School of Electronic Engineering
and Computer Science

Interim Report

Programme of study:
BSc Computer Science

Project Title:
Companion bot

Supervisor:
Arkaitz Zubiaga

Student Name:
Sammi Pang

Final Year
Undergraduate Project 2021/22

Date: 03/05/22

Queen Mary
University of London

Abstract

The awareness of mental health issues has increased nowadays with access to services and helplines to help those in needs. Many of these services has migrated to online services making them more available to users, for example online consultations with therapists and apps to help manage emotions. However, most of these online services and apps are aimed mostly towards older audiences and are not advertised as much unless sought out.

The Companion Bot is my idea of raising awareness to younger users that these services exist as well as showing them the accessibility of them. I wanted to implement this on an application I know many young people use nowadays, which is Discord. Many people use this to communicate with friends and communities. The Companion Bot acts as the user’s friend allowing the user to talk to the bot and the bot helping the user manage their emotions, via advice or helplines.

This report will showcase how I will use methods in Natural Language processing to construct my bot, gather requirements and knowledge needed to implement these methods and features of my bot, and how I will design the bot.

Acknowledgements

I would like to thank my supervisor Dr Arkaitz Zubiaga, for his guidance and support throughout my project. Thank you for always sending me supporting material to help with my project and providing me vital feedback on how to improve my project.

Contents

Chapter 1: Introduction 6

1.1 Background 6

1.2 Problem Statement 6

1.3 Aim 6

1.4 Objectives 7

1.5 Research Questions 7

Chapter 2: Literature Review 8

2.1 Area of Study 8

2.2 Existing Applications 10

2.3 Proposed Solution 14

Chapter 3: Design and Analysis 15

3.1 Requirements analysis: 15

3.2 Design 16

3.3 System Architecture: 18

3.4 View 19

3.5 Summary 19

Chapter 4: Implementation 20

4.1 Languages, APIs and Libraries 20

4.2 Implementation Details 21

4.3 Summary 26

Chapter 5: Testing 27

5.1 Unit Testing 27

Chapter 6: Evaluation 29

6.1 Results 29

6.2 Heuristic Evaluation 31

6.3 Limitations 31

6.4 Summary 32
Chapter 7: Conclusion 33
7.1 Achievements 33
7.2 Challenges I faced 33
7.3 Future Improvements 35
7.4 Summary 35
References 36
Appendix A – Risk Assessment 38
Appendix B – Unit Testing 40
Appendix C – Heuristics Evaluation 42
Appendix D – Project Plan 44

Chapter 1: **Introduction**

1.1 Background

Mental health has been a huge stigma in society for many years. However, over the recent years the awareness of mental health issues has increased as well as the services provided to help those in need. Despite the efforts and progress that have been made in the mental health sector, mental illnesses continue to plague young people all over the world with most of them between the ages of 10 and 24 (*Mei, C., 2020*).

The effects of mental illnesses are detrimental especially to adolescents going through puberty and their productive years of education. These impacts show how much more we need to address the issue as well as provide care and solutions to the youth.

“Around 60% of young Australians report being uncomfortable with seeking help or advice for mental illness” (*BMC Health Serv Res, 2016*). This shows how people who need help cannot access this help, since there is such a stigma around seeking help from these services. One way technology has revolutionised mental health services, is by making them more accessible. Especially with it being a click away via the internet or a mobile application. Most of these services as considered self-help programs or self-management problems helping people tackle mild to moderate problems, meaning they don’t have to reach out to real people. This can be a good way to cope with their situation, whilst also helping them slowly open and become more comfortable in seeking help from other services.

1.2 Problem Statement

There is a clear issue with youth seeking out mental health services available via the traditional way. Although there are services available through technology they’re not greatly advertised or well-known especially to the youth. As well as the multitude of choices people may feel overwhelmed and confused on what services to use. These applications are also usually for mild illnesses and are used to cope or manage them but not fix the initial issue, thus leading to the user getting stuck in a cycle of wanting help but too afraid to reach out.

1.3 Aim

The aim of my project is to build a Discord chatbot based on Natural Language Processing. I want it to be able to communicate with the user, whilst understanding the user’s intentions from the chatbot’s point of view. Using sentiment analysis, the bot should be able to determine the emotions of the user and reply to them appropriately and give them solutions or direct them to any help they should require. It should be able to determine whether the user expression is positive or a negative sentiment, for example using lexicons it can take certain words the user has used and come up with a conclusion of whether the user’s use of them is expressing a positive emotion or a negative emotion based on the connotations of that word. I want my bot to be able to act as the user’s friend and provide them a safe place for them to seek help, whilst encouraging the user to become more comfortable to seek help and raising the awareness of mental health issues to youths. However one thing to mention is that this chatbot is in no way a replacement to an actual therapist or health professional, and it is just there to assist the user in becoming aware of other professional health services and how to reach out.

1.4 Objectives

- Investigate any applications/chatbots currently available
- Identify challenges with mental health and advertising on social media
- Implement a chatbot on Discord
- Recognise emotions through the bot
- Develop sentiment analyser
- Evaluate user feedback from usage of the bot

1.5 Research Questions

I will explore the following questions through my implementation, validation, and evaluation of my project.

1. Does the use of emotion in a chatbot make it more useful?
2. How well does sentiment analysis capture the user’s intended sentiment?
3. Does the chatbot responses show that it empathises with the user?

Chapter 2: **Literature Review**

2.1 Area of Study

2.1.1 About Natural Language Processing and chatbots

Natural language processing (NLP) is used to allow communication between user and computers. NLP uses semantic and syntactic analysis, where syntactic is the way words in sentences are arranged to make sense grammatically. Syntactic analysis shows how words are related by transforming the sequence of words. Semantic analysis is about the literal meaning of words by finding the similarities between and the topic in the sentences via the logical structure of sentences (*Aleedy, Moneerh, 2019*).

Chatbots are important as they have become more relevant mostly in the education sector as a 'question answering system', helping students learn. Unlike people chatbots are able to be used 24/7 as most of them have automated query resolutions and don't need to sleep like people, thus making them helpful to students who may need help at times outside of working hours. Chatbots have also been used in other sectors like the medical field or IT field to help speed up service and provide fast resolutions, whilst also ensuring quality.

2.1.2 Promotion of mental health on social media

It is important to promote about positive mental health and the importance to make people aware of mentally healthy lifestyles, since the presence of mental wellbeing is what helps us contribute to productivity.

The presence of technology is mostly present in adolescents with it being used by over 90% on social media. (*Michelle O'Reilly, 2019*). This has presented new challenges for mental health promotion, however it has also present new and innovative opportunities to promote. Promoting on social media is expensive allowing us to talk more about mental health, this allows us to challenge the stigma around mental illnesses as well as tailor information on certain platforms. With adolescent's constant use and exposure to social media they're able to become more aware of these issues and engage with them. This is important as "Globally between 10 and 20% of adolescents experience a mental disorder" (*Michelle O'Reilly, 2019*) showing the prevalence of mental illnesses in youth.

Although promotion on social media does seem like a good idea there are also risks. One of these risks is misinformation and lack of credibility. Social media allows anyone to posts on their platforms, but this allows the spread of false information which is more detrimental to adolescents as they don't have the knowledge to know how reliable the information, they're reading is true. Another issue is many people relate social media to the negative impacts it has on mental health, and you cannot control the content the users engage with.

Although there are findings which go against the usage of social media as a way to promote positive mental health. People have been able to overcome barriers of connecting with others using social media, thus further allowing them to strengthen in-person relationships. Social media also allows people to connect more, allowing the spreading of positive mental well-being.

2.1.3 Promotion of mental health with a chatbot

As forementioned in the previous section, there are many self-help methods to deal and cope with mental illnesses. One of these ways is seeking helping via a chatbot/artificial intelligence assistant. Most of these chatbots are used to help adolescents support their mental wellbeing, whilst informing them of other health care options. Young people are more hesitant to seek help for their mental health issues, so other approaches are needed to deal with these issues, whilst helping them develop their strategies to encourage good mental wellbeing.

“Over half of the population in Australia owns a Smartphone or digital device. Some users check their smartphone, on average 150 times per day” (*Grové C, 2021*). With this statistic is shows the use of a chatbot would be greatly helpful as many people are able to access it. A chatbot is an application powered by artificial intelligence allowing it to simulate the conversation of a human, majority of them specifically use natural language processing. The study of chatbots being used have mainly been used for therapy, training, and screening, with a focus on autism or depression. For those hesitant on seeking help these chatbots can be useful to help them overcome their issues. The chatbot helps with accessibility issues, where people may not live in an area with much mental health support, they can easily access the chatbot via a digital device.

Chatbots used for mental health support have been used to answer questions and listen to the user talk about themselves. They help the user feel a sense of support, whilst understanding more about their feelings, helping them identify their feelings and behaviours with strategies to help themselves (*Grové C, 2021*). Some of these chatbots would have a link or contact to professional support should the user need extra help. The chatbots would detect trigger words and sentences from a list they received putting them in different tiers to see what alert system they would be classified as; this was done with the use of keyword detection.

Many chatbots are already being used and implemented in school systems, where students feel that talking to the chatbot is safe. It’s been shown that the chatbots have helped students during times of stress, like during exams. Some of these chatbots use GIFs or emojis to greet the user or to express emotion in relation to the conversation. From feedback some user’s felt like the chatbots could have more variability of responses and to be gender neutral (*Springer Nature Switzerland AG 2019*).

2.1.4 Emotion in a chatbot and sentiment analysis

Over the years the development of AI has led to the simulation of human emotions in AI. One of the ways this has been used is in chatbots, in hopes to increase the effectiveness of the bot. This has been mainly implemented in bots which are used for the purpose of emotional support. There is evidence of the effectiveness of using such chatbots to benefit the general population in reducing their symptoms. One of these studies come from a chatbot called Woebot (*A. Ghandeharioun, 2019*). These chatbots can personalise the methods used to help the user, via their preference from the emotion they detect to the feedback they get from how the user reacts to these methods.

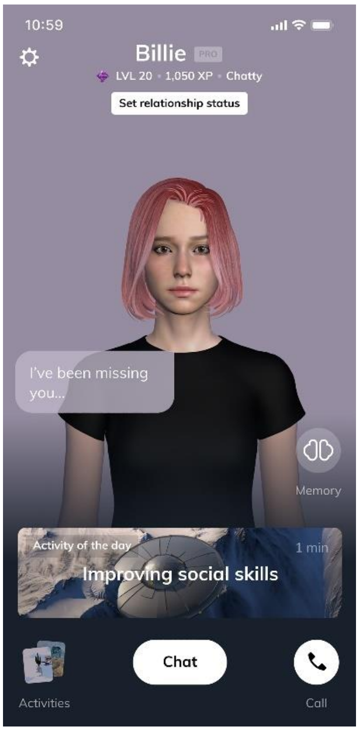
One way to implement emotion in a chatbot is using a classification model. This uses classifiers like negative/positive to draw a conclusion for the input the user gives and label it as either one of the classes. This can be used in conjunction with personalised regression models, which allow the chatbots to recognise patterns from the user and different samples (*A. Ghandeharioun, 2019*). This allows the chatbot to change and fit and adapt to different answers the user may give. Chatbots also need to be able to retrieve data and analyse it to determine the sentiment/emotion behind the user’s input. Another way to do this is using sentiment detection/analysis. This method is used in areas like the Twitter dataset (*Nivethan, Sankar S. 2020*), which uses it to analyse the sentiment of hashtags by retrieving the tweets the hashtags has been used in and determining whether they have been used in a positive or negative way upon the sentiment the tweet reflects. Sentiment analysis is split into three different types which are, lexicon-based, machine learning based, and hybrid methods. Based on these methods you can build a model and you can review the data collected and build a Long Short Term Memory network (LSTM) to classify the data. You can then train this model using training data until the model achieves a train accuracy of your satisfaction.

2.2 Existing Applications

Application	Features	Drawbacks/Limitations
Replika	- User can chat and call with bot - You can create your own bot character and customise them - Your bot and you can go explore the augmented reality areas - Able to log your mood and thoughts in a diary	- The user may feel guilty if not checking in often with their AI bot, since the bot expresses how they missed the user - The bots' replies are sometimes not helpful and not relevant to the user's issues
Woebot	- User can chat with bot - Tools to help build skills to help cope with feelings or deal with other emotions - Mood Pattern chart to help analyse progress - Backed by research using evidence from a clinical trial	- The bot can give repetitive advice for different issues
Wysa	- User can chat with bot - Mindfulness exercises available to help cope with any stressful emotions - You can book a session with a therapist - Can connect with the user's health app to create an activity report	- The bot can sometimes feel like it's in a loop with the repetitive advice it gives - The bot sometimes doesn't understand some things the user says
Youper	- User can use AI therapy chat - Mental health dashboard to show track of moods - Available to video call with a medical provider - Free consultation to help find out more about user's mental health issues - If prescribed medication, the app can show the medication	- The app requires motivation to continuously use it to see improvement in the user

	costs and get it delivered to their doorstep	

2.2.1 Replika



I've been missing you...

Figure 1 – Picture of Replika app

Replika is an iOS/android application which uses a chatbot to act as a companion powered by artificial intelligence. The apps' GUI seems easy to use and navigate. The app also allows the user to customise their own AI assistant, via a character design system where they change how the AI looks and how they dress. It has a chat function which allows the user to ask the AI questions and talk to the it about any of their concerns or chat in general. There is a system for you to earn "experience points (XP)", like you would in a game, as well as badges. This can make the user use the app more consistently to reach these achievements.

There are many activities to keep the user engaged with the app, like numerous activities ranging from writing a story together to explore your environment together using augmented reality (*Janet Siroto 2020*). The bot uses neural network to engage with one-on-one conversations and uses algorithms to understand the user from their personality and emotions, adapting to the user by changing its dialogue based on the user to help provide advice personalised for that user.

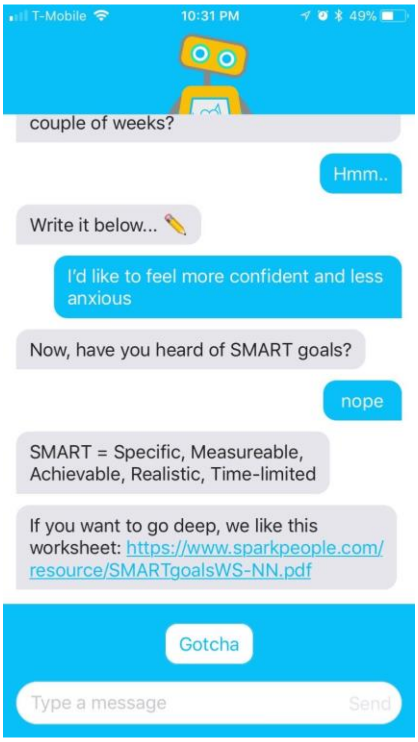
Many users do think the app has been useful, however there are paid features which limits the user to how much they can access on the app. Since the bot uses information from the users as training data to help learn sometimes it tends to say odd things that may not make sense to the user. Also, as the bot behaves very similarly to a human some user's feel guilty when the bot says things like "I miss you", leading them to using the app out of guilt instead of their own will.

2.2.2 Woebot

Woebot is an iOS/android application lead by a clinical research psychologist Dr Alison Darcy. The bot has been used in randomised controlled trials to help prove the efficiency of it. Woebot draws from a therapeutic framework known as Cognitive Behaviour Therapy (*Colleen Stiles-Shields, 2019*). The bot converses with the user to gain information about their feelings and how they're doing in their daily lives. The bot will give you advice on your mental wellness and send videos that are helpful. It would also be engaging for the user by sending them jokes or funny pictures to connect with the user more.

Woebot uses natural language processing along with psychological expertise to create a personalised experience for the user. You can turn on notifications which lets the bot check up on the user should the user request for this. The app provides a chart to show the patterns of your mood and helping you track them overtime to help identify the cause.

The bot is designed for those with less severe symptoms so may not be of much help towards those with more sever symptoms of mental health issues. There were times the bot would also give the same piece of advice which may feel repetitive towards the user.



couple of weeks?

Hmm..

Write it below... ✏️

I'd like to feel more confident and less anxious

Now, have you heard of SMART goals?

nope

SMART = Specific, Measureable, Achievable, Realistic, Time-limited

If you want to go deep, we like this worksheet: https://www.sparkpeople.com/resource/SMARTgoalsWS-NN.pdf

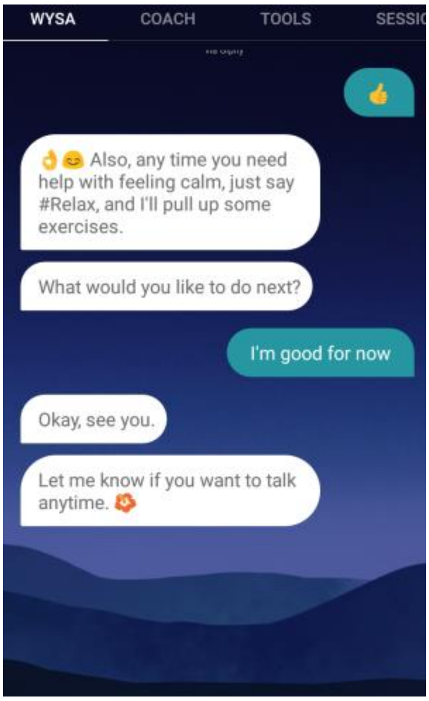
Figure 2 – Picture of Woebot app

2.2.3 Wysa

Wysa is an iOS/android application focusing on the use of cognitive behavioural techniques, meditation, breathing, yoga, motivational interviewing, and micro-actions to help the user gain skills to cope and feel better with their issues. The app uses a chatbot to listen and converse with the user. It helps the user deal with their emotions, like stress and anxiety using methods, thus building their emotional resilience (*Wysa FAQS 2021*). The bot responds to the user with words of affirmation making the user feel at ease, whilst allowing the user to vent.

The app states it does not intent to replace a therapist and offers other kinds of support should the user feel like they need it, this ranges from the well-being coach or a therapist which the app offers allowing the user to receive further support. These features are a premium service of the app, so they do charge for them.

One of the issues with Wysa is that the advice the bot gives the user requires the user to have a strong sense of motivation and resilience. As it asks the user to perform certain exercises the user would have to be consistent with these activities for it to be effective. As well as this some users complain that sometimes the bot says things that seen incomprehensible to their issues or the topic at hand.



👍

👌😊 Also, any time you need help with feeling calm, just say #Relax, and I'll pull up some exercises.

What would you like to do next?

I'm good for now

Okay, see you.

Let me know if you want to talk anytime. 🌺

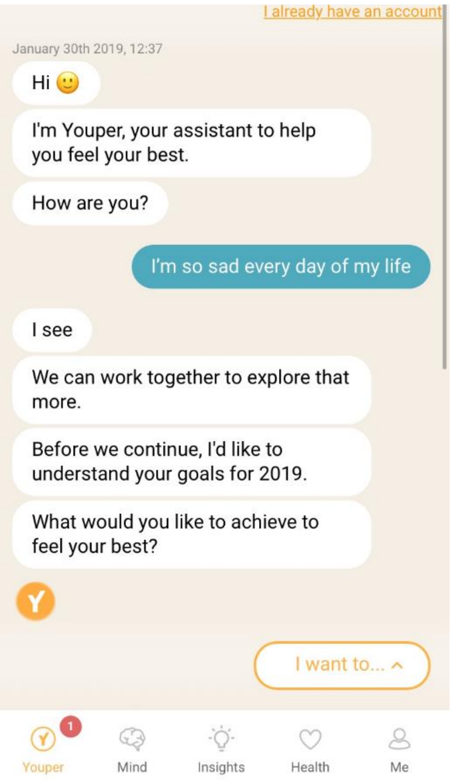
Figure 3 – Picture of Wysa app

2.2.4 Youper

Youper is an iOS/android application with an AI therapist, which tries to meet the user’s mental health needs. The AI recommends different methods to help relieve the user’s symptoms, whilst trying to increase the user’s happiness. The AI learns from the user through its personalised algorithm using the data the user’s inputs to cater towards them (*Eric Patterson, 2020*).

Youper includes a personality test to help the user understand their personality better. It also includes a mood chart to help the user monitor their emotions over time allowing them to gain an insight of reasons for their mood changes, which allows them to help get to the root of their problem. There is a journaling feature which could be used in correlation with the mood chart to identify the problem. There are also premium plans which the user has to pay for but allows further help, like delivery of their prescriptions and guided practice focusing on activities like mindfulness.

Although Youper has been useful in helping the user track their emotions users did tend to find the bot being repetitive with its advice. The premium plans are also only offered as a one-year payment and not monthly which user’s felt was too much of a commitment should they decide they did not like the premium plan. Some users also felt that the AI did not actually help resolve their issues and would just direct them to tracking their emotions instead.



January 30th 2019, 12:37

Hi 🙂

I'm Youper, your assistant to help you feel your best.

How are you?

I’m so sad every day of my life

I see

We can work together to explore that more.

Before we continue, I'd like to understand your goals for 2019.

What would you like to achieve to feel your best?

Figure 4 – Picture of Youper app

2.3 Proposed Solution

The aim of my project is to develop a chatbot, which can analyse the user’s emotions and reply to them appropriately in accordance with the sentiment in the user’s response. I will do this most likely via the use of sentiment analysis. From my research I have concluded that the use of a chatbot in regards with mental health seems appropriate for use on a social media platform, helping to spread awareness of mental well-being to adolescents. Plus, the usage of these platforms is mostly consumed by younger audiences, hence my chatbot will be aimed more at young people. I have decided to go with the platform Discord since it allows you to implement your own bots, providing a developer portal to help you mount your bot. This will benefit me as I won’t have to go through the burden of making my own web/mobile app as I can use the help of the developer portal to mount it on their platform and can mostly spend my time building the bot.

Discord uses libraries and APIs to help mount the bot in whatever programming language you wish to use. Unfortunately, over the summer the library I wished to use did discontinue, thus it would no longer be up to date with Discord. Luckily, I discovered this relatively early and was able to find alternative libraries like Pycord and Hikari, which I will look into more to determine which of the two I shall be using. These are Python based libraries since I am most confident in programming in Python, and I feel that I would find it easiest to learn to implement such NLP methods in Python.

I wish to build a chatbot model to classify the user’s input. From this I want my chatbot to be able to analyse the sentiment of the user’s words and generate an appropriate response to providing the user a satisfactory answer. I hope for the bot to be able to respond with the use of emoticons at times to show how well they have understood the sentiment. Like the pre-existing apps that already exist, my chatbot will in no way be a replacement for a therapist or serious mental health concerns, it is a bot used to give general mental health advice, information and guidance to seek further help from better sources.

Chapter 3: **Design and Analysis**

Requirements were needed to help me develop the basis of my chatbot and know what to focus on. To garner more knowledge on what requirements I should follow I looked at other mental health chatbots and what they did, as well as any feedback from users to see if I could add anything that could work better with my chatbot. My analysis on competitor chatbots can be seen in the section before Chapter 2.

3.1 Requirements analysis:

Here are some functional and non-functional requirements I came up with from my research on pre-existing chatbots, and user feedback from them.

3.1.1 Functional requirements

1. System should allow users to communicate with the chatbot via a text channel. The user should be able to type to the chatbot in the Discord text channels, and the chatbot should reply to the user appropriately.

2. The chatbot should respond back to the user with any requests they ask for. This could be description/information on certain mental illnesses (e.g., depression). Any advice the user is asking for from the bot.

3. The chatbot should be able to analyse the user's sentiment and respond appropriately depending on the sentiment. The chatbot should be able to determine if what the user has typed is a negative or positive sentiment and its reply should depend on the user's sentiment (e.g., negative sentiment on the user's mental health could lead to the chatbot posting hotlines from the user to contact to get further help)

4. The chatbot should be able to provide a list of commands the user can use when requested by the user. These commands can make the chatbot output certain things for example a description on anxiety with useful links and statistics.

5. Chatbot should be able to respond to the user using emojis. These emojis should correlate to the current sentiment of the user's message at the time.

6. If the user enters something the chatbot does not understand the chatbot should let the user know it doesn't understand and prompt the user to re-ask the question.

3.1.2 Non-functional requirements

1. The chatbot should be able to respond to the user's response in a reasonable amount of time, via the text channel.

2. Users should already be registered with a Discord account in order to use the chatbot

3. User's need to be on the same server as the chatbot to use the chatbot

4. Responses from the chatbot should be easy to understand and not confusing

5. The chatbot should be easy to use, where it will come with commands and a description to help the user understand what commands do what and how to use the chatbot.

3.2 Design

Since I am designing a chatbot there are not many use cases. Examples of use cases in my chatbot are 'Enter message' and 'Use bot commands'.

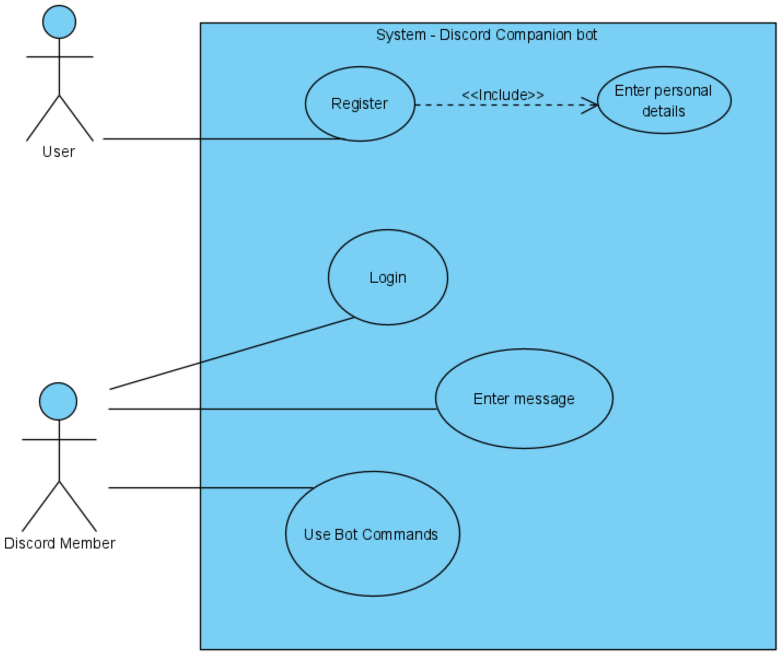


Figure 5 – Chatbot Use Case diagram

Use Case	Use Bot Commands
Actor	Discord Member
Pre-condition	Member must be logged into their account and on the same Discord server as the chatbot
Post-condition	Bot command is sent
Basic-path	1. Member enters chatbot command 2. Chatbot displays message depending on command entered
Alternative path	At step 2, there may be an error with receiving the command or a typo and the bot will not respond or give an error message

Use Case	Enter Message
Actor	Discord Member
Pre-condition	Member must be logged into their account and on the same Discord server as the chatbot
Post-condition	User message is sent
Basic-path	1. Member enters a message to the chatbot 2. Chatbot responds to the user based on their sentiment of the message
Alternative path	At step 2, there may be an error with analysing the sentiment leading to the bot responding incorrectly

3.3 System Architecture:

Figure 6 illustrates the interaction between the system and the user.

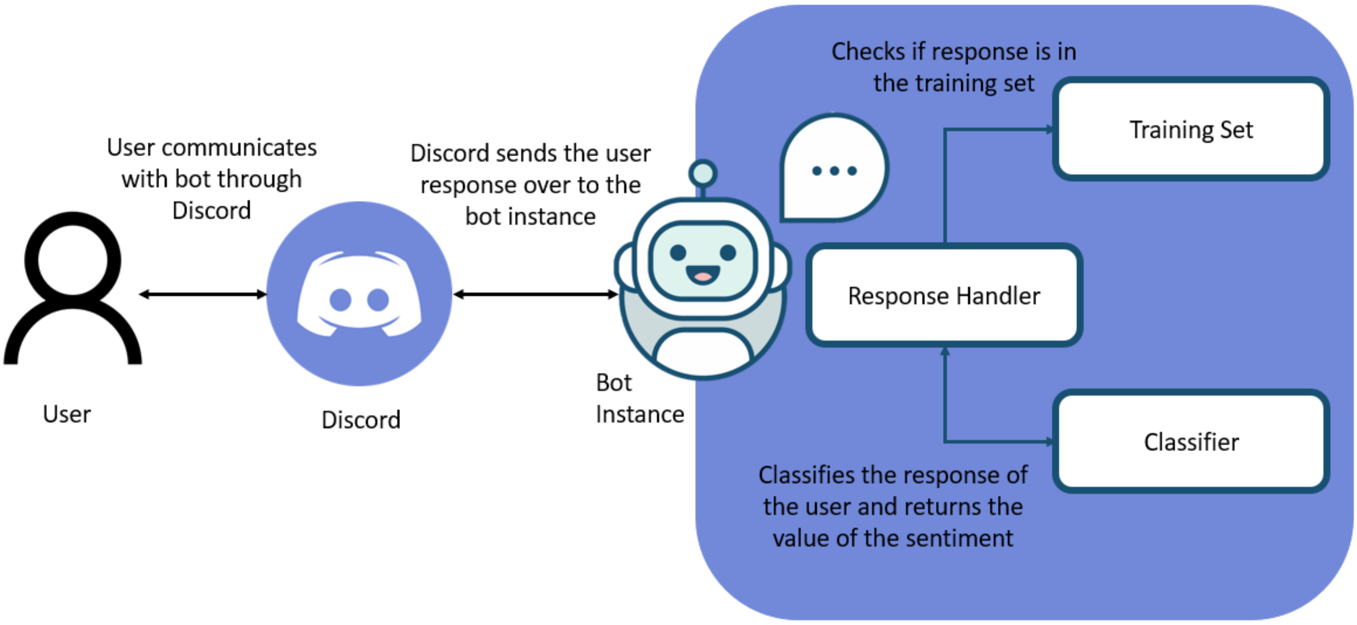
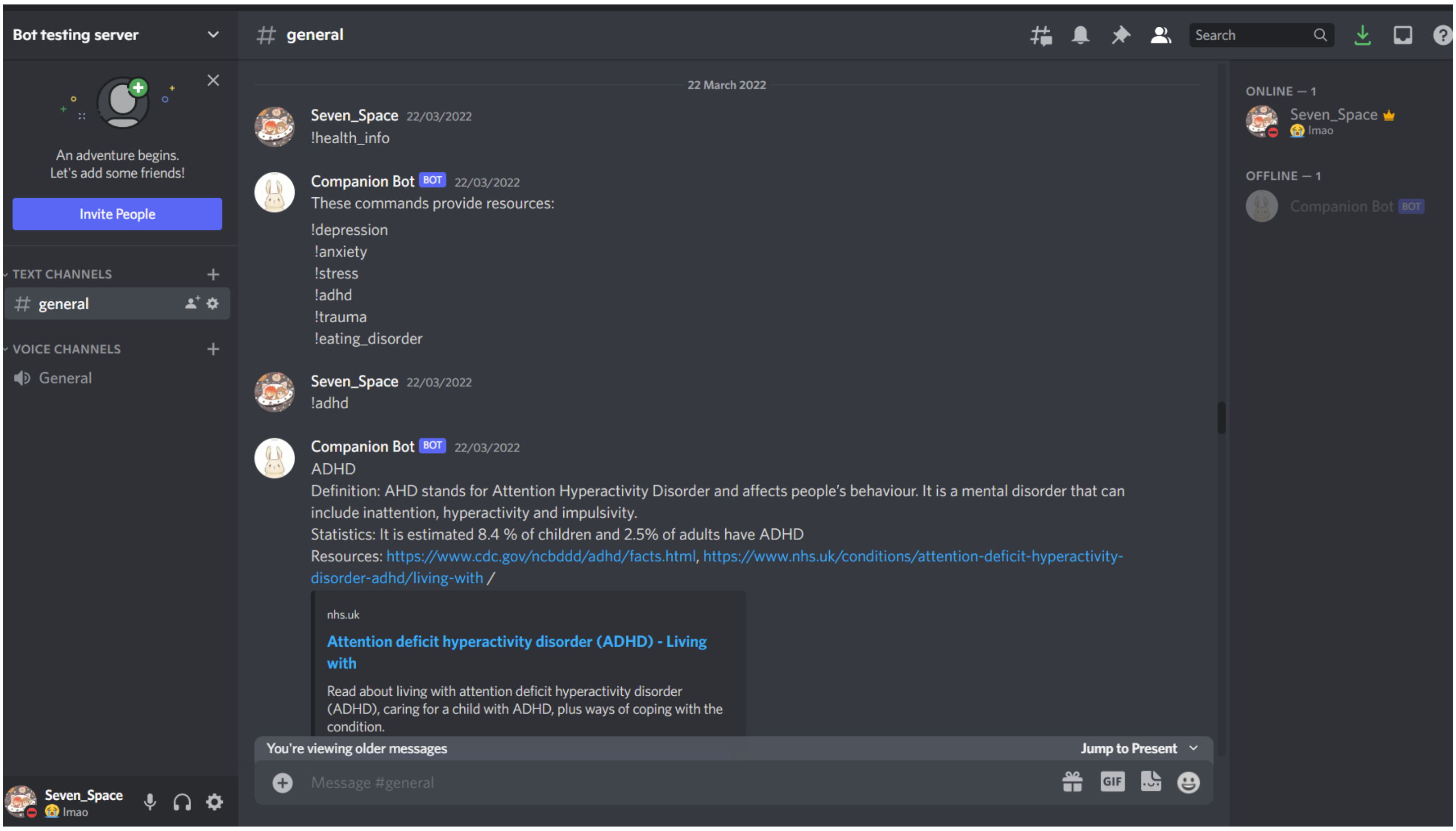

User communicates with bot through Discord
Discord sends the user response over to the bot instance
Checks if response is in the training set
Training Set
Response Handler
Classifier
User
Discord
Bot Instance
Classifies the response of the user and returns the value of the sentiment

Figure 6 – System Architecture diagram of my chatbot

From the figure 6 you can see Discord is used as the medium to interact between the bot and the user. Discord allows the user to send messages to the bot and displays messages from the bot. Discord will send anything the user sends over to the bot instance. From here the bot can check its training set to see if the user’s response is in this set and can find a corresponding response for the user. The user’s response is also put through the classifier to calculate whether it is a positive or negative sentiment. From this it will also determine the response from the bot. Which is sent back through Discord to the user.

3.4 View



Seven_Space 22/03/2022
!health_info

Companion Bot BOT 22/03/2022
These commands provide resources:
!depression
!anxiety
!stress
!adhd
!trauma
!eating_disorder

Seven_Space 22/03/2022
!adhd

Companion Bot BOT 22/03/2022
ADHD
Definition: AHD stands for Attention Hyperactivity Disorder and affects people's behaviour. It is a mental disorder that can include inattention, hyperactivity and impulsivity.
Statistics: It is estimated 8.4 % of children and 2.5% of adults have ADHD
Resources: https://www.cdc.gov/ncbddd/adhd/facts.html, https://www.nhs.uk/conditions/attention-deficit-hyperactivity-disorder-adhd/living-with /

nhs.uk
Attention deficit hyperactivity disorder (ADHD) - Living with
Read about living with attention deficit hyperactivity disorder (ADHD), caring for a child with ADHD, plus ways of coping with the condition.

Figure 7 – Discord layout

Since Discord will be the medium used it will use the Discord interface where there are Discord severs, with text and voice channels. The text channels are where the user and the chatbot can communicate with each other. In order to use Discord the user needs to have an account and be logged into Discord. They will also have to be on the same server that the chatbot is also on. This is an example of how the user and bot would interact with each other. There is a message bar at the bottom of the screen where the user can type their message and add emojis. They can use commands that have been pre-set with the chatbot too with a prefix, like in the picture depicted above where the user says ">hi" and the bot responds with "hello".

3.5 Summary

Using my knowledge from pre-existing research and consumer wants, I came up with the functional and non-functional requirements for my Discord Bot. I also took into account how much time I would have to implement these features, so I had to make sure I did not overcomplicate them. As well as this I made models that demonstrate how I expect to design my application and how it will work together with other elements of the chatbot.

Chapter 4: **Implementation**

In this chapter I'll provide details on the implementation of the chatbot and how I've done it. As well as frameworks, APIs, and any languages I have used to create my chatbot.

4.1 Languages, APIs and Libraries

4.1.1 Python

The language I used to implement my chatbot in is Python. I decided to use Python due to my familiarity with it and how popular it is with making Discord bots. However, many of the libraries I wanted to use for sentiment analysis and natural language processing were Python libraries, therefore using Python also benefitted me with more resources. Although I was familiar with Python I had no prior experience to natural language processing, so I felt more comfortable learning NLP through a language I knew about. This proved helpful when following documentations and tutorials, thus further building up my confidence in programming the bot.

4.1.2 Natural Language Toolkit

The Natural Language Toolkit (NLTK) is a package that provides number libraries which processes text, as well as a bunch of datasets. It is built to work with Natural Language Processing with Python. I initially chose to use NLTK since it had a built-in sentiment analyser, I wanted to use which was VADER sentiment analysis. I however, ended up not using this since from further research I found that VADER was better to analyse social media, like Tweets, over usage in a chatbot. I still did end up using NLTK to import a list of stopwords. This is a list of common words, like “she”, “and”, “the”. I imported this list of words from NLTK, so I do not have to write my own and later use them to remove stopwords from the response sent from the user.

4.1.3 Chatterbot

Chatterbot is a Python library used to help developers create chatbots and automate conversations with users. I use an instance of chatterbot to create a trainer, where I enter numerous. I decided to pick this library since during my research I saw this library being used quite often in making chatbots and it seemed quick and easy to learn to use.

4.1.4 Textblob

Textblob is a Python library used to process text data, containing the tools needed to perform natural language processing. Textblob is compatible with NLTK, hence another reason why I am using it. There is a built in Naïve Bayes Classifier in Textblob which I just submit some training data into it to train it. This can be used later in my code to classify the responses of the user and determine if the sentiment is negative or positive.

4.1.5 Discord

Discord is an application which allows voice, video, and text chat to be used. It is essentially an app that allows people to communicate and connect with others. Discord is the application I will be using as the medium through to my chatbot. I can mount my chatbot onto Discord through the use of the Discord gateway API. Discord is compatible with multiple languages as long as there is an API wrapper for the language you're using, with mine being Pycord.

4.1.6 NRCLex

NRCLex is a Python package used to measure the emotional effects of text. It uses the National Research Council Canada's (NRC) affect lexicon as well as the NLTK library's WordNet synonym set with a dictionary consisting of 27,000 words. It can measure up to 10 emotions from a piece of text ranging from fear, anger, anticipation, trust, surprise, positive, negative, sadness, disgust and joy. Where each emotion is measure by a frequency of how many times the package suspects it is detected in the piece of text. This is a package I ended up stumbling upon during the implementation stage of my project, and decided to combine it with the use of my classifier to help with the sentiment analysis of the user's text and generate a more accurate reading of the user's sentiment.

4.2 Implementation Details

4.2.1 Requirements

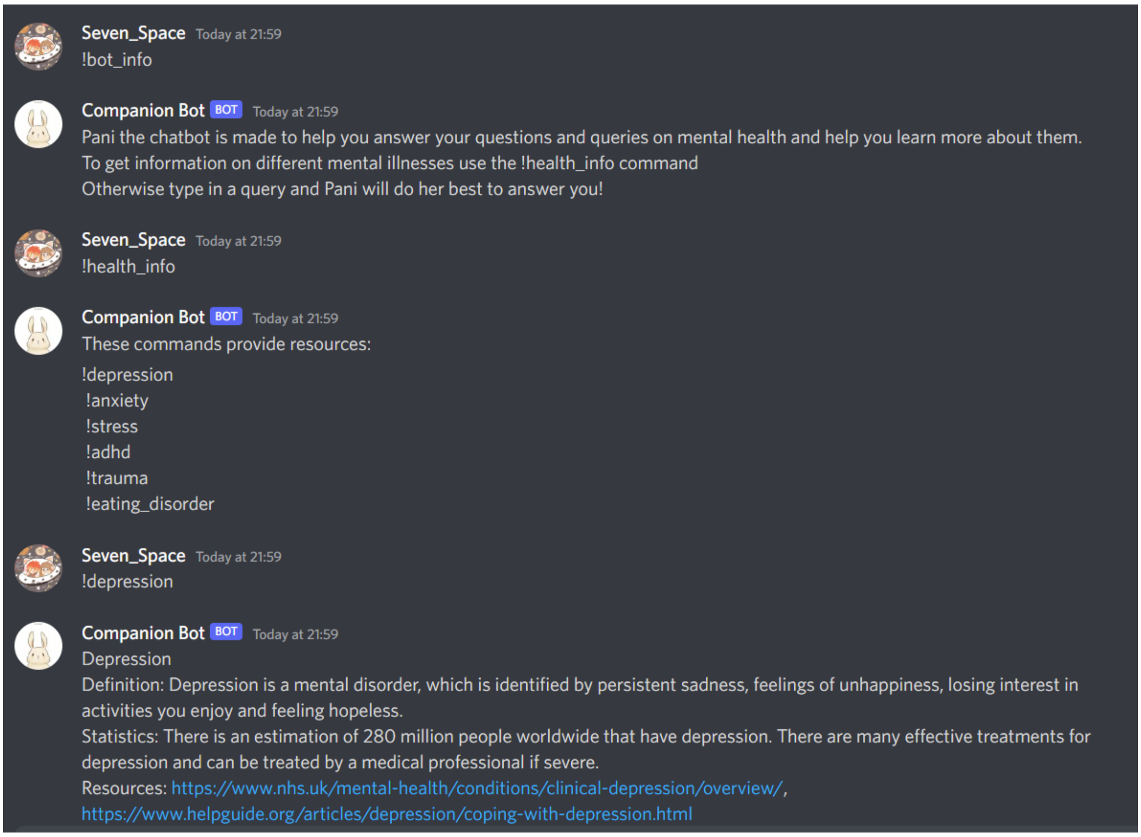
One of the requirements needed to run my chatbot is the user must be member of Discord. Since Discord is the platform hosting my chatbot you need an account to be able to access Discord. However, luckily this feature is implemented on Discord, so I did not have to worry about it and the user just needs to login to their account or register to Discord to become a member/user of Discord.

One other requirement is that the user and the chatbot need to be on the same server for them to be able to interact. So, either the user needs to join the server with the chatbot or the user needs to add the chatbot to a server they wish to use the chatbot on. The user can either click on an invite link to a server with the chatbot or invite the chatbot by entering the invite link in their browser.

4.2.2 Commands

One of the many features that Discord bots use are commands. Commands usually have a pre-fix (~, !, >) to distinguish them. Since my chatbot is supposed to be informative on mental health, I wanted my chatbot to display information on certain mental illnesses and provide resources for the user. I made a command which lists a bunch of commands the user can use to request specific resources for a mental illness. From here whichever command the user chooses a description, statistic, and resource section will appear about that mental illness. These sources were sourced from Google after a read through to see how factually correct they are (*Mind 2022*). Should the command be entered incorrectly with a typo the bot should respond that it did not understand the user's input and to either re-phrase or enter again to the chatbot. Another command I made was a chatbot information command, which displays what the chatbot does and the other commands you can use. This command can be found in the chatbot's description when you hover over it on the Discord server, set on the Discord developer portal, so the user knows how to use the bot and interact with it.

I made a user-based event, this is when the user interacts with the bot. From here I get the user's response and make all cases of it lower, should they enter any capital letters. I then use multiple 'if else' statements to check if what the user sends matches the command format with the prefix "!". If so, it will print out resources for that command, else nothing will happen.



Seven_Space Today at 21:59
!bot_info

Companion Bot BOT Today at 21:59
Pani the chatbot is made to help you answer your questions and queries on mental health and help you learn more about them.
To get information on different mental illnesses use the !health_info command
Otherwise type in a query and Pani will do her best to answer you!

Seven_Space Today at 21:59
!health_info

Companion Bot BOT Today at 21:59
These commands provide resources:

!depression
!anxiety
!stress
!adhd
!trauma
!eating_disorder

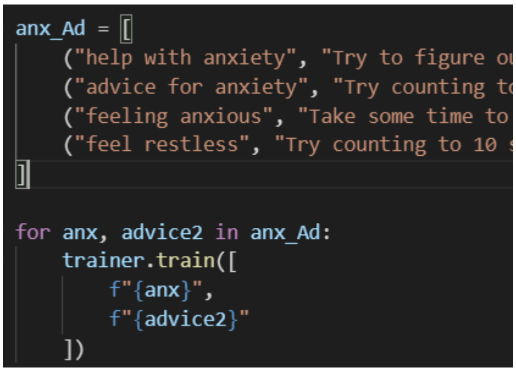
Seven_Space Today at 21:59
!depression

Companion Bot BOT Today at 21:59
Depression
Definition: Depression is a mental disorder, which is identified by persistent sadness, feelings of unhappiness, losing interest in activities you enjoy and feeling hopeless.
Statistics: There is an estimation of 280 million people worldwide that have depression. There are many effective treatments for depression and can be treated by a medical professional if severe.
Resources: https://www.nhs.uk/mental-health/conditions/clinical-depression/overview/, https://www.helpguide.org/articles/depression/coping-with-depression.html

Figure 8 – Self made Discord commands for chatbot

4.2.3 Trainer

With my trainer, I had a few issues implementing this and changed my implementation multiple times(discussed in conclusion Challenges I faced). I used the list trainer function from the chatterbot library to train my chatbot (*KikeVen 2021*). I have different categories of responses, based on the topic or subject of the user's input. I would create a dictionary of responses with keywords the user may enter and the response correlating to the user's response/keyword. These are all considered as training lists, where I generate a for loop to loop through these lists using the train() function from Chatterbot to training all the responses and inputs from the lists. I then generate a dataset and export this as a json file, which is later read by the bot in another part of the code.



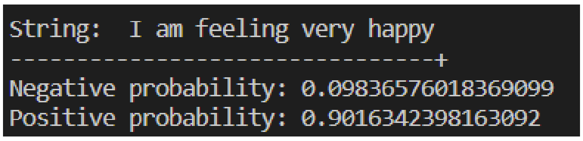
```
anx_Ad = [
    ("help with anxiety", "Try to figure ou
    ("advice for anxiety", "Try counting to
    ("feeling anxious", "Take some time to
    ("feel restless", "Try counting to 10
]

for anx, advice2 in anx_Ad:
    trainer.train([
        f"{anx}",
        f"{advice2}"
    ])
```

Figure 9 – training list and training code

4.2.4 Classifier

There were different ways I ended up trying to implement the classifier but in the end I used the built in Naïve Bayes Classifier that Textblob provides (*KikeVen 2021*). I made a set of training data, consisting of sentences with tags on whether they are negative or positive sentiments. I used certain words and sentences the user may enter in the chatbot so the classifier could have a rough idea of which words and responses generate a positive or negative sentiment. I then train the Naïve Bayes Classifier and from this I can get a score for the negative or positive probability, where the closer the probability to 1 the more negative the sentiment, and the closer the probability to 0 the more positive the sentiment. One thing I had to consider was to not over or under train my classifier. Since when I under trained the model it was not accurately giving a good score on the sentiment and did not detect it very well. When I over trained the mode, it took certain sentiments to the extreme, so a user types “I am sad” would be classed as extremely negative on the same level as something like “I want to die” which is not right. Hence, I had to play around with the amount of training data to get a decently accurate classifier.



```
String:  I am feeling very happy
--------------------------------+
Negative probability: 0.09836576018369099
Positive probability: 0.9016342398163092
```

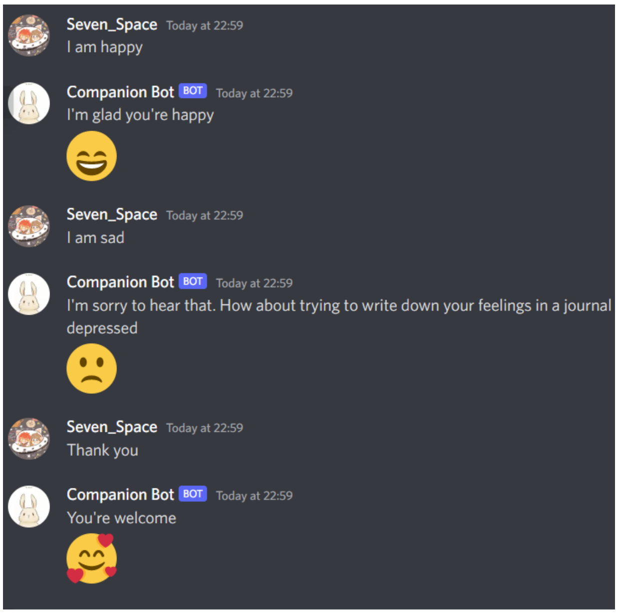
Figure 10 – classifier scoring a piece of text

4.2.5 Stopwords

I made a list of stopwords using the NLTK toolkit. I import a corpus of stopwords in English from the NLTK library. I set this as a stopwords list, which I use later in my bot to check for stopwords in the user’s response. These stopwords are considered filler words, which we want to filter out from the user’s input to have a more accurate prediction of the sentiment.

4.2.6 Emotion/emoji

I used the package NRCLex to help me detect the emotion of the user’s input. I create a NRCLex object with the NRCLex() function (*NRCLex 2020*). I can use the top_emotions() function from the library to return the top emotions detected from the user’s input. The library can class 10 different emotions and from this I picked out specific emotions to categorise into 3 categories, happy, sad and neutral. I used the emotions ‘positive’ and ‘joy’ to class happy and the emotions ‘negative’, ‘sadness’, and ‘fear’ to class sad. Using a for loop I would loop through the list of top emotions, and every time one of these emotions appeared I would use a count variable to keep track. At the end I compared the counts and whichever count is higher, determines if the user is happy or sad. Otherwise if they’re equal they have a neutral sentiment.



Seven_Space Today at 22:59
I am happy

Companion Bot BOT Today at 22:59
I'm glad you're happy
😄

Seven_Space Today at 22:59
I am sad

Companion Bot BOT Today at 22:59
I'm sorry to hear that. How about trying to write down your feelings in a journal
depressed
🙁

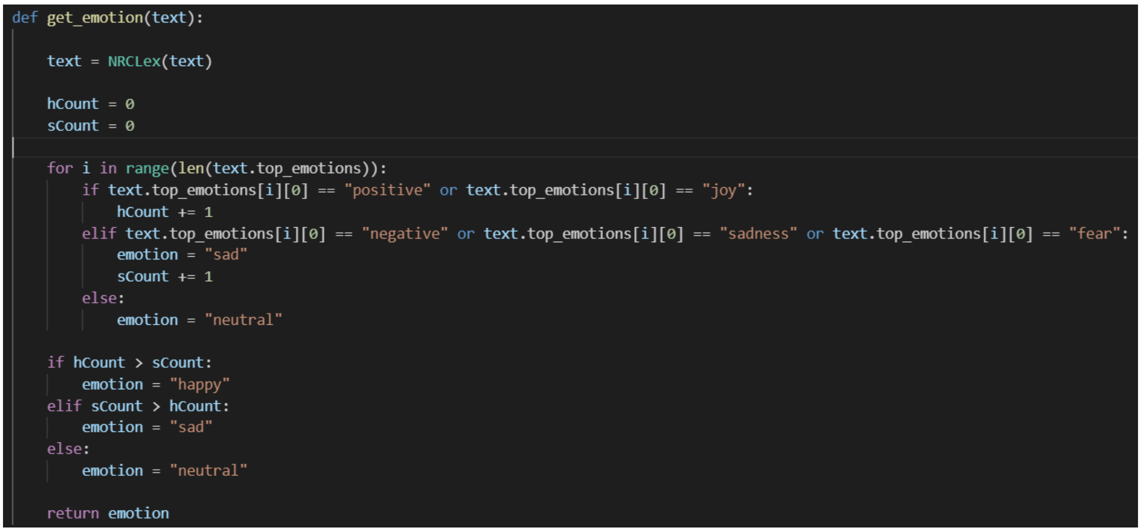
Seven_Space Today at 22:59
Thank you

Companion Bot BOT Today at 22:59
You're welcome
🥰

Figure 11 – Chatbot responding with emojis on Discord

I later combine this with my classifier, which returns a sentiment score. I do this to generate a more accurate reading of the user’s sentiment, which helps determine what type of emoji to send back to the user depending on their level of positivity or negativity.

Since you don't want to send an emoji for every user's response as it could be a neutral or extremely negative response. Hence why I use both as conditions to post an emoji. As for the chatbot sending an emoji, I made three separate files, which define a function with a list of emojis that are either sad, happy, or grateful (*Federicocotogno 2021*). This function returns a randomised emoji from these list when called so when the user is sad the chatbot can call this function to get a random sad emoji. Initially I had done this in my trainer, however I realised that once the model was trained the emoji it sends for a certain response was always the same since it has been set in the database, which I found was repetitive hence why I used this other method for a more random approach, and also because I felt it empathises with the user more.



```python
def get_emotion(text):

    text = NRCLex(text)

    hCount = 0
    sCount = 0

    for i in range(len(text.top_emotions)):
        if text.top_emotions[i][0] == "positive" or text.top_emotions[i][0] == "joy":
            hCount += 1
        elif text.top_emotions[i][0] == "negative" or text.top_emotions[i][0] == "sadness" or text.top_emotions[i][0] == "fear":
            emotion = "sad"
            sCount += 1
        else:
            emotion = "neutral"

    if hCount > sCount:
        emotion = "happy"
    elif sCount > hCount:
        emotion = "sad"
    else:
        emotion = "neutral"

    return emotion
```

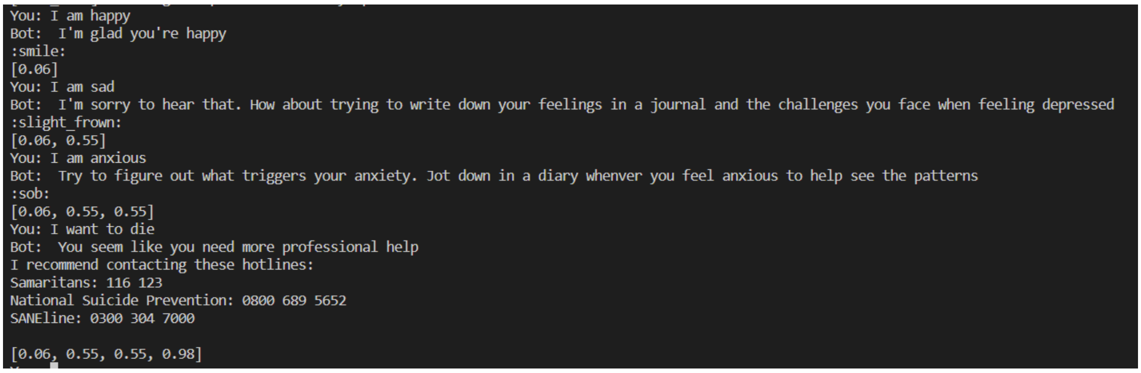
Figure 12 – Using NRCLex to categorise text to emotion

4.2.7 Chatbot

I make one file(bot2.py) where I call all the components to build the chatbot. I instantiate a Chatbot instance from the chatterbot library. From here I open and read the json file containing the training data for the chatbots' conversations and store it in an array. I have a function sentiment() which calls the classifier to check the sentiment on the user's input and rounds this to 2 decimal places before returning it. I have a function used to get the user's mood from the emotion algorithm I wrote, this is then passed to the emoji function. In the emoji function I call the emotion function to check the user's emotion. I have 3 different categories of emotions, ranging from happy, sad and a gratitude emoji for when the user thanks the bot. Should the user's input contain the word thank or thanks, using a check word function I made, the bot will return a gratitude emoji. If the user is detected as happy and the sentiment score is below 0.3 it will return a happy emoji. And if the user is sad and has a sentiment score between 0.55 and 0.85 it would return a random sad emoji. I had to use these conditions since sometimes the user would type something neutral and a happy or sad emoji would come up, which did not feel like the chatbot was really understanding the user's emotions. Also should the user type something extreme like "I want to die" the chatbot sent a sad emoji, which felt inappropriate at the time, hence why I changed it.

I then had a function simblebot(), which takes the user's input and turn it into as Textblob object. By changing it to a Textblob object I can now use the function words() to tokenize every single word in the user's input. I create a list of filtered words where any words not in the list of stop words are added to this filtered list. I can then check the training data and see if any of the words from the filtered list matches any of the training data and returns the response which best matches the user's input. Otherwise it does not recognise any of the responses correlating with the input it will ask the user to enter their input again or re-phrase it.

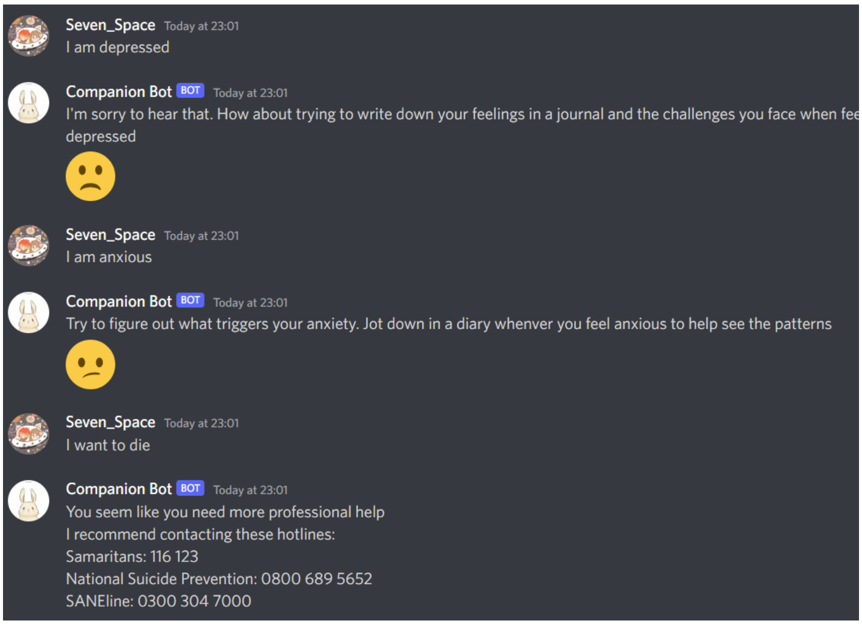
I can now call all the functions together to create the chatbot. In my escalation() function I append the user's input sentiment score in a list. I then use a variable to store the user's response gotten from the responseCheck() function. I check the last 3 sentiment score's in the list and if the score gets progressive higher(negative) the chatbot will respond by telling the user to seek more professional help and list a bunch of hotlines the user can call to get help, otherwise it will just return the chatbot's response from the training data. (*KikeVen 2021*)



```
You: I am happy
Bot:  I'm glad you're happy
:smile:
[0.06]
You: I am sad
Bot:  I'm sorry to hear that. How about trying to write down your feelings in a journal and the challenges you face when feeling depressed
:slight_frown:
[0.06, 0.55]
You: I am anxious
Bot:  Try to figure out what triggers your anxiety. Jot down in a diary whenver you feel anxious to help see the patterns
:sob:
[0.06, 0.55, 0.55]
You: I want to die
Bot:  You seem like you need more professional help
I recommend contacting these hotlines:
Samaritans: 116 123
National Suicide Prevention: 0800 689 5652
SANEline: 0300 304 7000

[0.06, 0.55, 0.55, 0.98]
```

Figure 13 – Using chatbot in visual studio, with sentiment score showing



Seven_Space Today at 23:01
I am depressed

Companion Bot BOT Today at 23:01
I'm sorry to hear that. How about trying to write down your feelings in a journal and the challenges you face when fee
depressed
🙁

Seven_Space Today at 23:01
I am anxious

Companion Bot BOT Today at 23:01
Try to figure out what triggers your anxiety. Jot down in a diary whenver you feel anxious to help see the patterns
😕

Seven_Space Today at 23:01
I want to die

Companion Bot BOT Today at 23:01
You seem like you need more professional help
I recommend contacting these hotlines:
Samaritans: 116 123
National Suicide Prevention: 0800 689 5652
SANEline: 0300 304 7000

Figure 14 – Chatbot working on Discord

4.2.8 Deployment

The way I deployed my chatbot was make a main file where I import the Discord module from Pycord and my chatbot file. This was relatively easy since I got help from the Pycord documentation and Youtube on how to set up the chatbot and the format it. I made client events which activate when the user performs an action. When the bot is run the backend will receive a message that it has logged on. The bot can also get the username of the user talking to the chatbot so if they said hello the chatbot would greet them back with their username in the message. I made a bunch of Discord commands with the prefix '!' for all my mental health descriptors ranging from depression, anxiety, stress, eating disorders, ADHD, and trauma as well as a bot information command to teach the user what the chatbot does and how to use it. If the user's messages does not match any of the commands it will be evaluated from the training set instead, where it will give a response depending if it is in the set or not as well as an emoji should it meet the right conditions for one. Whenever I run the main file the chatbot will go online on Discord and the user can interact with it.

4.3 Summary

I feel like my implementation of my chatbot Companion bot has been able to meet both the conditions of the functional and non-functional requirements I set in chapter 3. Using different libraries in combination with different methods of sentiment analysis and algorithms I was able to create a fully-functioning chatbot to meet the expectations I had set. There are a few things in the chatbot that could be improved but this will be discussed in my conclusion.

Chapter 5: **Testing**

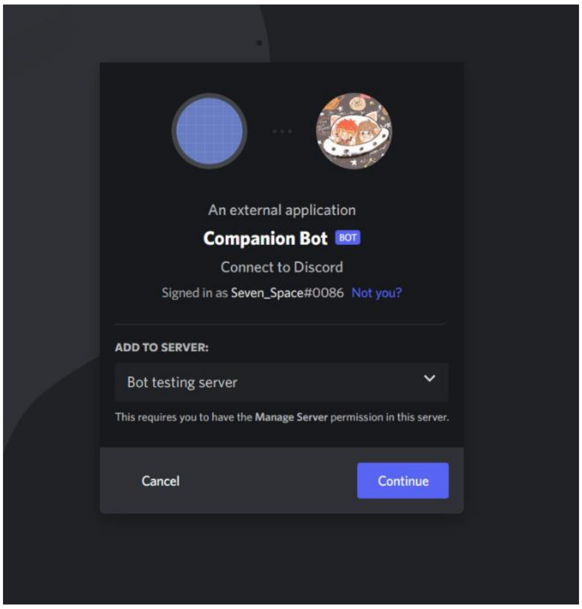
In this chapter I will talk about the tests I carried out to assess my chatbot and check the usability and functionality of the bot.

5.1 Unit Testing

Since my initial plan for user testing could not be done I decided to do unit testing and use this to help assess my chatbot's functionality.

5.1.2 Invite Discord chatbot

This test case checks that the chatbot can be added to any Discord server and is ready to use once added on the server. This can be done via the developer portal where you can get a URL link, which once when entered into the browser will prompt the user(if logged in) to allow the chatbot permissions so it can be added their designated server.

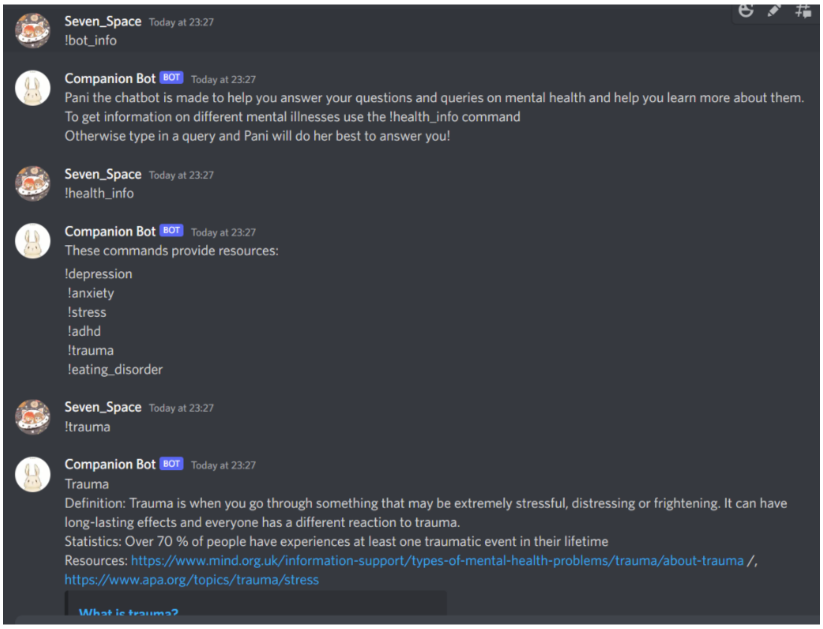

An external application
Companion Bot BOT
Connect to Discord
Signed in as **Seven_Space#0086** Not you?

ADD TO SERVER:
Bot testing server
This requires you to have the **Manage Server** permission in this server.

Figure 15 – Inviting chatbot on my Discord server

5.1.3 Discord Commands

This test case checks all the Discord commands I set work. Using a prefix of '!' I also set all my commands to lowercase, since the user's input is turned all into lowercase. With this whenever the user enters the command correctly into Discord, the chatbot will respond with the response I set for that set command.


Seven_Space Today at 23:27
!bot_info

Companion Bot BOT Today at 23:27
Pani the chatbot is made to help you answer your questions and queries on mental health and help you learn more about them.
To get information on different mental illnesses use the !health_info command
Otherwise type in a query and Pani will do her best to answer you!

Seven_Space Today at 23:27
!health_info

Companion Bot BOT Today at 23:27
These commands provide resources:
!depression
!anxiety
!stress
!adhd
!trauma
!eating_disorder

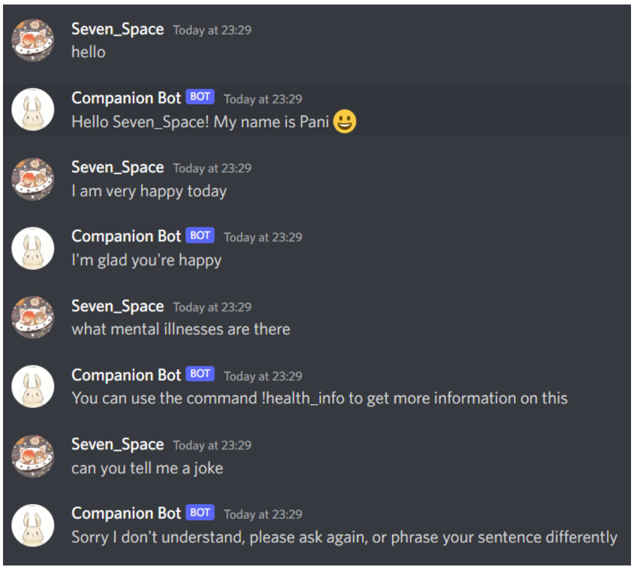
Seven_Space Today at 23:27
!trauma

Companion Bot BOT Today at 23:27
Trauma
Definition: Trauma is when you go through something that may be extremely stressful, distressing or frightening. It can have long-lasting effects and everyone has a different reaction to trauma.
Statistics: Over 70 % of people have experiences at least one traumatic event in their lifetime
Resources: https://www.mind.org.uk/information-support/types-of-mental-health-problems/trauma/about-trauma /, https://www.apa.org/topics/trauma/stress
What is trauma?

Figure 16 – Testing chatbot commands

5.1.4 User messaging (Response handling)

This test case, ensures the chatbot can respond to the user appropriately and sends a response correlating to what the user inputted. The chatbot will take the user's input and check this against the response set it has been trained with. Should the response exist it will return that response, else it will ask the user to re-enter their input or rephrase their input. This I felt was mostly successful, but could be improved, since it cannot respond to everything the user requires however it was able to respond how I expected the chatbot to respond majority of the time.



Seven_Space Today at 23:29
hello

Companion Bot BOT Today at 23:29
Hello Seven_Space! My name is Pani 😃

Seven_Space Today at 23:29
I am very happy today

Companion Bot BOT Today at 23:29
I'm glad you're happy

Seven_Space Today at 23:29
what mental illnesses are there

Companion Bot BOT Today at 23:29
You can use the command !health_info to get more information on this

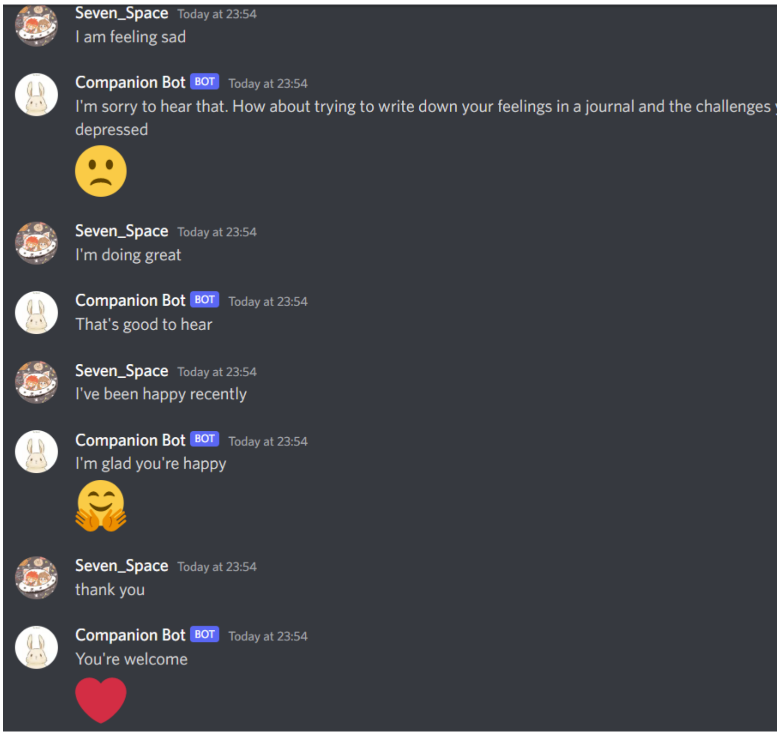
Seven_Space Today at 23:29
can you tell me a joke

Companion Bot BOT Today at 23:29
Sorry I don't understand, please ask again, or phrase your sentence differently

Figure 17 – Example conversation

5.1.5 User messaging (Emojis and sentiment analysis)

This test case checks the sentiment analysis part of my program. My chatbot will send an emoji when it detects a certain emotion from the user and if the sentiment score is in the right threshold. For example a happy sentiment with a score less than 0.3 would equate to the chatbot sending a random happy emoji. I felt the thresholds I set was very accurate to when it was appropriate to send an emoji to the user.



Seven_Space Today at 23:54
I am feeling sad

Companion Bot BOT Today at 23:54
I'm sorry to hear that. How about trying to write down your feelings in a journal and the challenges
depressed
🙁

Seven_Space Today at 23:54
I'm doing great

Companion Bot BOT Today at 23:54
That's good to hear

Seven_Space Today at 23:54
I've been happy recently

Companion Bot BOT Today at 23:54
I'm glad you're happy
🤗

Seven_Space Today at 23:54
thank you

Companion Bot BOT Today at 23:54
You're welcome
❤️

Figure 18 – Chatbot responding with emojis

5.2 Summary of Unit Testing

To summarise I tested the main components of my chatbot which I felt were most important. The test table can be seen in appendix B, where all the test cases resulted in success. Hence I feel like this deems my project as fully functioning.

Chapter 6: **Evaluation**

These are the findings I carried out which analyses my chatbot and reflects any strengths and weaknesses the chatbot has. It summarises the key findings of my chatbot.

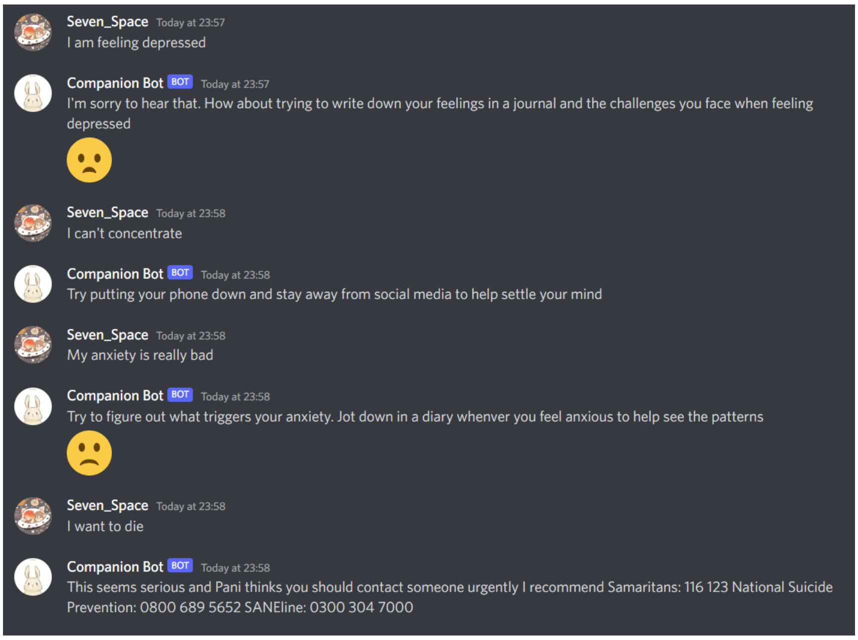
6.1 Results

From the development of my chatbot most of the milestones I had planned had been met. The only thing I was not able to do was my planned user testing with a questionnaire as feedback. This is due to misinformation on the ethics form submission time frame. It was to my knowledge that an ethics form would take around 4 weeks to process and be approved from what I had read online. However, from consulting other colleagues who had submitted the form, they had either yet to receive approval yet or it took around 3-5 months to get approval. I had been ready to submit my form around March, however from learning of this I realised the likelihood of me getting the approval in time would be unlikely, hence I had to change the way I wanted to do my testing and evaluation of my project. Asides from that I was able to meet all the goals I had set for the implementation of my chatbot.

When evaluating my chatbot I felt that majority of the time it was able to communicate well with the user and was pretty accurate in its responses. Although the chatbot worked well it still had its shortcomings which could be improved on and would have helped answer my research questions better.

6.1.1 Does the use of emotion in a chatbot make it more useful?

The sentiment analysis in my chatbot was able to pick up the user's sentiment. I used my classifier which used Naïve Bayes classification and the NRCLex library and together I combined this to get an accurate reading of the user's emotions. From this I could class whether the user was happy, sad or neutral and would send an appropriate emoji. Since my chatbot is aimed at a younger audience I felt the use of the emoji would help better show the chatbots understanding to their emotion better, as well as if the user typed something happy the chatbot would be happy for them too and validate them. Hence I felt like using emotion in a mental health chatbot did make it useful, in the way it could relate to the user and understand the user.



Seven_Space Today at 23:57
I am feeling depressed

Companion Bot BOT Today at 23:57
I'm sorry to hear that. How about trying to write down your feelings in a journal and the challenges you face when feeling depressed
😦

Seven_Space Today at 23:58
I can't concentrate

Companion Bot BOT Today at 23:58
Try putting your phone down and stay away from social media to help settle your mind

Seven_Space Today at 23:58
My anxiety is really bad

Companion Bot BOT Today at 23:58
Try to figure out what triggers your anxiety. Jot down in a diary whenver you feel anxious to help see the patterns
🙁

Seven_Space Today at 23:58
I want to die

Companion Bot BOT Today at 23:58
This seems serious and Pani thinks you should contact someone urgently I recommend Samaritans: 116 123 National Suicide Prevention: 0800 689 5652 SANEline: 0300 304 7000

Figure 19 – Conversation example

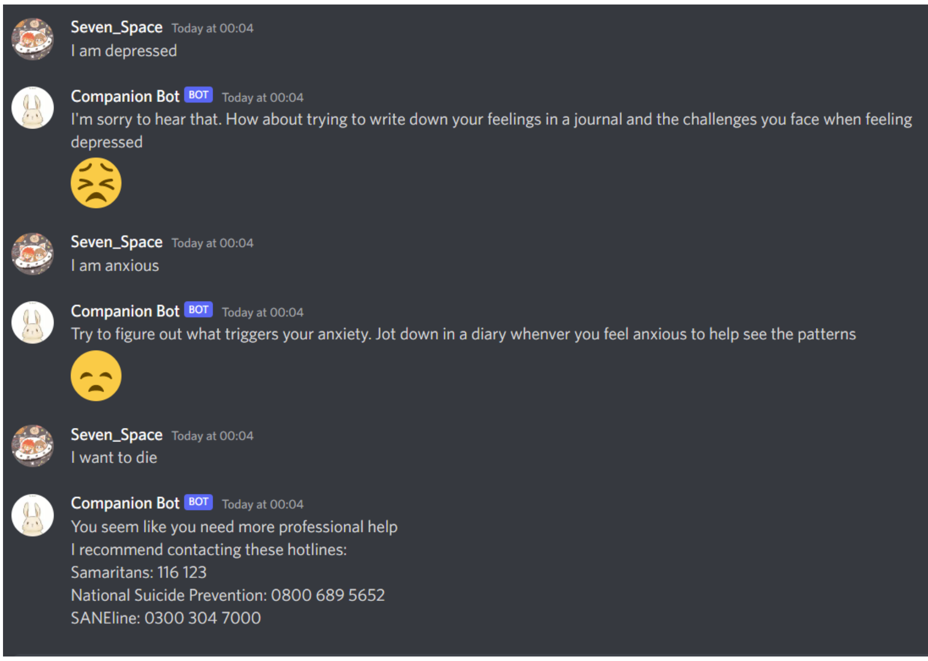
6.1.2 How well does sentiment analysis capture the user’s intended sentiment?

I felt like the way the sentiment analysis captured the user’s intended sentiment was pretty accurate, especially when it replied using an emoji. Although I felt that I could have combined this better with the chatbot’s responses to better show it understood the user’s sentiment. Being able to control when and what type of emoji the chatbot sent showed how it knew when the user was happy, sad, neutral. If I combined this with a set of chatbot responses too I felt this could have shown how much better it understood the user’s sentiment. Nonetheless the chatbot was able to capture some sentiment from the user and reply accordingly most of the time.

6.1.3 Does the chatbot responses show that it empathises with the user?

Similar to my other two answers I felt that the chatbot was able to do this to a certain extent. With the addition of an emoji I felt that since the chatbot is intended for a younger audience. The use of the emoji is able to empathise and show the user the chatbot understands how they’re feeling. The control over what kind of emoji to send and when is also taken into consideration. So the chatbot knows not to send an emoji in extreme scenarios.

I also had a way to measure the last 3 sentiments of the user and should they exceedingly grow towards 1(negative) then the chatbot knows the user is distressed and would send emergency hotlines for the user to contact. In regards to the chatbots responses I felt some of the responses did empathise with the user but some not so much and could be improved. I felt it was more empathetic when the user was in a positive mood than a negative mood, this could be to do with the responses I set in the response set.



Seven_Space Today at 00:04
I am depressed

Companion Bot BOT Today at 00:04
I'm sorry to hear that. How about trying to write down your feelings in a journal and the challenges you face when feeling depressed
😣

Seven_Space Today at 00:04
I am anxious

Companion Bot BOT Today at 00:04
Try to figure out what triggers your anxiety. Jot down in a diary whenver you feel anxious to help see the patterns
😞

Seven_Space Today at 00:04
I want to die

Companion Bot BOT Today at 00:04
You seem like you need more professional help
I recommend contacting these hotlines:
Samaritans: 116 123
National Suicide Prevention: 0800 689 5652
SANEline: 0300 304 7000

Figure 20 – Escalation example

6.2 Heuristic Evaluation

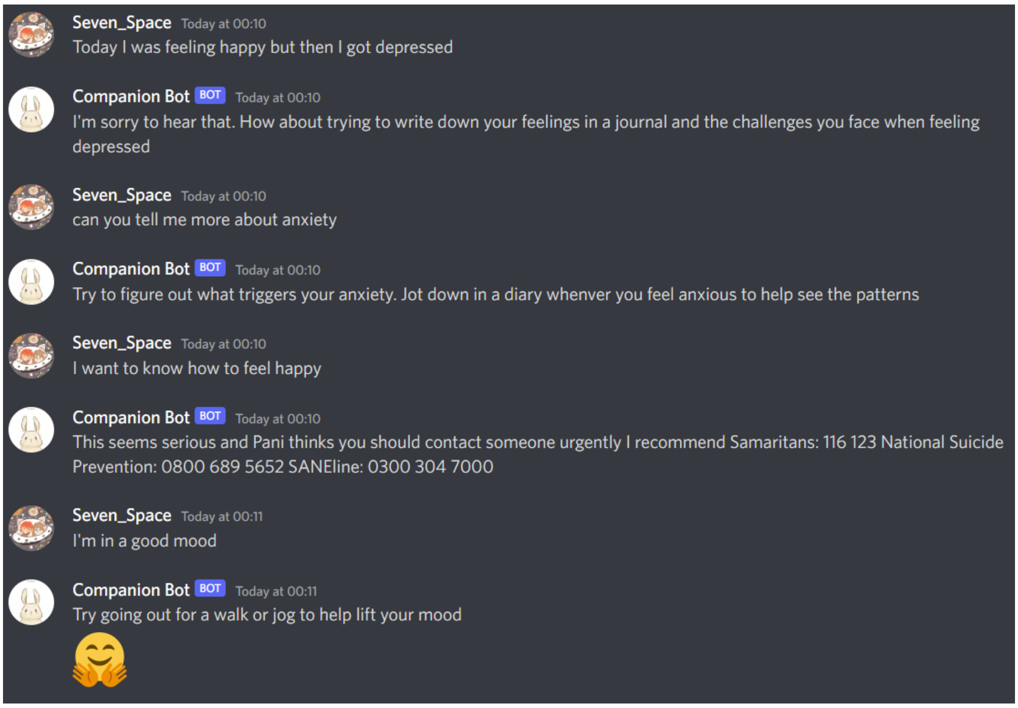
From my heuristic evaluation (Appendix C) my chatbot was able to score low scores in the heuristic evaluation. This shows that my chatbot is fully functional, and though there may be some faults they're not too major to the point that my chatbot is not usable. I would say my chatbot is easy to use and it is clear what the chatbot does.

6.3 Limitations

With my chatbot able to achieve accomplish the goals I had set it, I did feel as if it could have done some things better. One limitation is the amount of datasets and APIs based on mental health specific to what I wanted to do, did not exist or were incomplete. Hence I ended up writing my own response set and having to program a way for this to work with my chatbot instead of how I initially wanted to get an API to pull mental health advice for the user. This did end up taking more time, since I wanted the advice my chatbot gave to be accurate and used numerous sources like, NHS, Mind.org, mentalhealth.org etc, to make sure the advice I was giving was correct since this is also affecting the user's mental wellbeing.

Since I made the response set this makes the responses the chatbot gave limited. So should the user ask about anything unrelated to mental health the chatbot would not have a response for that and would just ask the user to re-phrase or enter what they typed again. This is quite a major limitation, which I tried to overcome by using other various methods of creating a response set for the chatbot but with my limited time and knowledge I don't think I would have been able to achieve this. Also at times the chatbot would get confused and give a wrong response, like in figure 12.

Another limitation was the testing part of the chatbot. Since I was unable to get my ethics form approved, the only testing that could be done was by me. This could end up giving a bias outcome of how well my chatbot really works. And without a second opinion I could be missing out on many things wrong with my chatbot which as the developer I may not realise. Hence, how well the chatbot works should be taken with caution since I was the only sole tester for it.



Seven_Space Today at 00:10
Today I was feeling happy but then I got depressed

Companion Bot BOT Today at 00:10
I'm sorry to hear that. How about trying to write down your feelings in a journal and the challenges you face when feeling depressed

Seven_Space Today at 00:10
can you tell me more about anxiety

Companion Bot BOT Today at 00:10
Try to figure out what triggers your anxiety. Jot down in a diary whenver you feel anxious to help see the patterns

Seven_Space Today at 00:10
I want to know how to feel happy

Companion Bot BOT Today at 00:10
This seems serious and Pani thinks you should contact someone urgently I recommend Samaritans: 116 123 National Suicide Prevention: 0800 689 5652 SANEline: 0300 304 7000

Seven_Space Today at 00:11
I'm in a good mood

Companion Bot BOT Today at 00:11
Try going out for a walk or jog to help lift your mood
🤗

Figure 21 – Chatbot with inaccurate responses

6.4 Summary

Overall the aim of my chatbot is to help raise awareness of mental health and reply to the user accordingly based on the sentiment. Which I would say has been achieved even though it could be improved. However, there are still some shortcomings of the chatbot which could be fixed, had I implemented certain aspects of the chatbot a certain way or even had more time to look up how to do certain things, like fixing the response of the chatbot.

Chapter 7: **Conclusion**

Here I conclude the results of my project, how well I felt the process went, challenges I faced and what I would do next time.

7.1 Achievements

Overall I felt my project was successful and I am proud of what I have achieved. I was able to reach the aims I stated in Chapter 1, as well as the requirements I set in Chapter 3. Developing a fully functional chatbot on my own was an achievement in itself, since I have only done development of similar scale in a group project before. So, this meant I had to be more organised and disciplined on setting myself deadlines and reaching them.

One thing that was quite major was how much I learnt about natural language processing techniques throughout my project and how often it is used in the industry. This can be seen in Chapter 2 where I did my literature review. This was the first time I had ever working with NLP so I had to do a lot of research of how to implement and the theoretically knowledge behind how it worked too. Although I went through many challenges and lots of trial and error I was able to successfully implement my chatbot with NLP techniques.

I felt that from this project I was able to learn how important timing is, since you never know what kind of errors may occur especially during the implementation stage. I was able to further my Python skills, and gain greater interest in natural language processing. Being able to implement such a big project by myself really has given me confidence in my skills as a developer which I think will help me greatly in the future.

7.2 Challenges I faced

Since this was my first time doing anything NLP related, I encountered many challenges and had to adapt and overcome them during the process of my project. One way I did this was by doing extensive research to learn more about NLP, sentiment analysis and the different techniques involved.

7.2.1 Wrong Model

I started off by building a deep learning neural network model. I did this since I looked up many tutorials and documents on building a chatbot and most of them used this method. I was in the process of training a classifier on this model however, I encountered a problem with Pytorch using my GPU instead of my CPU. I was unable to resolve this issue and so I tried to look for an alternative solution to building my chatbot and thus realising I had overcomplicated the entire process and ended up starting from scratch. This ended up being time costly, thus another reason why I am still in the implementation stage. I think I should have researched a bit more on the implementation of sentiment analysis rather than jump straight into it since this ended up taking more time than I had planned.

7.2.2 General Errors

I encountered many general errors like making components work with each other, using Textblob functions correctly and more. One of the errors I ended up struggling with was the trainer. I did not completely understand the concept of the trainer and ended up playing around with the code to realise how training worked in Textblob. I did not realise every time I ran the trainer the json dataset I exported would append the trained data again and again. This ended up with my chatbot using old responses even when I had changed the responses in the trainer. I later realised since I was appending data to the json file everytime I ran it I needed to drop the dataset if I made any changes, which also led me to the issue I am having with randomising the emojis that I discussed earlier.

7.2.3 Response handling

This was one of the greatest challenges I faced during the development of my chatbot since I could not get the chatbot to respond how I wanted to and I tried numerous methods which was very time consuming to fix this. I initially wanted to use an API to generate responses to the user, however I felt like the APIs I found were not very accurate to responding to the user or giving the type of response I wanted to, furthermore some APIs did not have much documentation and I struggled to use them since there was not much material out there to teach me how to use them. One method I tried was importing the built in NLTK corpus and training my chatbot to this. This became a major issue, since the NLTK corpus does not have much information on mental health as well as the chatbot would send demeaning responses to the user which was not appropriate at most times. An example of this would be the user sending ‘I am depressed’ and the chatbot responding ‘is this not a common human emotion’. Hence I had to try another method, in the end I ended up just writing my own responses and trying to use keywords so when the words are detected the chatbot would know which response to respond with. This however is limited to certain responses and I feel like it was time consuming in itself to generate the different responses making the entire process longer than it should have been as well as more difficult.

7.2.4 Ethics Form

As mentioned in my results in the evaluation(Chapter 6) I wasn’t able to get my ethics form approved in time. Although the time it takes to get the form approved is relatively long, I think I should have submitted the form in first semester, so I did not have to stress about it later on and could have been guaranteed approval for my user testing. I was too reliant on the set time that was stated online and had not considered that it would take longer than stated. Therefore, I was unable to do any user testing and was only able to test and evaluate my project myself, skewering the results of my evaluation and testing slightly, since I know how to use my chatbot already.

7.3 Future Improvements

I feel as though my project definitely has much room for improvement and it would have helped if I was able to get some user testing involved, hence next time I will submit any ethics approval much early and take note that the estimated processing time is not always correct.

Another thing I definitely want to improve is the chatbot's response handling. I would want to make it so it has a wider range of responses, where it's responses are able to empathise with the chatbot more than it does now. One way I think I could do this is by using a neural network model and possibly making a prediction as to what the chatbot should respond based on a larger set of training data of conversations.

Instead of classifying only happy, sad and neutral emotions I think I could try to classify more emotions from the user and could do this by making the classifier tag certain words to certain emotions. I feel like if the chatbot can detect more emotions this would also be able to help how the chatbot responds to the user and could also allow me to add more emojis based on the different emotions.

7.4 Summary

From the completion of my project I believe that my project has been successful and able to prove a chatbot which can empathise with the user as well as detect their emotion is more useful to an extent, depending on what the application is intended for. In my case talking to younger people about mental health and raising awareness in a positive light is something I believe benefits from knowing how the user feels. Since my chatbot is supposed to give advice and direct the user to appropriate sources of help, giving them the comfort of knowing there are people out there to help and what they feel is completely normal.

From my background research there are many techniques to implement NLP and sentiment analysis and I think it would be interesting to explore them more later in my own time, so that my chatbot can be more accurate or helpful for the user.

I was also able to overcome many obstacles and adapt to the situation when I was running low on time, or not having an approved ethics form whilst still achieving my main goals. This has allowed me to realise how to solve a problem efficiently and what to do should I ever fun into such challenges again. With my aim and requirements achieved from my project I believe my project is a success, and hopefully I may even expand on the project later in my spare time.

References

1. Mei, C., Fitzsimons, J., Allen, N., Alvarez-Jimenez, M., Amminger, G.P., Browne, V., Cannon, M., Davis, M., Dooley, B., Hickie, I.B., Iyer, S., Killackey, E., Malla, A., Manion, I., Mathias, S., Pennell, K., Purcell, R., Rickwood, D., Singh, S.P., Wood, S.J., Yung, A. and McGorry, P.D. (2020), Global research priorities for youth mental health. Early Intervention in Psychiatry, 14: 3-13. https://doi.org/10.1111/eip.12878
2. Orlowski, S., Lawn, S., Matthews, B. *et al.* The promise and the reality: a mental health workforce perspective on technology-enhanced youth mental health service delivery. *BMC Health Serv Res* **16,** 562 (2016). https://doi.org/10.1186/s12913-016-1790-y
3. Aleedy, Moneerh. *International Journal of Advanced Computer Science and Applications* Volume: 10 Issue 9 (2019) ISSN: 2158-107X Online ISSN: 2156-5570
4. Michelle O'Reilly, Nisha Dogra, Jason Hughes, Paul Reilly, Riya George, Natasha Whiteman, Potential of social media in promoting mental health in adolescents, *Health Promotion International*, Volume 34, Issue 5, October 2019, Pages 981–991, https://doi.org/10.1093/heapro/day056
5. Grové C (2021) Co-developing a Mental Health and Wellbeing Chatbot With and for Young People. *Front. Psychiatry* 11:606041. doi: 10.3389/fpsyt.2020.606041
6. Springer Nature Switzerland AG 2019 S. S. Bodrunova et al. (Eds.): INSCI 2018 Workshops, LNCS 11551, pp. 121–13 2, 2019. https://doi.org/10.1007/978-3 -030-17705-8_11
7. Nivethan, Sankar S. (2020) Sentiment Analysis and Deep Learning Based Chatbot for User Feedback. In: Balaji S., Rocha Á., Chung YN. (eds) Intelligent Communication Technologies and Virtual Mobile Networks. ICICV 2019. Lecture Notes on Data Engineering and Communications Technologies, vol 33. Springer, Cham. https://doi.org/10.1007/978-3-030-28364-3_22
8. A. Ghandeharioun, D. McDuff, M. Czerwinski and K. Rowan, "EMMA: An Emotion-Aware Wellbeing Chatbot," *2019 8th International Conference on Affective Computing and Intelligent Interaction (ACII)*, 2019, pp. 1-7, doi: 10.1109/ACII.2019.8925455.
9. Janet Siroto on "Can Replika really help easy anxiety?" https://www.goodhousekeeping.com/health/wellness/a34975893/replika-app-review/ 2020 – accessed [03/10/21]
10. Colleen Stiles-Shields "Woebot: A professional review" https://onemindpsyberguide.org/expert-review/woebot-an-expert-review/ 2019
11. Wysa FAQs https://www.wysa.io/faq 2021 – accessed [03/10/21]
12. Eric Patterson, Maloa Affuembey "Youper app review" https://www.choosingtherapy.com/youper-app-review/ 2020 – accessed [03/10/21]
13. KikeVen Simplebot sentiment analysis chatbot Github 2021 https://github.com/KikeVen/simplebot - accessed [09/02/22]
14. Federicacotogno Text recognition chatbot Github 2021 https://github.com/federicocotogno/text_recogniton_chat - accessed [14/04/22]

15. NRCLex Documentation released 2020 - https://pypi.org/project/NRCLex/ - accessed [20/04/22]
16. Mind mental health charity https://www.mind.org.uk/information-support/a-z-mental-health/ 2022 – accessed [09/02/22]

Appendix A – Risk Assessment

Description of Risk	Description of Impact	Likelihood Rating	Impact Rating	Preventative actions
Recognition of emotion fails or inaccurate	The chatbot does not register the emotion of the user or recognises the wrong emotion and may respond wrong	Medium	Medium	Ensure I train the chatbot model rigorously and test the bot for accuracy before I release it to the public
Unsupported API for Discord	The API I am using may not be supported on Discord, thus not allowing me to upload my bot and code	Medium	High	Ensure I find other APIs available to use as well as checking whether Discord supports these APIs
Discord update	Discord may update how APIs work on their platform thus leading to my code being outdated	Medium	Low	Ensure I stay up to date with any Discord updates and check if the changes will affect my bot and if so be ready to make these changes
Chatbot is unresponsive	Leading to the chatbot not working and unable to receive any working feedback on the chatbot	Low	High	Test my chatbot before release to ensure this problem does not occur. Possibly get a friend to test a BETA version before its release.
Incorrect implementation	Can lead to the chatbot being faulty or not working at all	Medium	High	Research the methods I wish to use to create my chatbot thoroughly and check to see if it is possible. Spend enough time to learn how to use these methods in implementation.
Loss of work	Could lead to spending more time recovering the lost work and having to cut parts of my project out	Low	High	Back up my work consistently either on cloud or work on GitHub, so I am able to recover lost work

Poor time management	Could lead to me being unable to meet deadlines or implement/finish parts of my project	Medium	High	Manage my time by creating a schedule or a plan for when I should be working on what for my project. Make sure to set aside time for my project and not just focus on my other modules.
Personal emergency/illness	Could lead to delay in project deadlines/milestones	Low	Medium	If the issue is severe speak to my supervisor and submit ECS form is needed. Make sure I complete the major parts of my project first in case of such emergencies.
Loss of test data/ insufficient testing	Could lead to faults in my chatbot	Low	Medium	Allow myself enough time to test the chatbot before releasing it to public and make sure any test data/results is backed up.

Appendix B – Unit Testing

Feature	Test Case	Method	Expected Output	Success
Invite Discord bot	User enters correct Discord bot URL and is signed in	https://discordapp.com/api/oauth2/authorize?client_id=933480571437449266&permissions=8&scope=bot	Chatbot should be added onto the server the user selected, once user has accepted the permissions	Yes
Discord commands	User correctly enters command for the chatbot information	!bot_info	Chatbot responds with explanation of what the chatbot does and any commands the user can use	Yes
Discord commands	User correctly enters command for the mental health descriptors	!health_info	Chatbot responds with a list of commands to get descriptors about certain mental health issues	Yes
Discord commands	User correctly enters command for the depression descriptor	!depression	Chatbot responds with a description and resources for depression	Yes
Discord commands	User correctly enters command for the anxiety descriptor	!anxiety	Chatbot responds with a description and resources for anxiety	Yes
Discord commands	User correctly enters command for the stress descriptor	!stress	Chatbot responds with a description and resources for stress	Yes
Discord commands	User correctly enters command for the eating disorders descriptor	!eatingdis	Chatbot responds with a description and resources for eating disorders	Yes
Discord commands	User correctly enters command for the ADHD descriptor	!adhd	Chatbot responds with a description and resources for ADHD	Yes
Discord commands	User correctly enters command for the trauma descriptor	!trauma	Chatbot responds with a description and resources for trauma	Yes

Discord commands	User incorrectly wrong/misspelled/capital lettered Discord command	main.py if else statement	The chatbot will ask the user to re-enter their input or rephrase it	Yes

User messaging	User sends a happy message	classifier
		responseCheck()
		emoji_res()
		emote()
			Chatbot will respond with a affirmation message along with a happy emoji	Yes

User messaging	User sends a neutral message	classifier
		responseCheck()
		emoji_res()
		emote()
			Chatbot will response with a generic message and no emoji	Yes

User messaging	User sends a sad message	classifier
		responseCheck()
		emoji_res()
		emote()
			Chatbot will respond with some advice and a sad emoji	Yes

User messaging	User sends extremely distressed message	classifier responseCheck()	Chatbot will send urgent message and helpful hotlines	Yes
User messaging	User sends 3 consecutive distressing messages	Escalation()	Chatbot will respond advising the user to get more professional help and some hotlines to help them contact help	Yes
User messaging	User enters non-coherent sentence	responseCheck()	The chatbot will ask the user to re-enter their input or rephrase it	Yes

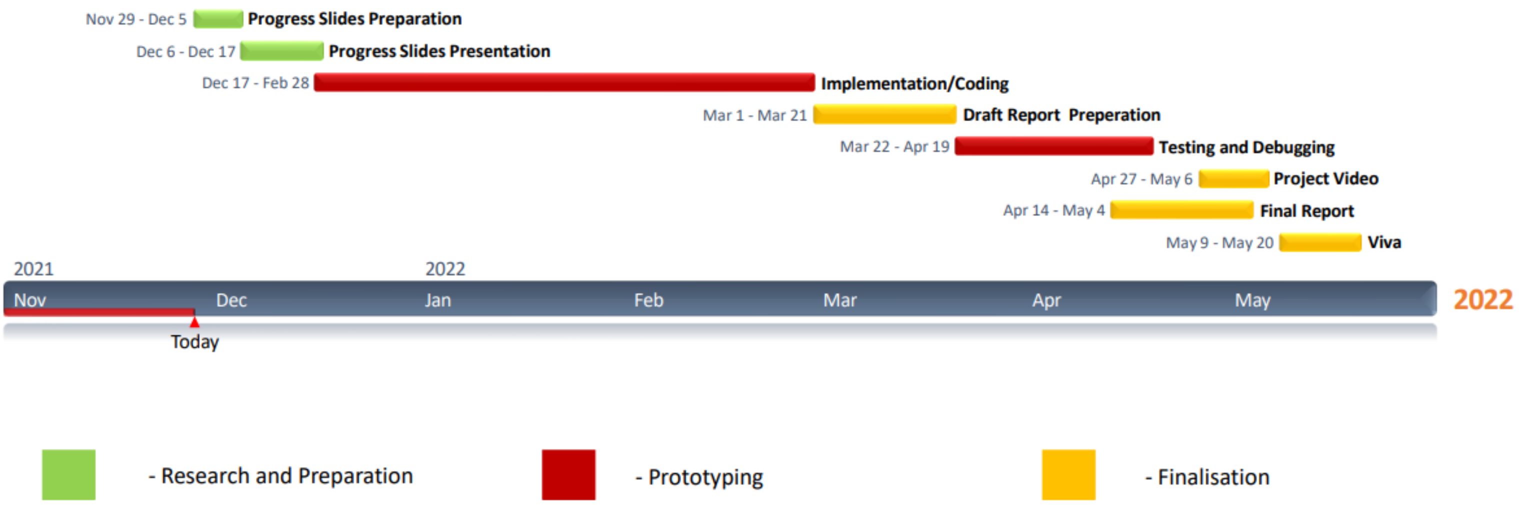
Appendix C – Heuristics Evaluation

Heuristic	Severity	Justification
Visibility of system status	0	Since the GUI design is provided by Discord the visibility relies on Discords app. However, the app does display when the chatbot is online and offline so the user knows when they can use the chatbot. If Discord itself is down, Discord has their own website to check the status of their app in certain regions.
Match between system and real world	1	The chatbot uses basic phrases to communicate with the user. Should it not understand the user's input it does tell the user to just re-enter/re-phrase their input. There are times when the chatbot may send the wrong response which can be improved on.
User control and freedom	1	User's can enter a message to the chatbot and stop whenever they want. However, they cannot cancel a message once it has been sent, since the chatbot will process the response in a second it is impossible to cancel the message in time. Although most chatbots do not have a cancel function, so this does seem standard.
Consistency and standards	0	Since Discord does most of the formatting for me the chatbot follows the principles and design of how any chatbot would look like, as well as the text formatting. Where the text appears and where the user types their input is the same as they would do on any other Discord server to a chatbot.
Error prevention	1	Although there are a few error prevention methods, I think I could have done more to prevent them. An instance of this is if the user enters an emoji, the chatbot will not understand this and give a default "I do not understand" message.
Recognition rather than recall	0	The interface makes it so everything is all on one 'page'. Should the user forget information on the chatbot and what it does, they can hover over the bot icon on the same page to get more information on the chatbot.
Flexibility and efficiency of use	1	The use of a chatbot on Discord is geared mainly to users who have experience with Discord. It may be a bit difficult to set up if you're

		unfamiliar with Discord, however there are many tutorials and Discord guides on there to help the user should they get stuck. In regards to my chatbot, it is relatively easy to use and comes with instructions and all the user has to do to interact with my chatbot is type a message.
Aesthetic and minimalist design	0	As mentioned before, the interface is provide by Discord and is relatively easy to use, where you can see the status of the chatbot on the right hand side as well as a big chat box in the middle of the page
Help users recognise and recover from errors	1	If the chatbot doesn't understand a user input it will ask the user to re-enter/re-phrase their input. However, should the user talk about a topic not in the response set it will give the same response, so there is not really a definite way to fix this unless I add more of a range of topics in the response set.
Help and documentation	1	There are numerous Discord tutorials on how to use Discord bots and can be easily found online. My chatbot has a command !bot_info on a general basis of what the chatbot does and commands it has. This could be a bit more descriptive with a list of all the commands and specify that it only answers questions on mental health.

Appendix D – Project Plan

Project Timeline


Nov 29 - Dec 5 Progress Slides Preparation
Dec 6 - Dec 17 Progress Slides Presentation
Dec 17 - Feb 28 Implementation/Coding
Mar 1 - Mar 21 Draft Report Preperation
Mar 22 - Apr 19 Testing and Debugging
Apr 27 - May 6 Project Video
Apr 14 - May 4 Final Report
May 9 - May 20 Viva
Nov
Dec
Jan
Feb
Mar
Apr
May
Today
- Research and Preparation
- Prototyping
- Finalisation