MOBILE APP FOR MONITORING MENTAL HEALTH STATUS

A PROJECT REPORT

Submitted by

KEERTHIGA .S **(Reg. No 421320104016)**

PAVITHRA .K **(Reg. No 421320104024)**

SHARMILA .B **(Reg. No 421320104041)**

SIVATHAVASI .V **(Reg. No 421320104044)**

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE & ENGINEERING

KRISHNASAMY

COLLEGE OF ENGINEERING & TECHNOLOGY

CUDDALORE 607 109

ANNA UNIVERSITY :: CHENNAI 600 025

MAY 2024

ANNA UNIVERSITY :: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report "**MOBILE APP FOR MONITORING MENTAL HEALTH STATUS**" is the bonafide work of **"KEERTHIGA.S (Reg.No.421320104016), PAVITHRA.K (Reg.No.421320104024), SHARMILA.B (Reg.No.421320104041), SIVATHAVASI.V (Reg.No.421320104044)**" who carried out the project work under my supervision.

SIGNATURE

Er. C. REIKHA,M.E., M.B.A.,

Associate Professor

HEAD OF DEPARTMENT

Department of Computer Science & Engineering,

Krishnasamy College of Engineering & Technology,

Cuddalore - 607 109.

SIGNATURE

Er. S. RAMYA,M.E.,

Assistant Professor

SUPERVISOR

Department of Computer Science & Engineering,

Krishnasamy College of Engineering & Technology,

Cuddalore - 607 109.

Submitted for the project Viva - Voce held on ……

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We take this privilege to express a few words of gratitude and respect to all those who helped us in this completion of our project.

We feel honored to place our warm salutation to **Krishnasamy College of Engineering & Technology, Cuddalore and Department of Computer Science & Engineering,** which has given us the opportunity to obtain chaired goal of beckoning professional.

We would like to extend our heartfelt and sincere thanks to our chairman **Dr. K. Rajendran, M.S., F.I.C.S., F.A.I.S.,** who has given us an excellent opportunity to exhibit our mode of creativity by our project.

We would like to avail this opportunity to express our gratitude, sincere and heartfelt thanks to our Principal **Dr. G. Elango, M.E., Ph.D.,**Vice Principal **Dr. K. Raghu, M.Sc., M.Phil., Ph.D.,** and our Administrative Officer **Mr. G. Balakrishnan, M.C.A., M.Phil.,** who has provided all help in executing the project successfully,

We acknowledge our heartfelt and sincere thanks to the Head of the Department, **Er.C.Reikha,M.E., M.B.A.,**Associate Professor, Department of Computer Science& Engineering, who has been so helpful in completing the project at the stipulated time and encouraged us to complete our project successfully. We take this opportunity to express our sincere thanks to our guide **Er. S. Ramya, M.E.,**Assistant Professor, Department of Computer Science & Engineering, who encouraged us in each and every step of this project which helped us to make our task a full-fledged one.

Finally, We would like to use this opportunity to express our sincere thanks to all teaching and non-teaching staff for providing all help for successful completion of our project.

ABSTRACT

In an era where mental health awareness is gaining prominence, technological solutions are emerging to address the growing need for accessible support.

This project introduces a novel application—a mental health status monitor application—that leverages Natural Language Processing (NLP) and Machine Learning (ML) techniques to assess and monitor individuals' mental well-being in real-time.

The application provides a user-friendly interface for individuals to express their emotions, concerns, and experiences confidentially. Through sophisticated algorithms, it analyses linguistic patterns, sentiment, and contextual cues to detect potential indicators of mental health issues such as anxiety, depression, or stress.

Additionally, the application offers personalized recommendations, resources, and coping strategies based on the user's identified needs and preferences. By providing continuous support and early intervention, this innovative tool aims to promote mental wellness, facilitate timely access to mental health services, and ultimately contribute to the improvement of overall mental health outcomes in society.

TABLE OF CONTENT

LIST OF ABBREVIATIONS

AI	:	Artificial Intelligence
GDPR	:	General Data Protection Regulation
GPL	:	General Public License
HIPAA	:	Health Insurance Portability and Accountability Act
MH	:	Mental Health
ML	:	Machine Learning
NLP	:	Natural Language Processing

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

In recent years, mental health has gained significant attention as a critical aspect of overall well-being. The World Health Organization (WHO) estimates that globally, one in four people will be affected by mental or neurological disorders at some point in their lives. Despite the prevalence of mental health issues, there are significant barriers to accessing timely support and treatment, including stigma, limited resources, and a shortage of mental health professionals. However, advancements in technology offer promising solutions to address these challenges and improve mental health outcomes.

One such technological innovation is the development of mental health status monitor applications. These application leverage Artificial Intelligence (AI) technologies, such as Natural Language Processing (NLP) and Machine Learning (ML), to provide real-time support and monitoring of individuals' mental well-being. By utilizing this application, which are accessible through various digital platforms such as smartphones and computers, individuals can access support and resources discreetly and conveniently, without the fear of judgment or stigma.

The purpose of this project is to provide mental health status monitor applications. This application would leverage cutting-edge AI techniques, including deep learning algorithms, to provide more accurate and personalized assessments of users' mental health status. The application would utilize data from various sources, such as social media activity or wearable devices, to provide a comprehensive view of the user's mental well-being over time.

1.2 OBJECTIVE

The main objective of the project is to provide a mental health status monitor application which aims to enhance the existing landscape by incorporating advanced features and capabilities.

It would leverage cutting-edge AI techniques, including deep learning algorithms, to provide more accurate and personalized assessments of users' mental health status.

It analyses users mental status and provides suggestions and activities which improves the mental health status if the user need they can also communicate with the psychologist.

The application would utilize data from various sources, such as social media activity or wearable devices, to provide a comprehensive view of the user's mental well-being over time.

It employs validated assessment tools or questionnaires to screen for symptoms of common mental health disorders such as depression, anxiety, or stress.

It can analyse historical user data and feedback, the machine learning module continuously improves the application performance and accuracy. It can adapt its responses based on individual user preferences, past interactions, and behavioural patterns and works as a best companion application for the user.

CHAPTER 2
LITERATURE REVIEW

2.1 THE WHO SPECIAL INITIATIVE FOR MENTAL HEALTH (2019–2023)

Mental, neurological and substance use disorders account for more than 10% of the global disease burden. The lost productivity resulting from depression and anxiety, two of the most common mental disorders, costs the global economy US$ 1 trillion each year. Commonly, in low- and middle- income countries, 85% of people with mental disorders receive no treatment at all for their disorder. This is why, in 2018, the WHO Director-General, Dr Tedros Adhanom Ghebreyesus, identified mental health as an area for which action should be accelerated. The result was the establishment of the WHO Special Initiative for the Mental Health, covering the 5-year period 2019-2023.

2.2 THE RELATIONSHIP BETWEEN USE OF CBT SKILLS AND DEPRESSION TREATMENT OUTCOME

Cognitive and behavioural therapies emphasize the importance of skill acquisition and use, and these skills are proposed to mediate treatment outcomes. Despite its theoretical importance, research on skill use as a mechanism of change in CBT and its measurement is still in its infancy. A search of online databases was conducted to identify and review the literature testing the meditational effect of CBT skills on treating depression in adults. Additionally, we reviewed the various methods to assess a patient's use of CBT skills. We identified 13 studies examining the frequency of CBT skill use and 11 studies examining the quality of CBT skill use.

2.3 EFFECTIVENESS OF CBT FOR CHILDREN AND ADOLESCENTS WITH DEPRESSION

Cognitive-behavioural therapy (CBT) is first choice of treatment for depressive symptoms and disorders in adolescents, however improvements are necessary because overall efficacy is low. Insights on CBT components and contextual and structural characteristics might increase the efficacy. The aim of our approach is to evaluate the efficacy of CBT for youth with depression and investigate the influence of specific components, contextual and structural factors that could improve effects. Outcomes were meta-analysed and confidence in results was assessed using the GRADE-method. Meta-regression was used to pinpoint components or other factors that were associated with an in- or decrease of effects of CBT.

2.4 WHY COGNITIVE BEHAVIOURAL THERAPY IS THE CURRENT GOLD STANDARD OF PSYCHOTHERAPY.

Taking into account the number of publications/studies, academic programs, and/or practicing professionals, cognitive behavioural therapy (CBT) is arguably the *gold standard* of the psychotherapy field. However, recently, some colleagues have argued for plurality in psychotherapy, questioning the status of CBT as the *gold standard* in psychotherapy (1), because many studies are of low quality and/or the comparator conditions are weak (i.e., wait list rather than active comparators), thus challenging CBT's prominent status among academic programs and practitioners. We think that many issues factor into the *gold-standard* designation. At the same time, there is clearly room for further improvement, both in terms of CBT's efficacy/effectiveness and its underlying theories/mechanisms of change.

CHAPTER 3
PROBLEM DESCRIPTION

3.1 EXISTING SYSTEM

The existing system of mental health status monitor applications encompasses a range of solutions designed to address various aspects of mental well-being. These applications typically utilize artificial intelligence (AI) technologies such as natural language processing (NLP) and machine learning (ML) to interact with users and assess their mental health status.

Some existing applications only focus on providing emotional support and conversation, allowing users to express their feelings and receive empathetic responses in return. Others offer screening tools based on validated psychological assessments to identify symptoms of common mental health conditions like depression, anxiety, or stress.

Additionally, there are chatbots integrated into therapy platforms or mental health services which does not provide a proper health related suggestions, enabling users to access professional support, counselling, or guided self-help interventions remotely. Overall, the existing system reflects a growing trend towards leveraging technology to democratize mental health care, decreases accessibility, and does not provide timely support to individuals seeking assistance for their mental well-being.

3.1.1 DISADVANTAGES

- In existing system, It does not provide a proper data security, it deal with very sensitive personal information.
- Due to accessibility issues, users struggle to use the application.
- It does not provides customization options so the users are less engaged with the application.

3.2 PROPOSED SYSTEM

The proposed system for a mental health status monitor application aims to enhance the existing landscape by incorporating advanced features and capabilities. This system would leverage cutting-edge AI techniques, including deep learning algorithms, to provide more accurate and personalized assessments of users' mental health status.

Unlike traditional application, the proposed system can analyses users mental status and provides suggestions and activities which improves the mental health status if the user need they can also communicate with the psychologist using the provided contact details.

Furthermore, the application would utilize data from various sources, such as social media activity or wearable devices, to provide a comprehensive view of the user's mental well-being over time.

Additionally, the proposed system would prioritize user privacy and data security, implementing robust encryption protocols and ensuring compliance with relevant regulations such as GDPR or HIPAA. By offering proactive monitoring, timely interventions, and tailored support resources, the proposed system aims to empower users to better manage their mental health and improve overall well-being in the digital age.

3.2.1 ADVANTAGES

- The proposed system would prioritize user privacy and data security.
- It provide more accurate and personalized assessments of users' mental health status.
- The application would utilize data from various sources and provide a comprehensive view of the user's mental health status.

CHAPTER 4
SYSTEM REQUIREMENT SPECIFICATION

4.1 SOFTWARE REQUIREMENT

- **Operating system** : Windows 10
- **Coding Language** : Python
- **Database** : SQLite

4.2 HARDWARE REQUIREMENT

- **Processor** : Intel i3 processor based or higher
- **Hard Disk** : 10 GB (minimum)
- **Ram** : 6 GB (minimum)

CHAPTER 5
SOFTWARE DESCRIPTION

5.1 PYTHON

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive** − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido

van Rossum still holds a vital role in directing its progress.

Python Features

Python's features include −

- **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.
- **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** − Python provides interfaces to all major commercial databases.
- **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** − Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below −

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type

checking.

- It supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

Python is available on a wide variety of platforms including Linux and Mac OS Let's understand how to set up our Python environment.

Getting Python

The most up-to-date and current source code, binaries, documentation, news, etc., is available on the official website of Python.

Windows Installation

Here are the steps to install Python on Windows machine.

- Windows installer python-XYZ.msifile where XYZ is the version you need to install.
- To use this installer python-XYZ.msi, the Windows system must support Microsoft Installer 2.0. Save the installer file to your local machine and then run it to find out if your machine supports MSI.
- Run the downloaded file. This brings up the Python install wizard, which is really easy to use. Just accept the default settings, wait until the install is finished, and you are done.

The Python language has many similarities to Perl, C, and Java. However, there are some definite differences between the languages.

5.2 SQLite

SQLite is an in-process library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. It is a popular choice as an embedded database for local/client storage in application software such as web browsers. It is also used in many other applications that need a lightweight, embedded database. JSP seamlessly integrates with other Java EE (Enterprise

Edition) technologies, such as Servlets, JDBC (Java Database Connectivity), and EJBs (Enterprise JavaBeans). This enables developers to build robust, enterprise-grade web applications.

JSP supports the use of custom tag libraries, which encapsulate reusable components and functionality. Tag libraries abstract complex tasks into simple

SQLite History

SQLite was created in the year 2000 by D. Richard Hipp, who continues to lead the development of the software today. SQLite was designed to be a lightweight and simple database engine that could be easily embedded into other applications. It was created as an alternative to more complex and heavyweight database engines, such as MySQL and PostgreSQL. Over the years, SQLite has gained widespread adoption and is now one of the most widely used database engines in the world. It is used in many applications, including web browsers, mobile phones, and a wide variety of other software.

Why SQLite?

1. **Ease of use:** SQLite is very easy to get started with, as it requires no setup or configuration. You can simply include the library in your project and start using it.
2. **Embeddability:** SQLite is designed to be embedded into other applications. It is a self-contained, serverless database engine, which means you can include it in your application without the need for a separate database server.
3. **Lightweight:** SQLite is a very lightweight database engine, with a small library size (typically less than 1MB). This makes it well-suited for use in applications where the database is embedded directly into the application binary, such as mobile apps.
4. **Serverless:** As mentioned earlier, SQLite is a serverless database engine,

which means there is no need to set up and maintain a separate database server process. This makes it easy to deploy and manage, as there are no additional dependencies to worry about.

5. **Cross-platform:** SQLite is available on many platforms, including Linux, macOS, and Windows, making it a good choice for cross-platform development.
6. **Standalone:** SQLite stores all of the data in a single file on the filesystem, which makes it easy to copy or backup the database.
7. **High reliability:** SQLite has been widely tested and used in production systems for many years, and has a reputation for being a reliable and robust database engine.

An easy way to get started with SQLite

Because SQLite is an embedded database, you actually don't need to 'download' it in the same way that you would download MySQL or PostgreSQL for example. You can create and interact with SQLite databases using a range of tools.

An easy way to get started would be to:

1. Download an example SQLite dataset
2. Download a GUI program to access the database, like Beekeeper Studio, or DBeaver.
3. Double-click the '.db' file to open it.
4. Using these tools, you can navigate your SQLite file in a spreadsheet-like way.

Features of SQLite

1. The transactions follow ACID properties i.e. atomicity, consistency, isolation, and durability even after system crashes and power failures.
2. The configuration process is very easy, no setup or administration is needed.
3. All the features of SQL are implemented in it with some additional features like partial indexes, indexes on expressions, JSON, and common table expressions.
4. Sometimes it is faster than the direct file system I/O.
5. It supports terabyte-sized databases and gigabyte-sized strings and blobs.
6. Almost all OS supports SQLite like Android, BSD, iOS, Linux, Mac, Solaris, VxWorks, and Windows (Win32, WinCE, etc. It is very much easy to port to other systems.
7. A complete database can be stored in a single cross-platform disk file**.**

Applications of SQLite

1. Due to its small code print and efficient usage of memory, it is the popular choice for the database engine in cell phones, PDAs, MP3 players, set-top boxes, and other electronic gadgets.
2. It is used as an alternative for open to writing XML, JSON, CSV, or some proprietary format into disk files used by the application.
3. As it has no complication for configuration and easily stores file in an ordinary disk file, so it can be used as a database for small to medium sized websites.
4. It is faster and accessible through a wide variety of third-party tools, so it has great applications in different software platforms.

CHAPTER 6
SYSTEM DESIGN

6.1 SYSTEM ARCHITECTURE

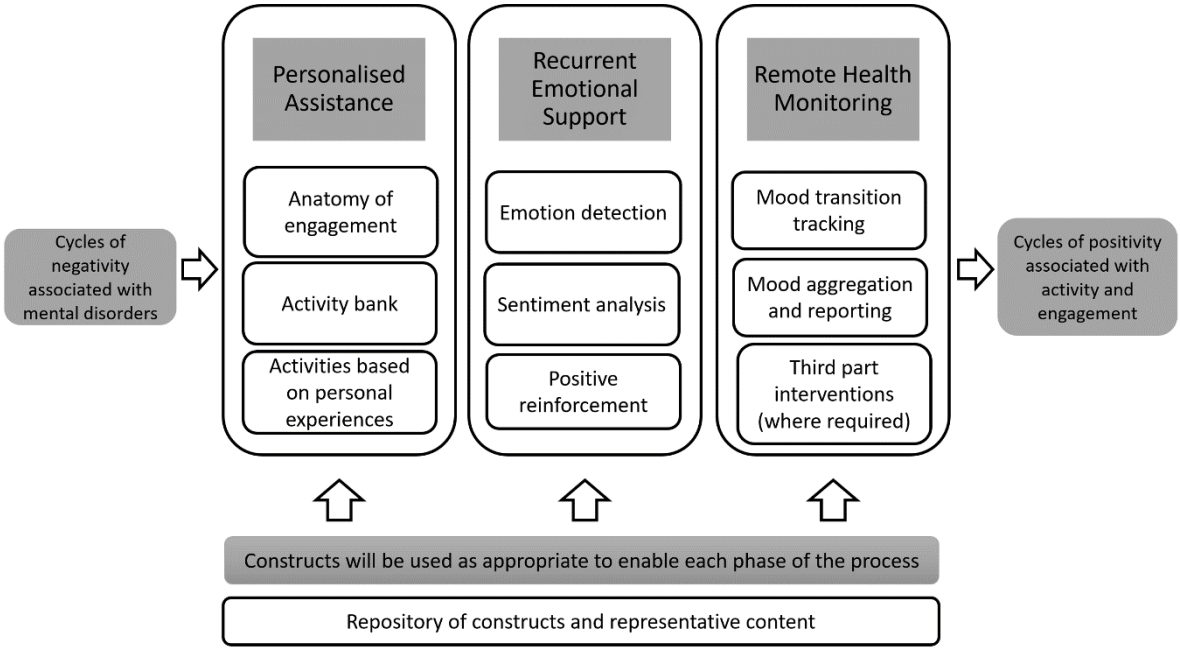


Figure 6.1 Architecture Diagram

Figure 6.1 illustrates that, these modules collaborate to facilitate application which typically utilize AI technologies such as NLP and ML to interact with users and assess their mental health status. Application focus on providing emotional support and conversation, allowing users to express their feelings and receive empathetic responses in return. And also offer screening tools based on validated psychological assessments to identify symptoms of common mental health conditions like depression, anxiety, or stress. It integrated into therapy platforms or mental health services, enabling users to access professional support, counselling, or guided self-help interventions remotely.

CHAPTER 7
MODULE DESCRIPTION

7.1 LIST OF MODULES

- Login/Register Module
- Mental Health Assessment Module
- Resource and Recommendation Module
- Feedback and Monitoring Module
- Psychologist Contact Module

7.1.1 LOGIN / REGISTER MODULE

- **Description:** This module serves as the front-end interface through which users interact with the application. It includes components such as Login or Registering page using email and password.
- **Functionality:** Users can easily login into their account to get a regular mental health status.

7.1.2 MENTAL HEALTH ASSESSMENT MODULE

- **Description:** This module evaluates the user's mental health status based on the input provided.
- **Functionality:** It employs validated assessment tools or questionnaires to screen for symptoms of common mental health disorders such as depression, anxiety, or stress. The module may also consider additional factors such as sleep patterns, social interactions, and lifestyle habits to provide a comprehensive assessment.

7.1.3 RESOURCE AND RECOMMENDATION MODULE

- **Description:** This module provides users with relevant resources, coping strategies, and recommendations based on their mental health assessment.
- **Functionality:** It offers tailored suggestions for self-care activities, relaxation techniques, mental health apps, support groups, or professional services such as therapy or counseling. The module may also integrate with external databases or APIs to fetch up-to-date information and recommendations.

7.1.4 FEEDBACK AND MONITORING MODULE

- **Description:** This module collects feedback from users and monitors their progress over time.
- **Functionality:** It prompts users to provide feedback on the application responses and effectiveness. It also tracks changes in the user's mental health status, mood fluctuations, or adherence to recommended strategies. This module can generate reports or insights for users to review their progress and identify areas for improvement.

7.1.5 PSYCHOLOGIST CONTACT MODULE

- **Description:** This module ensures the users psychological needs though communication with psychologists.
- **Functionality:** It implements psychologists contact details that if the user needs urgent psychologist help then he or she can visit this page.

CHAPTER 8
UML DIAGRAMS

8.1 UML DIAGRAM DESCRIPTION

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

8.1.1 DATAFLOW DIAGRAM

The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.

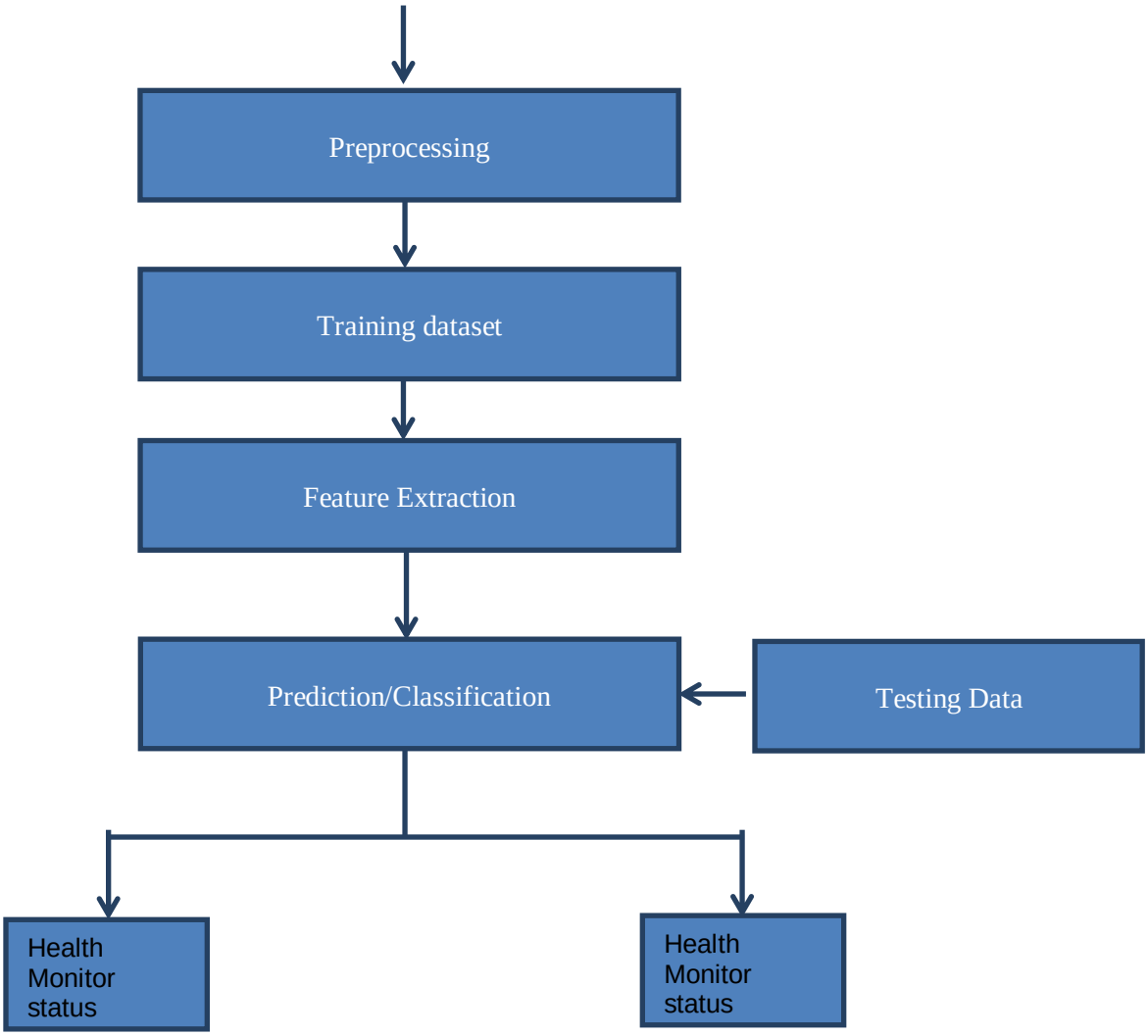


Figure 8.1.1 Data Flow Diagram

8.1.2 USECASE DIAGRAM

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.

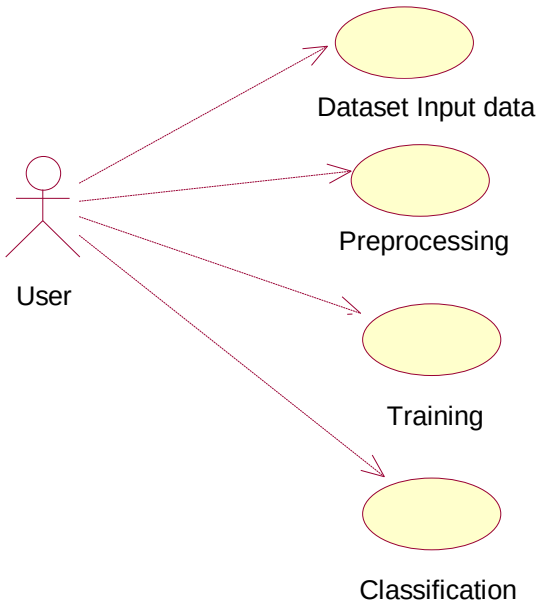


Figure 8.1.2 Use Case Diagram

8.1.3 CLASS DIAGRAM

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods).

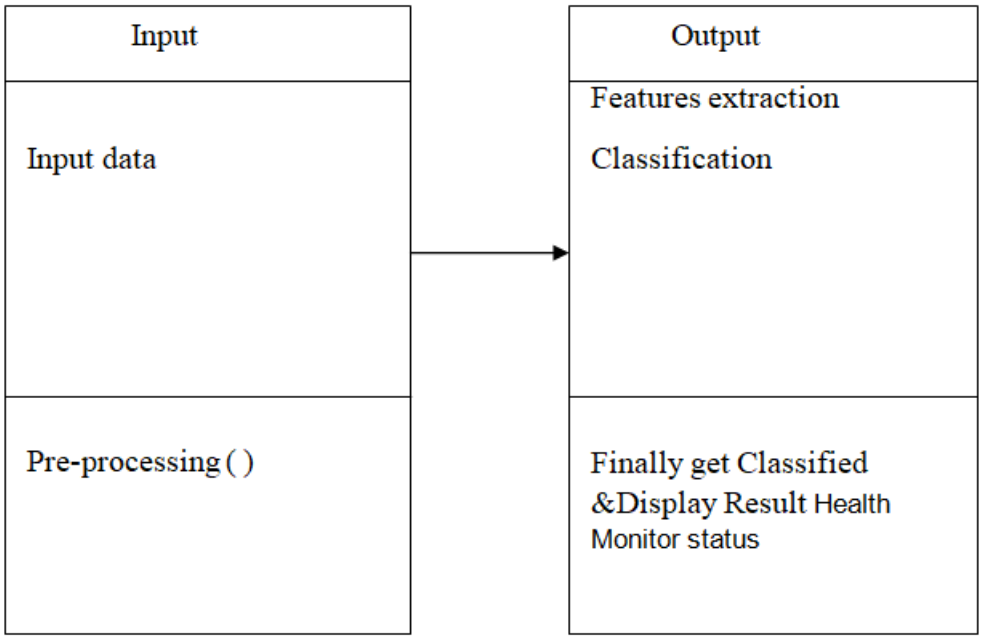


Figure 8.1.3 Class Diagram

8.1.4 SEQUENCE DIAGRAM

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagram are,

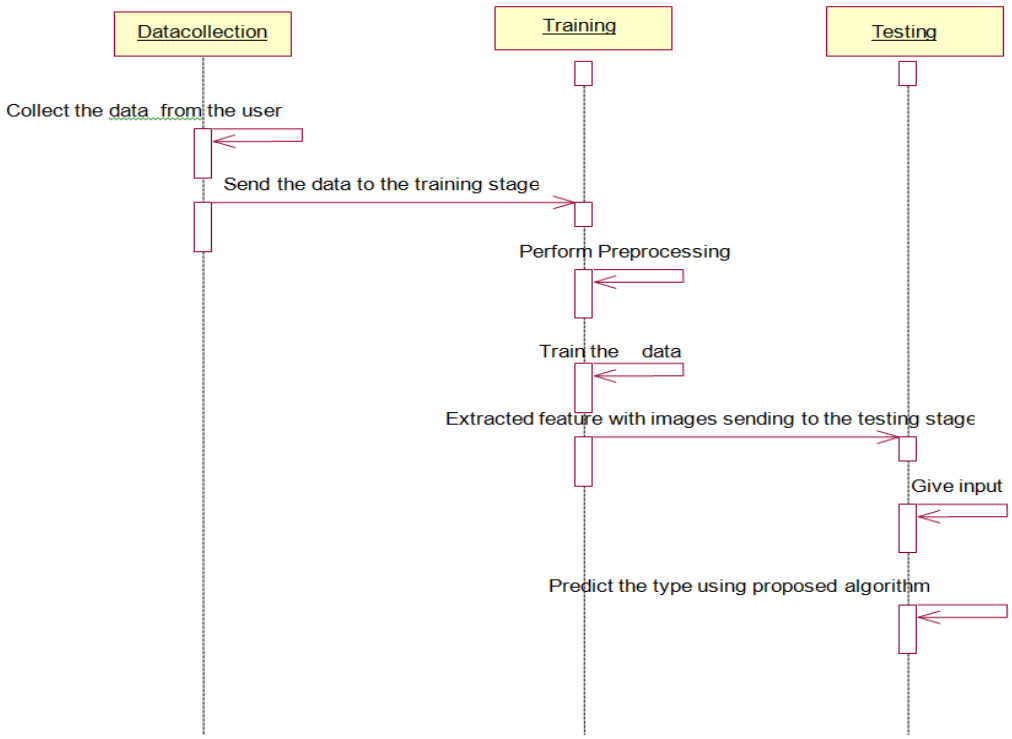


Figure 8.1.4 Sequence Diagram

8.1.5 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

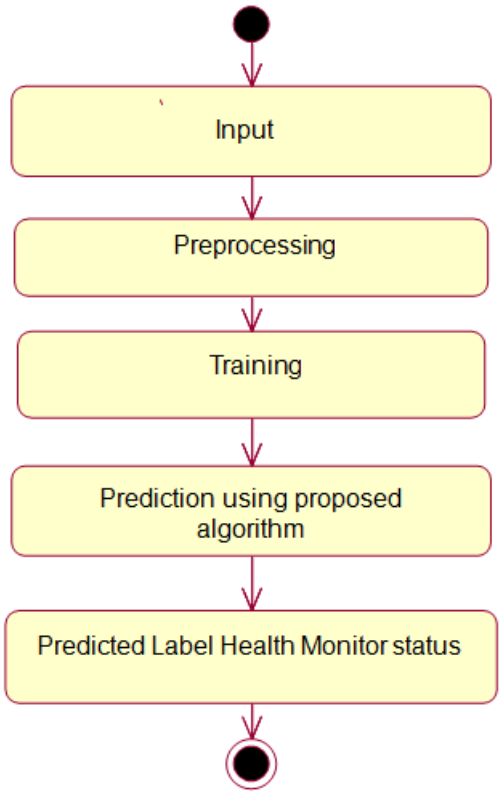


Figure 8.1.5 Activity Diagram

CHAPTER 9

SYSTEM TESTING & FEASIBILITY STUDY

9.1 SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

9.2 TYPES OF TESTS

9.2.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

9.2.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

9.2.3 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

9.2.4 WHITE BOX TESTING

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

9.2.5 BLACK BOX TESTING

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

9.2.6 ACCEPTANCE TESTING

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

Test Results

All the test cases mentioned above passed successfully. No defects encountered.

9.3 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are,

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

9.3.1 ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

9.3.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

9.3.3 SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

CHAPTER 10

CONCLUSION & FUTURE ENHANCEMENT

10.1 CONCLUSION

In conclusion, mental health status monitor chatbot applications represent a promising avenue for improving access to mental health support and monitoring. By harnessing the power of artificial intelligence, these chatbots offer a convenient and discreet way for individuals to engage with mental health resources and receive personalized assistance. While there are challenges and ethical considerations to address, including data privacy and algorithmic biases, the potential benefits of chatbots in mental health care are significant. With further research, development, and collaboration between technologists, mental health professionals, and policymakers, mental health status monitor chatbots have the potential to play a vital role in promoting mental well-being, reducing stigma, and ultimately improving mental health outcomes for individuals worldwide.

10.2 FUTURE ENHANCEMENT

Mobile apps have significant potential to deliver high-efficacy mental health interventions. Given the global shortage of psychiatrists and the lack of mental health care access in rural regions, apps have emerged as a viable tool to bridge the mental health treatment gap. Technology is well-poised to transform how mental health treatment is delivered and accessed, but this transformation requires the combined mobilization of science, regulation, and design. AI will give us the ability to start practicing preventative mental health interventions. Oftentimes in mental health care we are limited to reacting to severe outcomes like suicidal behavior. With AI in future, it’s likely we will arrive at indicators that are reliable and tested across a lot of data sources to identify when a person is at very high risk for depression. Then we can intervene earlier.

APPENDIX 1

SOURCE CODE

app.py

```
from flask import Flask, render_template, request
from chatterbot import ChatBot
from chatterbot.trainers import ChatterBotCorpusTrainer
import os
from chatterbot import ChatBot
from chatterbot.trainers import ListTrainer
filenumber=int(os.listdir('saved_conversations')[-1])
filenumber=filenumber+1
file= open('saved_conversations/'+str(filenumber),"w+")
file.write('bot : Hi There! I am a medical chatbot. You can begin conversation by typing in a message and pressing enter.\n')
file.close()
app = Flask(__name__)
english_bot = ChatBot('Bot',
            storage_adapter='chatterbot.storage.SQLStorageAdapter',
            logic_adapters=[
   {
      'import_path': 'chatterbot.logic.BestMatch'
   },
],
trainer='chatterbot.trainers.ListTrainer')
english_bot.set_trainer(ListTrainer)
@app.route("/")
```

```
def home():
    return render_template("index.html")
@app.route("/get")
def get_bot_response():
    userText = request.args.get('msg')
    response = str(english_bot.get_response(userText))
    appendfile=os.listdir('saved_conversations')[-1]
    appendfile= open('saved_conversations/'+str(filenumber),"a")
    appendfile.write('user : '+userText+'\n')
    appendfile.write('bot : '+response+'\n')
    appendfile.close()
    return response
if __name__ == "__main__":
    app.run()
```

train.py

```
from chatterbot import ChatBot, utils
from chatterbot.trainers import ListTrainer
import os
try:
    os.remove("db.sqlite3")
    print("Old database removed. Training new database")
except FileNotFoundError:
    print('No database found. Creating new database.')
english_bot = ChatBot('Bot')
english_bot.set_trainer(ListTrainer)
for file in os.listdir('data'):
    print('Training using ' + file)
```

```
    convData = open('data/' + file, 'r').readlines()
    english_bot.train(convData)
    print("Training completed for " + file)
```

launch.json

```
    "version": "0.2.0",
    "configurations": [
        {
            "name": "Python: Flask",
            "type": "python",
            "request": "launch",
            "module": "flask",
            "env": {
                "FLASK_APP": "app.py",
                "FLASK_DEBUG": "1"
            },
            "args": [
                "run",
                "--no-debugger",
                "--no-reload"
            ],
            "jinja": true,
            "justMyCode": true
        }
    ]
}
```

Loginactivity.java

```
package com.example.moody;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.Button;
public class loginActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        Button signIn = findViewById(R.id.signinBtn);
    }
}
```

Mainactivity.java

```
package com.example.moody;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import com.firebase.ui.auth.AuthUI;
import java.util.Arrays;
import java.util.List;
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

Moodycalenderactivity.java

```
package com.example.moody;
import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity;
public class MoodCalendarActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_moodcalendar);
    }
}
```

Profileactivity.java

```
package com.example.moody;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import com.firebase.ui.auth.AuthUI;
import java.util.Arrays;
import java.util.List;
import android.os.Bundle;
public class ProfileActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_profile);
    }
}
```

Ratemymoodactivity.java

```
package com.example.moody;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
public class RateMyMoodActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_ratemymood);
    }
}
```

Activitylist.java

```
package com.example.moody;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
public class FriendsListActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_friendslist);
    }
}
```

Homeactivity.java

```
package com.example.moody;
import android.content.Intent;
```

```
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import androidx.appcompat.app.AppCompatActivity;
public class HomeActivity extends AppCompatActivity {
    Button profilebtn;
    Button friendsbtn;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_home);
        profilebtn = findViewById(R.id.HomePageToProfileBtn);
        friendsbtn = findViewById(R.id.btnFriends);
        //Event listener to direct user to their profile
        profilebtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                launchProfilePage();
            }
        });
        //Event listener to direct user to their friendslist
        friendsbtn.setOnClickListener(new View.OnClickListener(){
            @Override
            public void onClick(View v) {
                launchFriendsList();
            }
        });
    }
```

```
    private void launchFriendsList() {
        Intent i = new Intent(HomeActivity.this, FriendsListActivity.class);
        startActivity(i);
    }
    //Creates intent to take user to their profile
    private void launchProfilePage() {
        Intent i = new Intent(HomeActivity.this, ProfileActivity.class);
        startActivity(i);
    }
}
```

Exampleunittext.java

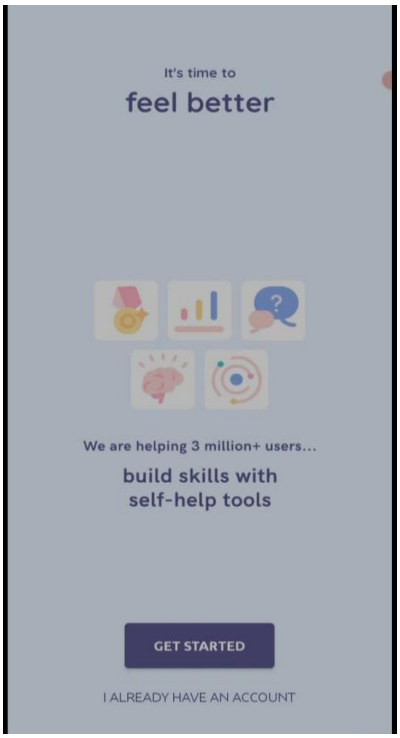
```
package com.example.moody;
import org.junit.Test;
import static org.junit.Assert.*;
/**
 * Example local unit test, which will execute on the development machine (host).
 *
 * @see <a href="http://d.android.com/tools/testing">Testing documentation</a>
 */
public class ExampleUnitTest {
    @Test
    public void addition_isCorrect() {
        assertEquals(4, 2 + 2);
    }
}
```

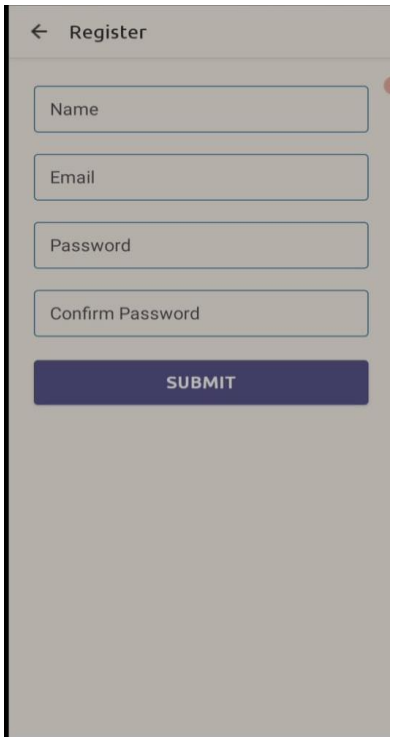
ActivityMain.xml

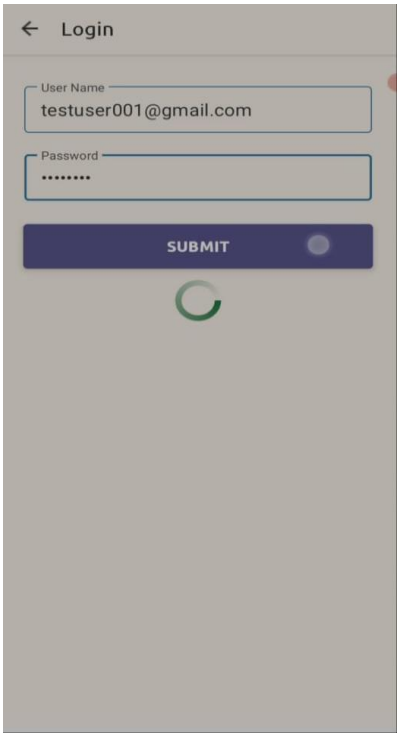
```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.moody">
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.Moody">
        <activity android:name=".loginActivity">
        </activity>
        <activity android:name=".HomeActivity">
            <intent-filter>
                        <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".ProfileActivity">
        </activity>
        <activity android:name=".MoodCalendarActivity">
        </activity>
        <activity android:name=".RateMyMoodActivity">
        </activity>
        <activity android:name=".MainActivity">
        </activity>
    </application>
</manifest>
```

APPENDIX 2
SCREENSHOTS

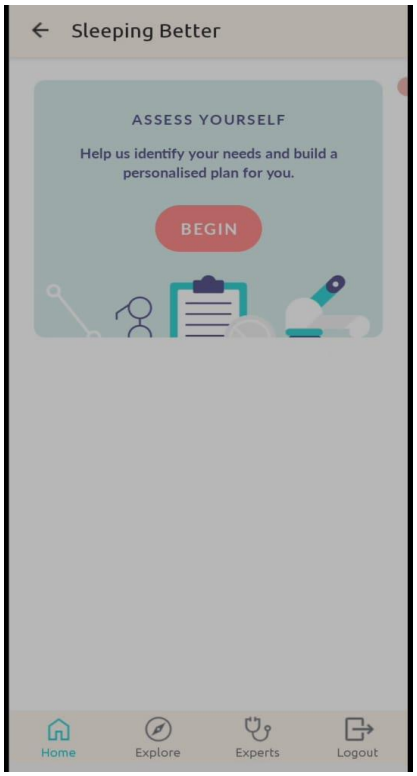
Register / Signup page

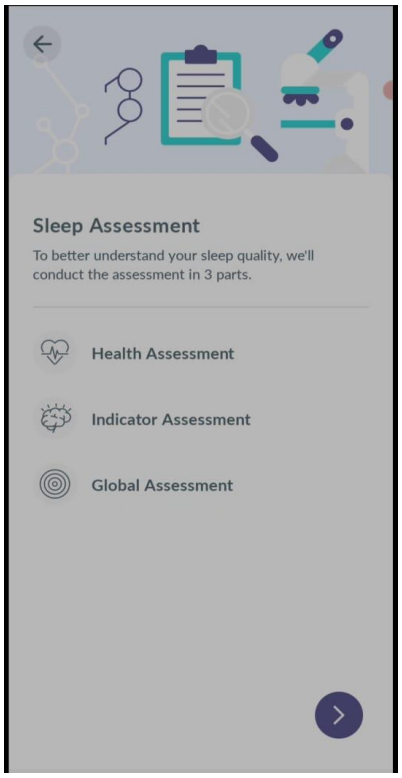


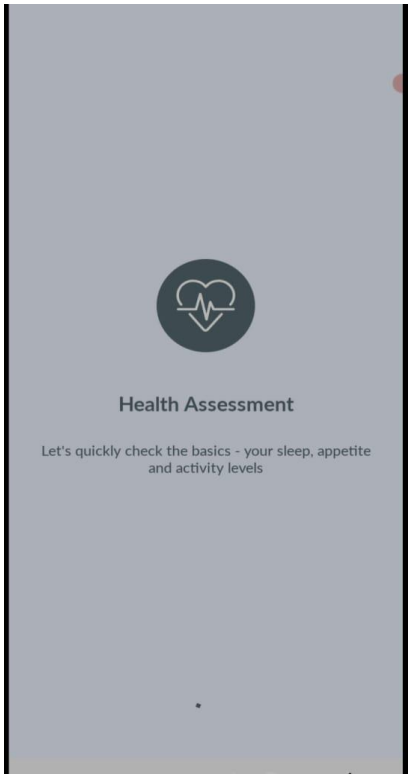


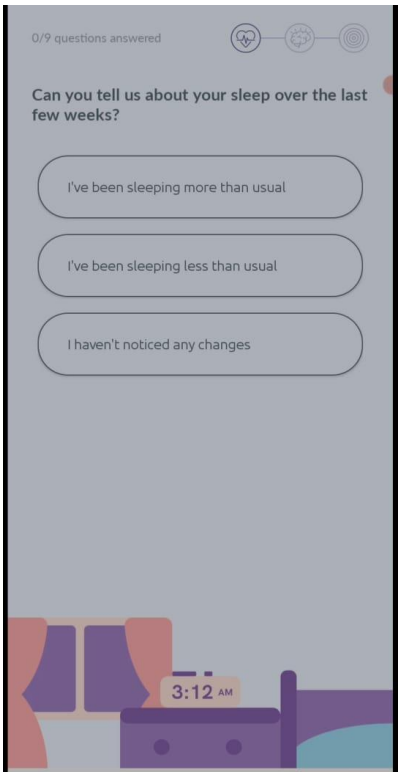


Assessment page

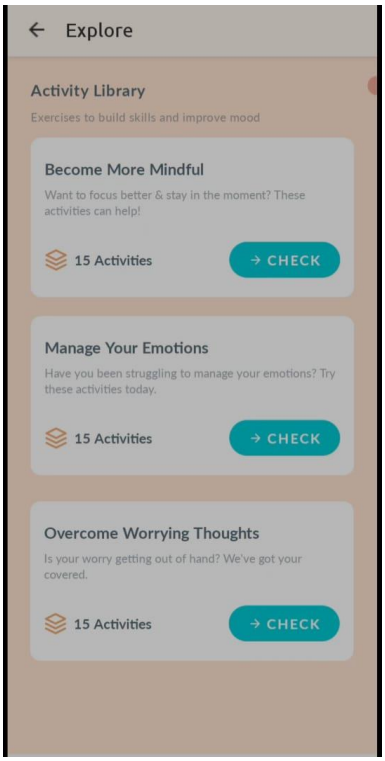


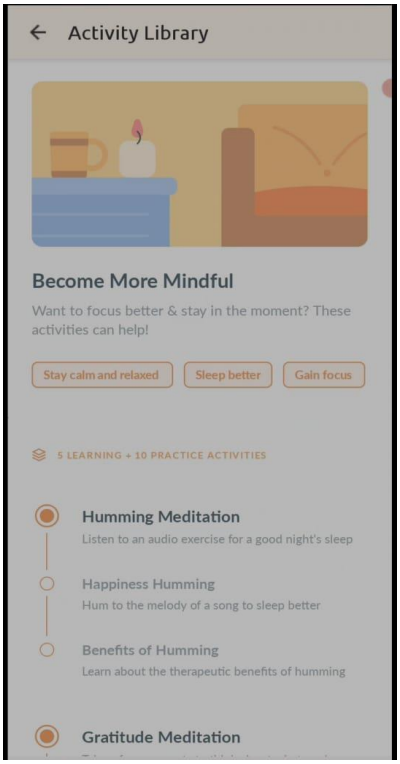


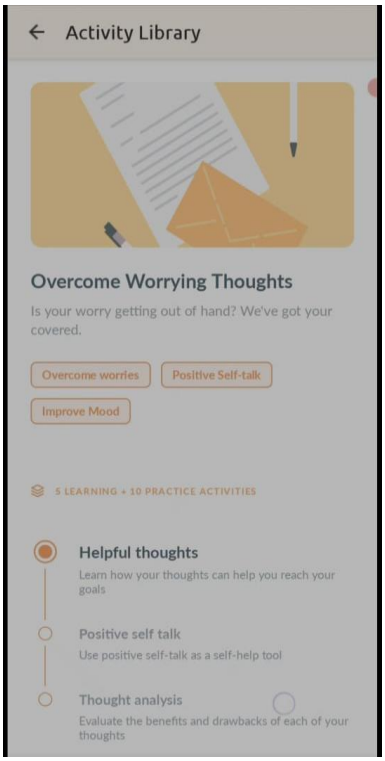


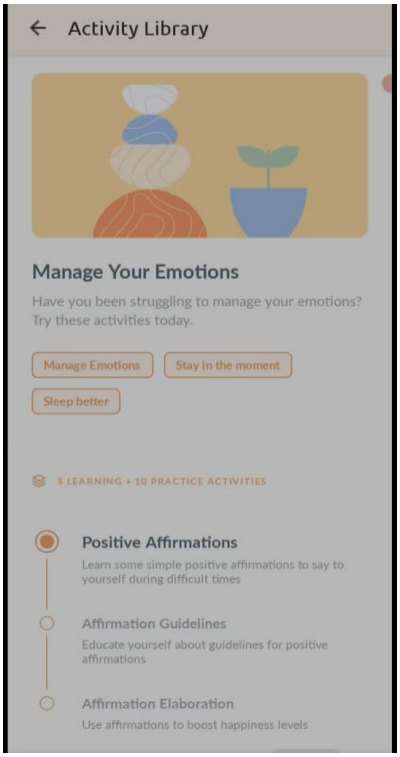


Suggestion page

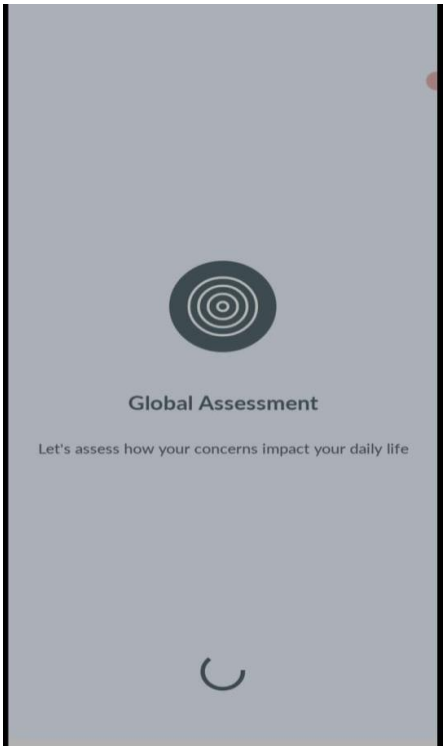


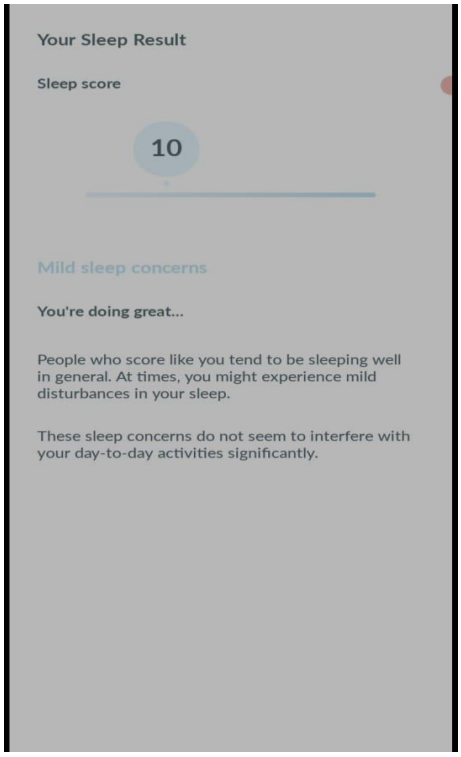


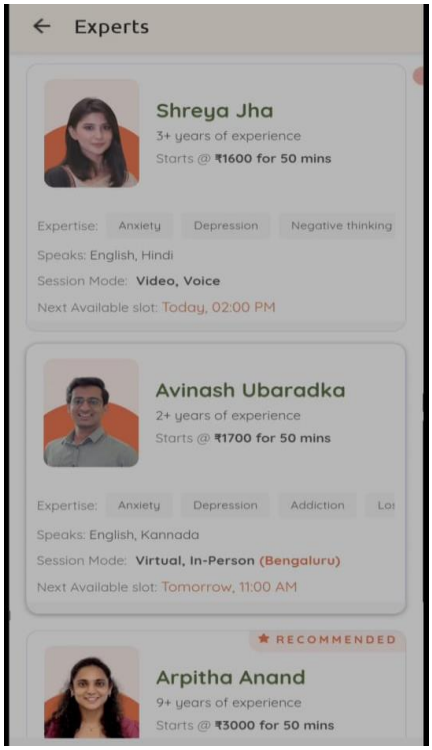




Progress Report & Psychologist Contact page







REFERENCES

[1]T.L.G. Mental health matters. Lancet Glob. Health 2020, 8, e1352. [Google Scholar] [CrossRef].

[2] World Health Organization. The WHO Special Initiative for Mental Health Technical Documents; World Health Organization: Geneva, Switzerland, 2019. [Google Scholar]WHO.

[3] International Classification of Diseases; WHO [Internet]: Geneva, Switzerland, 1992. [Google Scholar].

[4] [Google Scholar]Arch, J.J.; Eifert, G.H.; Davies, C.; Vilardaga, J.C.P.; Rose, R.D.; Craske, M.G. Randomized clinical trial of cognitive behavioural therapy (CBT).

[5] Di Mauro, J.; Domingues, J.; Fernandez, G.; Tolin, D.F. Long-term effectiveness of CBT for anxiety disorders in an adult outpatient clinic sample: A follow-up study.

[6] Hundt, N.E.; Mignogna, J.; Underhill, C.; Cully, J.A. The relationship between use of CBT skills and depression treatment outcome.

[7] M.; De Winter, L.; Vermeulen-Smit, E.; Bodden, D.; Nauta, M.; Stone, L.; Van Den Heuvel, M.; Al Taher, R.; De Graaf, I.; Kendall, T.; et al. Effectiveness of CBT for children and adolescents with depression.

[8] Sigurvinsdóttir, A.L.; Jensínudóttir, K.B.; Baldvinsdóttir, K.D.; Smárason, O.; Skarphedinsson, G. Effectiveness of cognitive behavioural therapy (CBT) for child and adolescent anxiety disorders across different CBT modalities and comparisons.

[9] David, D.; Cristea, I.; Hofmann, S.G. Why cognitive behavioural therapy is the current gold standard of psychotherapy.

[10] Thase, M.E.; Wright, J.H.; Eells, T.D.; Barrett, M.S.; Wisniewski, S.R.; Balasubramani, G.; McCrone, P.; Brown, G.K.

[11] Watts, S.; Mackenzie, A.; Thomas, C.; Griskaitis, A.; Mewton, L.; Williams, A.; Andrews, G. CBT for depression.

[12] Andersson, G. The Internet and CBT: A Clinical Guide; CRC Press: Boca Raton, FL, USA, 2014. [Google Scholar]Lukas, C.A.; Eskofier, B.; Berking, M. A gamified smartphone-based intervention for depression: Randomized controlled pilot trial.

[13] Fitzpatrick, K.K.; Darcy, A.; Vierhile, M. Delivering cognitive behaviour therapy to young adults with symptoms of depression and anxiety using a fully automated conversational agent (Woebot).

[14] Demirci, H.M. User Experience over Time with Conversational Agents: Case Study of Woebot on Supporting Subjective Well-Being. Master's Thesis, Middle East Technical University, Ankara, Turkey, 2018. [Google Scholar].

[15] Prochaska, J.J.; Vogel, E.A.; Chieng, A.; Kendra, M.; Baiocchi, M.; Pajarito, S.; Robinson, A. A therapeutic relational agent for reducing problematic substance use (Woebot).

[16] Monnier, D. Woebot: A continuation of and an end to psychotherapy? Psychotherapies 2020, 40, 71–78.

[17] Newitz, A. I bonded with a robot. New Sci. 2022, 253, 22. [Google Scholar] [CrossRef].

[18] Campbell-Sills, L.; Liverant, G.I.; Brown, T.A. Psychometric evaluation of the behavioural inhibition/behavioural activation scales in a large sample of outpatients with anxiety and mood disorders.

[19] Richards, D.A. Cost and Outcome of Behavioural Activation versus Cognitive Behavioural Therapy for Depression (COBRA).

[20] Jacobson, N.S.; Martell, C.R.; Dimidjian, S. Behavioral activation treatment for depression: Returning to contextual roots.