Mental HealthCare Application

A project submitted in partial fulfillment of the

requirements for the award of the degree of

Bachelor of Technology

In

Information Technology

Submitted by: Abhijit Mishra

Roll number:12012047

Group Number: - 120

Supervised by:

Dr. MUKESH MANN

Assistant professor

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY,
SONEPAT – 131201, HARYANA, INDIA

Acknowledgment

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of people whose ceaseless cooperation made it possible, whose constant guidance and encouragement crown all efforts with success. I would like to express my greatest appreciation to Dr. Mukesh Mann Assistant Professor (IT), and IIIT Sonepat for giving me the opportunity to undertake this project. He has helped and supported me throughout the project.

I would also like to thank my classmates who helped me in completing the project by exchanging interesting ideas and sharing their experiences.

I wish to thank my parents as well for their undivided support and interest who inspired me and encouraged me to go my own way, without whom I would be unable to complete my project.

Abhijit Mishra

SELF DECLARATION

I hereby declare that the work contained in the project file titled "**Mental HealthCare Application**" is original. I have followed the standards of research/project ethics to the best of my abilities. I have acknowledged all sources of information that I have used in the project.

Name: Abhijit Mishra

Roll No.: 12012047

Department of Information Technology,

Indian Institute of Information Technology,

Sonepat-131201, Haryana, India.

CERTIFICATE

This is to certify that Mr. Abhijit Mishra has worked on the project entitled “Mental HealthCare Application” under my supervision and guidance.

The contents of the project, being submitted to the Department of Information Technology, IIIT SONEPAT, HARYANA, for the award of the degree of B.Tech in Information Technology, are original and carried out by the candidate himself. This project has not been submitted in full or part for award of any other degree or diploma to this or any other university.

Dr. Mukesh Mann

Supervisor

Department of Information Technology,

Indian Institute of Information technology,

Sonepat-131201, Haryana, India

Abstract

Name of the student: **Abhijit Mishra**

Roll No.: **12012047**

Degree for which submitted: **B.Tech (IT)**

Department of **Information Technology, IIIT Sonepat.**

Project Title: **Mental HealthCare Application**

Name of the thesis supervisor: **Dr. Mukesh Mann**

Month and year of the thesis submission: **December 2021**

Mental illness is the most neglected health problem in the developing world. Mental disabilities are pervasive, affecting approximately 8% of the world's population. Additionally, mental health is perceived as a luxury good but if left untreated mental illness can lead to other commodity disorders, such as depression, substance abuse, and even early mortality.

My project Mental HealthCare App is an android application that provides psychoeducation and counseling regarding a wide range of problems including depression, anxiety, trauma, and relationship dissolutions with the help of an interactive chatbot. It also recommends recreational activities for users based on their field of interest. This helps in tackling loneliness and improving impaired social relationships. The application also provides the user with an option to speak to a licensed therapist which directs the user to an online counseling website (https://www.7cups.com/online-therapy/).

The programming languages and frameworks used till now are python (for the chatbot), kotlin(for android app development), and chaquopy (for integrating and running python code on android).

LIST OF ABBREVIATIONS

S/W	Software
UML	Unified Modeling Language
JSON	JavaScript Object Notation
API	Application Programming Interface
UI	User Interface

LIST OF FIGURES

TABLE OF CONTENTS

Chapter 1

Introduction

1.1 INTRODUCTION

The project that I have undertaken is a Mental HealthCare Application that aims to provide moral support and share useful information with people undergoing stress, anxiety, and depression. The application has a chatbot system that responds to a wide range of problems related to psychological well-being.

It also aims to address the feeling of isolation and impaired social relationships by suggesting recreational activities to the user. It also gives the user a range of fields of interest to decide what they feel like doing.

The motive of choosing this topic as my practicum project is the fact that mental disability and mental health care have been neglected in the discourse around health, human rights, and equality. During the COVID-19 era, economic recession and lockdowns have escalated the symptoms of anxiety and depressive disorders. This application serves as a novice attempt to tackle this pervasive problem. The application currently is focused on software development rather than research but is highly scalable and works as a base for further enhancements.

1.2 PROBLEM OUTLINE

Mental illnesses are health conditions involving changes in emotion, thinking, or behavior (or a combination of these). Mental illnesses are associated with distress and/or problems functioning in social, work, or family activities. Mental health issues do not get better on their own. Untreated anxiety may escalate to panic attacks, and failing to address trauma can lead to post-traumatic stress disorder.

Mental health conditions are increasing worldwide. According to WHO, Suicide is the second leading cause of death among 15-29-year-olds. Covid 19 has further escalated this problem. The distress in the pandemic probably stems from people's limited social interactions, tensions among families in lockdown together, and fear of illness.

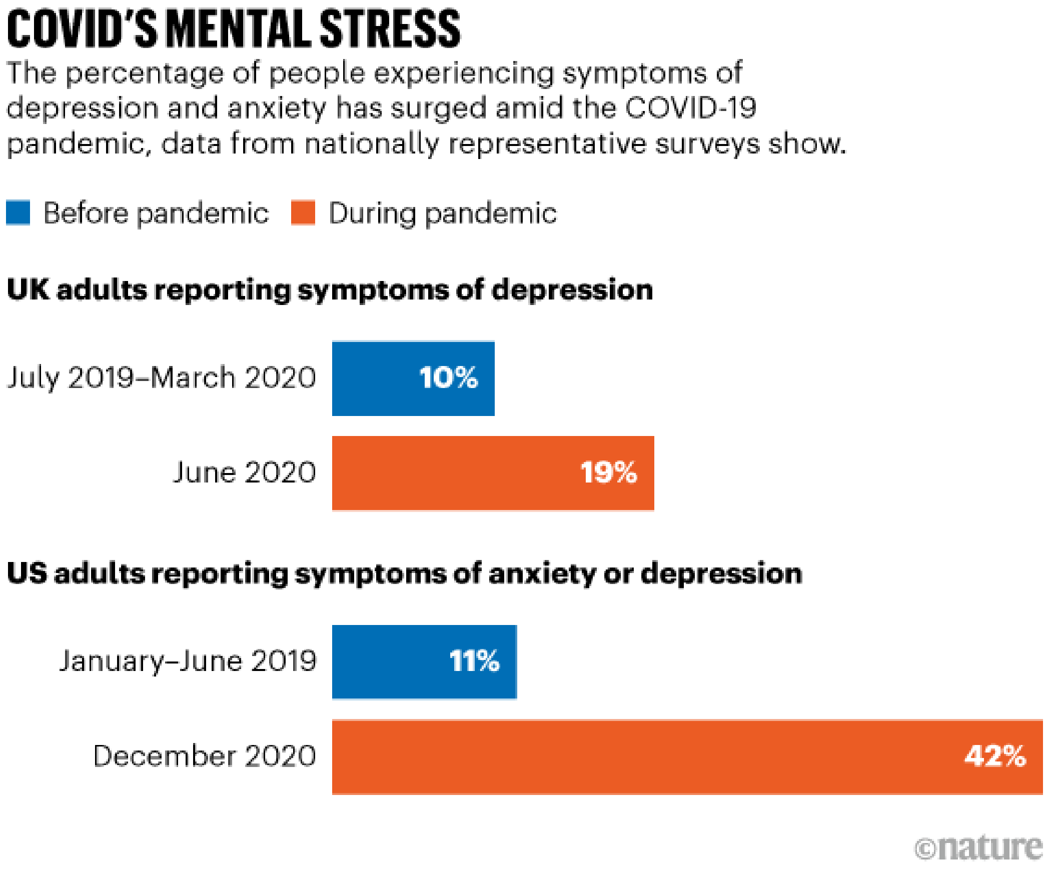


Fig 1.1 Covid Mental Health Bar graph

Furthermore, there are just a handful of competent applications undertaking this issue. My application tends to offer a simple and user-friendly interface as a solution to this problem by answering a wide range of questions and suggesting activities to the user.

1.3 PROJECT OBJECTIVES

The principal objective of my project is to provide valuable information to the user and help them in improving their psychological condition. Besides that, the project also aims to help develop hobbies among individuals which further helps in tackling loneliness and isolation, a major precursor to various psychiatric disorders. The application has a simple user interface that makes it convenient for a diverse range of users.

1.4 PROJECT METHODOLOGY

The Mental HealthCare App has two main functionalities:

- **Chatbot**

  Firstly the question asked by the user transformed after several steps to an embedding vector.

  The embedding vector is then checked with the embedding vector of each question already present in the dataset via cosine similarity.

  The question with the highest similarity score is chosen and the answer relative to that question is given as the output.

  This operation is performed in the python programming language.

- **Activities Recommender**

  The user chooses the field of interest.

  The bored api returns the response to our request url in the form of a JSON file.

  We extract the activity part from that JSON file and display it to the user.

Along with these, the application has a user-friendly interface which is accomplished with the help of XML which sets up the layout of the application. The application is written in the kotlin programming language. Python is integrated into java with the help of Chaquopy which is a python sdk for android.

Requirements:-

Chaquopy version used 9.1.0

Android devices with minimum API level 16.

Supported Android Gradle plugin versions 3.4 - 4.1

Included Python version 3.8.6

DATASET

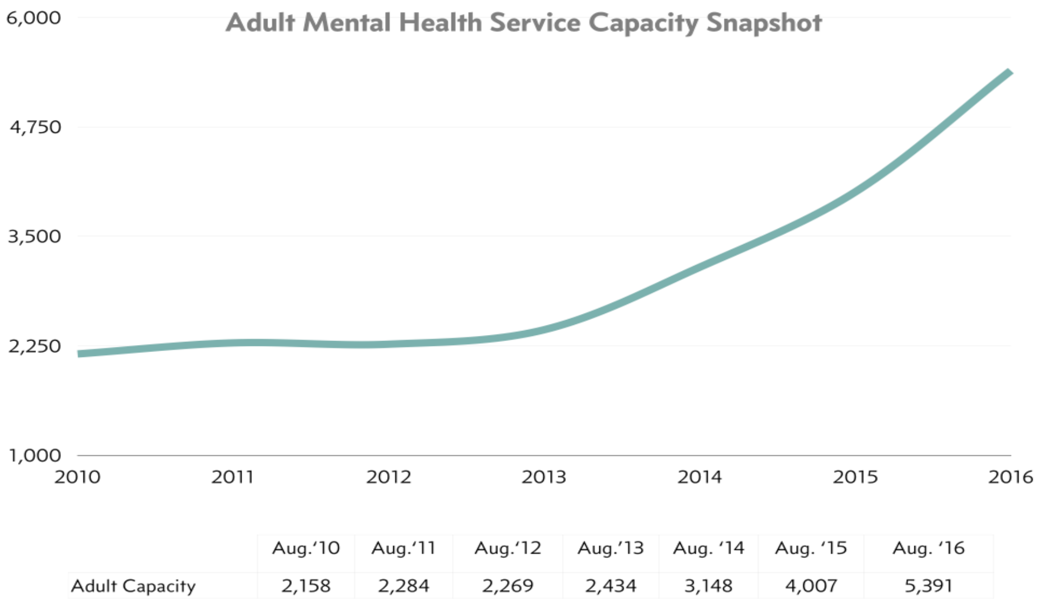
Part of the dataset is taken from Kaggle(Link- https://www.kaggle.com/narendrageek/mental-health-faq-for-chatbot)

The rest of the dataset is from Counselchat.com (Link- https://github.com/nbertagnolli/counsel-chat/tree/master/data)

1.5 SCOPE OF PROJECT WORK

Advancements in technology have opened doors to new directions in mental health care. Mental health has a stigma attached to it. If you suffer from a mental illness, the common conception is that there’s something wrong with you, as a person, as a human being. People often restrain from talking about their mental problems. The emergence of mobile health applications has made mental health care more accessible to people suffering from mental illness.

With the financial, social, and personal burden of mental and substance use disorders on the rise globally the need for effective mental healthcare apps is clearly evident. Although the challenge is to provide cheap but reliable guidance to people who cannot afford regular counseling sessions from a licensed therapist. My application tries to solve this problem by using the dataset from counselchat.com where therapists have given answers to a wide range of questions. There is a broad scope of ideas that can be implemented to keep track of users’ well-being.



	Aug.’10	Aug.’11	Aug.’12	Aug.’13	Aug. ‘14	Aug. ‘15	Aug. ‘16
Adult Capacity	2,158	2,284	2,269	2,434	3,148	4,007	5,391

Fig 1.2 Adult Mental Service Chart

1.6 ORGANISATION OF PROJECT

The project has been divided into several phases:

- Firstly the front-end of the android application has been created with XML to design the user interface.

- Then the required dataset has been collected. Then an efficient chatbot system has been built that can work on android.

- Finally, the backend along with the api call to bored api has been written in kotlin.

1.7 SUMMARY

In the first chapter, we have discussed:

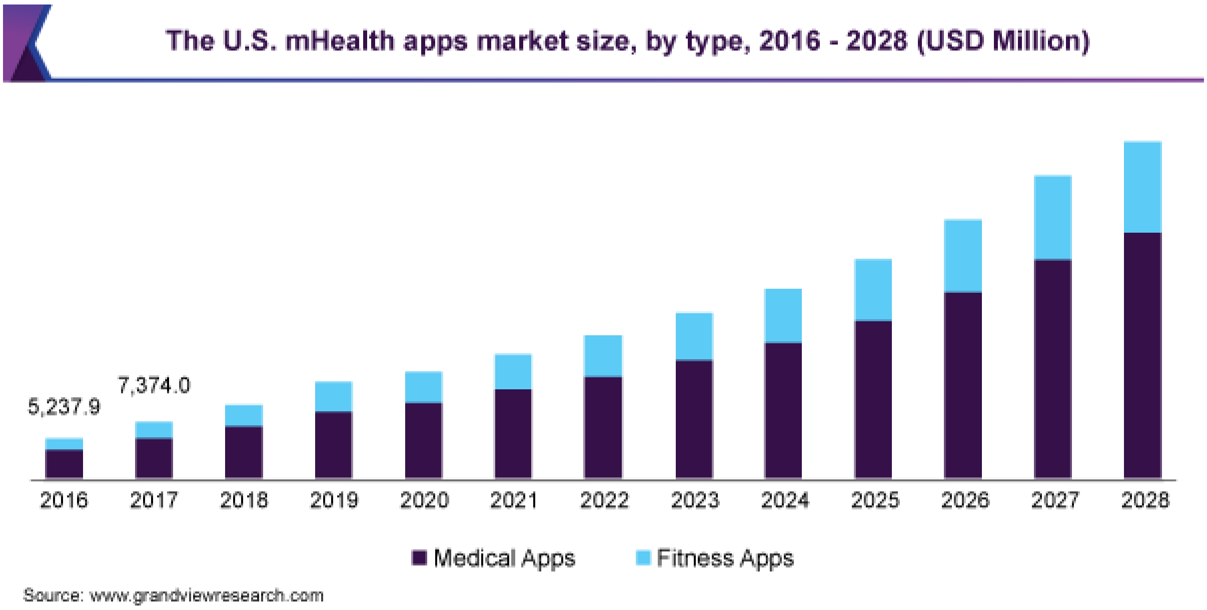
Purpose of the project:- to provide valuable information to the user and help them in improving their psychological condition.
Problem Outline:- The increasing mental health related problems in the COVID 19 pandemic era.
Design/Approach:- Designing a simple but effective UI for our android application and afterward a chatbot and an activity recommender.
Scope of the Project:- Advancements in technology opening new gates in finding solutions to mental health issues.

Chapter 2

Study and Review of Literature

2.1 INTRODUCTION

Advancements and innovations in technology have led to enormous progress in the field of health care. The emergence of mobile health applications has made mental health care more accessible to people suffering from mental ailments.



Source: www.grandviewresearch.com

Fig 2.1 Mental Health Apps Market Size

As shown in the graph the market for mental health apps is predicted to grow exponentially in a few years. This shows the importance of the information the user needs to have of some of the technologies used and services provided by these applications. The following chapter discusses about some of the tools and languages used to build these systems.

My application uses Kotlin programming language and XML to design the user interface. Meanwhile, the chatbot system is built in python programming language. Chaquopy is used to integrate python code in Android studio.

2.2 UI DEVELOPMENT

XML:- XML stands for Extensible Markup Language. XML is a markup language much like HTML used to describe data. XML tags are not predefined in XML. In Android, we use xml for designing our layouts because xml is a lightweight language so it doesn't make our layout heavy. In Android, there are several xml files used for several different purposes like the Layout xml file, the Manifest xml file, the string and style xml files, etc.

KOTLIN:- Kotlin is a cross-platform, modern statically typed programming language with type inference used by over 60% of professional Android developers that helps boost productivity, developer satisfaction, and code safety. Kotlin is object-oriented and supports functional programming features. It is designed for the JVM (Java Virtual Machine).

2.3 BACKEND DEVELOPMENT

PYTHON:- Python is an interpreted high-level general-purpose programming language. It was created by Guido van Rossum during 1985- 1990. Its design philosophy emphasizes code readability with its use of significant indentation. It supports functional and structured programming methods as well as OOP. Python is also used for machine learning tasks. It has various libraries like numpy, tensorflow and scikit learn which provide all the necessary tools for coding machine learning algorithms.

THE PYTHON LIBRARIES USED FOR IMPLEMENTING CHATBOT

NUMPY:- NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

PANDAS:- Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series.

SCIKIT LEARN:- It provides simple and efficient tools for predictive data analysis. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistent interface in Python.

NLTK:- The Natural Language Toolkit, or more commonly NLTK, is a suite of libraries and programs for symbolic and statistical natural language processing for English written in the Python programming language.

CHAQUOPY:- Chaquopy provides everything you need to include Python components in an Android app, including full integration with Android Studio's standard Gradle build system. It provides simple APIs for calling Python code from Java/Kotlin, and vice versa. It also provides a wide range of third-party Python packages, including SciPy, OpenCV, TensorFlow, and many more.

2.4 SUMMARY

This chapter presents some discussions about the tools and technologies used in the development of our mental healthcare application.

Kotlin is a modern, concise and safe programming language. It is robust, statically typed and much less verbose than Java. Hence it has become the goto language for android development. XML is used to design the layout as it is a lightweight language so it doesn't make our layout heavy.

Python is an ideal programming language for machine learning as it is simple and consistent. It has a range of libraries and frameworks and is highly flexible. Hence it is used for implementing the chatbot.

Chaquopy was used for integrating the chatbot as it provides everything you need to include Python components in an Android app, including full integration with Android Studio's standard Gradle build system.

Chapter 3

Implementation

3.1 INTRODUCTION

The implementation phase plays the most important role in the software development process. It is at this stage that the physical source code of the system being built is created. Software design and implementation activities are invariably inter-leaved. Implementation is the process of realizing the design as a program. This chapter explores the principal implementation processes, the UML diagrams and displays all the important code snippets from the application.

3.2 UML ANALYSIS MODEL

The Unified Modeling Language (UML) is a graphical language for OOAD(object oriented analysis and design) that gives a standard way to write a software system’s blueprint. It helps to visualize, specify, construct, and document the artifacts of an object-oriented system. It is used to depict the structures and the relationships in a complex system. Star UML was used to model this application.

Systems and Models in UML

System − A set of elements organized to achieve certain objectives form a system. Systems are often divided into subsystems and described by a set of models.

Model − Model is a simplified, complete, and consistent abstraction of a system, created for better understanding of the system.

A UML diagram is a graphical representation of a system. It comprises a group of elements generally in the form of a graph.

USE CASE DIAGRAM

The purpose of a use case diagram in UML is to demonstrate the different ways that a user might interact with a system. More formally the use case diagrams illustrate the interactions that exist between users (actors) and use cases (actions) within the application. In our application there is only one actor- the user, therefore we have one use case diagram.

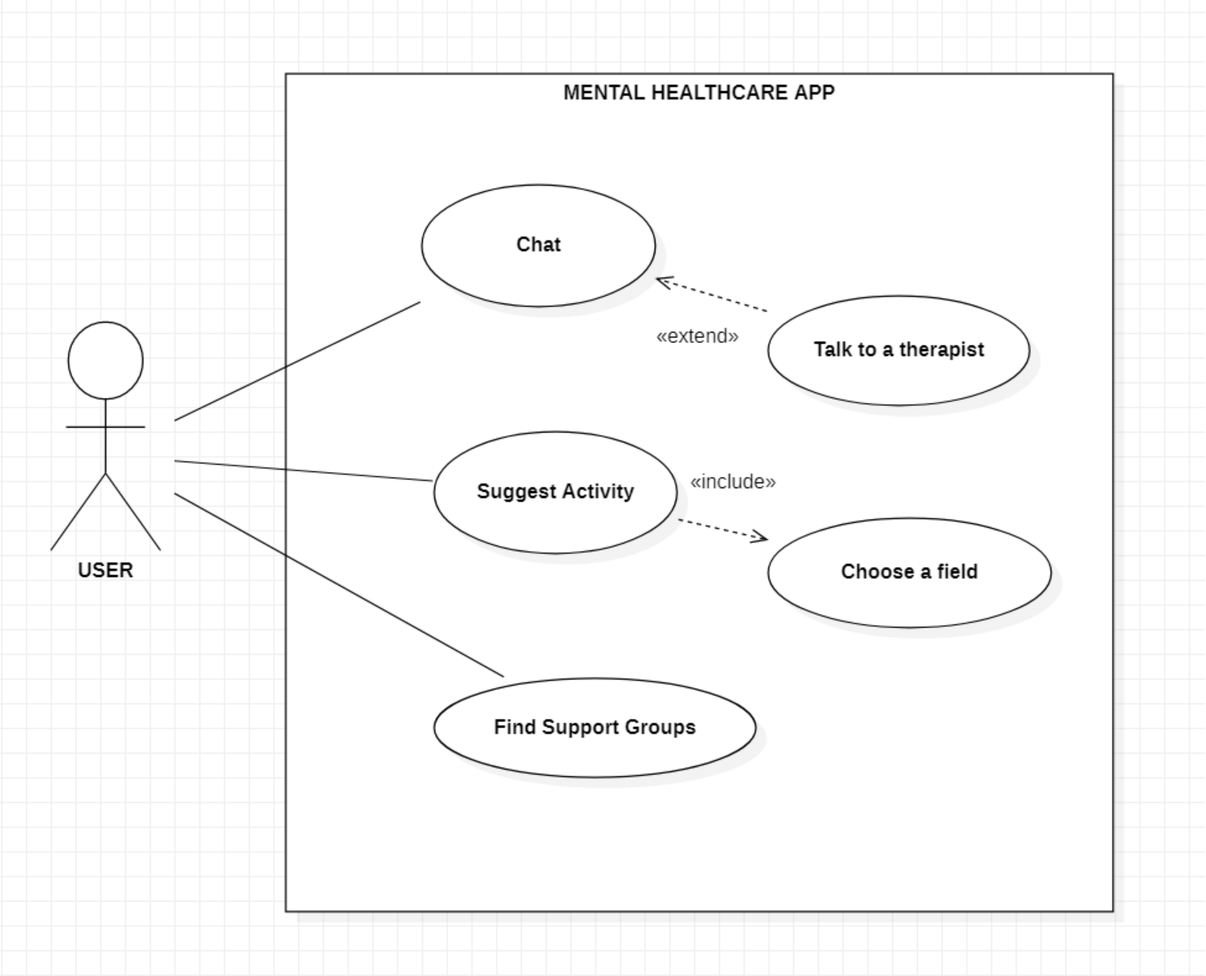


Fig 3.1 User Use Case Diagram

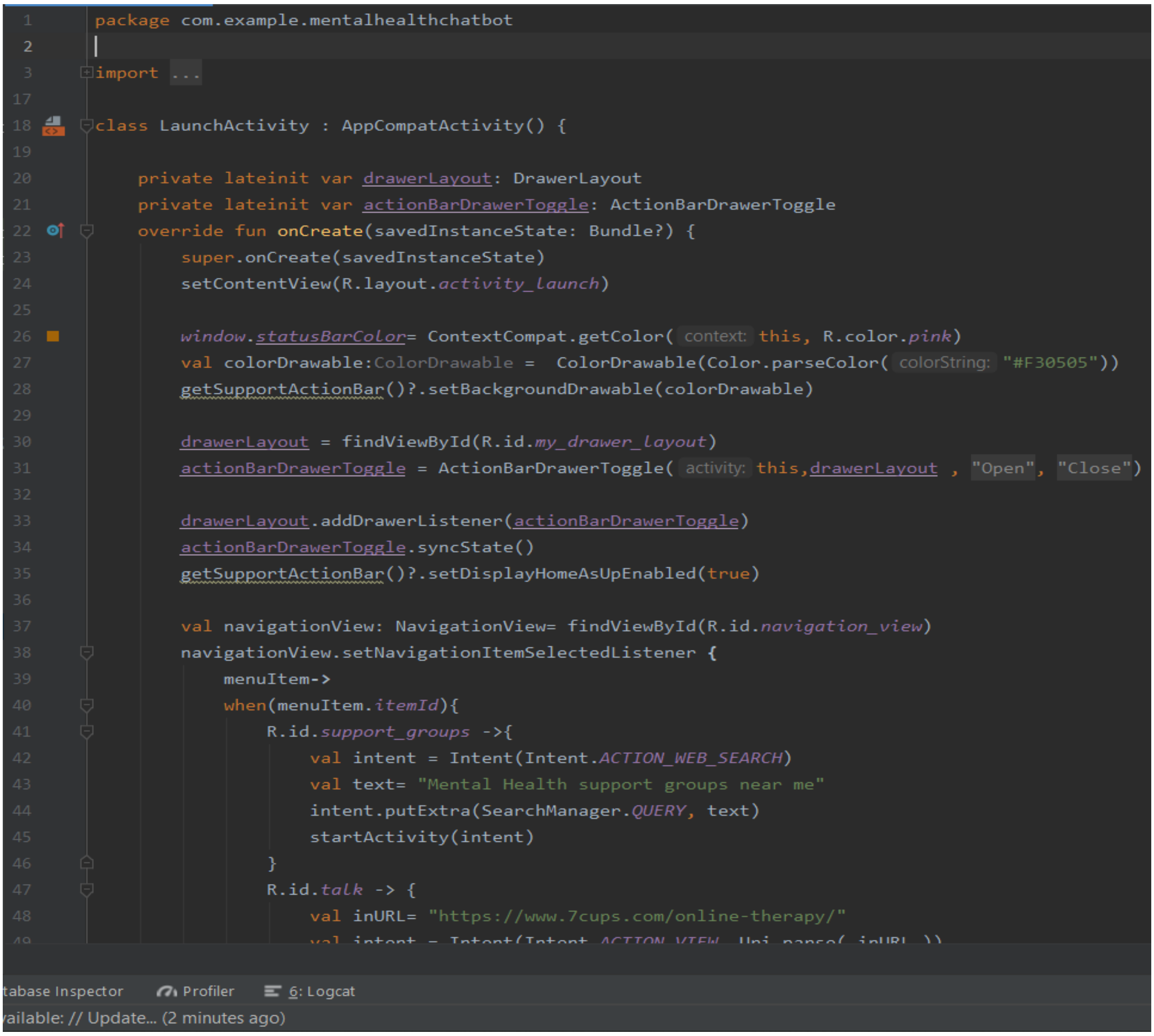
The above figure shows all of the interactions a user can have with the system. The user could chat with the chatbot and if he/she isn’t satisfied they could choose the option to talk to a licensed therapist. Similarly if the user wants activity recommendations they have to choose a field of interest.

3.3 CODE SNIPPETS

The code for the mental healthcare application was written on Android Studio. The snippets are taken as Screen Shots of the code open in Android Studio to explain the main concepts of implementation.

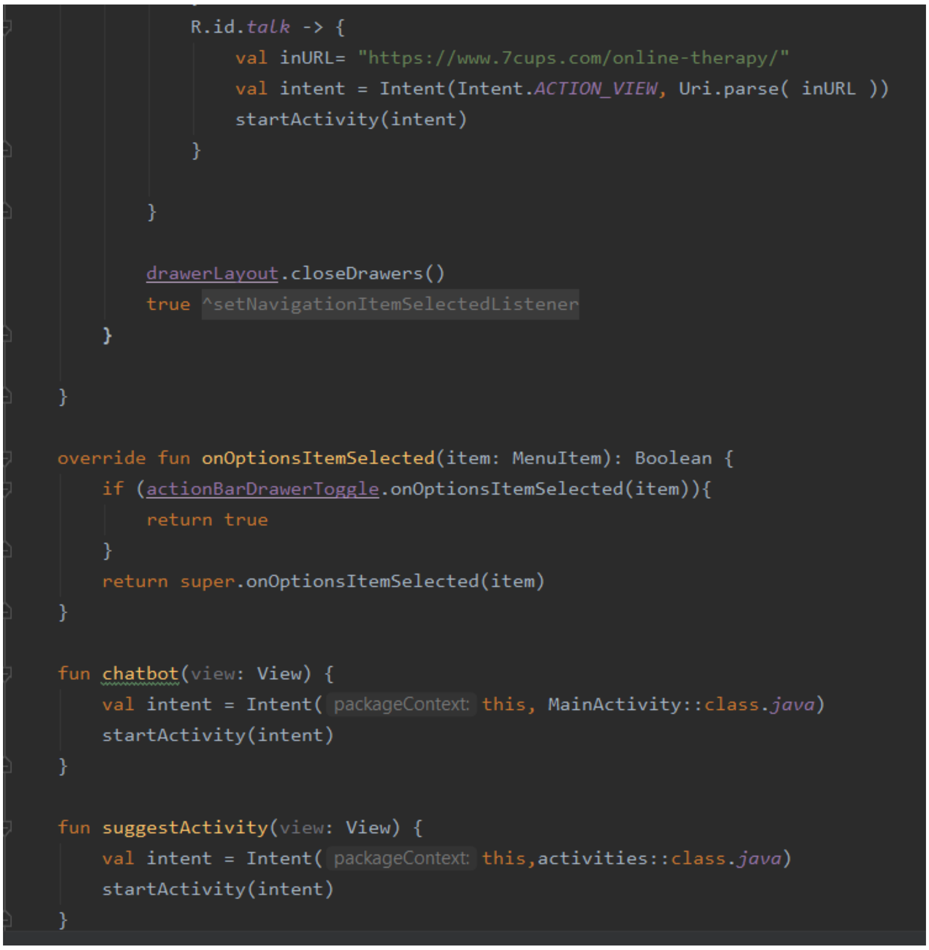
3.3.1 Launch Activity

This is the home page of the application. It gives the user option for browsing to either the chatbot or the activity recommender. It also has a navigation menu which gives the option for online therapy and finding support groups.



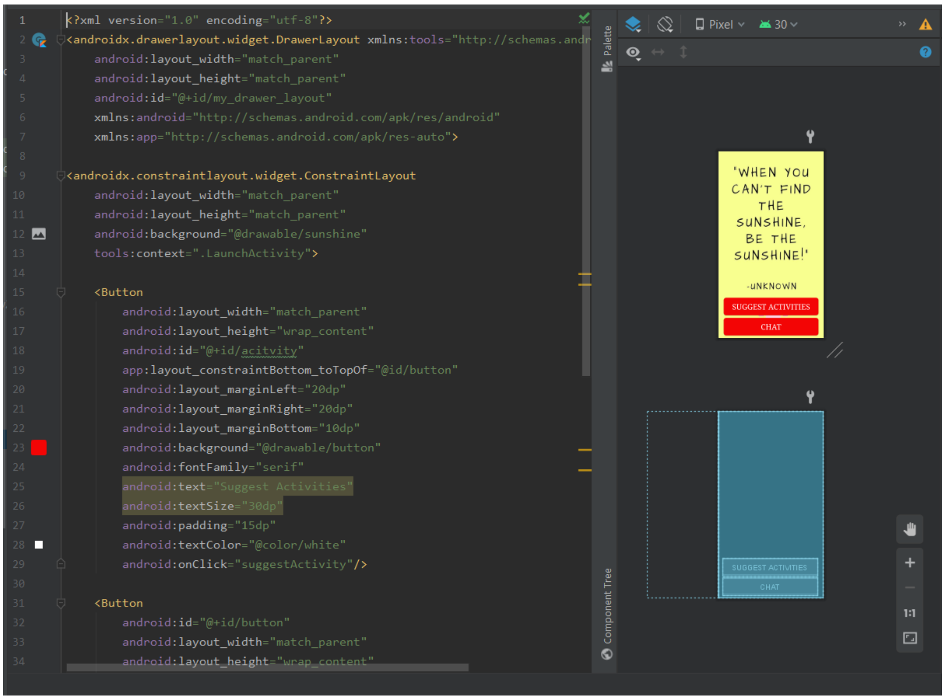
```kotlin
package com.example.mentalhealthchatbot

import ...

class LaunchActivity : AppCompatActivity() {

    private lateinit var drawerLayout: DrawerLayout
    private lateinit var actionBarDrawerToggle: ActionBarDrawerToggle
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_launch)

        window.statusBarColor= ContextCompat.getColor( context: this, R.color.pink)
        val colorDrawable:ColorDrawable =  ColorDrawable(Color.parseColor( colorString: "#F30505"))
        getSupportActionBar()?.setBackgroundDrawable(colorDrawable)

        drawerLayout = findViewById(R.id.my_drawer_layout)
        actionBarDrawerToggle = ActionBarDrawerToggle( activity: this,drawerLayout , "Open", "Close")

        drawerLayout.addDrawerListener(actionBarDrawerToggle)
        actionBarDrawerToggle.syncState()
        getSupportActionBar()?.setDisplayHomeAsUpEnabled(true)

        val navigationView: NavigationView= findViewById(R.id.navigation_view)
        navigationView.setNavigationItemSelectedListener {
            menuItem->
            when(menuItem.itemId){
                R.id.support_groups ->{
                    val intent = Intent(Intent.ACTION_WEB_SEARCH)
                    val text= "Mental Health support groups near me"
                    intent.putExtra(SearchManager.QUERY, text)
                    startActivity(intent)
                }
                R.id.talk -> {
                    val inURL= "https://www.7cups.com/online-therapy/"
                    val intent = Intent(Intent.ACTION_VIEW, Uri.parse( inURL ))
```

Fig 3.2 Code Snippet for Launch Activity (Kotlin)



```
            R.id.talk -> {
                val inURL= "https://www.7cups.com/online-therapy/"
                val intent = Intent(Intent.ACTION_VIEW, Uri.parse( inURL ))
                startActivity(intent)
            }

        }

        drawerLayout.closeDrawers()
        true ^setNavigationItemSelectedListener
    }

}

override fun onOptionsItemSelected(item: MenuItem): Boolean {
    if (actionBarDrawerToggle.onOptionsItemSelected(item)){
        return true
    }
    return super.onOptionsItemSelected(item)
}

fun chatbot(view: View) {
    val intent = Intent( packageContext: this, MainActivity::class.java)
    startActivity(intent)
}

fun suggestActivity(view: View) {
    val intent = Intent( packageContext: this,activities::class.java)
    startActivity(intent)
}
```

Fig 3.3 Code Snippet for Launch Activity (Kotlin) cont.

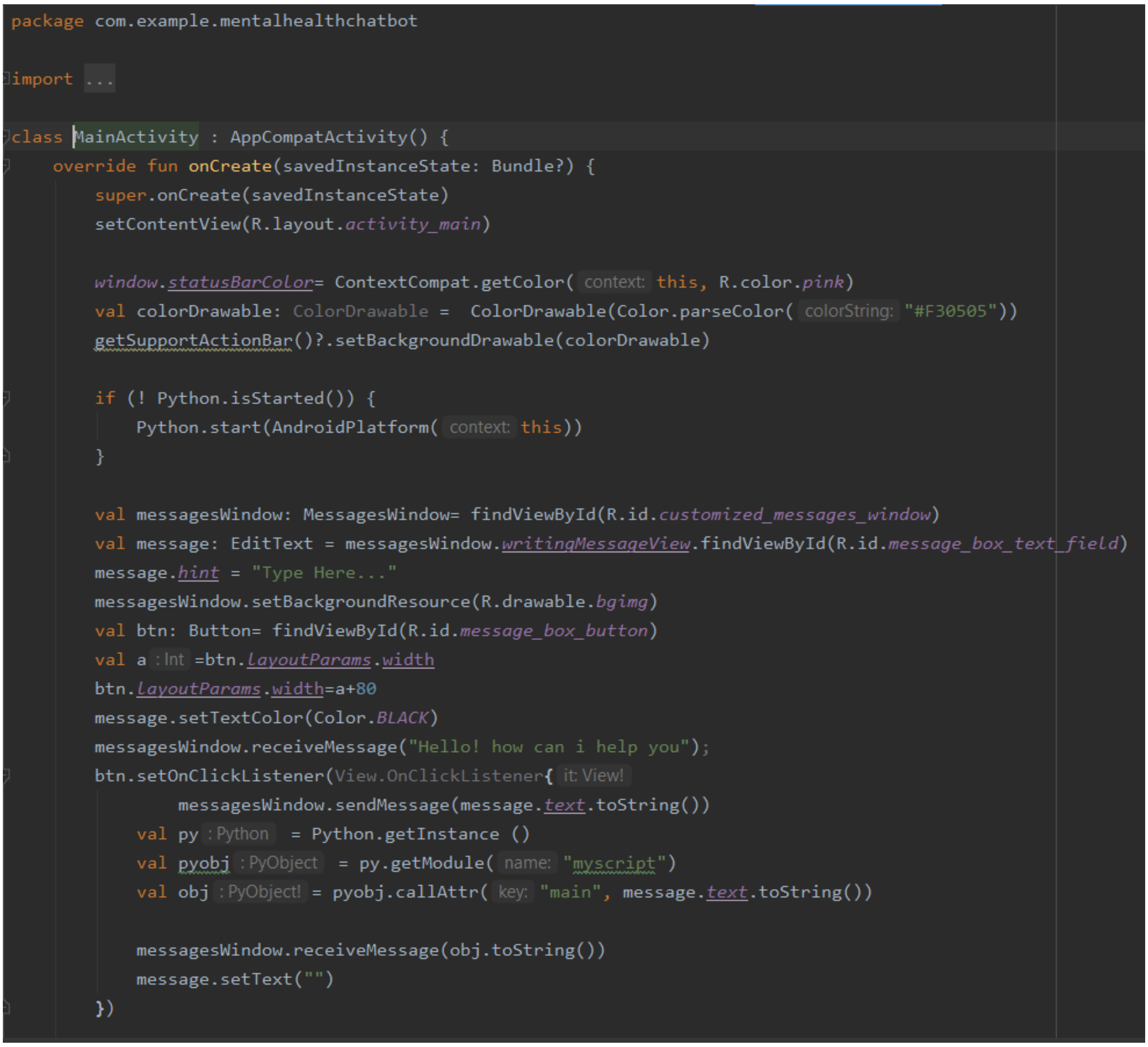


```
<?xml version="1.0" encoding="utf-8"?>
<androidx.drawerlayout.widget.DrawerLayout xmlns:tools="http://schemas.andr
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/my_drawer_layout"
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto">

<androidx.constraintlayout.widget.ConstraintLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/sunshine"
    tools:context=".LaunchActivity">

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/acitvity"
        app:layout_constraintBottom_toTopOf="@id/button"
        android:layout_marginLeft="20dp"
        android:layout_marginRight="20dp"
        android:layout_marginBottom="10dp"
        android:background="@drawable/button"
        android:fontFamily="serif"
        android:text="Suggest Activities"
        android:textSize="30dp"
        android:padding="15dp"
        android:textColor="@color/white"
        android:onClick="suggestActivity"/>

    <Button
        android:id="@+id/button"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
```

Fig 3.4 Code Snippet for Launch Activity (XML)

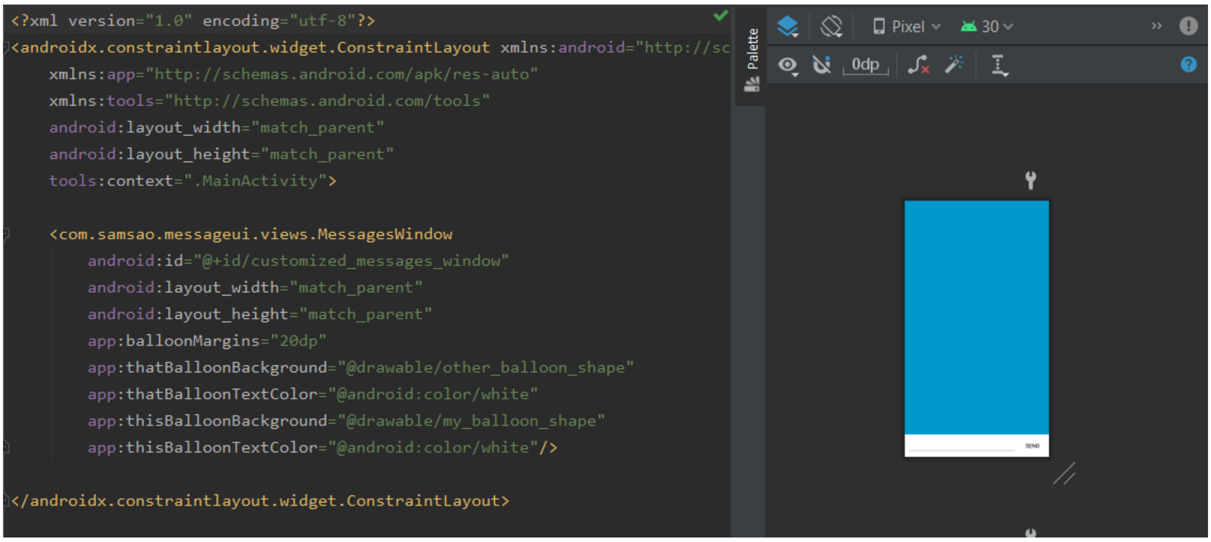
3.3.2 Chat Activity

This is the activity for the chatbot. It gives the user the option to interact with the chatbot by writing their issues in the message window and sending them. The application responds by returning the most likely answer in accordance with the dataset by calling a python function with the message as the argument.



```
package com.example.mentalhealthchatbot

import ...

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        window.statusBarColor= ContextCompat.getColor( context: this, R.color.pink)
        val colorDrawable: ColorDrawable =  ColorDrawable(Color.parseColor( colorString: "#F30505"))
        getSupportActionBar()?.setBackgroundDrawable(colorDrawable)

        if (! Python.isStarted()) {
            Python.start(AndroidPlatform( context: this))
        }

        val messagesWindow: MessagesWindow= findViewById(R.id.customized_messages_window)
        val message: EditText = messagesWindow.writingMessageView.findViewById(R.id.message_box_text_field)
        message.hint = "Type Here..."
        messagesWindow.setBackgroundResource(R.drawable.bgimg)
        val btn: Button= findViewById(R.id.message_box_button)
        val a : Int =btn.LayoutParams.width
        btn.LayoutParams.width=a+80
        message.setTextColor(Color.BLACK)
        messagesWindow.receiveMessage("Hello! how can i help you");
        btn.setOnClickListener(View.OnClickListener{ it: View!
                messagesWindow.sendMessage(message.text.toString())
            val py : Python  = Python.getInstance ()
            val pyobj : PyObject  = py.getModule( name: "myscript")
            val obj : PyObject! = pyobj.callAttr( key: "main", message.text.toString())

            messagesWindow.receiveMessage(obj.toString())
            message.setText("")
        })
```

Fig 3.5 Code Snippet for Chat Activity (Kotlin)

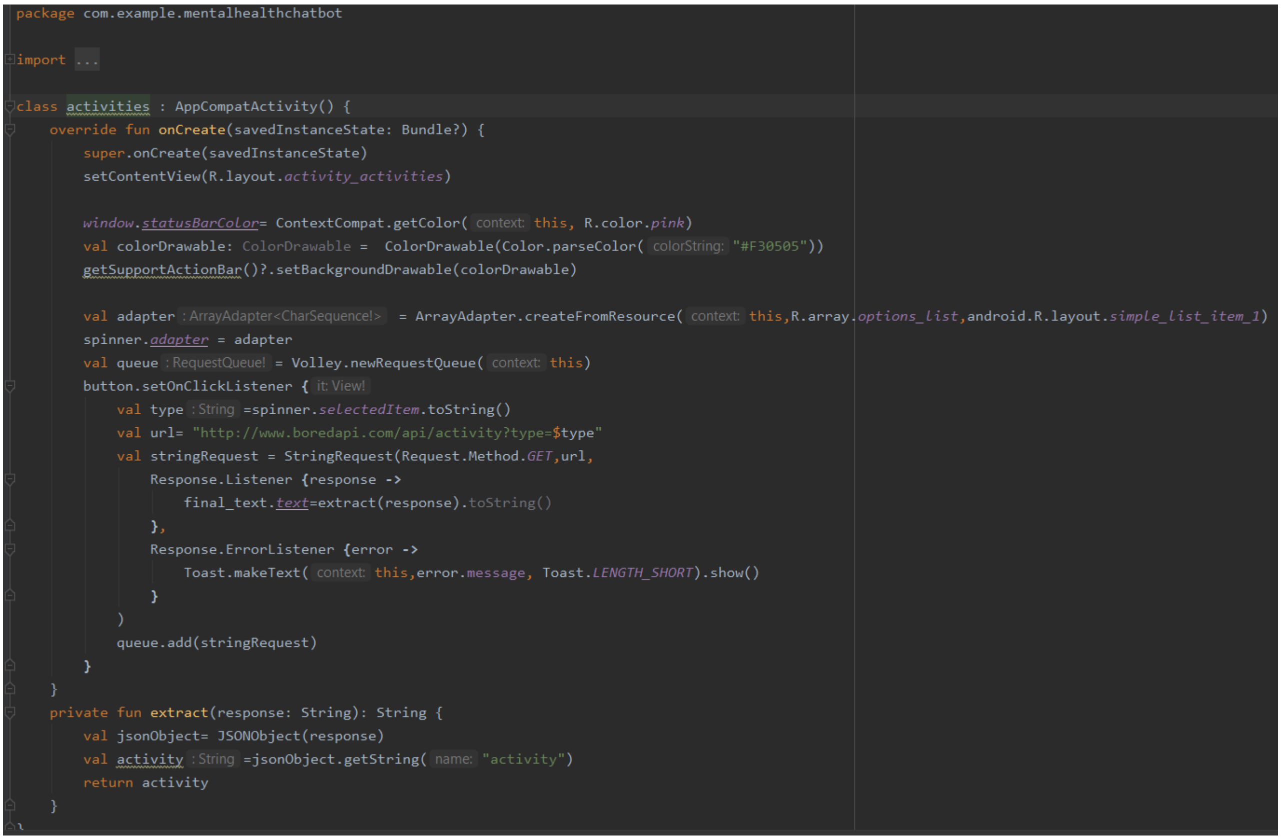


```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://sc
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <com.samsao.messageui.views.MessagesWindow
        android:id="@+id/customized_messages_window"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:balloonMargins="20dp"
        app:thatBalloonBackground="@drawable/other_balloon_shape"
        app:thatBalloonTextColor="@android:color/white"
        app:thisBalloonBackground="@drawable/my_balloon_shape"
        app:thisBalloonTextColor="@android:color/white"/>

</androidx.constraintlayout.widget.ConstraintLayout>
```

Fig 3.6 Code Snippet for Chat Activity (XML)

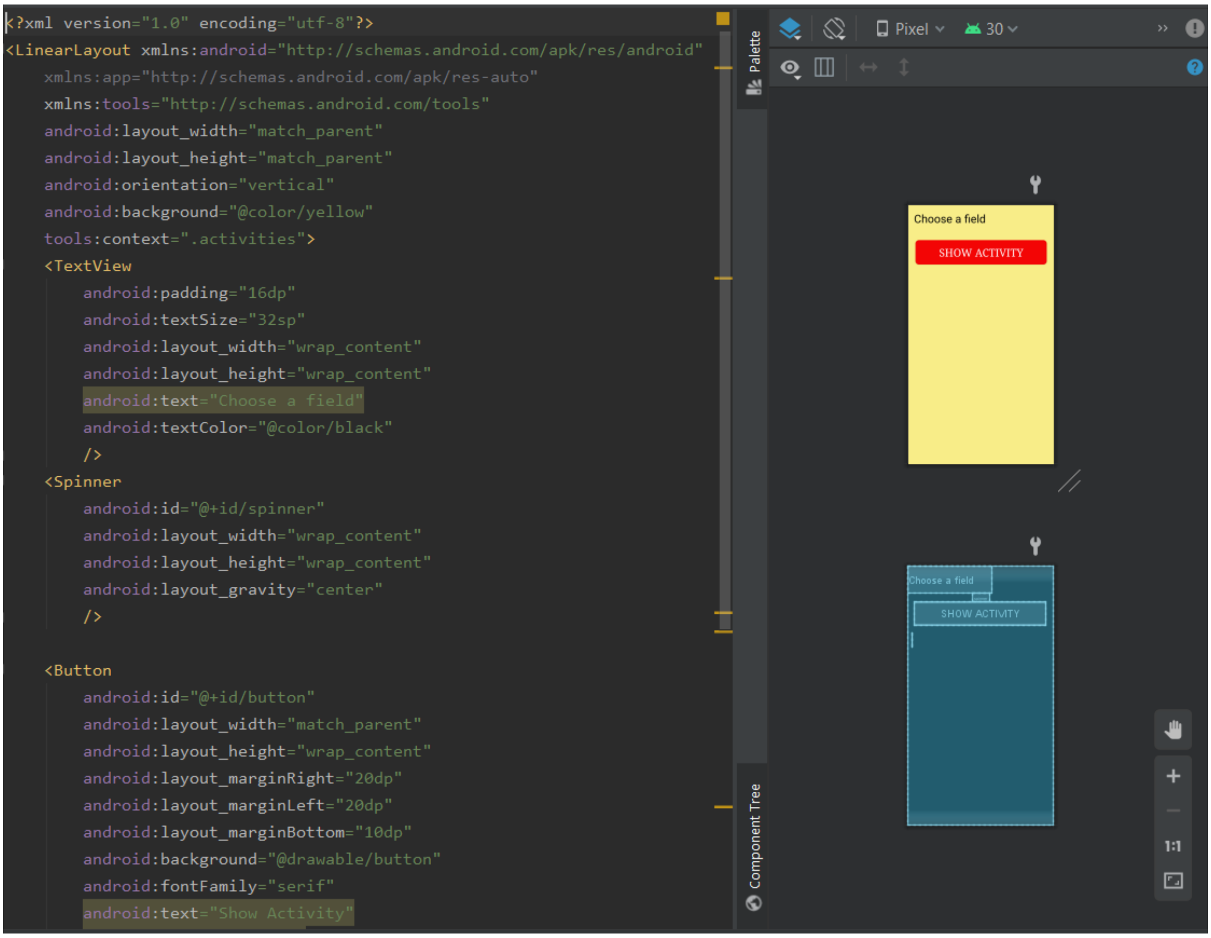
3.3.3 Suggest Activity

This is the activity for recommending recreational tasks. The user has to pick a field they are in and click on the show activity button. The application makes an api call to bored api to suggest activity based on user preference.



```kotlin
package com.example.mentalhealthchatbot

import ...

class activities : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_activities)

        window.statusBarColor= ContextCompat.getColor( context: this, R.color.pink)
        val colorDrawable: ColorDrawable =  ColorDrawable(Color.parseColor( colorString: "#F30505"))
        getSupportActionBar()?.setBackgroundDrawable(colorDrawable)

        val adapter : ArrayAdapter<CharSequence!>  = ArrayAdapter.createFromResource( context: this,R.array.options_list,android.R.layout.simple_list_item_1)
        spinner.adapter = adapter
        val queue : RequestQueue! = Volley.newRequestQueue( context: this)
        button.setOnClickListener { it: View!
            val type : String =spinner.selectedItem.toString()
            val url= "http://www.boredapi.com/api/activity?type=$type"
            val stringRequest = StringRequest(Request.Method.GET,url,
                Response.Listener {response ->
                    final_text.text=extract(response).toString()
                },
                Response.ErrorListener {error ->
                    Toast.makeText( context: this,error.message, Toast.LENGTH_SHORT).show()
                }
            )
            queue.add(stringRequest)
        }
    }
    private fun extract(response: String): String {
        val jsonObject= JSONObject(response)
        val activity : String =jsonObject.getString( name: "activity")
        return activity
    }
}
```

Fig 3.7 Code Snippet for Suggest Activity (Kotlin)

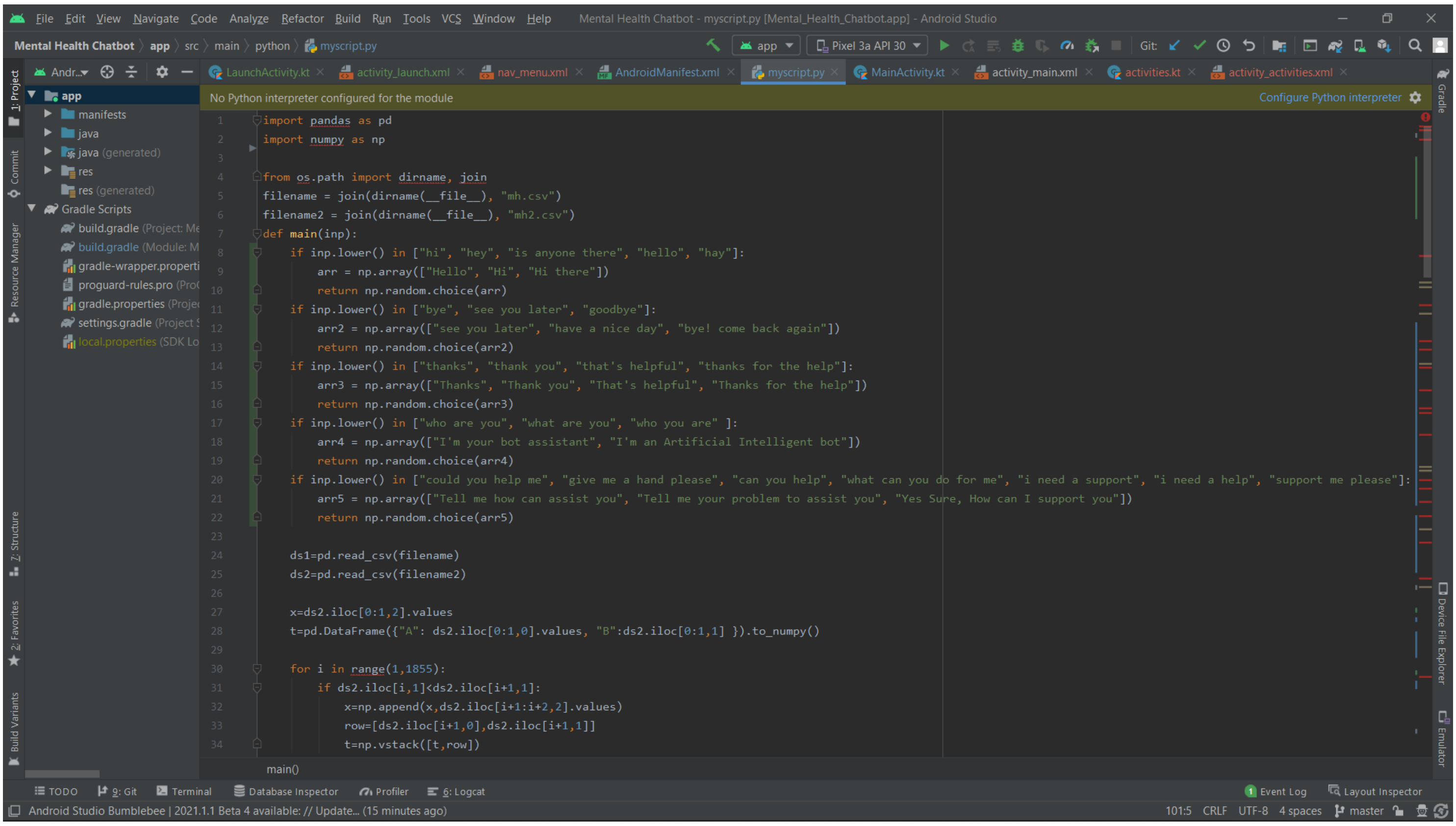


```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="@color/yellow"
    tools:context=".activities">
    <TextView
        android:padding="16dp"
        android:textSize="32sp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Choose a field"
        android:textColor="@color/black"
        />
    <Spinner
        android:id="@+id/spinner"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        />

    <Button
        android:id="@+id/button"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginRight="20dp"
        android:layout_marginLeft="20dp"
        android:layout_marginBottom="10dp"
        android:background="@drawable/button"
        android:fontFamily="serif"
        android:text="Show Activity"
```

Fig 3.8 Code Snippet for Suggest Activity (XML)

3.3.4 Python script

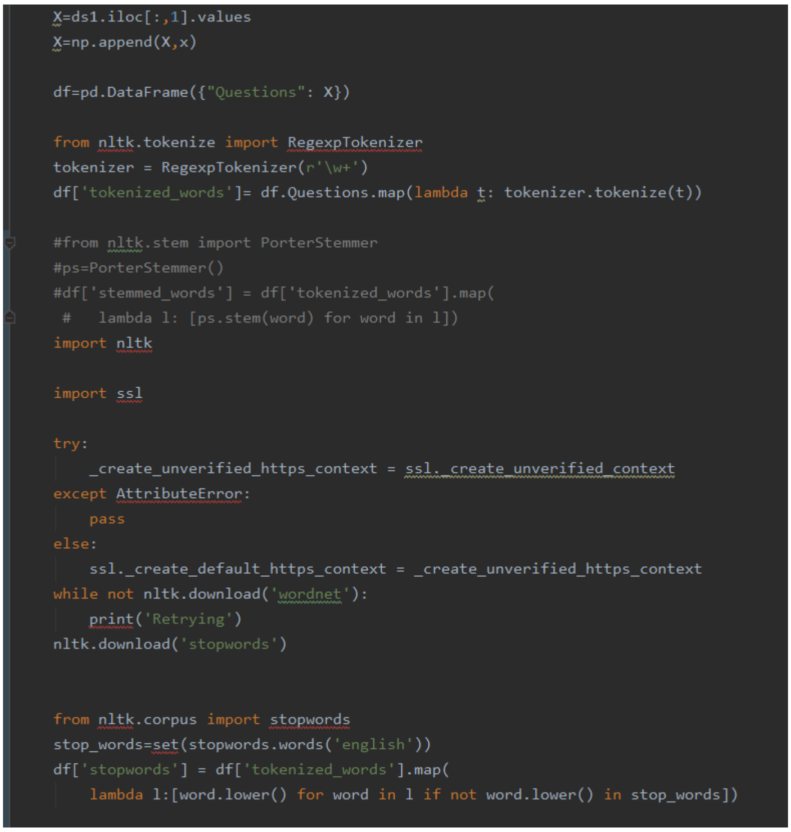
This is the python code for the chatbot system. The application has a pre-trained custom embedding matrix in accordance with the dataset which is used by the main() function to return the result to the input query.



No Python interpreter configured for the module

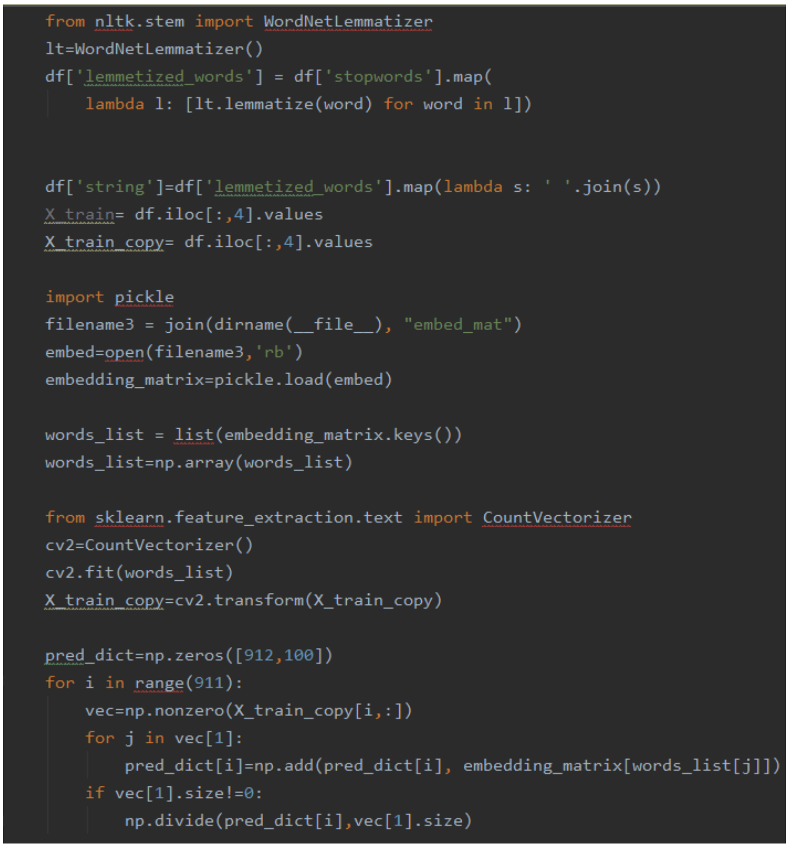
```python
import pandas as pd
import numpy as np

from os.path import dirname, join
filename = join(dirname(__file__), "mh.csv")
filename2 = join(dirname(__file__), "mh2.csv")
def main(inp):
    if inp.lower() in ["hi", "hey", "is anyone there", "hello", "hay"]:
        arr = np.array(["Hello", "Hi", "Hi there"])
        return np.random.choice(arr)
    if inp.lower() in ["bye", "see you later", "goodbye"]:
        arr2 = np.array(["see you later", "have a nice day", "bye! come back again"])
        return np.random.choice(arr2)
    if inp.lower() in ["thanks", "thank you", "that's helpful", "thanks for the help"]:
        arr3 = np.array(["Thanks", "Thank you", "That's helpful", "Thanks for the help"])
        return np.random.choice(arr3)
    if inp.lower() in ["who are you", "what are you", "who you are" ]:
        arr4 = np.array(["I'm your bot assistant", "I'm an Artificial Intelligent bot"])
        return np.random.choice(arr4)
    if inp.lower() in ["could you help me", "give me a hand please", "can you help", "what can you do for me", "i need a support", "i need a help", "support me please"]:
        arr5 = np.array(["Tell me how can assist you", "Tell me your problem to assist you", "Yes Sure, How can I support you"])
        return np.random.choice(arr5)

    ds1=pd.read_csv(filename)
    ds2=pd.read_csv(filename2)

    x=ds2.iloc[0:1,2].values
    t=pd.DataFrame({"A": ds2.iloc[0:1,0].values, "B":ds2.iloc[0:1,1] }).to_numpy()

    for i in range(1,1855):
        if ds2.iloc[i,1]<ds2.iloc[i+1,1]:
            x=np.append(x,ds2.iloc[i+1:i+2,2].values)
            row=[ds2.iloc[i+1,0],ds2.iloc[i+1,1]]
            t=np.vstack([t,row])
```

Fig 3.9 Code Snippet for python script(myscript.py)



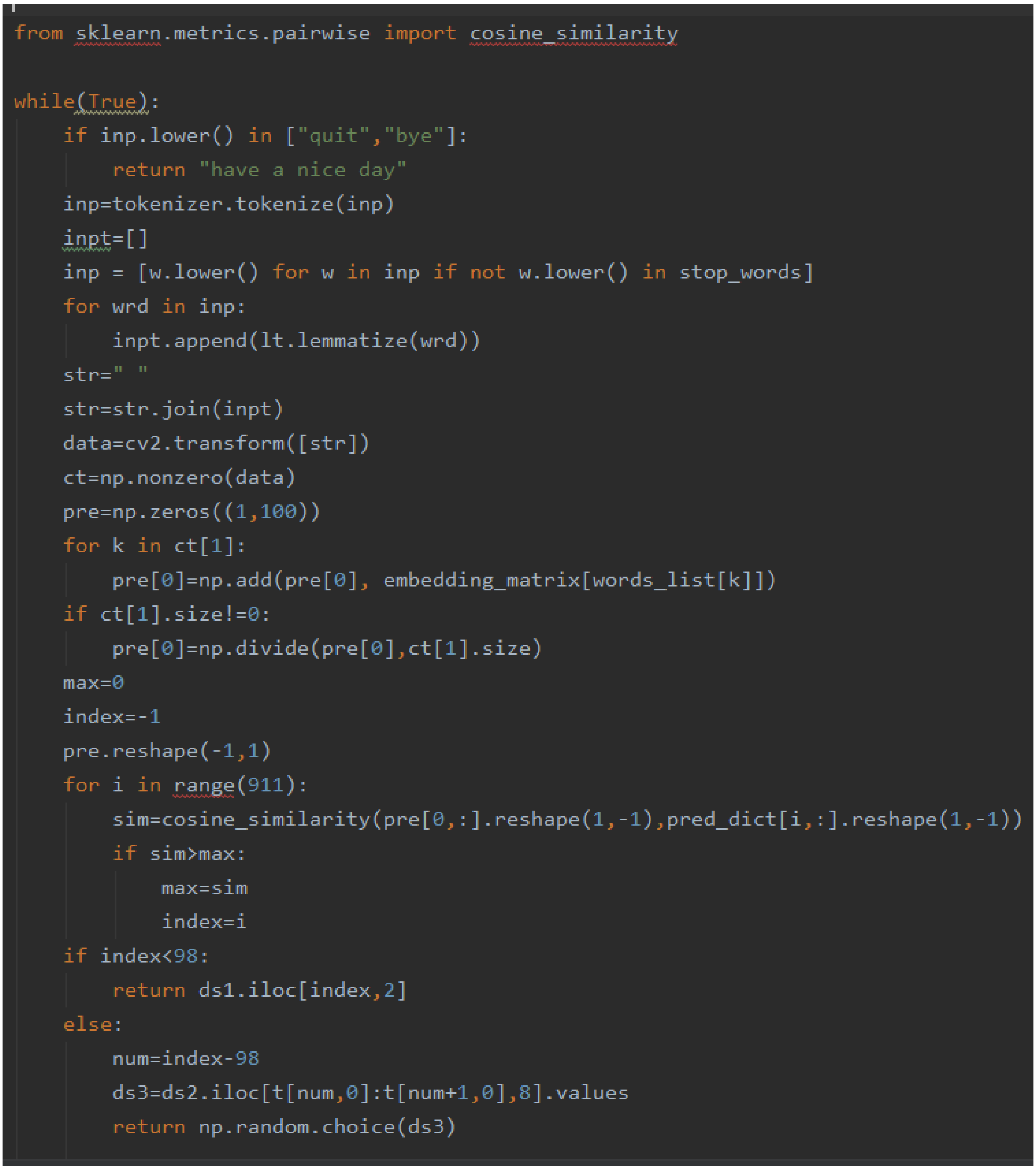
```
X=ds1.iloc[:,1].values
X=np.append(X,x)

df=pd.DataFrame({"Questions": X})

from nltk.tokenize import RegexpTokenizer
tokenizer = RegexpTokenizer(r'\w+')
df['tokenized_words']= df.Questions.map(lambda t: tokenizer.tokenize(t))

#from nltk.stem import PorterStemmer
#ps=PorterStemmer()
#df['stemmed_words'] = df['tokenized_words'].map(
 #   lambda l: [ps.stem(word) for word in l])
import nltk

import ssl

try:
    _create_unverified_https_context = ssl._create_unverified_context
except AttributeError:
    pass
else:
    ssl._create_default_https_context = _create_unverified_https_context
while not nltk.download('wordnet'):
    print('Retrying')
nltk.download('stopwords')


from nltk.corpus import stopwords
stop_words=set(stopwords.words('english'))
df['stopwords'] = df['tokenized_words'].map(
    lambda l:[word.lower() for word in l if not word.lower() in stop_words])
```

Fig 3.10 Code Snippet for python script(myscript.py) continued-1



```
from nltk.stem import WordNetLemmatizer
lt=WordNetLemmatizer()
df['lemmetized_words'] = df['stopwords'].map(
    lambda l: [lt.lemmatize(word) for word in l])


df['string']=df['lemmetized_words'].map(lambda s: ' '.join(s))
X_train= df.iloc[:,4].values
X_train_copy= df.iloc[:,4].values

import pickle
filename3 = join(dirname(__file__), "embed_mat")
embed=open(filename3,'rb')
embedding_matrix=pickle.load(embed)

words_list = list(embedding_matrix.keys())
words_list=np.array(words_list)

from sklearn.feature_extraction.text import CountVectorizer
cv2=CountVectorizer()
cv2.fit(words_list)
X_train_copy=cv2.transform(X_train_copy)

pred_dict=np.zeros([912,100])
for i in range(911):
    vec=np.nonzero(X_train_copy[i,:])
    for j in vec[1]:
        pred_dict[i]=np.add(pred_dict[i], embedding_matrix[words_list[j]])
    if vec[1].size!=0:
        np.divide(pred_dict[i],vec[1].size)
```

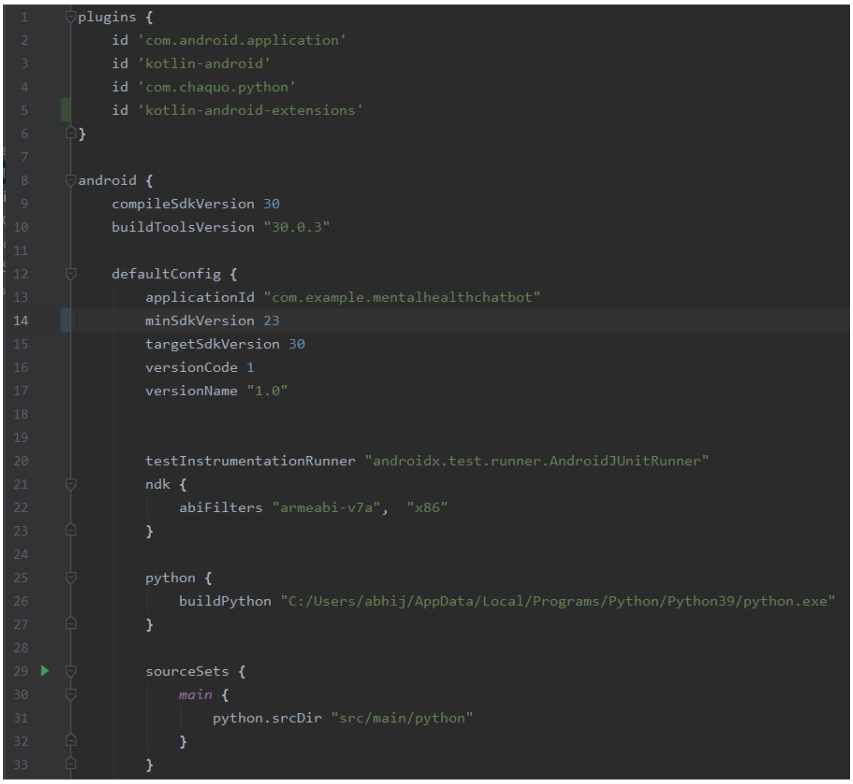
Fig 3.11 Code Snippet for python script(myscript.py) continued-2



```
from sklearn.metrics.pairwise import cosine_similarity

while(True):
    if inp.lower() in ["quit","bye"]:
        return "have a nice day"
    inp=tokenizer.tokenize(inp)
    inpt=[]
    inp = [w.lower() for w in inp if not w.lower() in stop_words]
    for wrd in inp:
        inpt.append(lt.lemmatize(wrd))
    str=" "
    str=str.join(inpt)
    data=cv2.transform([str])
    ct=np.nonzero(data)
    pre=np.zeros((1,100))
    for k in ct[1]:
        pre[0]=np.add(pre[0], embedding_matrix[words_list[k]])
    if ct[1].size!=0:
        pre[0]=np.divide(pre[0],ct[1].size)
    max=0
    index=-1
    pre.reshape(-1,1)
    for i in range(911):
        sim=cosine_similarity(pre[0,:].reshape(1,-1),pred_dict[i,:].reshape(1,-1))
        if sim>max:
            max=sim
            index=i
    if index<98:
        return ds1.iloc[index,2]
    else:
        num=index-98
        ds3=ds2.iloc[t[num,0]:t[num+1,0],8].values
        return np.random.choice(ds3)
```

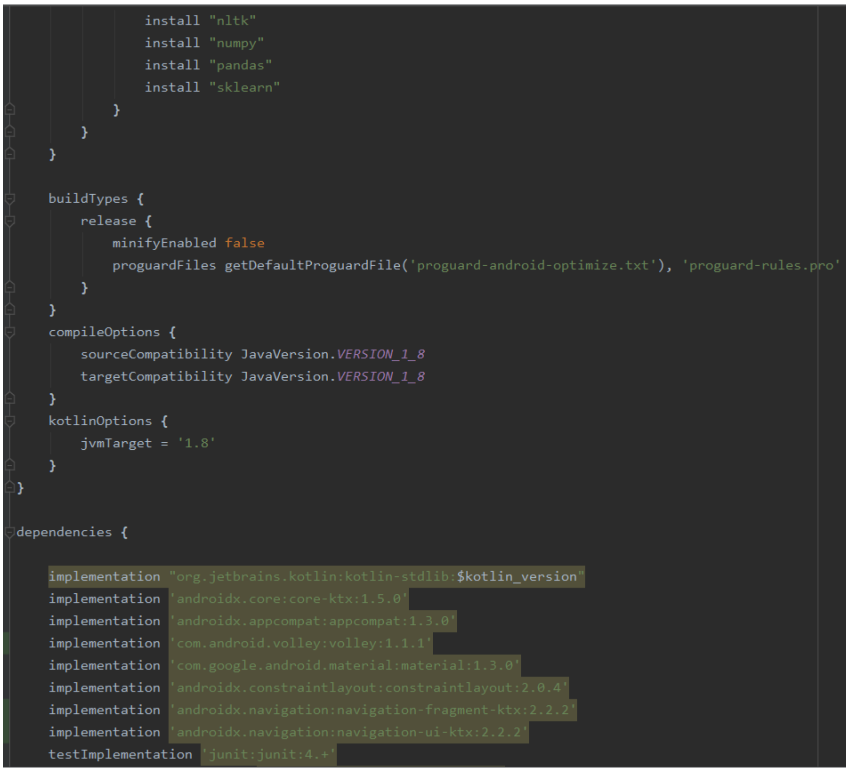
Fig 3.12 Code Snippet for python script(myscript.py) continued-3

3.3.5 Gradle build



```
plugins {
    id 'com.android.application'
    id 'kotlin-android'
    id 'com.chaquo.python'
    id 'kotlin-android-extensions'
}

android {
    compileSdkVersion 30
    buildToolsVersion "30.0.3"

    defaultConfig {
        applicationId "com.example.mentalhealthchatbot"
        minSdkVersion 23
        targetSdkVersion 30
        versionCode 1
        versionName "1.0"


        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
        ndk {
            abiFilters "armeabi-v7a",  "x86"
        }

        python {
            buildPython "C:/Users/abhij/AppData/Local/Programs/Python/Python39/python.exe"
        }

        sourceSets {
            main {
                python.srcDir "src/main/python"
            }
        }
```

Fig 3.13 Code Snippet for gradle build



```
                install "nltk"
                install "numpy"
                install "pandas"
                install "sklearn"
            }
        }
    }

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
        }
    }
    compileOptions {
        sourceCompatibility JavaVersion.VERSION_1_8
        targetCompatibility JavaVersion.VERSION_1_8
    }
    kotlinOptions {
        jvmTarget = '1.8'
    }
}

dependencies {

    implementation "org.jetbrains.kotlin:kotlin-stdlib:$kotlin_version"
    implementation 'androidx.core:core-ktx:1.5.0'
    implementation 'androidx.appcompat:appcompat:1.3.0'
    implementation 'com.android.volley:volley:1.1.1'
    implementation 'com.google.android.material:material:1.3.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.0.4'
    implementation 'androidx.navigation:navigation-fragment-ktx:2.2.2'
    implementation 'androidx.navigation:navigation-ui-ktx:2.2.2'
    testImplementation 'junit:junit:4.+'
```

Fig 3.14 Code Snippet for gradle build continued

3.4 SUMMARY

In this chapter we discussed the various procedures done to carry out the implementation of the design and realize it in the code. We used a use case diagram to give a brief overview of the interactions that take place between a user and the system. Some code snippets were also presented to show the implementation of the key functionalities.

Chapter 4

Result and Conclusion

4.1 Testing

Software Testing is a method to check whether the actual software product matches expected requirements and to ensure that software product is Defect free. It involves execution of software/system components using manual or automated tools to evaluate one or more properties of interest. The purpose of software testing is to identify errors, gaps or missing requirements in contrast to actual requirements. The application was tested on Pixel 3a API 30 emulator(Target: Android 11.0) and debugging was performed using Android Studio Debugger.

Software testing template and results

Test Description	Steps	Expected System Response	Status
Access the homepage	Open the application	The homepage is displayed	pass
Access the navigation menu	Slide the left of the homepage	Navigation menu is displayed	pass
Access the Suggest activity	Click on the suggest activity button on homepage	Suggest activity page is displayed	pass
Access the Chatbot	Click on the chat button on homepage	Chat activity page is displayed	pass
Talk to a therapist	Click on the Talk to a therapist button in the navigation drawer	User is sent to an online therapy website	pass
View Recreational	Choose a field in the suggest	An activity is shown	pass

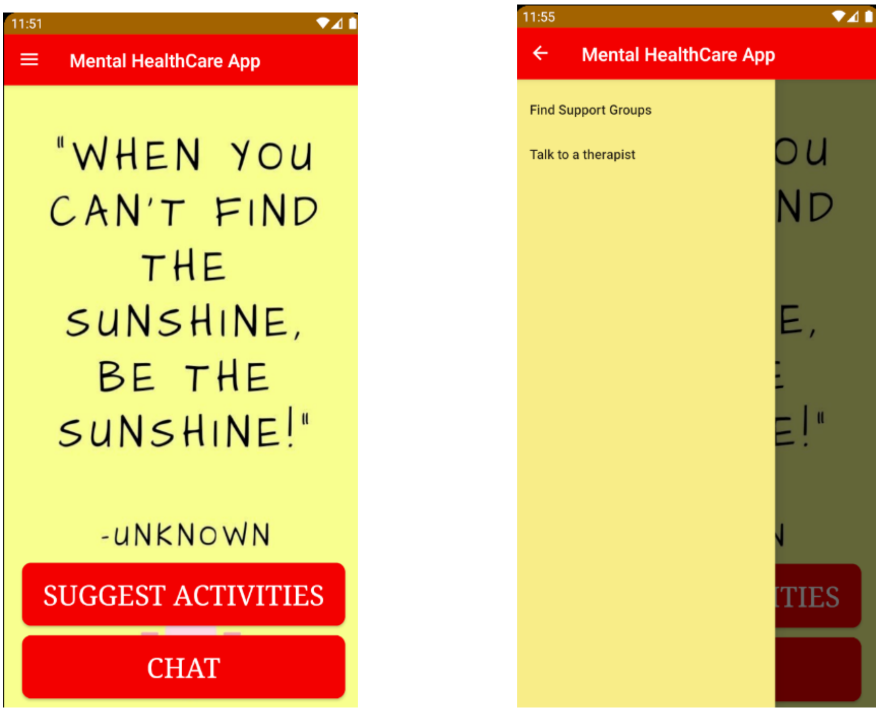
Activities	activity page and click show activity button		
Send a message to the chatbot	Write message in the message window and click send	Message is sent and shown as a balloon in user’s side(right) of chatbox	pass
Receive a message	Wait for response after sending the message	Message balloon is shown on the left side	pass
Find support groups	Click on the Find support groups button in the navigation drawer	User is shown nearest support groups on google	pass

4.2 Result and Screenshots

After successfully debugging and installing the application the final results achieved are shown below.

4.2.1 Launch Activity

This is the home page of the application. It gives the user option for browsing to either the chatbot or the activity recommender. It also has a navigation menu which gives the option for online therapy and finding support groups.



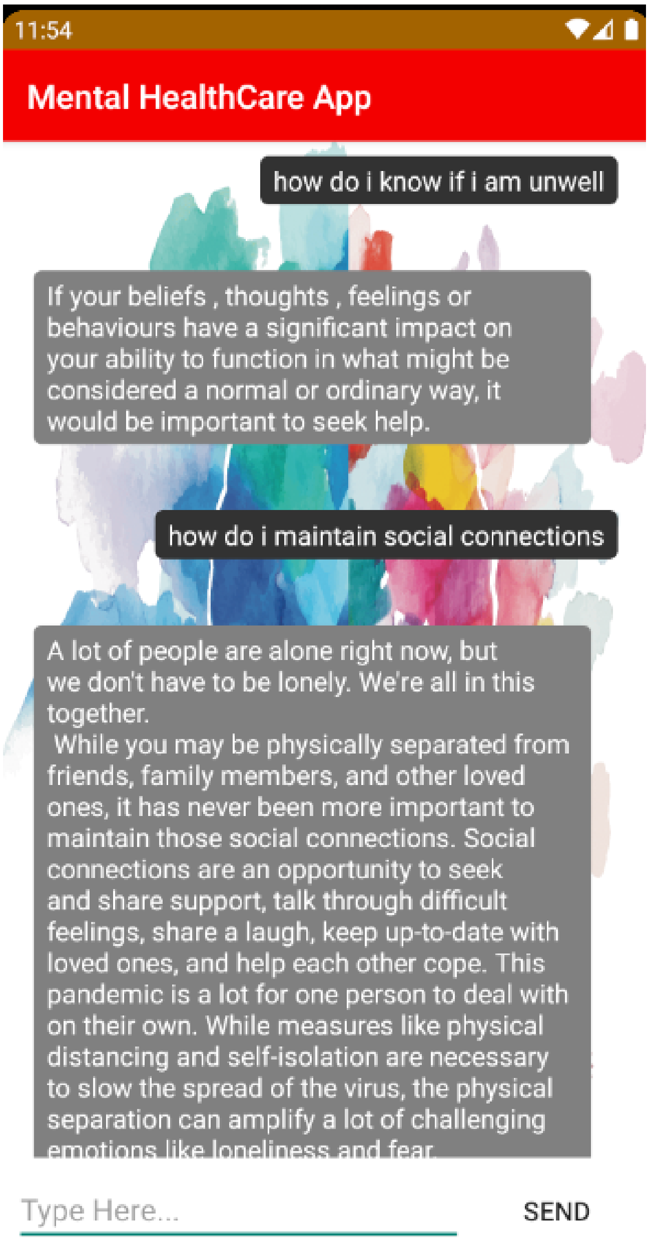
"WHEN YOU CAN'T FIND THE SUNSHINE, BE THE SUNSHINE!"

-UNKNOWN

Fig 4.1 Launch Activity Screenshot

4.2.2 Chat Activity

This is the activity for the chatbot. It gives the user the option to interact with the chatbot by writing their issues in the message window and sending them. The application responds by returning the most likely answer.



how do i know if i am unwell

If your beliefs , thoughts , feelings or behaviours have a significant impact on your ability to function in what might be considered a normal or ordinary way, it would be important to seek help.

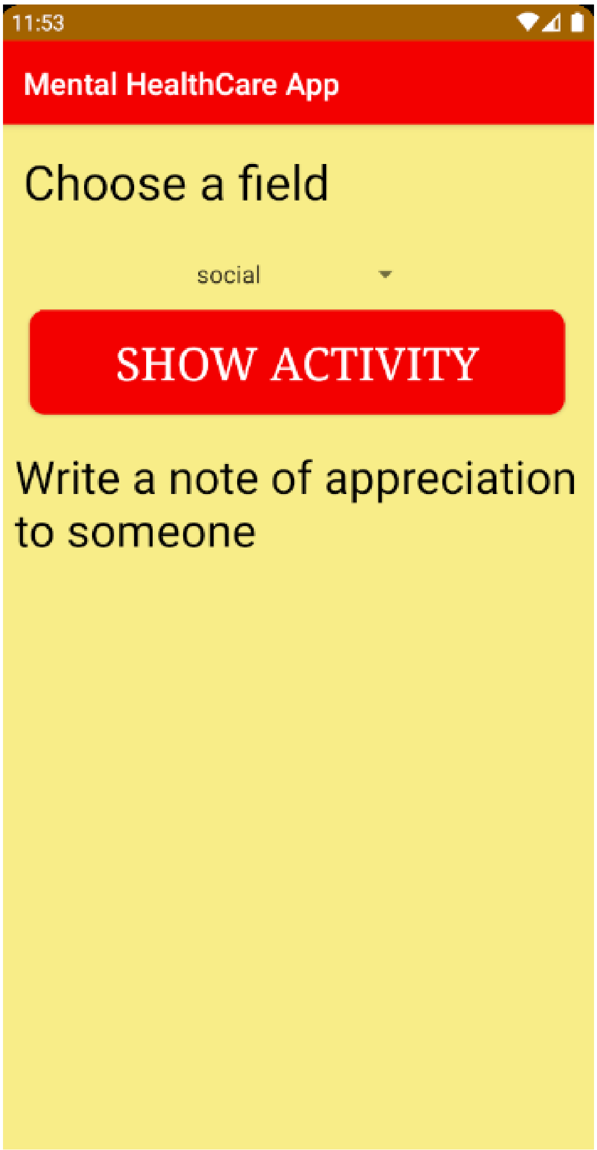
how do i maintain social connections

A lot of people are alone right now, but we don't have to be lonely. We're all in this together.
While you may be physically separated from friends, family members, and other loved ones, it has never been more important to maintain those social connections. Social connections are an opportunity to seek and share support, talk through difficult feelings, share a laugh, keep up-to-date with loved ones, and help each other cope. This pandemic is a lot for one person to deal with on their own. While measures like physical distancing and self-isolation are necessary to slow the spread of the virus, the physical separation can amplify a lot of challenging emotions like loneliness and fear

Fig 4.2 Chat Activity Screenshot

4.2.3 Suggest Activity

This is the activity for recommending recreational tasks. The user has to pick a field they are in and click on the show activity button. The application makes an api call to bored api to suggest activity based on user preference.



social

Write a note of appreciation to someone

Fig 4.3 Suggest Activity Screenshot

4.3 Limitations

The biggest limitation of the project is that the project does not work on a backend web server. All the computations take place on the android itself. Therefore the chatbot had to be made lightweight without recurrent neural network or transformer architecture to be able to work efficiently on the android.

4.4 Conclusion and Future Scope

The principal objective of this Practicum project was to develop a fully functioning project using the existing technologies and the knowledge obtained so far and if needed then further increasing that knowledge. This project helped me in increasing my knowledge about various facets of android studio and python. This application serves as a novice attempt to tackle this pervasive problem which could be a base for further enhancements.

Challenges

During the course of development of the application I encountered several challenges. The major challenge was to try and find a method to integrate the python code for chatbot with android. After looking through several methods including BeeWare, Kivy and QPython. I found chaquopy to work best for my application. I also had to find the most efficient method to look for the most likely response to the query in the dataset.

Future Scope

With the financial, social and personal burden of mental and substance use disorders on the rise globally, the need for effective mental healthcare applications is clearly evident. The emergence of mobile health applications has made mental healthcare more accessible and have opened doors in new directions in mental healthcare.
In the future we could include a daily activity tracker for users to keep track of their objectives thus helping them in achieving their goals of a better and healthier life.
We could also form groups of like minded people based on their hobbies which would further help in tackling the problem of loneliness and isolation.

References

[1] Android Studio Official Documentation https://developer.android.com/docs

[2] Chaquopy Documentation https://chaquo.com/chaquopy/documentation/

[3] Python Documentation https://docs.python.org/3/

[4] Jwala, K., G. N. V. G. Sirisha, and GV Padma Raju. "Developing a Chatbot using Machine Learning." International Journal of Recent Technology and Engineering (IJRTE) 8.1S3 (2019): 89-92. https://www.ijrte.org/wp-content/uploads/papers/v8i1S3/A10170681S319

[5] Kumar, Praveen, et al. "Designing and developing a chatbot using machine learning." 2018 International Conference on System Modeling & Advancement in Research Trends (SMART). IEEE, 2018. https://ieeexplore.ieee.org/document/8746972

[6]Raghav Agarwal, Build a simple Chatbot using NLTK Library in Python Available at: https://www.analyticsvidhya.com/blog/2021/07/build-a-simple-chatbot-using-python-and-nltk/

[7] Amila Viraj, How To Build Your Own Chatbot Using Deep Learning Available at: https://towardsdatascience.com/how-to-build-your-own-chatbot-using-deep-learning-bb41f970e281

[8] How to make Api calls in android studio. Available at: https://www.youtube.com/watch?v=3ZTjH9lEylI

[9] UML Documentation https://www.uml.org/

[10] Umer Farooq, Tools to run Python on Android Available at: https://medium.com/@umerfarooq_26378/tools-to-run-python-on-android-9060663972b4

[11] Natural Language Toolkit Available at: https://www.nltk.org/

[12] Siddhart M., July 2021, A Beginners Hands-on Guide to Natural Language Processing Available at: https://www.analyticsvidhya.com/blog/2021/07/nltk-a-beginners-hands-on-guide-to-natural-language-processing/

[13] Cosine Similarity Available at: http://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.cosine_similarity.html

[14] Data and file storage overview | Android Developers Available at: https://developer.android.com/training/data-storage

Appendix

Github: https://github.com/AbhijitMishra7/Mental-Health-Chatbot