, the flow of the conversation for the chatbot is considered hard to design. Apart from the reason that lack of psychology knowledge and experience, it is not guaranteed that the user will follow the instruction given. Although the chatbot still considers able to give a related response even it is out of the range of the conversation flow, but this makes the chatbot unable to perform all the functions completely and reduces the quality of the mental healthcare service.

However, there is some efforts had been done to improve user experience such as adding some emojis in the conversations. Furthermore, to simulate the human typing behavior, the typing indicator was also had been added. Since the color can indicate the basic mood, tone, concept, and connotation for a product, cold colors such as blue or green had been applied to the chatbot UI to let the user feel a sense of quietness and trust.

In short, the proposed chatbot now cannot be considered as a perfect chatbot that able to give 100% quality of the mental healthcare service. It still has a lot of room to be improved in the future. However, by reviewing back to our objectives, it indeed had achieved most of them.

6.3 Future Work

The proposed chatbot can be further improved by expanding its training data. Keep collecting the feedback of the user or even cooperate with human counselors to improve the content of the chatbot also recommend to be carried out. So that this chatbot can give more professional counseling services to the people.

Besides, this chatbot also can become more advance if it is added to face emotional recognition. Face recognition should work like voice emotional recognition as it will perform some action according to the facial expression of the user. By working together with both recognition modules, the chatbot should be able to predict the situation of the user more accurately and serve the user much better.

References

References

Chaib, F,(2017), "Depression: let's talk" says WHO, as depression tops list of causes of ill health [Online]. Available at: <http://www.who.int/news-room/detail/30-03-2017--depression-let-s-talk-says-who-as-depression-tops-list-of-causes-of-illhealth>. [Accessed 1 April 2021].

Mak, B., & Coniam, D. (2008). Using Wikis to enhance and develop writing skills among secondary school students in Hong Kong. System, An International Journal of Educational Technology and Applied Linguistics, 36, 437-455.

de Zwart, P. L., Jeronimus, B. F., & de Jonge, P. (2019). Empirical evidence for definitions of episode, remission, recovery, relapse and recurrence in depression: a systematic review. Epidemiology and psychiatric sciences, 28(5), 544–562. Available at: https://doi.org/10.1017/S2045796018000227 [Accessed 15 April 2021].

Google Cloud. (2019). *Cloud Speech-to-Text - Speech Recognition | Cloud Speech-to- Text | Google Cloud*. [online] Available at: https://cloud.google.com/speech-to-text/ [Accessed 16 Aug. 2019].

Google Cloud. (2019). *Cloud Text-to-Speech - Speech Synthesis | Cloud Text-to-Speech API | Google Cloud*. [online] Available at: https://cloud.google.com/text-to-speech/ [Accessed 16 Aug. 2019].

References

Hill, J., Randolph Ford, W. and Farreras, I. (2015). Real conversations with artificial intelligence: A comparison between human–human online conversations and human–chatbot conversations. *Computers in Human Behavior*, 49, pp.245-250.

Hoermann, S., McCabe, K. L., Milne, D. N., & Calvo, R. A. (2017). Application of Synchronous Text-Based Dialogue Systems in Mental Health Interventions: Systematic Review. Journal of medical Internet research, 19(8), e267. Available at: < https://doi.org/10.2196/jmir.7023>[Accessed 15 April 2021].

Pichora-Fuller, M., & Dupuis, K. (2010). Toronto emotional speech Set (tess). Available at: <https://dataverse.scholarsportal.info/dataset.xhtml?persistentId=doi%3A10.5683%2FSP2%2FE8H2MF>[Accessed 15 April 2021].

Livingstone, S. R., & Russo, F. A. (2018). The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English. PLoS ONE, 13(5), Article e0196391.

Saxena, S., Thornicroft, G., Knapp, M., & Whiteford, H. (2007). Resources for mental health: scarcity, inequity, and inefficiency. Lancet (London, England), 370(9590), 878–889. Available at: > https://doi.org/10.1016/S0140-6736(07)61239-2>[Accessed 15 April 2021].

Shawar, B. A., & Atwell, E. (2005). Using corpora in machine-learning chatbotsystems. International Journal of Corpus Linguistics, 10(4), 489–516.

WHO. (2018). Mental health: strengthening our response. [online] Available at: <https://www.who.int/en/news-room/fact-sheets/detail/mental-health-strengthening-our-response> [Accessed 15 April 2021].

References

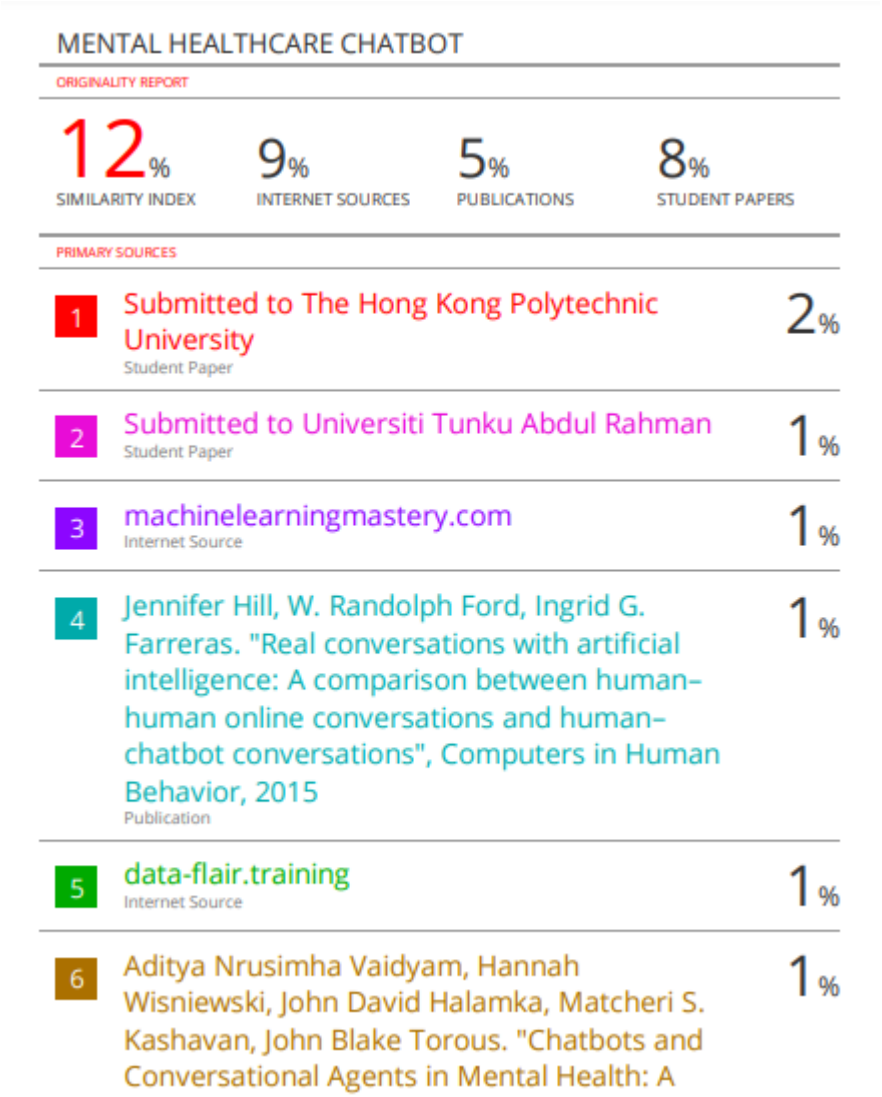
Woudenberg, A.V. (2014). A Chatbot Dialogue Manager - Chatbots and Dialogue Systems: A Hybrid Approach. Available at: < https://www.semanticscholar.org/paper/A-Chatbot-Dialogue-Manager-Chatbots-and-Dialogue-A-Woudenberg/33aeb980817c2031cea6bf85649e58906d8c4bae#citing-papers> [Accessed 15 April 2021].

Vaidyam AN, Wisniewski H, Halamka JD, Kashavan MS, Torous JB. (2019) Chatbots and Conversational Agents in Mental Health: A Review of the Psychiatric Landscape. Can J Psychiatry. 2019;64(7):456-464.

Appendix A - Poster

Poster

Plagiarism Check Result



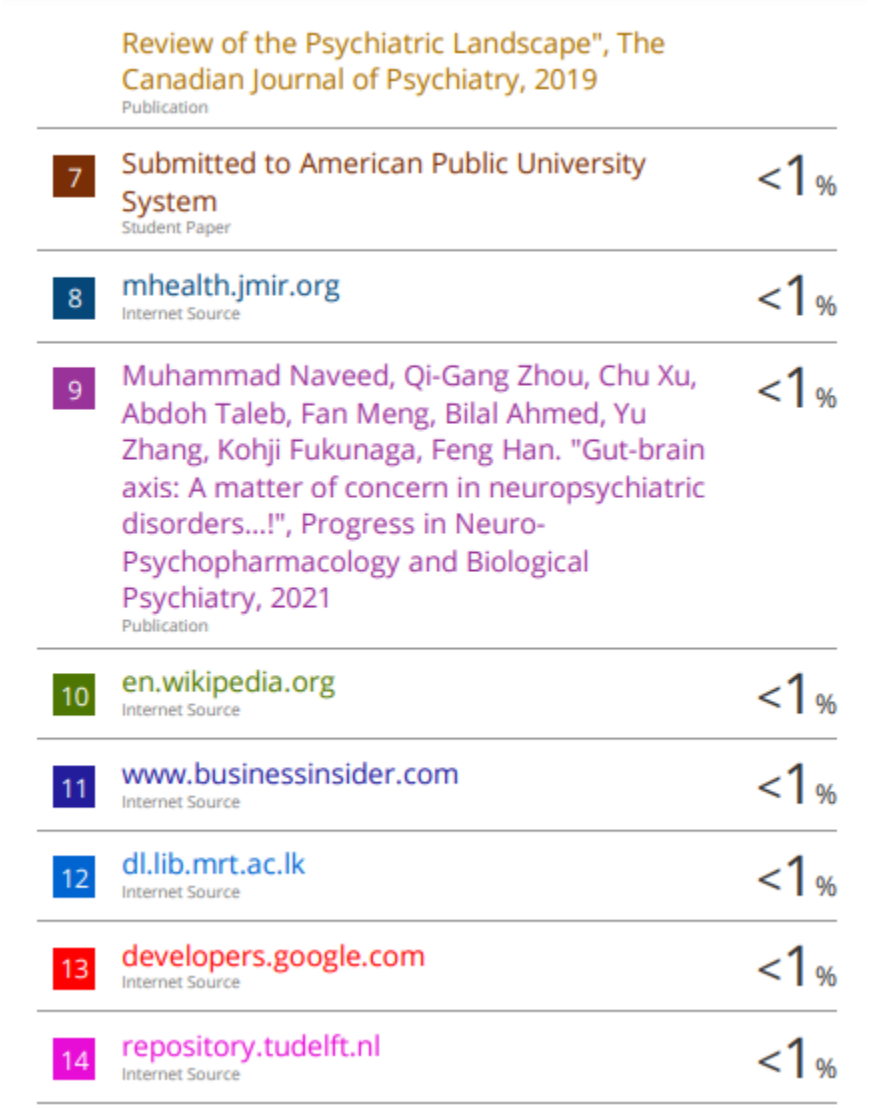
MENTAL HEALTHCARE CHATBOT

ORIGINALITY REPORT

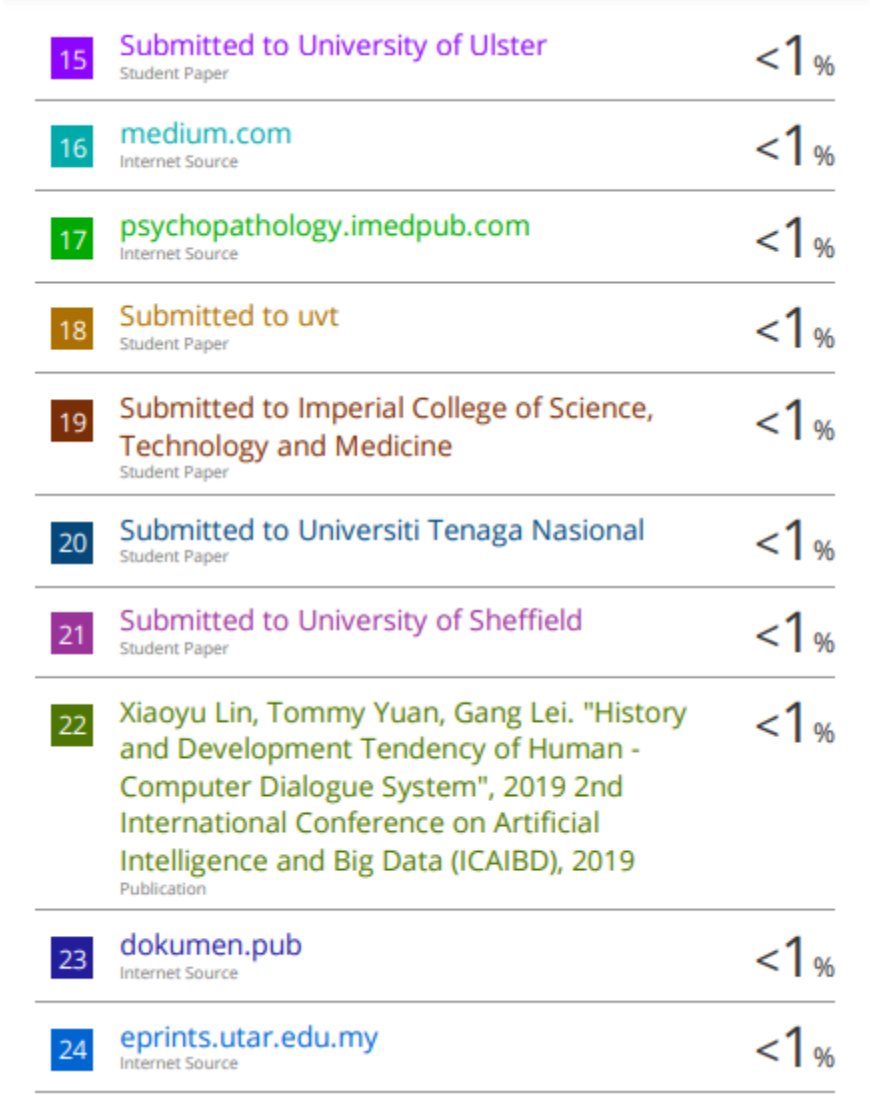
12%	9%	5%	8%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to The Hong Kong Polytechnic University Student Paper	2%
2	Submitted to Universiti Tunku Abdul Rahman Student Paper	1%
3	machinelearningmastery.com Internet Source	1%
4	Jennifer Hill, W. Randolph Ford, Ingrid G. Farreras. "Real conversations with artificial intelligence: A comparison between human–human online conversations and human–chatbot conversations", Computers in Human Behavior, 2015 Publication	1%
5	data-flair.training Internet Source	1%
6	Aditya Nrusimha Vaidyam, Hannah Wisniewski, John David Halamka, Matcheri S. Kashavan, John Blake Torous. "Chatbots and Conversational Agents in Mental Health: A	1%



	Review of the Psychiatric Landscape", The Canadian Journal of Psychiatry, 2019 Publication	
7	Submitted to American Public University System Student Paper	<1 %
8	mhealth.jmir.org Internet Source	<1 %
9	Muhammad Naveed, Qi-Gang Zhou, Chu Xu, Abdoh Taleb, Fan Meng, Bilal Ahmed, Yu Zhang, Kohji Fukunaga, Feng Han. "Gut-brain axis: A matter of concern in neuropsychiatric disorders...!", Progress in Neuro-Psychopharmacology and Biological Psychiatry, 2021 Publication	<1 %
10	en.wikipedia.org Internet Source	<1 %
11	www.businessinsider.com Internet Source	<1 %
12	dl.lib.mrt.ac.lk Internet Source	<1 %
13	developers.google.com Internet Source	<1 %
14	repository.tudelft.nl Internet Source	<1 %



15	Submitted to University of Ulster Student Paper	<1 %
16	medium.com Internet Source	<1 %
17	psychopathology.imedpub.com Internet Source	<1 %
18	Submitted to uvt Student Paper	<1 %
19	Submitted to Imperial College of Science, Technology and Medicine Student Paper	<1 %
20	Submitted to Universiti Tenaga Nasional Student Paper	<1 %
21	Submitted to University of Sheffield Student Paper	<1 %
22	Xiaoyu Lin, Tommy Yuan, Gang Lei. "History and Development Tendency of Human - Computer Dialogue System", 2019 2nd International Conference on Artificial Intelligence and Big Data (ICAIBD), 2019 Publication	<1 %
23	dokumen.pub Internet Source	<1 %
24	eprints.utar.edu.my Internet Source	<1 %

25	scholar.sun.ac.za Internet Source	<1 %
26	zenodo.org Internet Source	<1 %
27	www.insightpsychological.ca Internet Source	<1 %
28	web-tools.uts.edu.au Internet Source	<1 %
29	www.betterhelp.com Internet Source	<1 %
30	Takuya Iwamoto, Kohtaro Nishi, Taishi Unokuchi. "Chapter 15 Simultaneous Dialog Robot System", Springer Science and Business Media LLC, 2019 Publication	<1 %
31	pdfs.semanticscholar.org Internet Source	<1 %
32	pt.scribd.com Internet Source	<1 %
33	spod.tarc.edu.my Internet Source	<1 %
34	www.library.ln.edu.hk Internet Source	<1 %

FINAL YEAR PROJECT WEEKLY REPORT
(Project II)

Trimester, Year: Y4 S2	**Study week no.: 1-7**
Student Name & ID: OON XIN YI 1703418	
Supervisor: Dr. Jasmina Khaw Wen Min	
Project Title: Mental Healthcare Chatbot	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Review Back to what I had done in FYP1.

2. WORK TO BE DONE

Building a voice emotional recognition model

3. PROBLEMS ENCOUNTERED

-Difficult to build a voice emotional recognition model

-Both models is not compatible with each other, need to rebuild a text processing model

4. SELF EVALUATION OF THE PROGRESS

Slow Progress

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y4 S2	**Study week no.: 7-12**
Student Name & ID: OON XIN YI 1703418	
Supervisor: Dr. Jasmina Khaw Wen Min	
Project Title: Mental Healthcare Chatbot	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

-Build a voice emotional recognition model

-Roughly design the flow for the content of the chatbot

-Convert tf model into keras model

-Build Chatbot UI

2. WORK TO BE DONE

-Combine 2 models together and make sure both can be function well (Model Integration)

-Make use of both model into the functionality of chatbot

-Evaluate the chatbot performance

-Write Report

3. PROBLEMS ENCOUNTERED
-Accuracy of the model need to improved -Minor frontend error need to be amended
4. SELF EVALUATION OF THE PROGRESS
Slow Progress

Supervisor's signature

Student's signature

Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	OON XIN YI
ID Number(s)	17ACB03418
Programme / Course	BACHELOR OF COMPUTER SCIENCE
Title of Final Year Project	MENTAL HEALTHCARE CHATBOT

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)**
Overall similarity index: 12 % **Similarity by source** Internet Sources: 9 % Publications: 5 % Student Papers: 8 %	
Number of individual sources listed of more than 3% similarity: none	

Parameters of originality required and limits approved by UTAR are as Follows:

(i) Overall similarity index is 20% and below, and

(ii) Matching of individual sources listed must be less than 3% each, and

(iii) Matching texts in continuous block must not exceed 8 words

Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

Signature of Supervisor

Name: Jasmina Khaw Yen Min

Date: 16 April 2021

Signature of Co-Supervisor

Name: _____-_____

Date: _____-_____

Bachelor of Computer Science (Honours)
Faculty of Information and Communication Technology (Kampar Campus), UTAR

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	17ACB03418
Student Name	OON XIN YI
Supervisor Name	DR. JASMINA KHAW WEN MIN

TICK (√)	DOCUMENT ITEMS Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
√	Front Cover
√	Signed Report Status Declaration Form
√	Title Page
√	Signed form of the Declaration of Originality
√	Acknowledgement
√	Abstract
√	Table of Contents
√	List of Figures (if applicable)
√	List of Tables (if applicable)
√	List of Symbols (if applicable)
√	List of Abbreviations (if applicable)
√	Chapters / Content
√	Bibliography (or References)
√	All references in bibliography are cited in the thesis, especially in the chapter of literature review
√	Appendices (if applicable)
√	Poster
√	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report. _____ (Signature of Student) Date: 16 April 2021	Supervisor verification. Report with incorrect format can get 5 mark (1 grade) reduction. _____ (Signature of Supervisor) Date: 16 April 2021