BSCS-F22-30

03-134192-023 Muhammad Attiq Ur Rehman

03-134182-115 Aizaz Ahmed

Madad

In partial fulfilment of the requirements for the degree of

Bachelor of Science in Computer Science

Supervisor: Zupash Awais

Department of Computer Sciences

Bahria University, Lahore Campus

June 2023

C e r t i f i c a t e

We accept the work contained in the report titled

"Madad"

written by

Muhammad Attiq Ur Rehman

Aizaz Ahmed

as a confirmation to the required standard for the partial fulfilment of the degree of Bachelor of Science in Computer Science.

Approved by:

Supervisor: Zupash Awais

_____ (Signature)

June 20, 2023

DECLARATION

We hereby declare that this project report is based on our original work except for citations and quotations which have been duly acknowledged. We also declare that it has not been previously and concurrently submitted for any other degree or award at Bahria University or other institutions.

Enrolment	Name	Signature
03-134192-023	Muhammad Attiq Ur Rehman	
03-134182-115	Aizaz Ahmed	

Date : June 20, 2023

Specially dedicated to
my beloved grandmother, mother and father
(Muhammad Attiq Ur Rehman)
my beloved grandmother, mother and father
(Aizaz Ahmed)

ACKNOWLEDGEMENTS

We would like to thank everyone who had contributed to the successful completion of this project. We would like to express our gratitude to our research supervisor, Miss Zupash Awais for her invaluable advice, guidance, and her enormous patience throughout the development of the research.

In addition, we would also like to express my gratitude to our loving parent and friends who had helped and given us encouragement.

Muhammad Attiq Ur Rehman
Aizaz Ahmed

Madad

ABSTRACT

Mental health and illness are major health issues, affecting public health in all over the world. There is variety of mental health challenges such as depression, anxiety and eating disorders etc. Commonly People have shyness and hesitation while discussing their mental conditions with others because they fear that other people will label them as mental or psycho. To overcome this issue, we are providing individuals access to a platform where they can quickly assess state of their mental health by simply responding to a few questions which is a psychological test, they can easily evaluate their mental health situation and will be able to discover whether they are experiencing mental health illness or not through their test's score. We are going to develop an flutter application for those who are dealing with mental health issues. If they want to consult a psychologist, based on their results this platform will be helpful for them because it will allow them to contact with professional psychologist, they can contact them directly to deal with their mental health issues via application without any interruption. It will be helpful in giving concise information to the doctor that give them idea about patient. The agile model and the feature-driven development (FDD) process will be used to create this application.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Background

"Mental health is a state of mental well-being that enables people to cope with the stresses of life, realise their abilities, learn and work well, and contribute to their community[1]." Excessive stress brought on by a specific circumstance or series of events may be a contributing factor in mental health issues. Mental health issues are influenced by a variety of factors, including biological ones like heredity or brain chemistry. Traumatic or abusive experiences in life, as well as a family history of mental health issues [2].

If people seek treatment, mental health disorders are treatable. According to specialists, developing effective coping mechanisms is crucial for maintaining your general health. Each person reacts to stress in their own unique way. We must change our perceptions because there are times when the only things we can control are how we perceive events and how we choose to react to them. Increasing physical activity, talking to someone, and making time for someone are a few useful methods for managing an illness.

People with mental health concerns are generally embarrassed and hesitant to talk about their conditions because doing so frequently results in people labelling them crazy or psycho. We will create a mobile application to address this issue. The user must take the application's introductory test before continuing. Additionally, test findings will determine whether it is crucial to provide therapy to the user. Then, our app will give mental health professionals access to the user's fundamental information. The user can then chat with a licenced psychologist inside the app [3].

1.2 Problem Statements

Many people find it difficult to get in touch with mental health professionals for a variety of reasons, including financial limitations, social stigma, or a lack of resources. As a result, there is a rising need for readily available, reasonably priced mental health services that can help people recognise and take care of their mental health disorders.

Multilayer Perceptron (MLP), a machine learning algorithm, can be used to create an app that can help users assess their mental health, get coping methods, and interact with mental health specialists when needed in order to solve this issue.To comprehend user input and produce personalised assessments, the app will use MLP algorithms. This will give users a customised user experience that is catered to their particular mental health needs. By utilising contemporary technology to increase access to mental health resources, this initiative seeks to decrease the occurrence and impact of mental health illnesses while boosting general wellbeing.

1.3 Aims and Objectives

i) To develop a test-conducting app for persons with mental health issues that makes it simple to identify between someone who has a mental health issue and someone who has little or no mental health issue.

ii) To suggest a psychologist or psychiatrist in light of the findings.

iii) To provide the user with a setting in which they can speak with mental health professionals through the app or, if they choose, visit their clinic.

1.4 Scope of Project

This mental health app's main objective is to give users a platform to evaluate and manage their mental health conditions. The software will recommend coping mechanisms to users based on the assessment. If a user's condition warrants it, they can also get in touch with a mental health expert using the app.

The three main user types for the programme will be administrators, psychologists, and patients. Patients will have access to assessments, coping mechanisms, and, if necessary, connections with mental health professionals. Psychologists will be able to check the profiles of their patients and speak with them. Users' data will be accessible to administrators, who can also control the app's features and content.

The app will also have an appealing design and a user-friendly interface with simple navigation. It will make users feel cosy and interested by utilising the right colours, fonts, and layouts. Overall, the main goal of this app is to help those who are battling with mental health illnesses on their path to better mental health by offering them a useful resource.

CHAPTER 2

SOFTWARE REQUIREMENTS SPECIFICATION

2.1 System Features

The application will include a variety of functions centred around its topic that will meet certain needs and achieve the desired result. One of the crucial duties is:

• MLP Forecast

2.1.1 MLP Prediction

A questionnaire with various questions and multiple answer choices will be completed by the user. The application's prediction model will determine the outcome, which is divided into one of five classes based on the user's responses: Normal, Mild, Moderate, Severe, or Extremely Severe.

2.2 User classes and classification

Three different user roles can be accommodated by the application:

- Patient
- Psychologist
- Admin

2.2.1 Patient

The mental health application allows users to register and log in to access a questionnaire with multiple-choice questions. Based on their reactions, the MLP prediction model will ascertain their mental state and offer static suggestions. Patients can speak with a mental health specialist within the app for additional help if they are not happy with the results. The application offers a secure and safe environment for people to ask for assistance and advance their mental health.

2.2.2 Psychologist

Psychologists can log in and view patient profiles on the mental health application. Through the app's chat feature, they can interact with their patients and provide the necessary advice and care. The application's suggested coping mechanisms can be verified by psychologists, who can also offer additional input on their efficacy. The capabilities of the app give psychologists a practical and effective way to oversee and support the mental health of their patients.

2.2.3 Admin

Administrators can access the mental health application and control user profiles and accounts. They can also control the app's content and resources, such as coping mechanisms and instructional materials, to make sure they are current and useful. The application's security and functionality must be maintained by the admin to guarantee that patients and psychologists can connect and seek assistance on a secure and reliable platform. The features of the application give Admin the instruments and resources needed to manage it effectively and give all users a good experience.

2.3 Operating Environment

- Some basic operating environments are necessary to guarantee the application runs correctly. These consist of:
- Android Base Requisites
    - Required Android Version 7.0(minimum) or above
    - Google Play Services
- iOS Base Requisites
    - Required iOS 9.0(minimum) or above
    - iOS Simulator
- Model Training System
    - Python (TensorFlow, sklearn)
- Development Essentials
    - Visual Studio Code
    - Android Studio
    - Flutter Framework
    - Google Colab
- Databases
    - Firebase

2.4 Design and Implementation

The design and implementation of the mental health application must take a number of variables into account in order to succeed. To protect sensitive user data, first and foremost, strong security measures must be implemented. The app should also be made to be user-friendly, open to all users, and compatible with a variety of hardware and software. During development, it's important to take technical limitations like processor speed, storage space, and network connectivity into account.

The software should also adhere to data security laws to store sensitive user data securely and link smoothly with mental health professionals to enable users to do so quickly and safely. In order to accommodate growing usage and data storage as the user base expands, it should also be designed to be scalable. Finally, the app must

undergo extensive testing, including checks for bugs, usability issues, and security flaws, before being made available to the public.

2.5 Assumptions and Dependencies

- The accuracy and efficacy of the mental health exam used to evaluate whether users need therapy are crucial to the app's success.
- The success of the app depends on the experts' willingness and capacity to offer their services via the app.
- The app must work with a variety of mobile devices and operating systems, and users must have access to a smartphone with an active internet connection.
- The app's effectiveness depends on users' willingness to share their personal information with the app and mental health specialists.
- For the app to be successful, user information must be secure and secret, and users must be able to communicate with mental health specialists in a secure and private setting.

2.6 Interface Requirement

The following are some of the interface requirements for the application:

• An Android version of at least 7.0 or above is needed on the device.

• Installing the most recent Google Play services version is also advised.

• An iOS version of 9.0 or later must be installed on the device.

2.7 Use Case Diagrams and Description Table

The use case diagrams offered give a thorough understanding of the system used in the project. An actor will interact with the application using specific conditions, workflows, and modules, as described in the following tables.

2.7.1 System Use Case

A system use case diagram provides a summary of the functioning of the entire system and how its parts work together to produce the intended result.

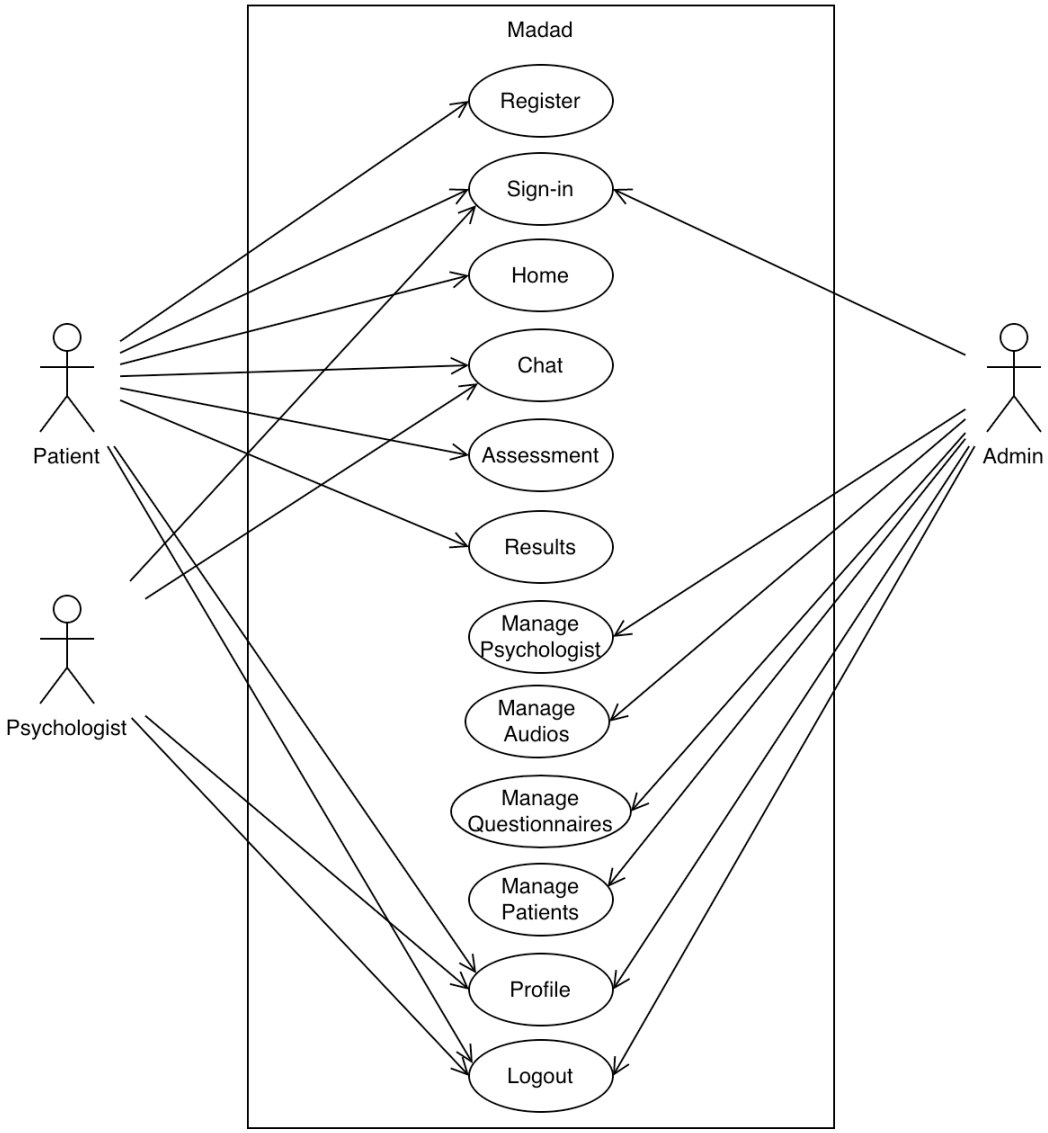


Figure 0.1: System Use Case

2.7.2 Register Use Case

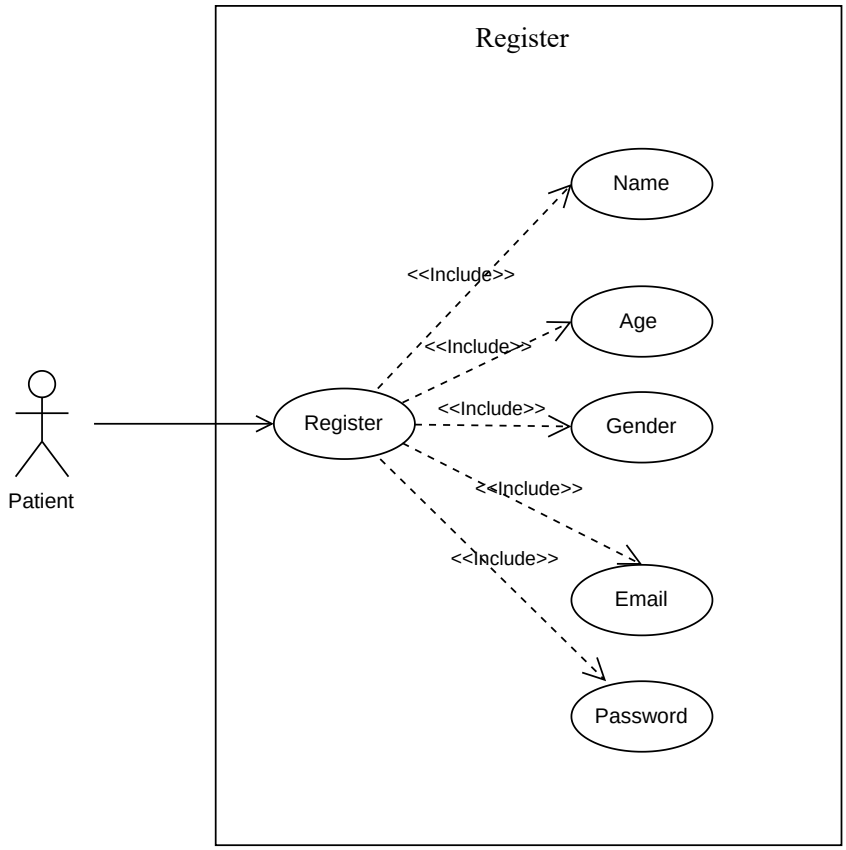


Figure 0.2: Register Use Case

In order to log into the application, the user needs create an account. The fields below are part of the registration procedure.

- Email
- Password
- Full Name
- Age
- Select Gender

Register Table

To register, the patient will need to provide accurate and valid information.
This use case table comprises the following components.

Table 0.1: Register

	Name	**Register**
1.	Use case ID	U1
2.	Objectives	Patient can register by filing the valid information.
3.	Priority	High
4.	Actor	Patient
5.	Flow of Events	• Open Application. • Fill up the required information. • Proceed with register.
6.	Alternate Flows	If patient is already registered, then it will go for sign-in.
7.	Exception Flow	• An error occurs in the application. • Application crashes. • Enter Invalid Information.
8.	Pre-condition	Patient not registered already.
9.	Post-condition	Patient registered successfully.

The patient can register, sign in, take the quiz, look at the list of psychologists, communicate with a psychologist, check the results, amend their profile information, and change their password. A psychologist can log in, amend their profile information, change their password, communicate with a patient, view the patient's profile, and log out. The administrator can log in, add a questionnaire, audio files, a psychologist, and log out.

2.7.3 Sign-in Use Case

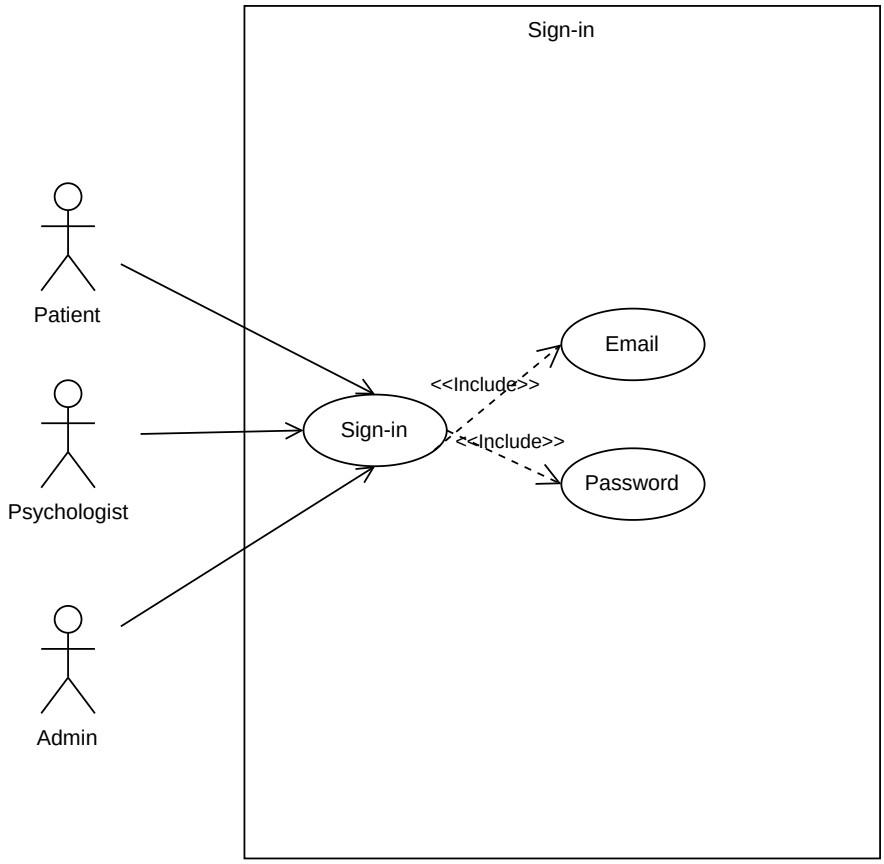


Figure 0.3: Sign-in Use Case

Sign-in Table

Table 0.2: Sign-in table

	Name	**Sign-in**
1.	Use case ID	U2
2.	Objectives	The patient, psychologist and admin can sign-in by filing the valid information.
3.	Priority	High
4.	Actor	Patient, Psychologist, Admin
5.	Flow of Events	• Open Application. • Fill up the required information. • Proceed with application.
6.	Alternate Flows	If the patient information is not stored in the database, then the patient will go for register.
7.	Exception Flow	• An error occurs in the application. • Application crashes. • Enter Invalid Information.
8.	Pre-condition	The patient and psychologist must register before sign-in.
9.	Post-condition	Sign-in successfully.

2.7.4 Home Use Case

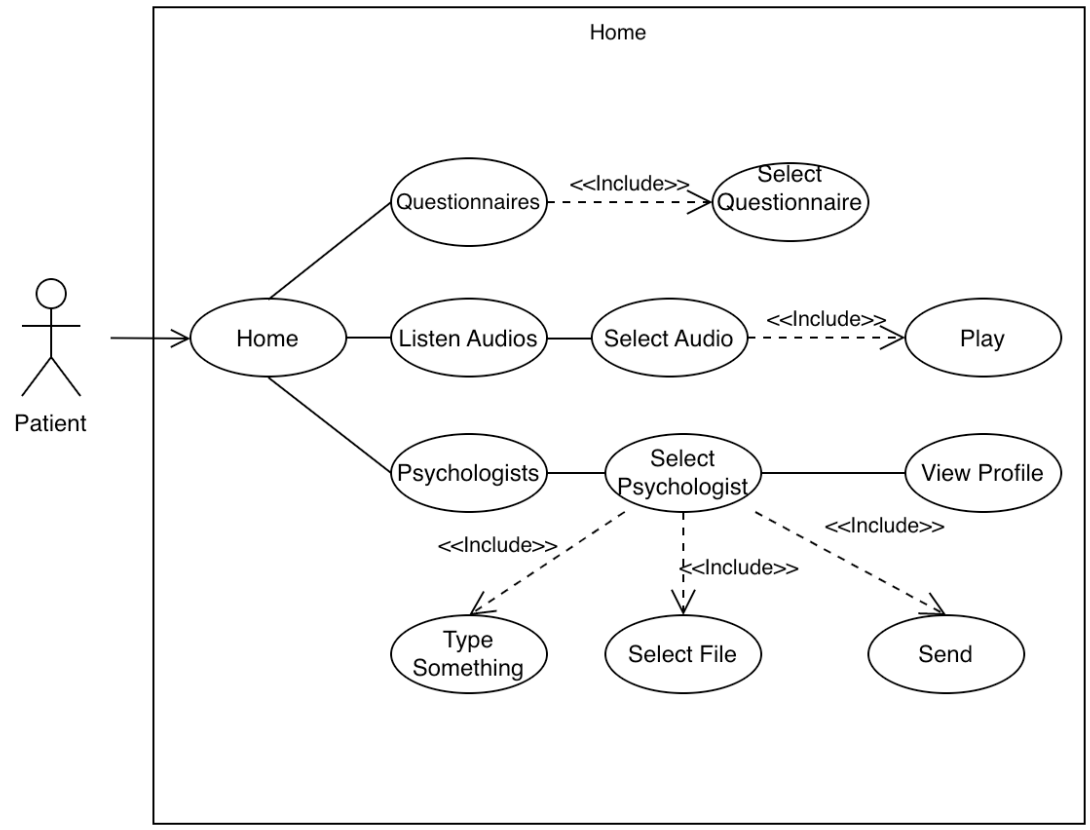


Figure 0.1: Home Use Case

Patients has the following options on home tab.

- Questionnaires
- Listen Audios
- Psychologists

Home Table

Table 0.3: Home

	Name	**Home**
1.	Use case ID	U3
2.	Objectives	Patients has the option to attempt different categories of questionnaire, listen audios to relax his/her mind and can see the list of psychologists.
3.	Priority	High
4.	Actor	Patient
5.	Flow of Events	• Open Application. • Click on home tab.
6.	Alternate Flows	No alternate flow.
7.	Exception Flow	No Exception flow.
8.	Pre-condition	Patient should be sign-in.
9.	Post-condition	No post condition.

The patient's home tab provides three options: add psychologist, add questionnaires, and add audios. These options allow the patient to interact with the application's features and resources to support their mental health. The patient can easily select the desired option and access the corresponding feature.

2.7.5 Chat Use Case

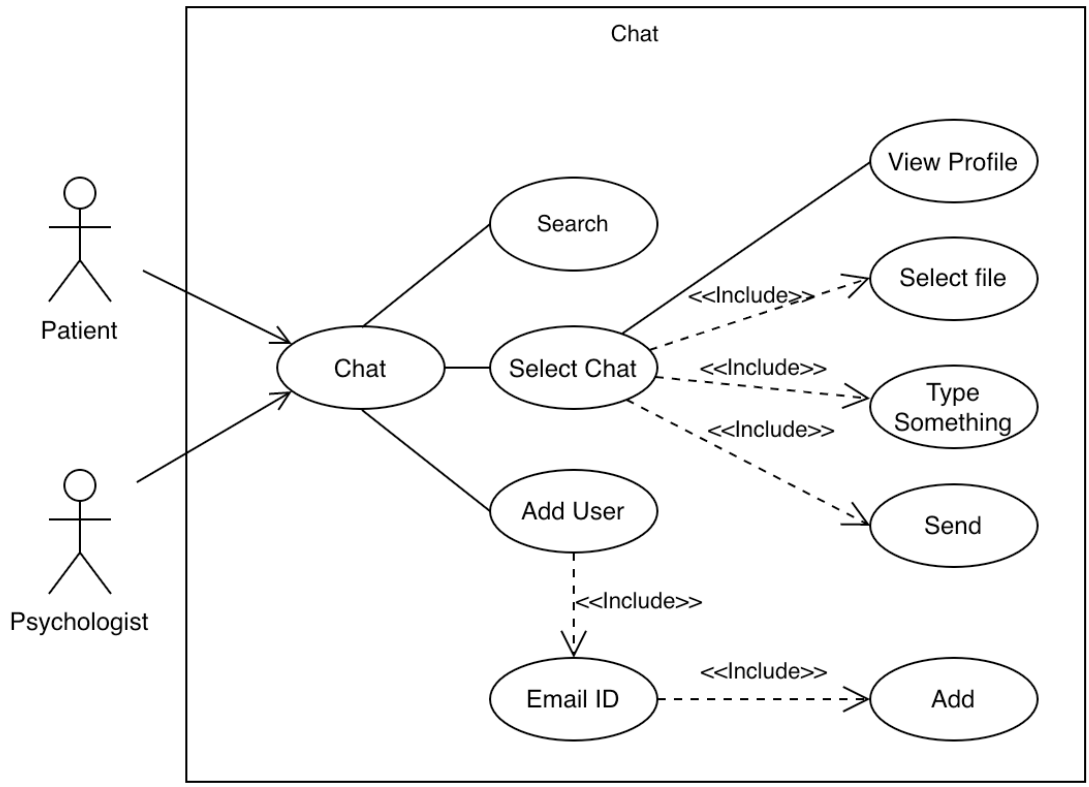


Figure 0.1: Chat Use Case

The chat feature enables confidential communication between patients and psychologists, facilitating support and guidance for patients.

Chat Table

Table 0.4: Chat

	Name	**Chat**
1.	Use case ID	U4
2.	Objectives	To create a connection between patient and psychologist.
3.	Priority	High
4.	Actor	Patient, Psychologist
5.	Flow of Events	• Open Application. • Click on chat tab.
6.	Alternate Flows	No alternate flow
7.	Exception Flow	No exception flows.
8.	Pre-condition	The patient and psychologist should be sign-in.
9.	Post-condition	No post condition.

The chat tab provides a secure and private communication channel for patients and psychologists to communicate with each other. It allows patients to seek guidance and support from their psychologist, and psychologists can respond to patient queries and concerns in real-time. The chat tab is designed to facilitate effective communication between patients and psychologists to support the patient's mental health needs.

2.7.6 Assessment Use Case

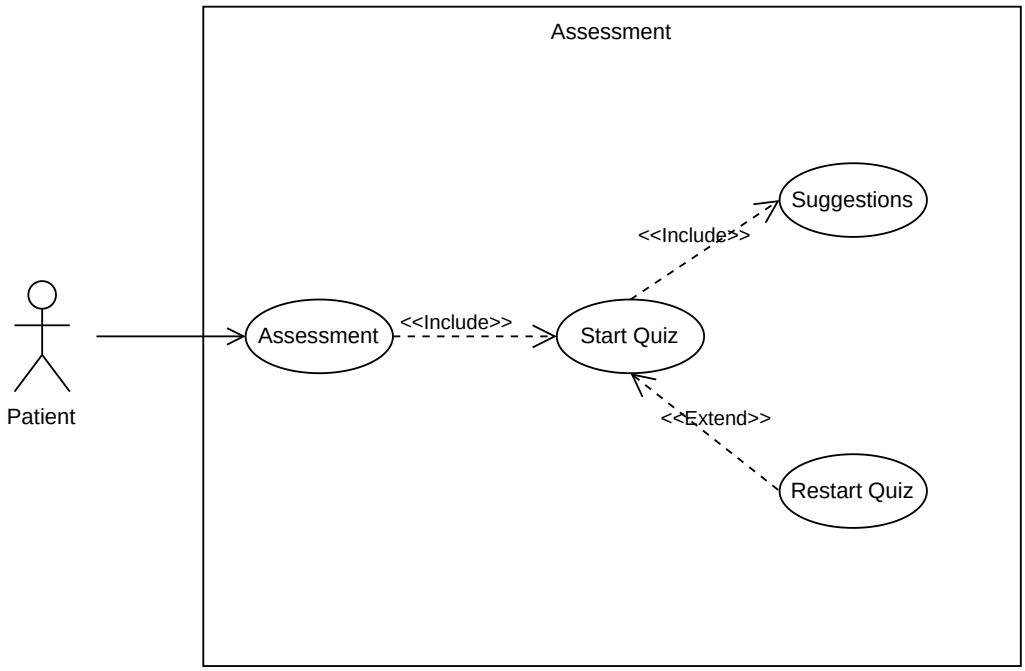


Figure 0.1: Assessment Use Case

The patient can assess their mental condition by answering several static questions.

Assessment Table

Table 0.5: Assessment

	Name	**Assessment**
1.	Use case ID	U5
2.	Objectives	Enable patients to chat with a licensed psychologist through the chat tab in the application.
3.	Priority	High
4.	Actor	Patient
5.	Flow of Events	• Open Application. • Click on assessment tab. • Start quiz. • Answer all the questions. • Suggestions
6.	Alternate Flows	No alternate flow.
7.	Exception Flow	No exceptional flow.
8.	Pre-condition	Patient should be sign-in.
9.	Post-condition	Can see the result.

The assessment tab provides patients with access to a mental health questionnaire designed to evaluate their current mental health status. This questionnaire is a comprehensive and validated tool that assesses various aspects of mental health, such as anxiety, depression, and stress levels.

2.7.7 Result Use Case

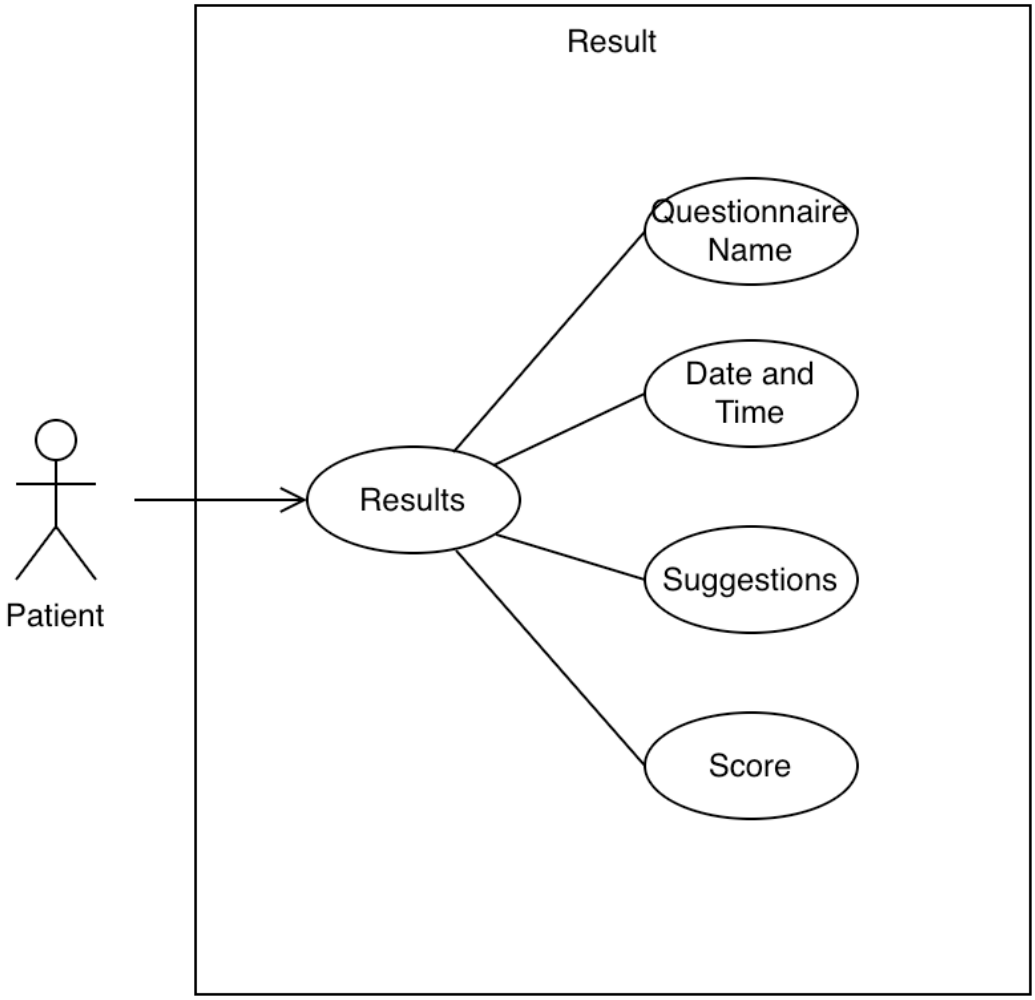


Figure 0.1: Results Use Case

Patient can check the results.

Result Table

Table 0.6: Result

	Name	**Result**
1.	Use case ID	U6
2.	Objectives	Allow patients to view their test results by accessing the result tab in the application.
3.	Priority	High
4.	Actor	Patient
5.	Flow of Events	• Open Application. • Click on result tab.
6.	Alternate Flows	No alternate flow.
7.	Exception Flow	No exceptional flow.
8.	Pre-condition	Patient should attempt questionnaire.
9.	Post-condition	No post condition.

The result tab displays the results of the mental health assessment completed by the patient. The result includes a breakdown of the patient's score on various mental health parameters, such as anxiety, depression, and stress levels. The result tab allows patients to track their mental health progress over time and make informed decisions about their treatment and care.

2.7.8 Manage Psychologist Use Case

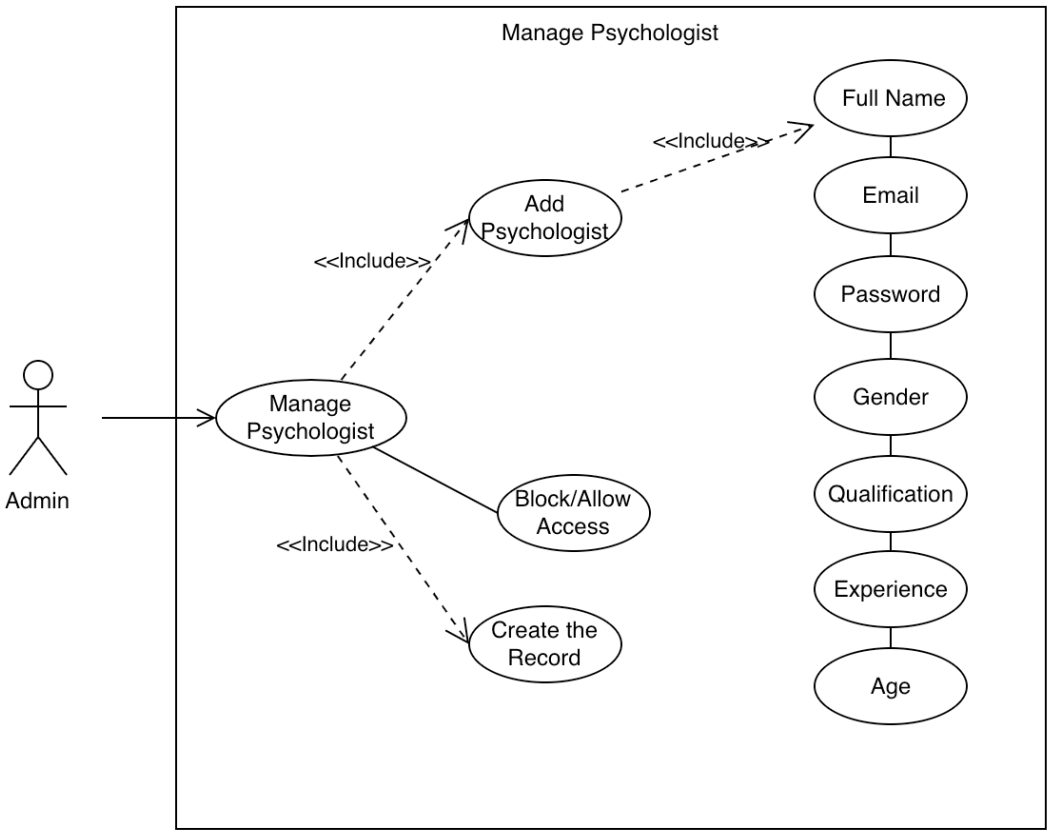


Figure 0.1: Manage Psychologist Use Case

Admin can add psychologist in the app.

Manage Psychologist Table

Table 0.7: Manage Psychologist

	Name	**Add Psychologist**
1.	Use case ID	U7
2.	Objectives	Enable the admin to add new psychologists to the application for patients to access their profiles and communicate with them.
3.	Priority	High
4.	Actor	Admin
5.	Flow of Events	• Open the application and click on “Add Psychologist” section. • Fill the required fields. • Click on the "Create Psychologist" button.
6.	Alternate Flows	No alternate flow.
7.	Exception Flow	• Display error message if the admin tries to create the psychologist with the same email. • Display an error message if the admin misses the required field.
8.	Pre-condition	Admin should sign-in.
9.	Post-condition	Psychologist uploaded successfully.

Admin can add a new psychologist by filling in their details such as full name, email, password, age, and gender information. The new psychologist will then be able to access the platform and their profile can be viewed in the psychologist's tab. Additionally, admin can also edit and delete the psychologist's account if needed.

2.7.9 Manage Audios Use Case

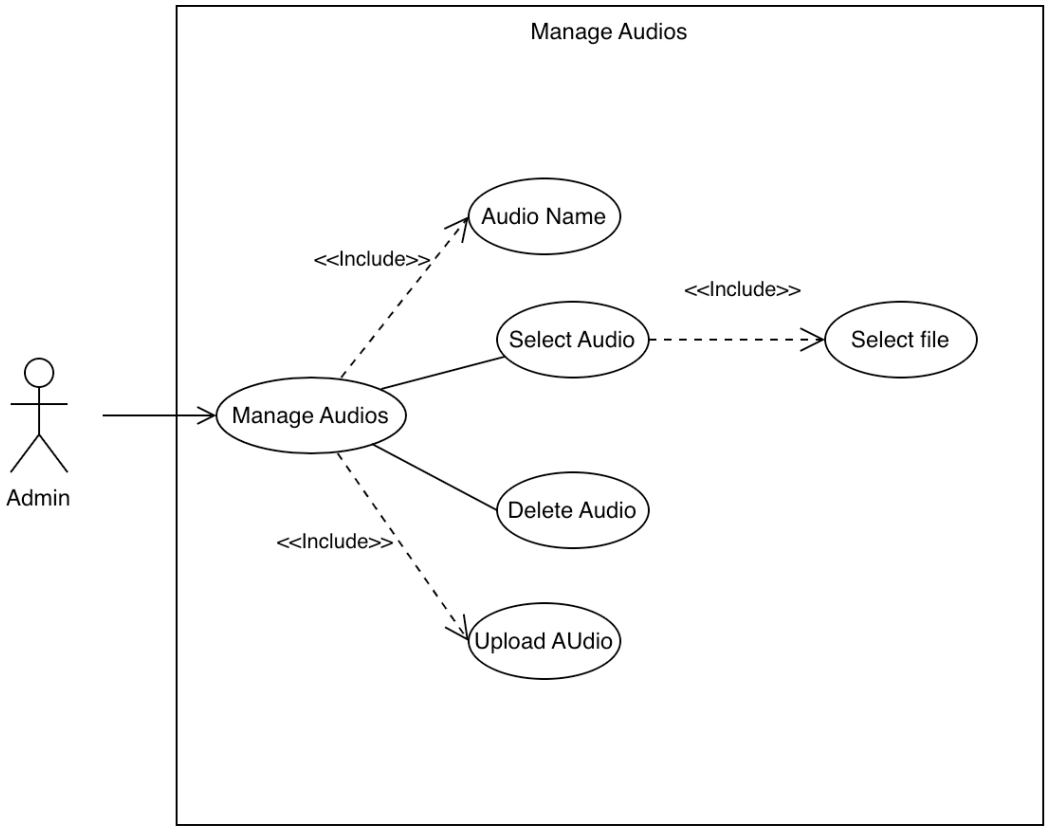


Figure 0.1: Manage Audio Use Case

Admin can upload audios in the system to keep the audios updated.

Manage Audios Table

Table 0.8: Manage Audios

	Name	**Add Audios**
1.	Use case ID	U8
2.	Objectives	Enable the admin to upload audio files within the application for patients to access.
3.	Priority	High
4.	Actor	Admin
5.	Flow of Events	• Open the application and click on “Upload Audios’ section. • Provide a name for the audio file. • Click on the "Add Question" button to begin creating a new question. • Write the name in the provided text box. • Click on the “Select Audio” button. • Select the audio from your device storage. • Click on the “Upload Audio” button. • Click on the "Submit" button to save the question and continue to add more questions as needed.
6.	Alternate Flows	No alternate flow.
7.	Exception Flow	• Display error message if file format is not supported. • display error message if file size exceeds maximum limit. • display error message and prompt admin to try again later if there is an issue with uploading the audio file.
8.	Pre-condition	Admin should be sign-in.
9.	Post-condition	Audio uploaded successfully.

In the "Add Audios" feature, the admin can upload and add new audio files to the application for patients to access. The admin can specify the audio's name.

2.7.10 Manage Questionnaire Use Case

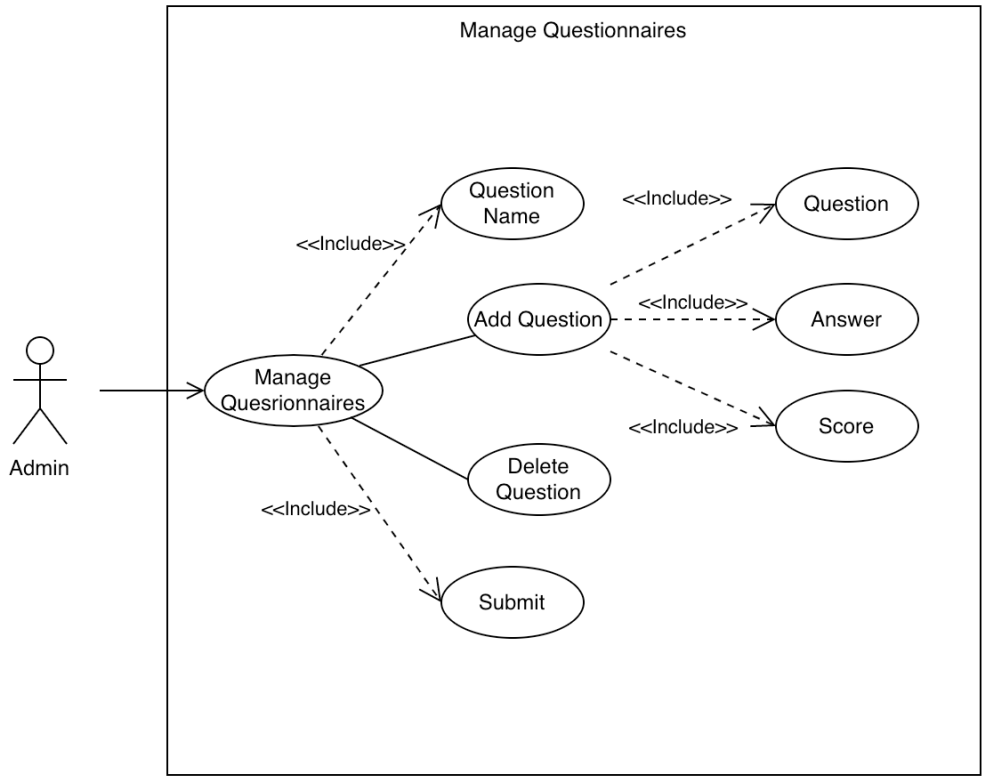


Figure 0.1: Manage Questionnaires Use Case

Admin can add questionnaires of different categories.

Manage Questionnaire Table

Table 0.9: Manage Questionnaire

	Name	**Add Questionnaire**
1.	Use case ID	U9
2.	Objectives	Allow the admin to add questionnaires of various categories within the application.
3.	Priority	High
4.	Actor	Admin
5.	Flow of Events	• Open the application and click on add questionnaires section. • Provide a name for the questionnaire. • Click on the "Add Question" button to begin creating a new question. • Write the question in the provided text box. • Enter multiple options for the question in the designated fields. • Assign a score to each option to provide a means of evaluating the response. • Click on the "Submit" button to save the question and continue to add more questions as needed.
6.	Alternate Flows	No alternate flow
7.	Exception Flow	• Prompt user to fill in missing fields before saving. • display error message when user attempts to save a duplicate question. • display error message and prompt user to try again later if there is an issue with saving the question.
8.	Pre-condition	Admin should be sign-in.
9.	Post-condition	Questionnaire uploaded successfully.

In the "Add Questionnaire" feature, the admin can create and add new questionnaires to the application. The admin can specify the questionnaire's name and category and add questions with multiple options and their corresponding scores.

2.7.11 Manage Patients

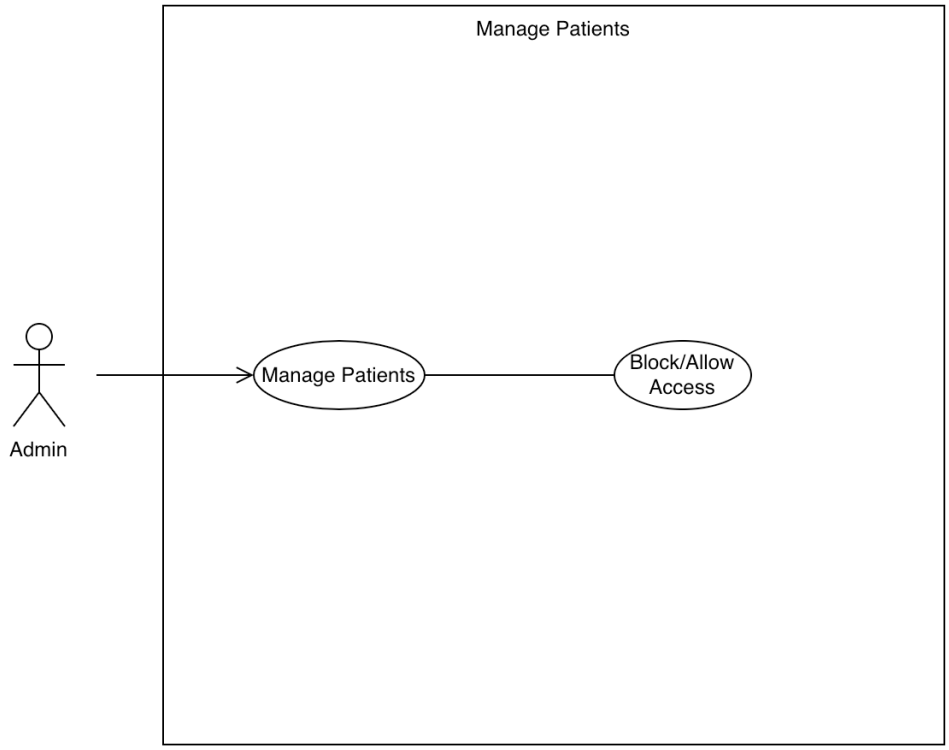


Figure 0.1: Manage Patients

Manage Patients Table

Table 0.10: Manage Patients

	Name	**Manage Patients**
1.	Use case ID	U10
2.	Objectives	Allow the admin to block/give app access to patient.
3.	Priority	High
4.	Actor	Admin
5.	Flow of Events	• Open the application and click on Manage Patients section. • Block / Allow Access app to patient.
6.	Alternate Flows	No alternate flow
7.	Exception Flow	• If user account is already block, then admin can re-give app access to patient
8.	Pre-condition	Admin should be sign-in.
9.	Post-condition	Blocked/Allowed successfully.

2.7.8 Profile Use Case

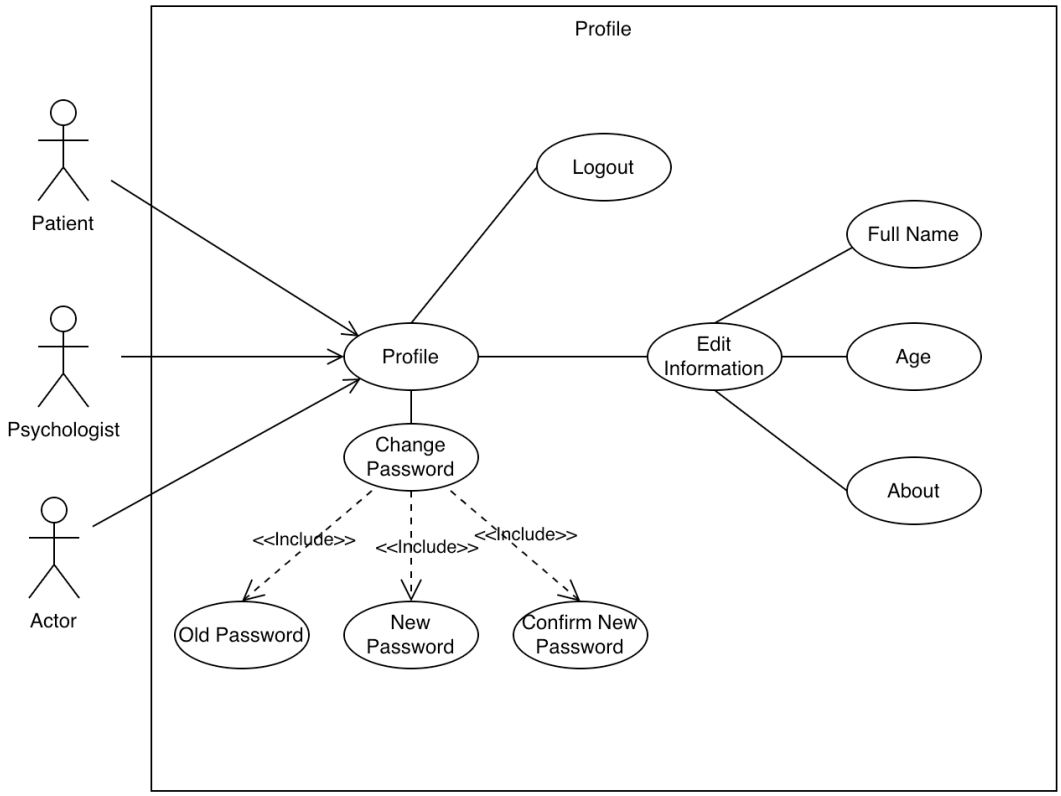


Figure 0.2: Profile Use Case

Profile Table

Table 0.11: Profile

	Name	**Profile**
1.	Use case ID	U11
2.	Objectives	Enable patients and psychologists to view and edit their profiles within the application, including personal information and the ability to change their passwords.
3.	Priority	High
4.	Actor	Patient, Psychologist
5.	Flow of Events	• Open Application. • Click on Profile tab. • Edit profile and click on “Save” button. • For new password click on “Change Password” button. • Fill the required fields. • Click on the “Change Password” button.
6.	Alternate Flows	No alternate flow.
7.	Exception Flow	Display error message if the patient or psychologist enter the wrong current password.
8.	Pre-condition	The patient and psychologist should sign-in.
9.	Post-condition	Profile updated successfully.

The profile tab allows patients and psychologists to view and edit their personal information, such as their full name, age, about, and profile picture. They can also change their password from this tab.

2.7.12 Logout Use Case

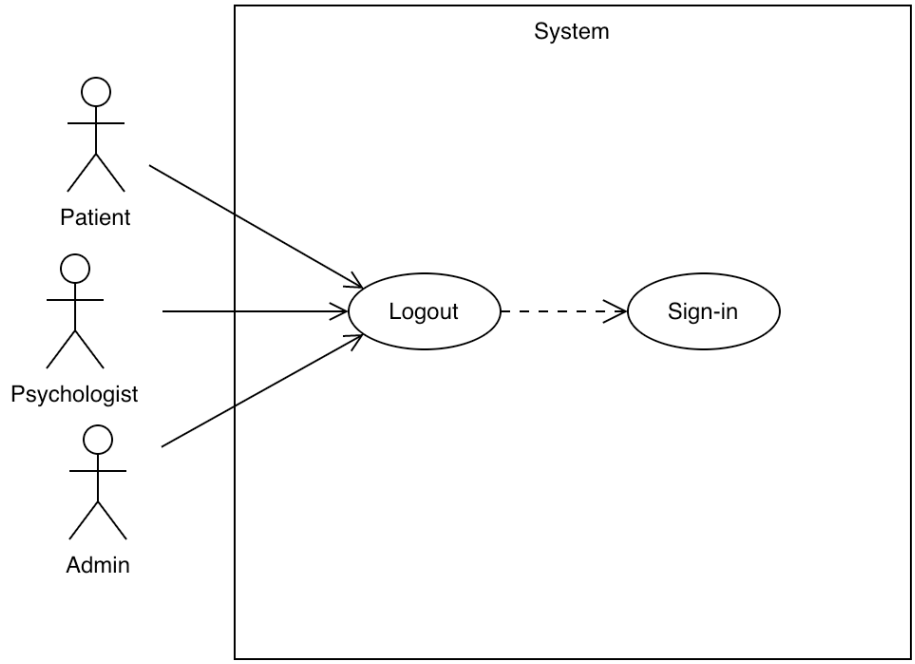


Figure 0.1: Logout Use Case

The patient, psychologist and admin have the option to logout of the application.

Logout Table

Table 0.12: Logout

	Name	**Logout**
1.	Use case ID	U12
2.	Objectives	To ensure that the user's account does not remain active in the background, the application will automatically transition to the sign-in page after the patient, psychologist, or admin logs out of their account. This security feature helps prevent unauthorized access to the user's account and protects their privacy.
3.	Priority	High
4.	Actor	Patient, Psychologist, Admin
5.	Flow of Events	• Open the application. • Navigate to profile section. • Click on the “logout” button.
6.	Alternate Flows	No alternate flow.
7.	Exception Flow	If the user's device loses network connection during the log out process, the application should display an error message to notify the user that the log out process was not completed successfully
8.	Pre-condition	The patient, psychologist, admin should be sign-in.
9.	Post-condition	No post condition after logout.

The logout feature allows patients, psychologists, and admins to end their current session and securely log out of the application. Once logged out, users will be directed back to the sign-in page and their account will no longer be accessible without proper credentials.

2.8 System Requirement

The subsequent table presents a comprehensive list of both functional and non-functional requirements that are essential for the successful completion of the project.

Table 0.13: System Requirement

ID	Priority	Use Case	Description
Functional Requirements			
FR1	High	Register (U1)	The functional requirement for the register use case is to provide the ability for patients to create new accounts by validating and ensuring that the email is unique, and the password meets the required complexity criteria.
FR2	High	Sign-in (U2)	The functional requirement for the sign-in use case is to get the correct credentials, such as username and password, to access their respective profiles. The system should authenticate the user's credentials and grant them access to the appropriate profile.
FR3	High	Home (U3)	The functional requirement for the home use case is to display options for users to access different categories of mental health questionnaires, listen to mindfulness audios, view a list of available psychologists, and initiate contact with them.
FR4	High	Chat (U4)	The functional requirements for the chat use case include establishing a real-time chat interface between patients and psychologists. The interface should allow sending and receiving text messages.

FR5	High	Assessment (U5)	The functional requirement for the Assessment use case is that the questionnaire must be designed to provide accurate and comprehensive evaluations of a patient's mental health status. It should also be user-friendly and accessible for patients to complete.
FR6	High	Results (U6)	The functional requirement for the result use case is to enable patients to access and view their mental health assessment results, which can be used for further discussion with their psychologist.
FR7	High	Add Questionnaires (U7)	The functional requirements for adding a questionnaire use case are to specify the questionnaire name, and category, add questions with multiple options and corresponding.
FR8	High	Add Audios (U8)	Functional requirements for the Add Audio use case include the ability for the admin to upload audio files and specify the name and details of the audio. These audio files should be accessible to patients on the platform.
FR9	High	Add Psychologists (U9)	Functional requirements include the ability for admin to input new psychologist's details such as full name, email, password, age and gender, view and edit their profiles, and delete their accounts if needed.
FR10	High	Profile(U10)	The functional requirements for the Profile use case include the ability for patients and psychologists to update and save changes to their personal information and password.

FR11	Medium	Logout (U11)	The functional requirements for the logout use case are to clear the user's session data and ensure that the user is logged out of the application. The user should also be redirected to the login page after logging out.
Non-Functional Requirements			
NFR1	High	Register (U1)	The registration process should be quick and responsive, with minimal loading times and delays to provide a smooth user experience.
NFR2	High	Sign-in (U2)	The system must ensure secure authentication and authorization processes to prevent unauthorized access to user accounts
NFR3	Medium	Home (U3)	It should provide quick access to the different options and functionalities available on the home page.
NFR4	High	Chat (U4)	The communication between patients and psychologists should be fast and responsive, with minimal latency or delay
NFR5	Medium	Assessment (U5)	The assessment should provide a smooth and intuitive user experience, with clear instructions and an easily navigable interface
NFR6	High	Results (U6)	Ensure that the results are displayed in a clear and easy-to-understand format.
NFR7	High	Add Questionnaires (U7)	The interface for adding questionnaires should be intuitive and user-friendly to ensure that administrators can efficiently create and upload new questionnaires without difficulty.
NFR8	Medium	Add Audios (U8)	To ensure that the audio files are of high quality and have good audio clarity.

NFR9	High	Add Psychologist (U9)	The system should provide secure storage and encryption of the psychologist's personal information to ensure confidentiality and privacy.
NFR10	High	Profile(U10)	The user interface should be intuitive and easy to navigate, with clear labels and instructions for updating personal information and changing passwords.
NFR11	Medium	Logout (U11)	The system should log out the user within a maximum of 5 seconds after clicking the logout button.

2.9 Literature Review

The study by Li et al. (2019) focuses on the recognition of depression using machine learning techniques and various feature generation strategies. The researchers explore the application of machine learning methods to accurately identify depression by examining different approaches to feature generation[2]. Their work contributes to the field of artificial intelligence in medicine by providing insights into the potential of machine learning for depression recognition, emphasizing the importance of feature generation strategies in achieving accurate results.

The study conducted by Narayanrao et al. (2020) focuses on the analysis of machine learning algorithms for predicting depression. The authors examine various machine learning techniques and their effectiveness in predicting depression. The research contributes to the field by evaluating the performance of different algorithms and providing insights into their predictive capabilities[3]. The study emphasizes the importance of machine learning in the field of depression prediction and highlights the potential for further advancements in this area.

Richter et al. (2020) employ machine learning-based analysis to differentiate between anxiety and depression based on behavioral patterns. The authors investigate the potential of machine learning techniques in distinguishing between these two commonly co-occurring mental health conditions. By leveraging behavioral data, the research contributes to a deeper understanding of the distinct characteristics of anxiety and depression. The study highlights the value of machine learning in identifying subtle behavioral differences and offers promising implications for improved diagnostic accuracy and personalized treatment approaches for individuals with anxiety and depression.[4]

Zulfiker et al. (2021) present an in-depth analysis of machine learning approaches for predicting depression. The authors extensively examine various machine learning techniques employed in depression prediction and provide a comprehensive overview of their strengths, limitations, and performance. By synthesizing existing research, the study contributes to the field by highlighting the potential of machine learning in

accurately identifying and predicting depression. The review underscores the importance of feature selection, data preprocessing, and model evaluation in achieving robust and reliable prediction outcomes. The findings of this research offer valuable insights for future studies aiming to develop effective and efficient machine learning models for depression prediction.

Haque et al. (2021) conduct a literature review on the detection of child depression using machine learning methods. The authors explore the application of machine learning techniques in identifying depression in children, highlighting the importance of early detection and intervention for improved mental health outcomes[5]. The study synthesizes existing research on feature selection, data preprocessing, and classification algorithms specifically tailored for detecting child depression. By analyzing the strengths and limitations of different machine learning approaches, the authors contribute to the field by providing insights into the potential of these methods in accurately identifying depressive symptoms in children. The findings of this review serve as a valuable resource for researchers and practitioners interested in developing effective and sensitive tools for early detection and intervention in child depression.

Fatima, A. et al. (2021) present a literature review on the topic of detecting depression, anxiety, and stress in texts using emotional recall, cognitive networks, and machine learning techniques. The authors critically analyze previous studies in this area, emphasizing the importance of accurately identifying and distinguishing these mental health conditions from text data. They explore the potential of emotional recall and cognitive networks in capturing the emotional and cognitive aspects of text, and discuss how machine learning can be leveraged to develop predictive models. The review sheds light on the advancements made in computational approaches for detecting mental health conditions through textual analysis[6]. The findings contribute to the understanding of this field, providing valuable insights for researchers and practitioners seeking to develop more effective and efficient tools for identifying depression, anxiety, and stress using big data and cognitive computing methods.

Ryu et al. (2022) focused on predicting poststroke depression using machine learning algorithms. The authors analyze previous research studies in this domain and examine the outcomes achieved through the application of machine learning techniques. Their review underscores the significance of accurately predicting poststroke depression as an essential aspect of stroke rehabilitation and mental health management. By assessing the performance of various machine learning algorithms, the authors provide insights into the predictive capabilities of these approaches and their potential for assisting in early identification and intervention for poststroke depression.[7] This literature review contributes to the field by consolidating knowledge on the use of machine learning algorithms in predicting poststroke depression, highlighting the need for further research in this area to improve patient outcomes and enhance stroke rehabilitation practices.

Table 0.14: Literature Review

Authors	Year	Paper	Algorithm	Results		Future Work
				F1 score	Accuracy	
Li et al.	2019	Depression recognition using machine learning methods with different feature generation strategies	Support Vector Machines (SVM), Random Forest (RF), and k-Nearest Neighbors (k-NN)	83.87%	-	Investigate the generalizability of the proposed method on other populations and contexts.
Narayanrao et al.	2020	Analysis of machine learning algorithms for predicting depression	Decision Tree, Random Forest, Logistic Regression, KNN, SVM	-	Random Forest 91.5%, KNN 86.5%, and Decision Tree 84.5%	Investigating and improving the use of machine learning algorithms for depression prediction.
Richter, T. et al.	2020	Using machine learning-based analysis for behavioral differentiation	Support Vector Machines (SVM), k-Nearest Neighbors (k-	-	77% for distinguishing between anxiety and depression based on behavioral measures, and up	Further research is needed to validate the findings with larger sample sizes and diverse populations, examine the generalizability of the findings across

		between anxiety and depression	NN), and Decision Trees (DT		to 81% accuracy when combining behavioral measures and self-report questionnaires	contexts, and identify potential underlying mechanisms of behavioral differences between anxiety and depression
Zulfiker, M. S. et al.	2021	An in-depth analysis of machine learning approaches to predict depression	SVM, Random Forest, KNN, Naive Bayes, Decision Tree, Logistic Regression, ANN	-	SVM 91.6%, ANN 89.9%	Further exploration of machine learning approaches, including the integration of additional features and evaluation on larger and more diverse datasets.
Haque, U. M. et al.	2021	Detection of child depression using machine learning methods	Random Forest, Decision Tree, Naïve Bayes, K-Nearest Neighbors, Support Vector Machine, Artificial Neural Network	-	Random Forest 97.36%, Decision Tree 93.74%, Naïve Bayes 85.89%, K-Nearest Neighbors 84.24%, Support Vector Machine 82.26%, and Artificial Neural Network 80.15%	Investigating and improving the performance of machine learning methods for detecting child depression.
Fatima, A. et al.	2021	Dasentimental: Detecting depression, anxiety, and stress in	Feature selection using Correlation-based Feature Selection (CFS)	76% for depression, 77% for	-	Investigating the use of other classifiers and feature selection methods, expanding the dataset to different

		texts via emotional recall, cognitive networks, and machine learning	and Principal Component Analysis (PCA), Support Vector Machines (SVM) and K-Nearest Neighbors (KNN) classifiers	anxiety, and 79% for stress detection		languages and populations, and evaluating the models' performance on social media data or other digital sources of text data
Ryu et al.	2022	Prediction of poststroke depression based on the outcomes of machine learning algorithms	Random Forest, KNN, SVM, Naïve Bayes, Logistic Regression		RF 84.5%, KNN 81.4%, SVM 79.4%, Naïve Bayes 74.2%, Logistic Regression 70.5%	Further research to explore and refine machine learning algorithms for predicting poststroke depression.

CHAPTER 3

DESIGN AND METHODOLOGY

3.1 Methodology

It helps the people by conducting tests related to mental health issues and based on their answers, it leads to the result that how much severity of the mental health issue the people are suffering from. The methodology we are using is agile model. In agile model, clients, designers and testers are in one correspondence with each other for the improvement of the project. We are using Feature Driven Development (FDD) method of agile model. It helps in the development of the project in features and divide complex features into subparts. The functions of this app are:

- Registration of user.
- Conduction of test to study user's mental situation.
- MLP for result prediction, which generates a result based on the user's answers after completing the assessment questionnaire.
- If needed more help, then the user can have the choice to consult with a mental health professional.

3.2 Multi-Layer Perceptron

Multi-Layer Perceptron is the abbreviation for a type of neural network used in machine learning. It has layers of interconnected nodes (neurons) and is trained using methods like gradient descent and backpropagation. MLPs can use methods like activation functions, regularization, dropout, batch normalization, learning rate

schedule, early stopping, and network architectural design to increase performance and avoid overfitting when learning complicated patterns in data.

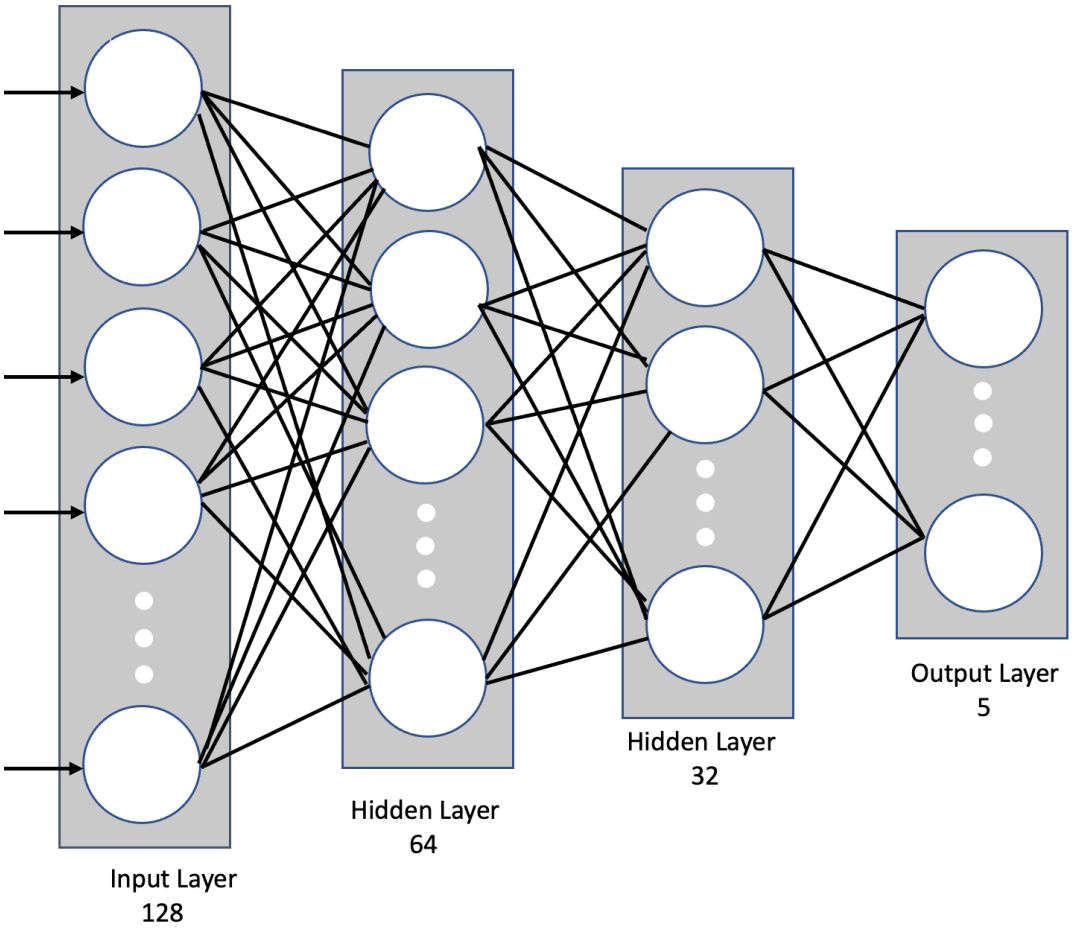


Figure 3.1: MLP Model Diagram

3.3 Design by Feature

- Sequential Diagram
- Methodology Diagram
- Class Diagram of a system

3.3.1 Sequential Diagram

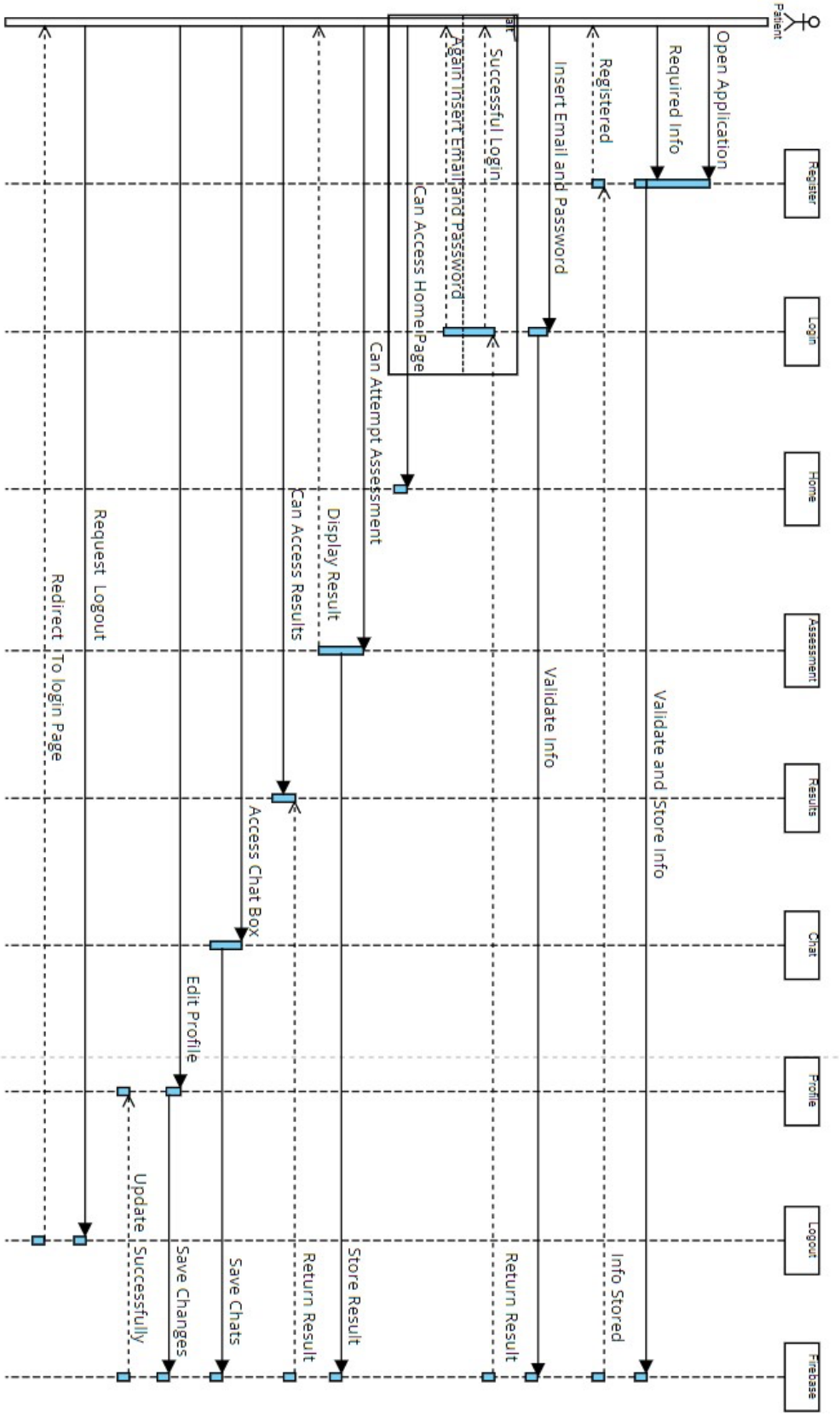


Figure 3.2: System Sequence Diagram

3.3.2 Methodology Diagram

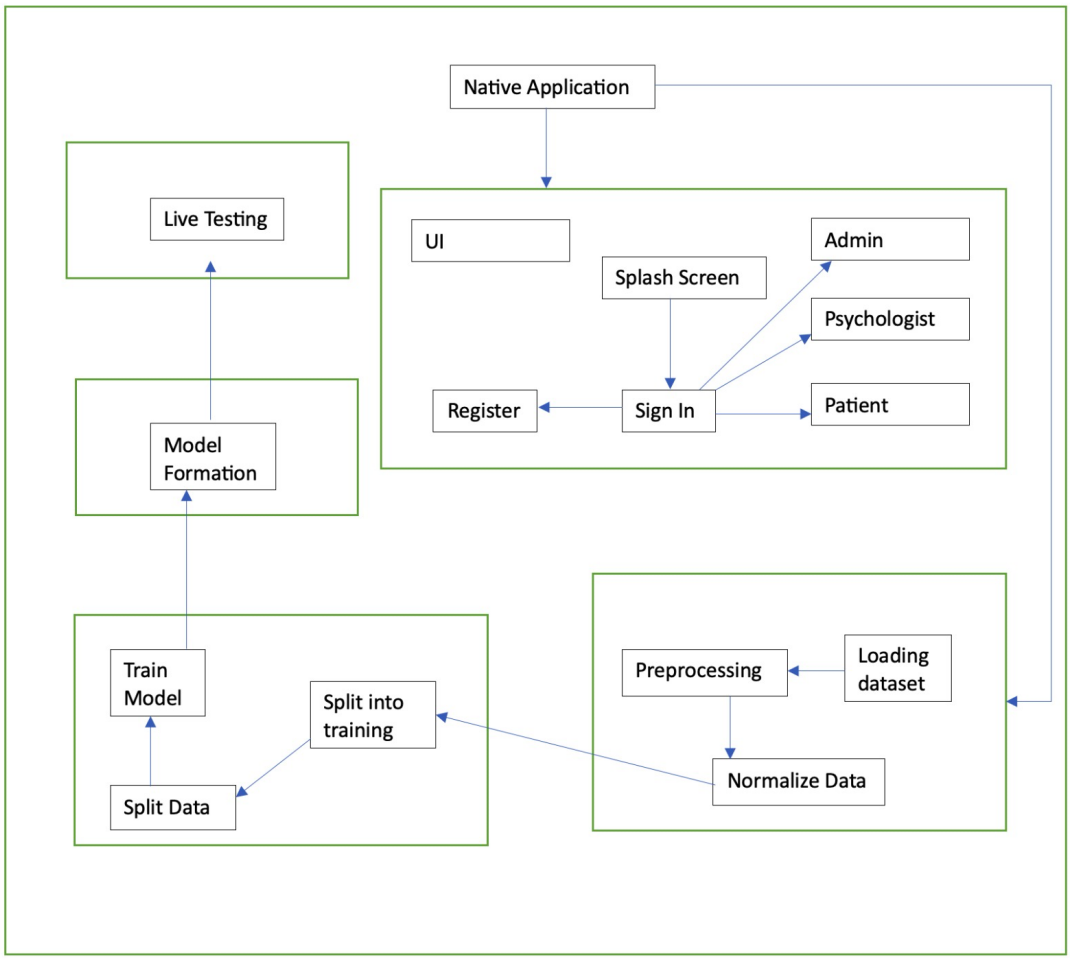


Figure 3.3: Methodology Diagram

3.3.3 Class Diagram

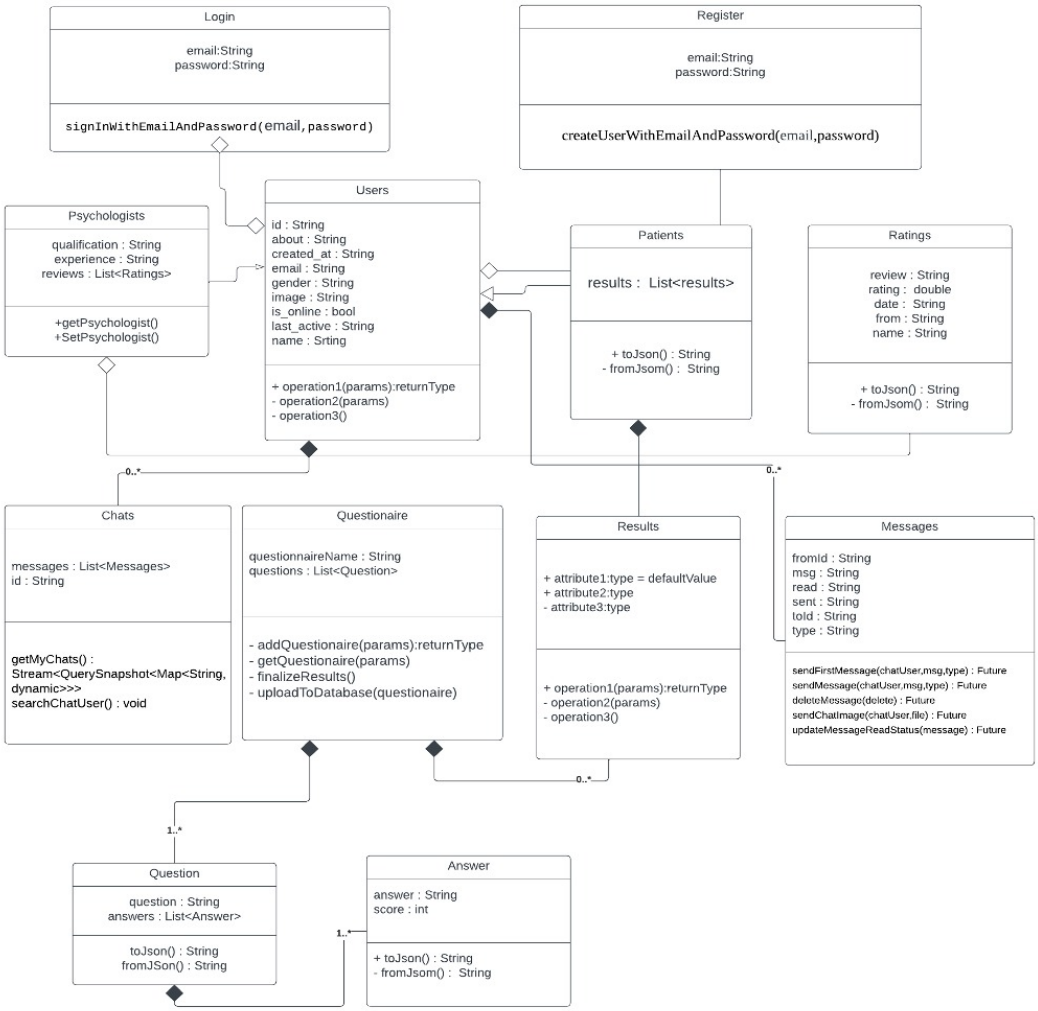


Figure 3.4: Class Diagram

CHAPTER 4

DATA, EXPERIMENTS AND IMPLMENTATION

4.1 Implementation

In this project we are working on two phases the first one is the training of deep learning model which include some steps. The first step is to gather dataset related to depression, anxiety, and stress. Dataset is available on Kaggle, but it is published by USNW. By visualizing the data, we came to know we have to perform feature engineering on data as it was in raw format. We removed the columns that were not required, performed data analysis, and applied binning and handling outlier's techniques to normalize the data.

4.1.1 Flutter Implementation

Our mobile application will consist of a framework known as flutter which is a software development framework used for designing mobile applications. Flutter is a front-end programming language which can be used for building native apps with the help of one coding syntax. Its application is portable which means it can be run on macOS, iOS, Android, Windows, and Web etc.

4.1.2 Dart Implementation

On any platform, Dart is a client-optimized language for creating quick apps. The most effective programming language for cross-platform development provides the functionality it aims to provide. Dart prioritizes both development (sub-second stateful

hot reload) and high-quality production experiences across a wide range of compilation targets (web, mobile, and desktop), and has been developed for a technological framework that is particularly suited to client development.

4.1.3 Firebase Implementation

The Google Firebase application development platform works with programmers to produce Android and web applications. It is intended to simplify life for developers by handling most of the data pushing and pulling. This frees up app developers from having to manage versions or locales through code. They can add the leftover pieces to Firebase, and the data will remain accurate across the entire structure.

4.1.4 Python (for machine learning)

Python is a popular programming language used to create various websites, and computer programs, and do information analysis. Python is a general-purpose language, which means it can be used to create a wide range of programs and is not focused on any problems. The multi-layered approach is then illustrated using representation learning and deliberation. We applied machine learning to the dataset for training, which incorporates Python, to train our model.

4.1.5 Tensor Flow

A complete open-source machine learning platform is called TensorFlow. The class concentrates on using a specific TensorFlow API to create and train machine learning models, even though TensorFlow is a robust system for managing all parts of a machine learning system.

4.1.6 Scikit-learn (Machine Learning Library)

The widely recognized machine learning library Scikit-learn is used for many machine learning methods. It was made using some technology you may be experienced with, including Numpy, Pandas, and Matplotlib. We divided our data into training and testing using this package.

4.2 Data and Experiments

As previously mentioned, the MLP model is being used in this project, and we are now talking about how it works and the conclusions we get from it.

4.2.1 Dataset

We utilized a dataset from Kaggle. Kaggle is a popular online platform that hosts a diverse collection of datasets for various machine learning tasks. Once we acquired the dataset, we processed it and organized it in line with the specifications of our model.

To evaluate the performance and generalization ability of our model, we divided the dataset into training and testing sets. To achieve this, we employed the preconfigured train-test split library provided by Scikit-learn (SK Learn). With this library, we set a test size of 0.2, indicating that 20% of the data would be allocated for testing, while the remaining 80% would be used for training our model.

Upon completing the initial stages of data pre-processing and partitioning, the next step involved designing the architecture of our model. Based on the picture provided, the model consists of four layers: an input layer with 128 neurons, followed by two hidden layers with 64 and 32 neurons, respectively, and finally, an output layer with 5 neurons. In these layers, we specified the activation functions as "relu" for the hidden layers and "softmax" for the output layer.

By setting up the layers and defining their activation functions, we established the structure and behaviour of our model. This configuration allows the model to learn and make predictions based on the patterns and relationships within the data.

```
# Defining Keras model
```

Figure 0.1: Processing Data

```
    keras.layers.Dense(64, activation='relu'),
    keras.layers.Dense(32, activation='relu'),
    keras.layers.Dense(5, activation='softmax')
])
```

The model is run to make predictions on the data towards the end of the training process. Our model's value of epochs is set to 40, suggesting that there will be 40 stages in the training procedure. The model iteratively learns from the data and modifies its internal parameters to increase accuracy with each epoch.

We give the model more opportunity to learn and improve its categorization skills by increasing the number of epochs. Over time, the model continuously improves its comprehension of the patterns and features present in the data, which results in an increase in accuracy.

```
Epoch 38/40
994/994 [==============================] - 3s 3ms/step - loss: 0.0710 - accuracy: 0.9708
Epoch 39/40
994/994 [==============================] - 3s 3ms/step - loss: 0.0704 - accuracy: 0.9714
Epoch 40/40
994/994 [==============================] - 2s 2ms/step - loss: 0.0808 - accuracy: 0.9677
249/249 [==============================] - 1s 3ms/step - loss: 0.4553 - accuracy: 0.8711
Test accuracy: 0.8711463212966919
```

Figure 0.2: Execution of model

CHAPTER 5

USER MANUAL

5.1 Splash Screen

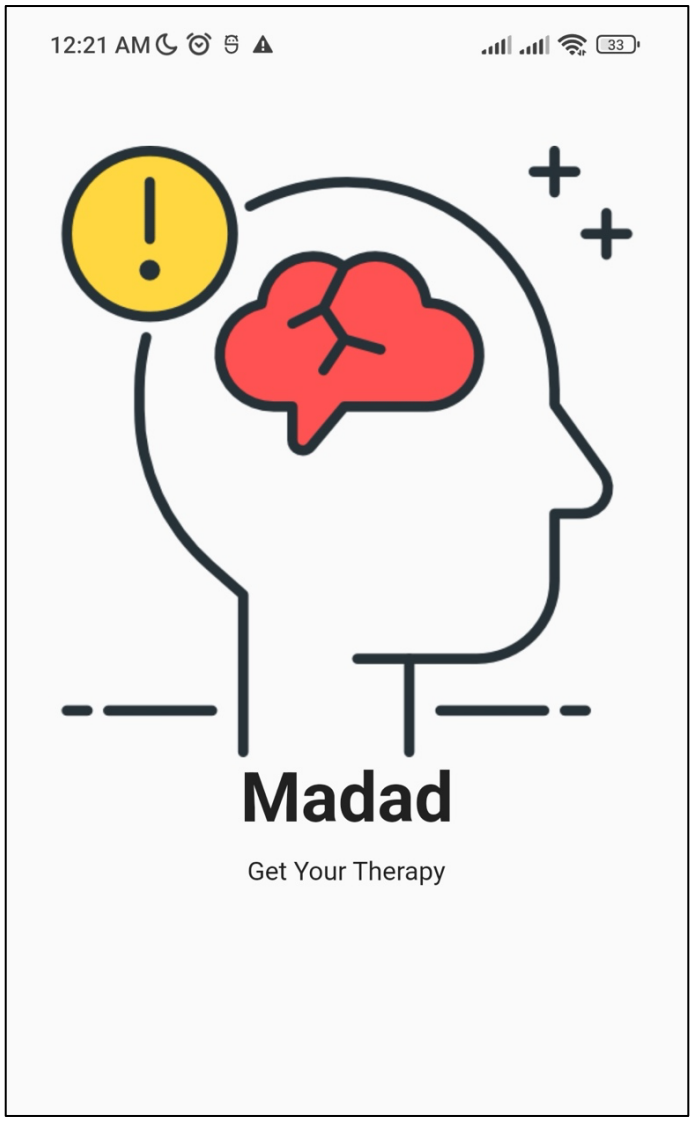


Figure 5.1: Splash Screen

The first screen that appears when the application is opened is called the Splash Screen. Users should enjoy a branded and visually appealing experience as the app loads its resources and gets ready to use.

5.2 Register Screen

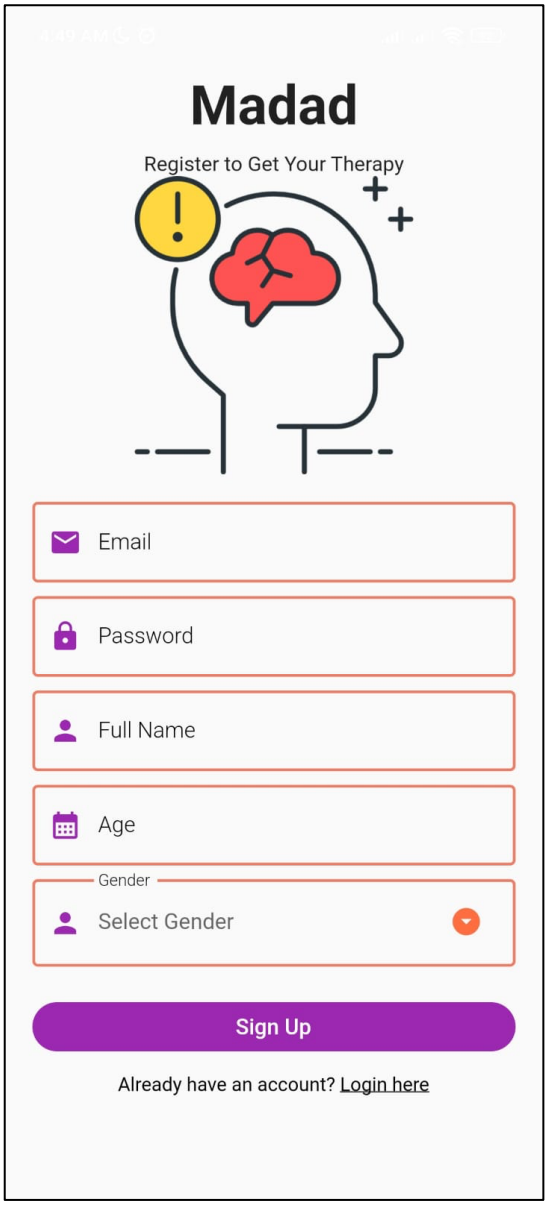


Figure 5.5.2: Register Screen

The Patient can establish a new account on the Register Screen to receive access to the application. The patient must enter their relevant data, including their full name, email address, password, age, and gender to register.

5.3 Login Screen

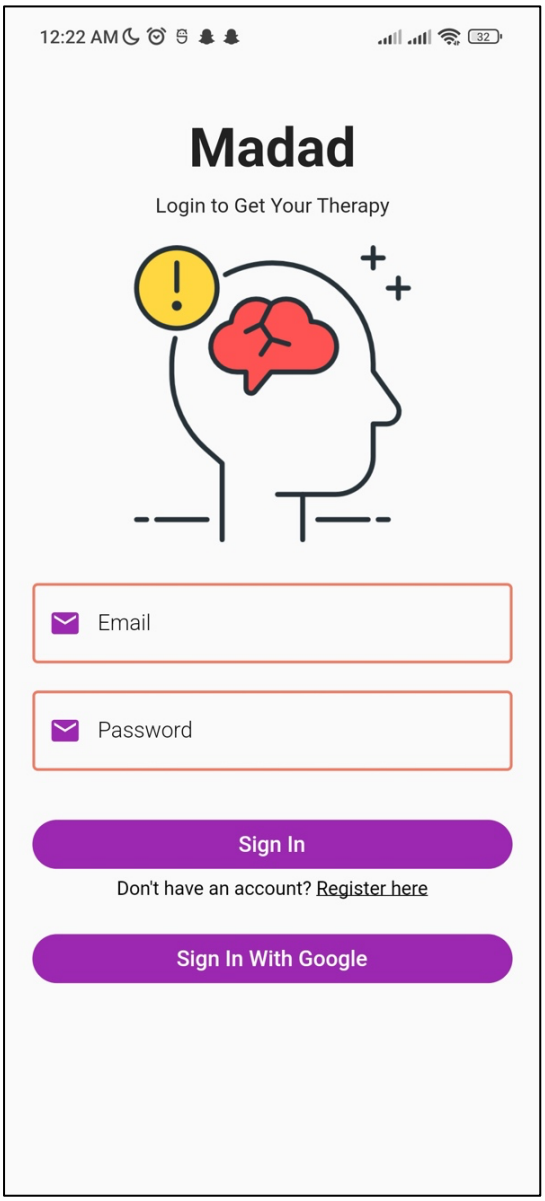


Figure 5.5.3: Login Screen

The patient, psychologist, and admin must enter their registered email address and password on the Sign-in Screen to log in. So, they can access their account by clicking the "Sign-in" button after providing their login information.

5.4 Home Screen

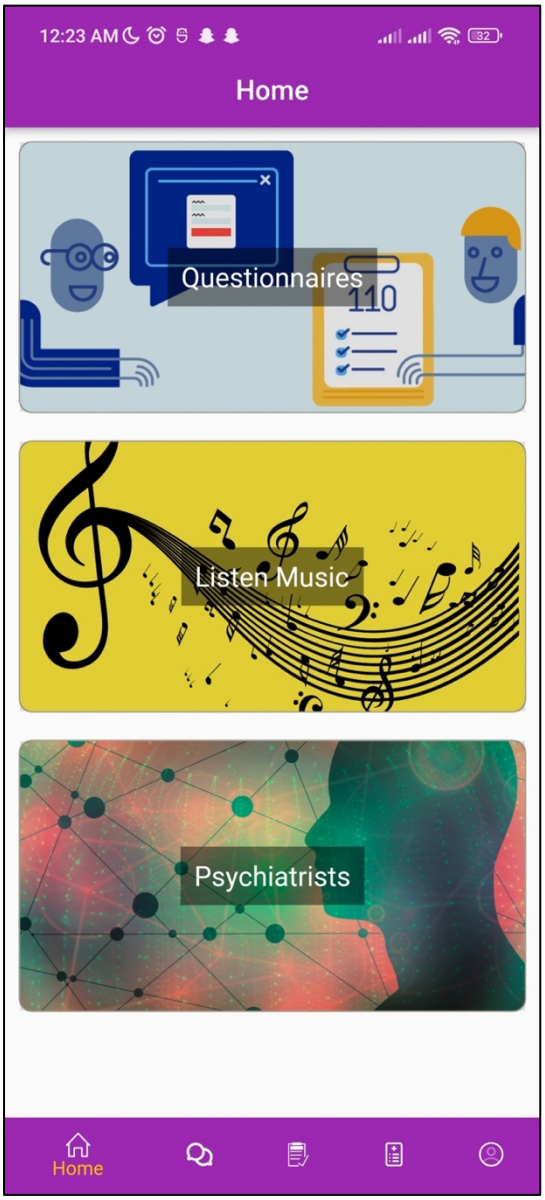


Figure 5.5.4: Home Screen

Patients have rapid access to key functions via the Home Tab. Patients can access the app's various sections, including those where they can add psychologists, access questionnaires, and discover audio materials. To explore and effectively use the features offered, they only need to touch on the appropriate selections under the Home Tab.

5.5 Questionnaires

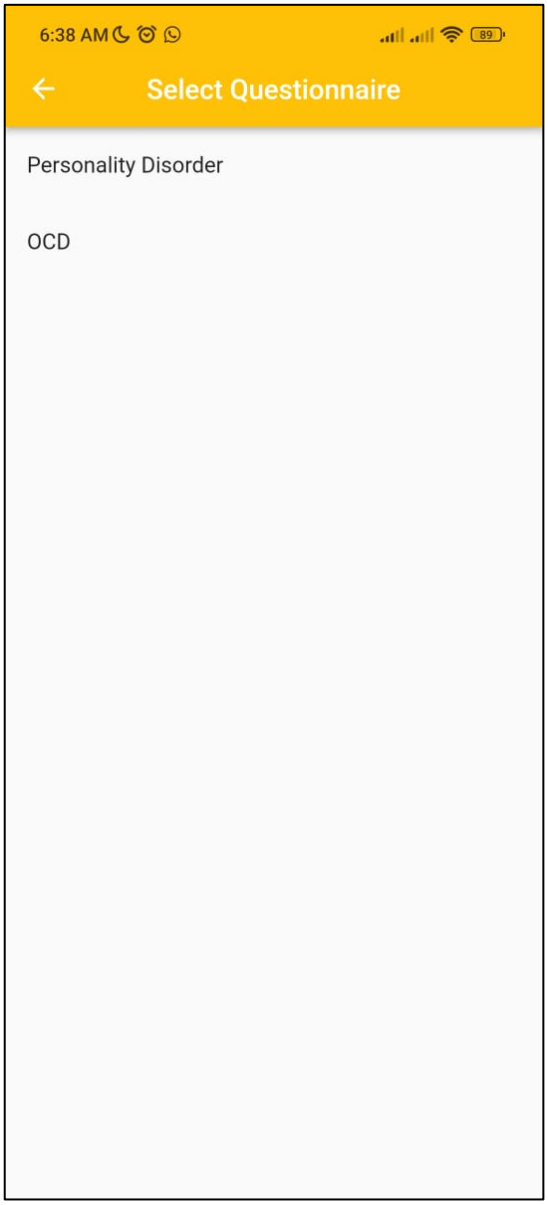


Figure 5.5.5: Questionnaires Screen

On this screen patients can select the desired questionnaire to assess their mental health.

5.6 Audio Player

Figure 5.5.6: Audio Player Screen

Patients can choose from a variety of audio files on the Audio Player screen to relax and improve their mental health. The accessible audios are simply navigable for patients, who can select their favourite and have a peaceful experience. Patients may easily play, pause, and change the audio settings thanks to the simple controls on the Audio Player interface.

5.7 Psychologists

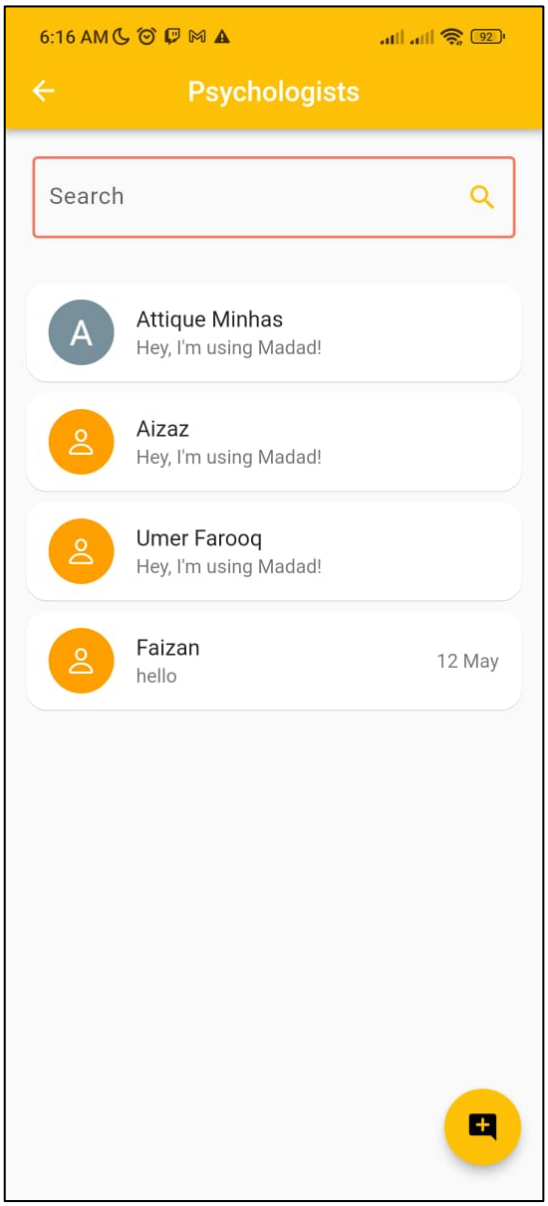


Figure 5.5.7: View Psychologists Screen

Patients can see any psychologist profile.

5.8 Chat Screen

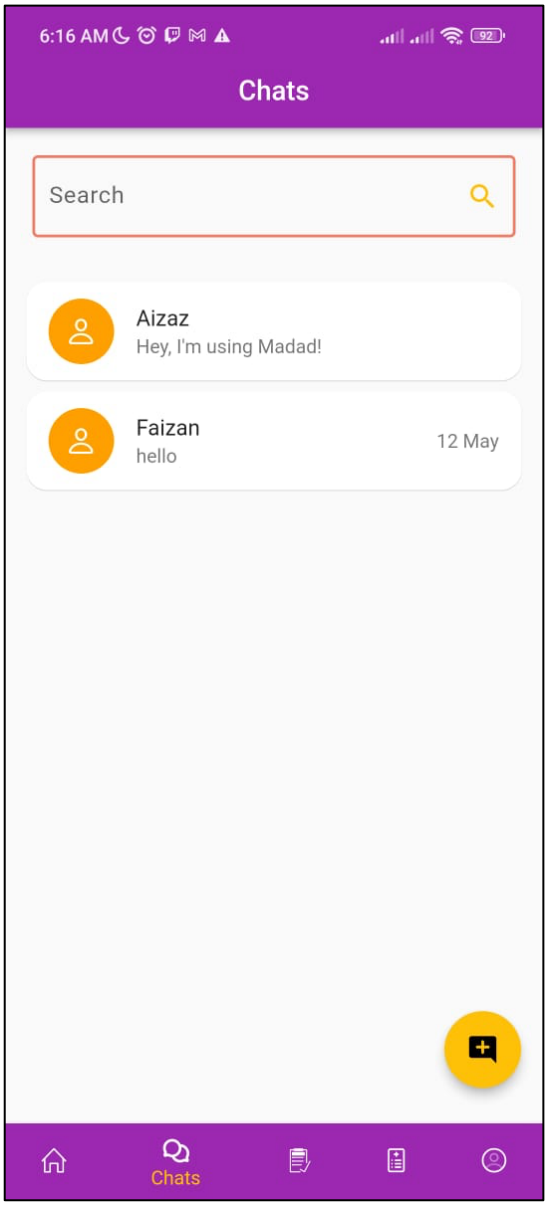


Figure 5.5.8: Chats Screen

Patients can see any previous chats and can create new chats with the psychologist.

5.9 Assessment Screen

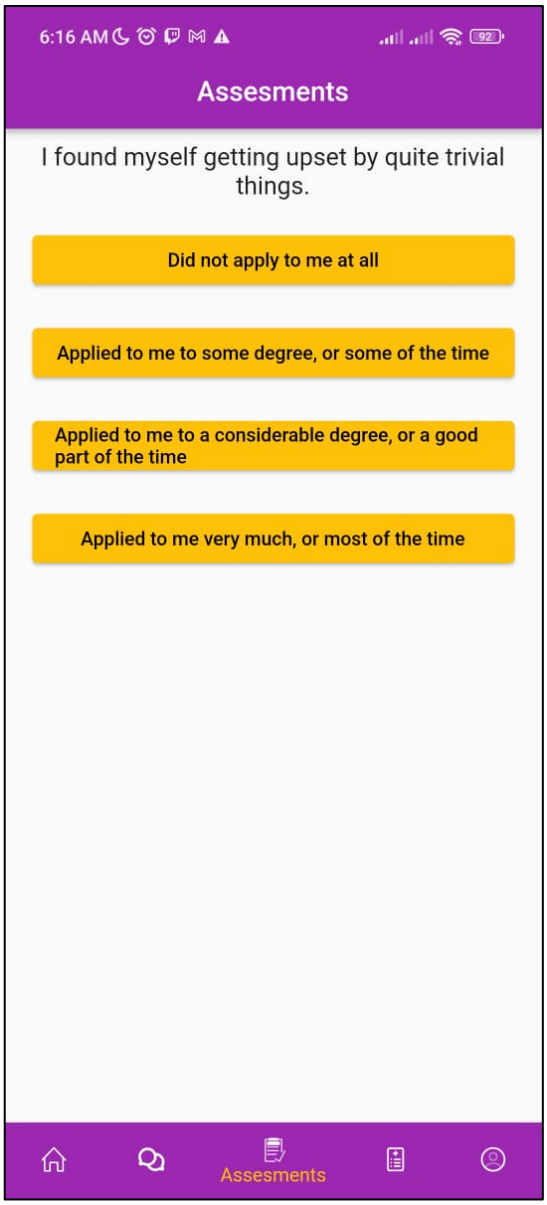


Figure 5.5.9: Assessments Screen

Patients can choose the assessment tab and attempt the questionnaire. After attempting the questionnaire, the patient can see the result.

5.10 Result Screen

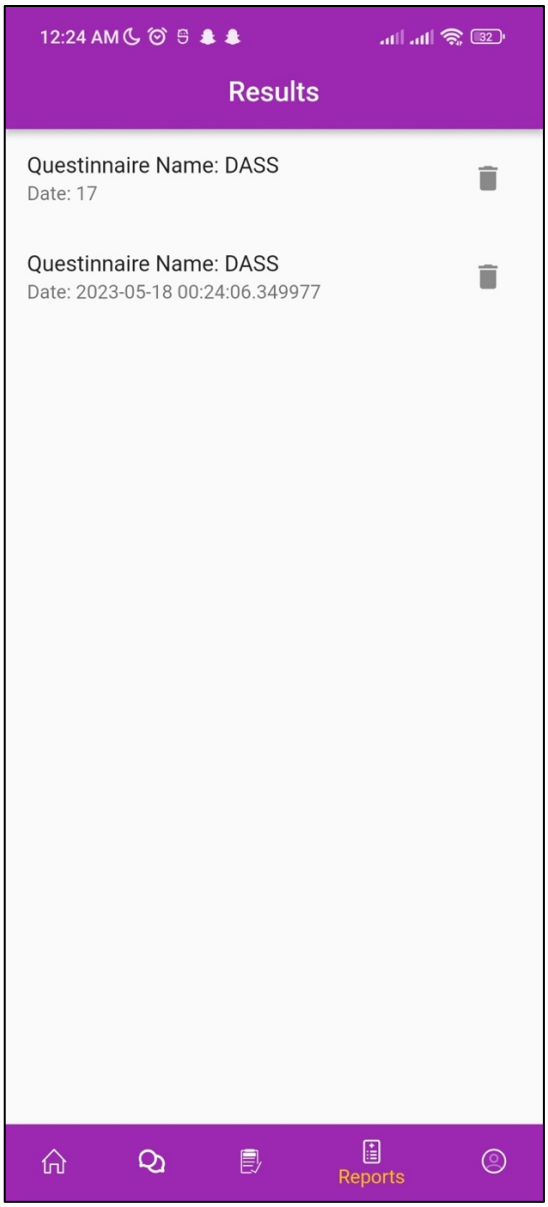


Figure 5.5.10: Results Screen

Patients can see any previous report and result of the previous assessment by the application. The patient can access any previous report and can access to the report information if needed.

5.11 Admin Screen

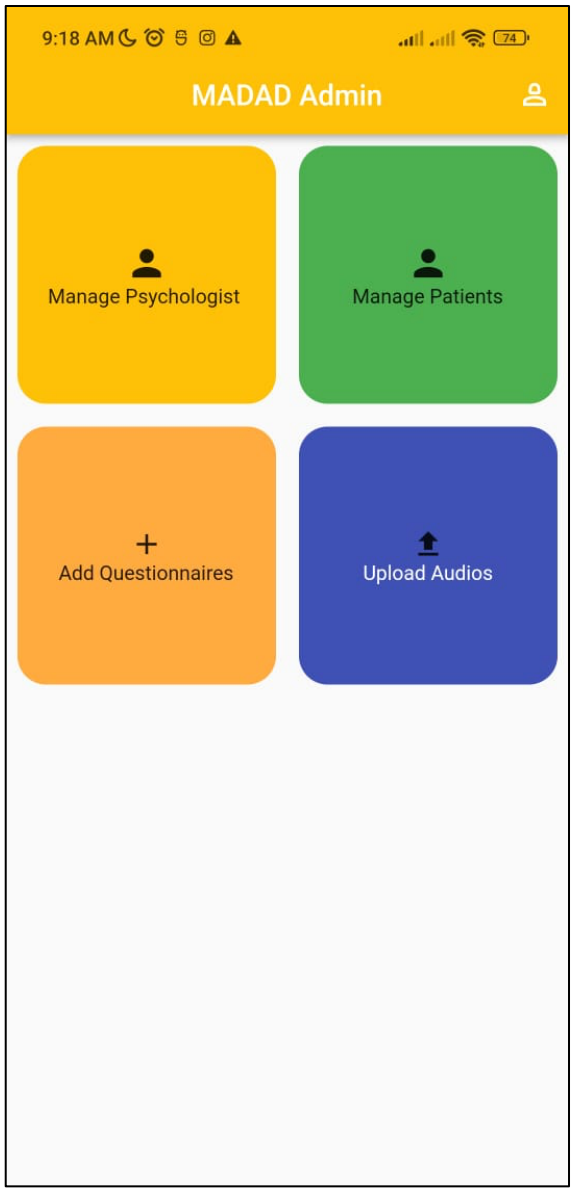


Figure 5.5.11: Admin Panel

This screen is the admin panel. From this screen, the admin can perform the given functions like manage patient and psychologist, add questionnaire and audios,etc.

5.12 Manage Psychologist

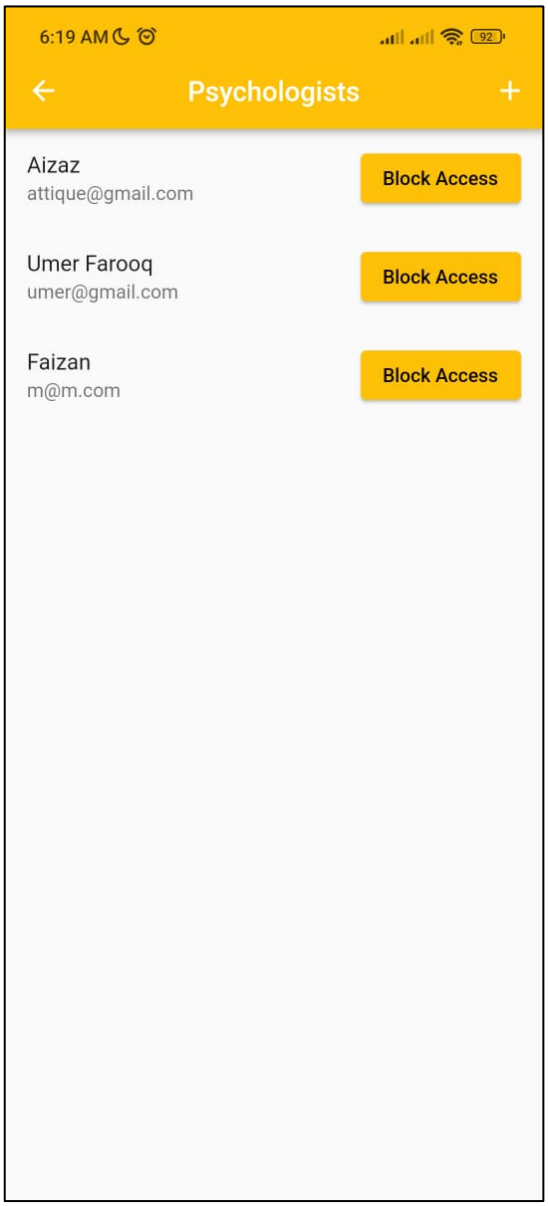


Figure 5.5.12: Manage Psychologists

From this panel, admin can add and block the psychologists.

5.13 Manage Audios

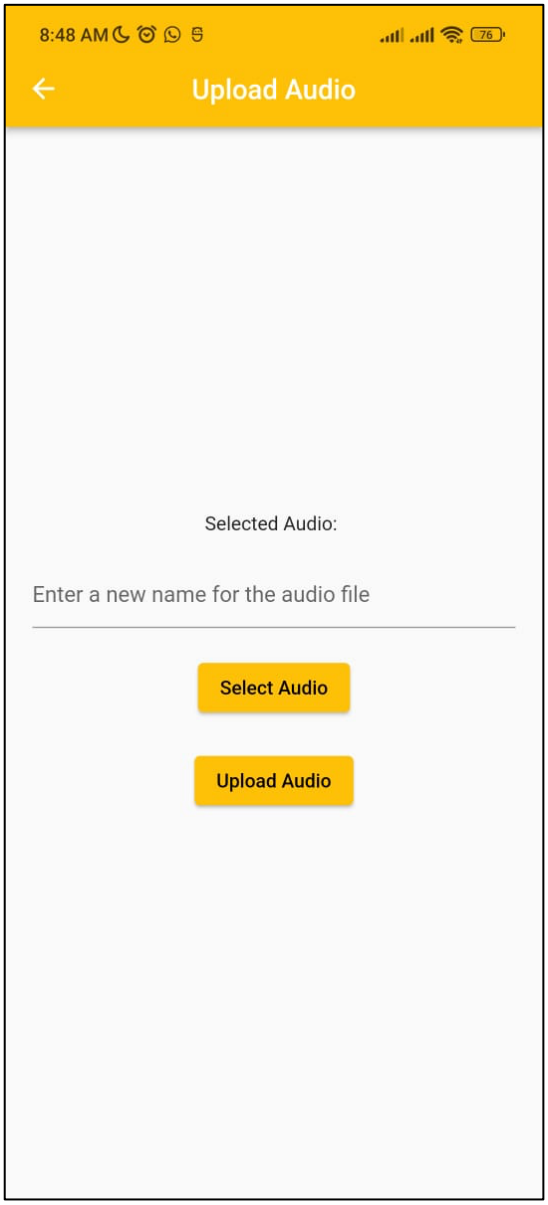


Figure 5.5.13: Manage Audios

From this panel, the admin can enter name of the audio and select and upload the audio as well.

5.14 Manage Questionnaires

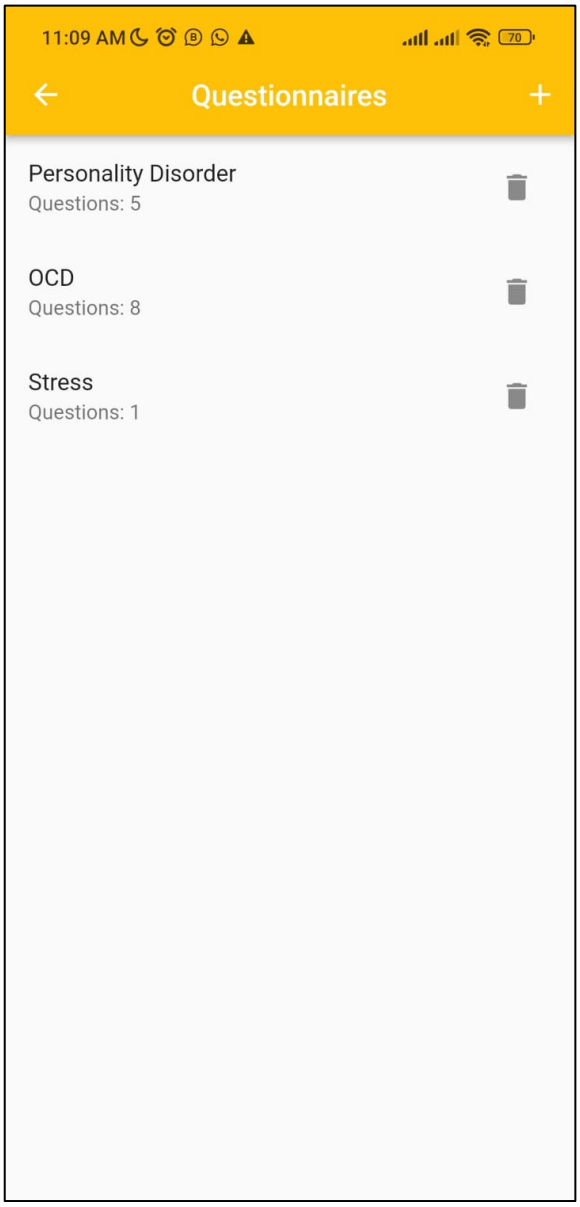


Figure 5.5.14: Manage Questionnaires

Questionnaire screen allows the admin to create a new questionnaire or delete the previous questionnaire.

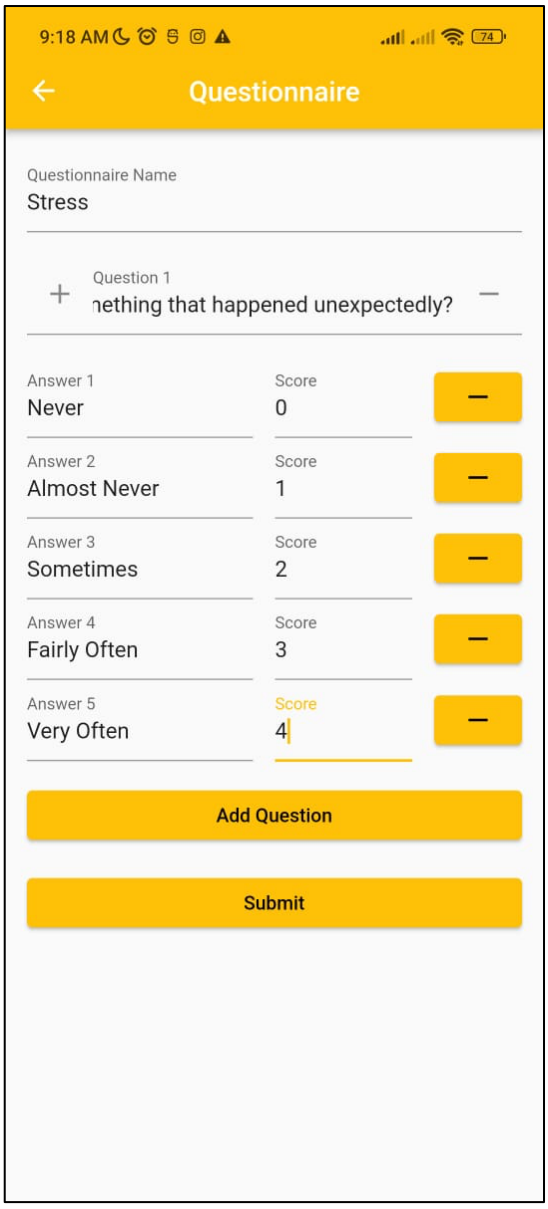


Figure 5.5.15: Uploading Questionnaire

After the admin click submit button the toast message will appear through firebase to notify the admin that the questionnaire is uploaded successfully.

Figure 5.5.16: Uploaded Successfully

5.15 Profile

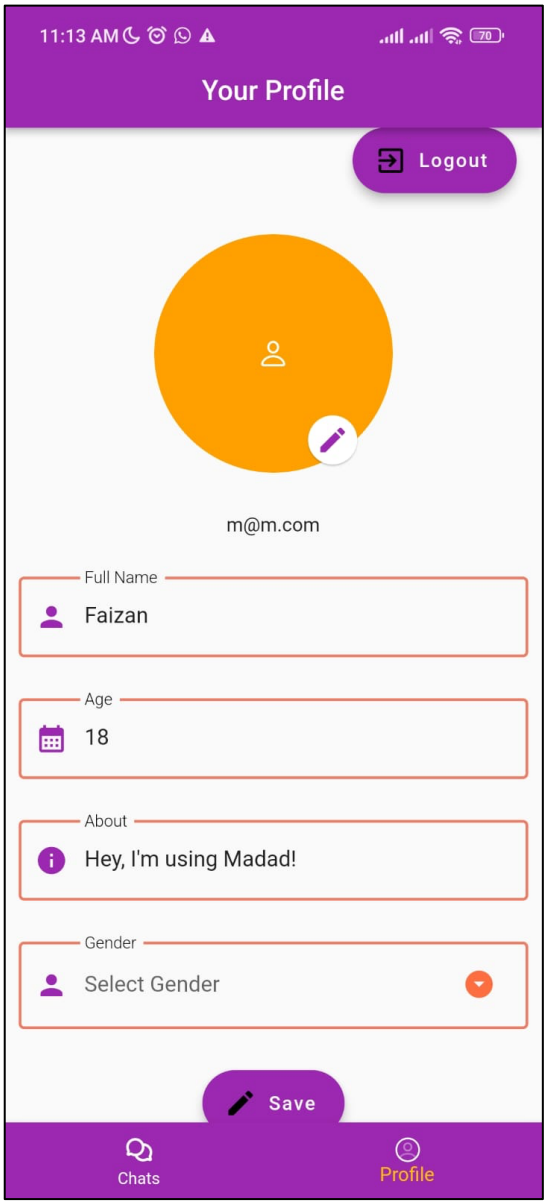


Figure 5.5.17: Profile Screen

Go to the "Profile" area of the application to check or edit your profile. From there, you may view all the details of your current profile, including your full name, age, and other relevant details. If you want to make any changes, choose to click on the respected field and remove previous data and enter new information. To make sure that your updated profile is successfully saved, don't forget to save the changes you've made.

5.16 Change Password

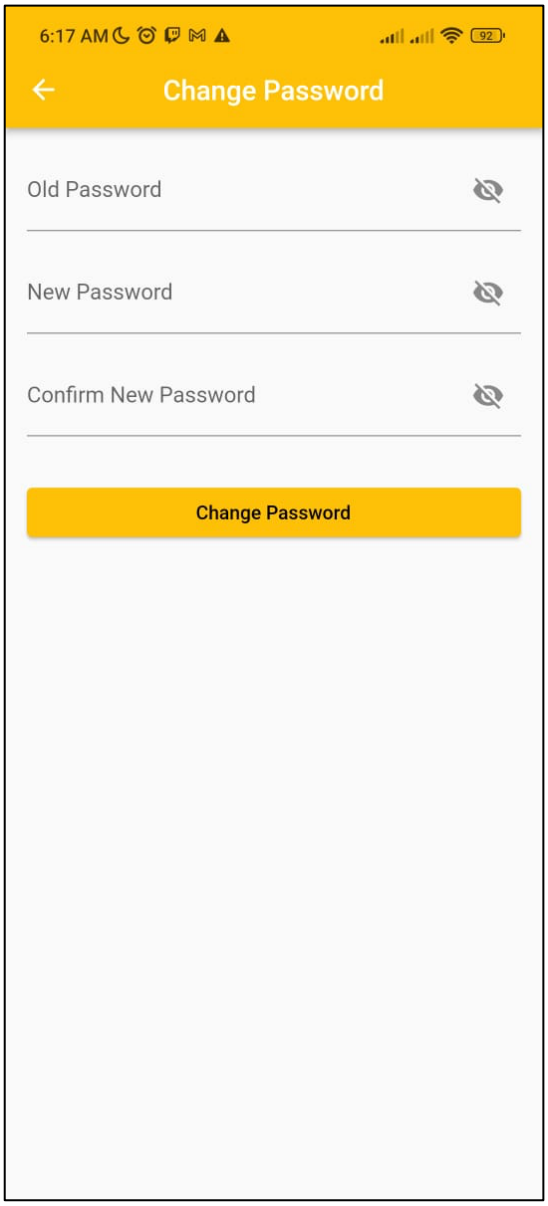


Figure 5.5.18: Change Password

Patients, psychologists, and administrators can all change their passwords within the application using the Change Password screen. Access the Change Password screen from the "Profile" section to modify your password.

CHAPTER 6

CONCLUSION AND RECOMMENDATIONS

6.1 Conclusion

In conclusion, Madad strives to remove obstacles related to mental health by utilizing contemporary technology and offering reachable mental health care. Our software gives users the tools to take charge of their mental health, have private chats, and access tailored resources. Madad works to create a society where everyone can prosper and live mentally healthy lives by fostering understanding, de-stigmatizing mental illness, and facilitating early intervention.

6.2 Recommendation

We will advise enhancing the user experience through ongoing design and navigation enhancements based on the research and analysis done during the creation and evaluation of the "Madad" mental health application. It is essential to broaden therapeutic options by incorporating new therapy methods like voice calls or video sessions. Building partnerships with organizations and professionals in the mental health field will aid in creating a vast network of skilled psychologists. To protect user information, it is crucial to give security and privacy measures top priority. To keep in line with consumer needs, changes and upgrades should be made often. It is advised to spread knowledge and awareness by using self-help tools and educational resources. The impact and reach of the application can be increased through cooperation with mental health initiatives and organizations.

REFERENCES

[1] M. S. Zulfiker, N. Kabir, A. A. Biswas, T. Nazneen, and M. S. Uddin, "An in-depth analysis of machine learning approaches to predict depression," *Current Research in Behavioral Sciences*, vol. 2, p. 100044, Nov. 2021, doi: 10.1016/J.CRBEHA.2021.100044.

[2] J. Chung and J. Teo, "Mental Health Prediction Using Machine Learning: Taxonomy, Applications, and Challenges," *Applied Computational Intelligence and Soft Computing*, vol. 2022, 2022, doi: 10.1155/2022/9970363.

[3] A. Fatima, Y. Li, T. T. Hills, and M. Stella, "DASentimental: Detecting Depression, Anxiety, and Stress in Texts via Emotional Recall, Cognitive Networks, and Machine Learning," *Big Data and Cognitive Computing 2021, Vol. 5, Page 77*, vol. 5, no. 4, p. 77, Dec. 2021, doi: 10.3390/BDCC5040077.

[4] U. M. Haque, E. Kabir, and R. Khanam, "Detection of child depression using machine learning methods," *PLoS One*, vol. 16, no. 12, p. e0261131, Dec. 2021, doi: 10.1371/JOURNAL.PONE.0261131.

[5] X. Luo and Y. Huang, "Mental Health Identification of Children and Young Adults in a Pandemic Using Machine Learning Classifiers," *Front Psychol*, vol. 13, p. 3759, Jul. 2022, doi: 10.3389/FPSYG.2022.947856/BIBTEX.

[6] T. Richter, B. Fishbain, A. Markus, G. Richter-Levin, and H. Okon-Singer, "Using machine learning-based analysis for behavioral differentiation between anxiety and depression," *Scientific Reports 2020 10:1*, vol. 10, no. 1, pp. 1–12, Oct. 2020, doi: 10.1038/s41598-020-72289-9.

[7] P. Vaishali Narayanrao and P. Lalitha Surya Kumari, "Analysis of Machine Learning Algorithms for Predicting Depression," *2020 International Conference on Computer Science, Engineering and Applications, ICCSEA 2020*, Mar. 2020, doi: 10.1109/ICCSEA49143.2020.9132963.

APPENDICES

APPENDIX A: Computer Programme Listing

```
import pandas as pd
import numpy as np
import tensorflow as tf
from tensorflow import keras
from sklearn.metrics import classification_report, roc_auc_score
import matplotlib.pyplot as plt
import seaborn as sns

SEED = 42
pd.set_option("display.max_columns", None)
```

```
df = pd.read_csv('data.csv', delimiter='\t')
```

```
df.head()
```

	Q1A	Q1I	Q1E	Q2A	Q2I	Q2E	Q3A	Q3I	Q3E	Q4A	Q4I	Q4E	Q5A	Q5I	Q5E
0	4	28	3890	4	25	2122	2	16	1944	4	8	2044	4	34	2153
1	4	2	8118	1	36	2890	2	35	4777	3	28	3090	4	10	5078
2	3	7	5784	1	33	4373	4	41	3242	1	13	6470	4	11	3927
3	2	23	5081	3	11	6837	2	37	5521	1	27	4556	3	28	3269
4	2	36	3215	2	13	7731	3	5	4156	4	10	2802	4	2	5628

```
df.shape
```

```
(39775, 172)
```

```
df.columns
```

```
Index(['Q1A', 'Q1I', 'Q1E', 'Q2A', 'Q2I', 'Q2E', 'Q3A', 'Q3I', 'Q3E', 'Q4A',
       ...
       'screensize', 'uniquenetworklocation', 'hand', 'religion',
       'orientation', 'race', 'voted', 'married', 'familysize', 'major'],
      dtype='object', length=172)
```

```
# build list of removed features to deleted from dataset
removedFeatures = [f'Q{i}E' for i in range(1, 43)] # add feature 'Q1E' to 'Q42E' to be removed
removedFeatures.extend([f'Q{i}I' for i in range(1, 43)]) # add feature 'Q1E' to 'Q42E' to be removed
removedFeatures.extend([f'VCL{i}' for i in range(1, 17)]) # add feature 'VCL1' to 'VCL16' to be removed
removedFeatures.extend([ 'source', 'introelapse', 'testelapse', 'surveyelapse', 'engnat', 'hand', 'orientation',
    'voted', 'country', 'screensize', 'uniquenetworklocation'])

# remove features from the dataset
depression = df.drop(removedFeatures, axis=1)
depression.head()
```

	Q1A	Q2A	Q3A	Q4A	Q5A	Q6A	Q7A	Q8A	Q9A	Q10A	Q11A	Q12A	Q13A	Q14A	Q15A	Q16A	Q17A	Q18A	Q19A	Q20A	Q21A
0	4	4	2	4	4	4	4	4	2	1	4	4	4	4	4	4	3	4	3	3	1
1	4	1	2	3	4	4	3	4	3	2	2	2	4	4	3	3	4	2	1	1	2
2	3	1	4	1	4	3	1	3	2	4	2	1	4	1	4	4	4	2	2	1	4
3	2	3	2	1	3	3	4	2	3	3	2	1	1	4	2	2	3	1	1	2	1
4	2	2	3	4	4	2	4	4	4	3	2	4	4	4	4	3	4	4	4	4	3

```
depression.shape
```

```
(39775, 61)
```

```
depression.isnull().sum()
```

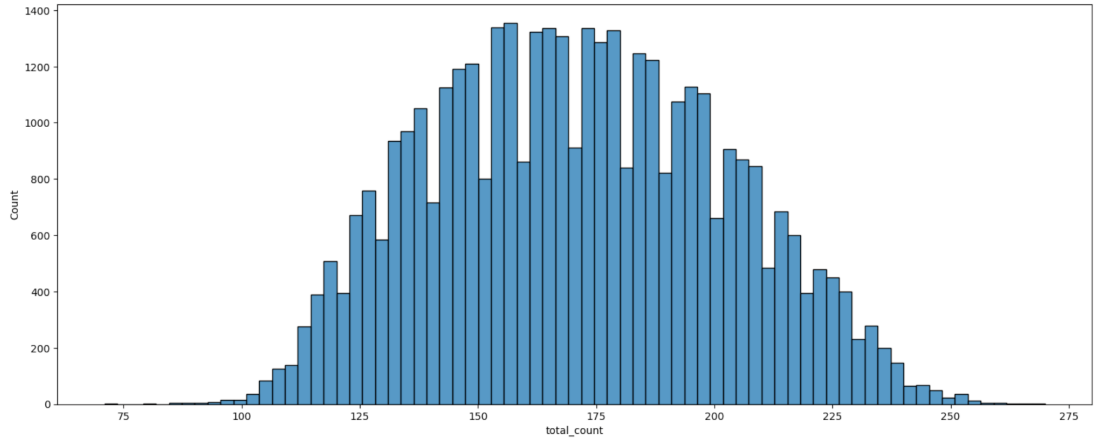
```
Q1A                 0
Q2A                 0
Q3A                 0
Q4A                 0
Q5A                 0
                 ...
religion            0
race                0
married             0
familysize          0
major           11403
Length: 61, dtype: int64
```

```
depression['total_count']  = depression.sum(axis=1)
depression.head()
```

	Q1A	Q2A	Q3A	Q4A	Q5A	Q6A	Q7A	Q8A	Q9A	Q10A	Q11A	Q12A	Q13A
0	4	4	2	4	4	4	4	4	2	1	4	4	4
1	4	1	2	3	4	4	3	4	3	2	2	2	4
2	3	1	4	1	4	3	1	3	2	4	2	1	4
3	2	3	2	1	3	3	4	2	3	3	2	1	1
4	2	2	3	4	4	2	4	4	4	3	2	4	4

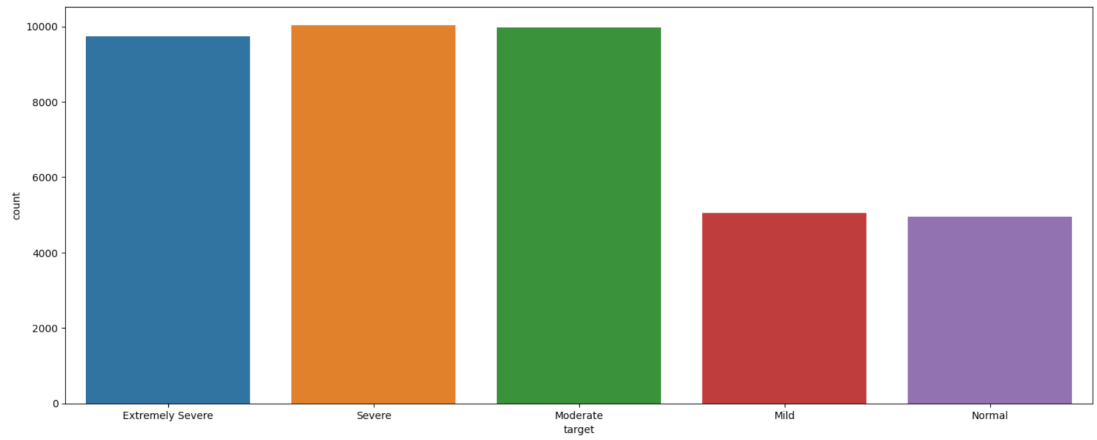
```
plt.figure(figsize=(18, 7))
sns.histplot(x=depression['total_count'])
```

```
<Axes: xlabel='total_count', ylabel='Count'>
```




```
plt.figure(figsize=(18, 7))
sns.countplot(x=depression['target'])
```

```
<Axes: xlabel='target', ylabel='count'>
```

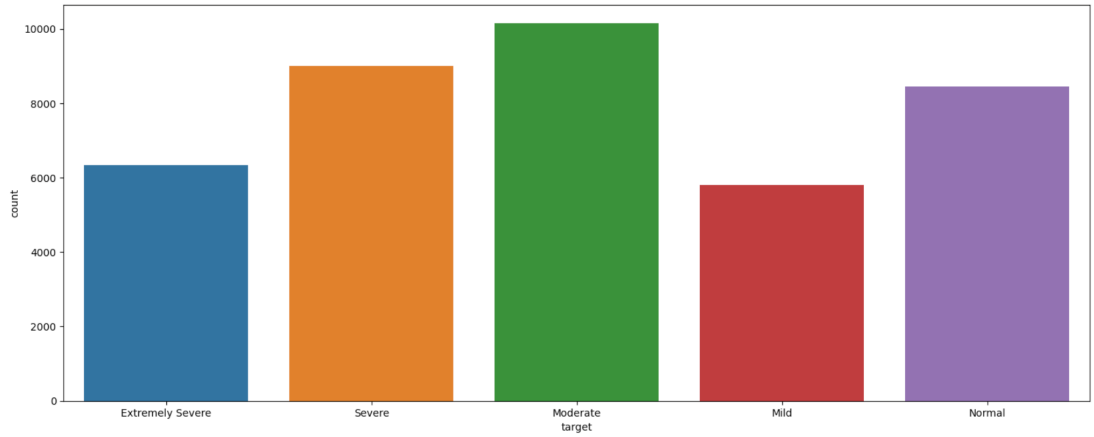



```
def buildTargetMove15Steps(value):
    if value <= 143:
        return 'Normal'
    if 143 < value <= 157:
        return 'Mild'
    if 157 < value <= 180:
        return 'Moderate'
    if 180 < value <= 204:
        return 'Severe'
    if value > 204:
        return 'Extremely Severe'


# build target feature
depression['target'] = depression['total_count'].apply(buildTargetMove15Steps)

# Let's visualize to see
plt.figure(figsize=(18, 7))
sns.countplot(x=depression['target'])
```

```
<Axes: xlabel='target', ylabel='count'>
```




```
feature_names = depression.columns
print(feature_names)
```

```
Index(['Q1A', 'Q2A', 'Q3A', 'Q4A', 'Q5A', 'Q6A', 'Q7A', 'Q8A', 'Q9A', 'Q10A',
       'Q11A', 'Q12A', 'Q13A', 'Q14A', 'Q15A', 'Q16A', 'Q17A', 'Q18A', 'Q19A',
       'Q20A', 'Q21A', 'Q22A', 'Q23A', 'Q24A', 'Q25A', 'Q26A', 'Q27A', 'Q28A',
       'Q29A', 'Q30A', 'Q31A', 'Q32A', 'Q33A', 'Q34A', 'Q35A', 'Q36A', 'Q37A',
       'Q38A', 'Q39A', 'Q40A', 'Q41A', 'Q42A', 'TIPI1', 'TIPI2', 'TIPI3',
       'TIPI4', 'TIPI5', 'TIPI6', 'TIPI7', 'TIPI8', 'TIPI9', 'TIPI10',
       'education', 'urban', 'gender', 'religion', 'race', 'married',
       'familysize', 'age_group', 'total_count', 'target'],
      dtype='object')
```

```
# split target from depression dataset
target = depression['target']
depression.drop(['target', 'total_count'], axis=1, inplace=True) # drop target and total_count
```

```
def buildTargetMove15Steps(value):
    if value == 'Normal':
        return 0
    if value == 'Mild':
        return 1
    if value == 'Moderate':
        return 2
    if value == 'Severe':
        return 3
    if value == 'Extremely Severe':
        return 4


# build target feature
target = target.apply(buildTargetMove15Steps)
```

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(depression, target, test_size=.2)

print(f'x_train: {x_train.shape}, y_train: {y_train.shape}')
print(f'x_test: {x_test.shape}, y_test: {y_test.shape}')
```

```
x_train: (31786, 60), y_train: (31786,)
x_test: (7947, 60), y_test: (7947,)
```

```
from sklearn.preprocessing import StandardScaler, MinMaxScaler

scaler = MinMaxScaler()

x_train_scaled = scaler.fit_transform(x_train)
x_test_scaled = scaler.transform(x_test)
```

```
# Define the Keras model
model = keras.Sequential([
    keras.layers.Dense(128, input_shape=(60,), activation='relu'),
    keras.layers.Dense(64, activation='relu'),
    keras.layers.Dense(32, activation='relu'),
    keras.layers.Dense(5, activation='softmax')
])
```

```
# Compile the model
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

# Train the model
model.fit(x_train_scaled, y_train, epochs=40)
test_loss, test_acc = model.evaluate(x_test_scaled, y_test)
print('Test accuracy:', test_acc)
```

```
Epoch 35/40
994/994 [==============================] - 3s 3ms/step - loss: 0.0844 - accuracy: 0.9664
Epoch 36/40
994/994 [==============================] - 2s 2ms/step - loss: 0.0527 - accuracy: 0.9784
Epoch 37/40
994/994 [==============================] - 2s 2ms/step - loss: 0.0643 - accuracy: 0.9740
Epoch 38/40
994/994 [==============================] - 2s 2ms/step - loss: 0.0638 - accuracy: 0.9746
Epoch 39/40
994/994 [==============================] - 2s 2ms/step - loss: 0.0563 - accuracy: 0.9782
Epoch 40/40
994/994 [==============================] - 2s 2ms/step - loss: 0.0635 - accuracy: 0.9747
249/249 [==============================] - 1s 1ms/step - loss: 0.1149 - accuracy: 0.9560
```

```
# Compile the model
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

# Train the model
model.fit(x_train_scaled, y_train, epochs=50)
test_loss, test_acc = model.evaluate(x_test_scaled, y_test)
print('Test accuracy:', test_acc)
```

```
Epoch 48/50
994/994 [==============================] - 2s 2ms/step - loss: 0.0452 - accuracy: 0.9819
Epoch 49/50
994/994 [==============================] - 3s 3ms/step - loss: 0.0525 - accuracy: 0.9797
Epoch 50/50
994/994 [==============================] - 3s 3ms/step - loss: 0.0426 - accuracy: 0.9835
249/249 [==============================] - 1s 1ms/step - loss: 0.0851 - accuracy: 0.9680
Test accuracy: 0.9680382609367371
```

```
# Compile the model
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

# Train the model
model.fit(x_train_scaled, y_train, epochs=60)
test_loss, test_acc = model.evaluate(x_test_scaled, y_test)
print('Test accuracy:', test_acc)
```

```
Epoch 59/60
994/994 [==============================] - 2s 2ms/step - loss: 0.0327 - accuracy: 0.9872
Epoch 60/60
994/994 [==============================] - 2s 2ms/step - loss: 0.0217 - accuracy: 0.9919
249/249 [==============================] - 1s 2ms/step - loss: 0.1094 - accuracy: 0.9653
Test accuracy: 0.965269923210144
```

```
# Compute the AUC score
y_test_onehot = keras.utils.to_categorical(y_test, num_classes=5)
y_pred_onehot = model.predict(x_test_scaled)
auc_score = roc_auc_score(y_test_onehot, y_pred_onehot, multi_class='ovr')
print('AUC score:', auc_score)

# Convert the Keras model to a TensorFlow Lite model
converter = tf.lite.TFLiteConverter.from_keras_model(model)
tflite_model = converter.convert()

# Save the TensorFlow Lite model to a file
with open('model.tflite', 'wb') as f:
    f.write(tflite_model)
```

```
249/249 [==============================] - 0s 1ms/step
AUC score: 0.9987061626659391
```

```
# Evaluate the model on the test data
y_pred = model.predict(x_test_scaled)
y_pred_classes = np.argmax(y_pred, axis=1)

# Print the accuracy classification report
print(classification_report(y_test, y_pred_classes))
```

```
249/249 [==============================] - 0s 1ms/step
              precision    recall  f1-score   support

           0       0.98      0.99      0.98      1643
           1       0.95      0.94      0.95      1132
           2       0.98      0.95      0.96      2047
           3       0.95      0.95      0.95      1845
           4       0.95      0.99      0.97      1280

    accuracy                           0.96      7947
   macro avg       0.96      0.96      0.96      7947
weighted avg       0.96      0.96      0.96      7947
```