**ÇANKAYA UNIVERSITY**

**FACULTY OF ENGINEERING**

**COMPUTER ENGINEERING DEPARTMENT**

# Project Report

## CENG 408

Innovative System Design and Development II

## *PROJECT ID:202421*

## *SenseAI*

*Ece Heval Ünal* - *202111014*

*Zeliha Aybüke Baştürk* - *202028034*

*Özge Alkan* - *202128406*

*Mustafa Tozman* – *202128014*

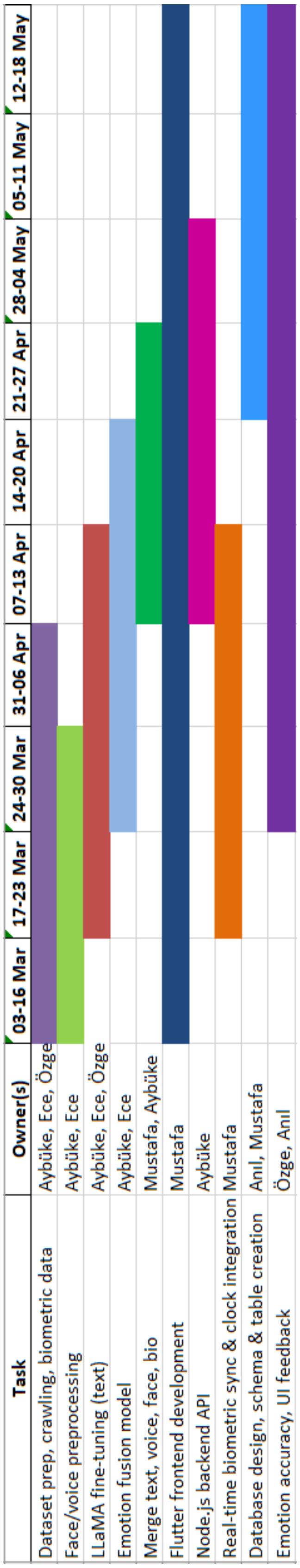
*Sadrettin Anıl Karaçay* - *202011046*

Advisors: *Dr. Atila BOSTAN*

*Dr. İsmail Bora ÇELİKKALE*

# Table of Contents

## Work Plan



| Task | Owner(s) | 03-16 Mar | 17-23 Mar | 24-30 Mar | 31-06 Apr | 07-13 Apr | 14-20 Apr | 21-27 Apr | 28-04 May | 05-11 May | 12-18 May |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Dataset prep, crawling, biometric data | Aybüke, Ece, Özge | ■ | ■ | ■ | ■ | | | | | | |
| Face/voice preprocessing | Aybüke, Ece | ■ | ■ | ■ | | | | | | | |
| LLaMA fine-tuning (text) | Aybüke, Ece, Özge | | ■ | ■ | ■ | ■ | | | | | |
| Emotion fusion model | Aybüke, Ece | | | ■ | ■ | ■ | ■ | | | | |
| Merge text, voice, face, bio | Mustafa, Aybüke | | | | | ■ | ■ | ■ | | | |
| Flutter frontend development | Mustafa | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |
| Node.js backend API | Aybüke | | | | | ■ | ■ | ■ | ■ | | |
| Real-time biometric sync & clock integration | Mustafa | | ■ | ■ | ■ | ■ | | | | | |
| Database design, schema & table creation | Anıl, Mustafa | | | | | | | ■ | ■ | ■ | ■ |
| Emotion accuracy, UI feedback | Özge, Anıl | | | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |

## Abstract

Mental health issues are becoming increasingly common, yet many individuals still face barriers in accessing timely and effective support. Long waiting times, lack of personalization, and the feeling of not being truly understood can worsen emotional distress. To address these challenges, we developed SenseAI — a smart, multimodal psychological chatbot designed to recognize and respond to a user's emotional state in real time.

SenseAI integrates several advanced technologies: natural language processing (powered by LLaMA), facial emotion recognition (using DeepFace), voice tone analysis, and heart rate variability monitoring via wearable devices such as Xiaomi Mi Band. These modalities are fused to form a unified emotional understanding, enabling the system to deliver contextually appropriate and empathetic responses.

Built with Flutter and Node.js, and integrated with services like Firebase and Zepp Life, SenseAI provides adaptive, human-like interaction. Our testing showed that multimodal input analysis significantly outperforms single-input systems in recognizing nuanced emotional states. By analyzing how users speak, what they say, their facial cues, and physiological signals, SenseAI offers more accurate and supportive mental health interactions.

During development, we tackled key challenges such as GPU memory limitations, multimodal data synchronization, and dataset optimization. Despite these, the system successfully achieved responsive emotion-driven dialogues and was recognized by TÜBİTAK 2209-A, advancing to the Teknofest 2025 Project Presentation Stage.

SenseAI demonstrates the promising potential of multimodal AI in delivering accessible, emotionally intelligent mental health support — offering a more immediate, personalized, and human-centered solution in a world where mental well-being is more critical than ever.

**Key words:** Multimodal emotion recognition, psychological chatbot, natural language processing, wearable devices, mental health AI, affective computing

# Özet

Ruh sağlığı sorunları giderek daha yaygın hale gelse de, birçok birey zamanında ve etkili destek alma konusunda hâlâ engellerle karşılaşmaktadır. Uzun bekleme süreleri, kişiselleştirme eksikliği ve gerçekten anlaşılamama hissi, duygusal sıkıntıları daha da derinleştirebilir. Bu sorunlara çözüm getirmek amacıyla, kullanıcıların duygusal durumlarını gerçek zamanlı olarak tanıyıp yanıt verebilen akıllı ve çok modlu bir psikolojik sohbet botu olan SenseAI geliştirildi.

SenseAI, birden fazla gelişmiş teknolojiyi entegre eder: doğal dil işleme (LLaMA tabanlı), yüz ifadesi tanıma (DeepFace kullanılarak), ses tonu analizi ve Xiaomi Mi Band gibi giyilebilir cihazlar üzerinden kalp atış hızı değişkenliği takibi. Bu modlar bir araya getirilerek bütünsel bir duygusal anlayış oluşturulur; böylece sistem, bağlama uygun ve empatik yanıtlar verebilir.

Flutter ve Node.js ile geliştirilen SenseAI; Firebase ve Zepp Life gibi servislerle entegre çalışarak, uyumlu ve insan benzeri bir etkileşim sunar. Yaptığımız testler, çok modlu girdi analizinin, yalnızca tek bir giriş türüne dayalı sistemlere kıyasla, duygusal durumları tanımada belirgin şekilde daha başarılı olduğunu ortaya koymuştur. Kullanıcının ne söylediğini, nasıl söylediğini, yüz ifadelerini ve fizyolojik sinyallerini birlikte analiz eden SenseAI, daha doğru ve destekleyici ruh sağlığı etkileşimleri sunar.

Geliştirme sürecinde GPU bellek sınırlamaları, çok modlu veri senkronizasyonu ve veri seti optimizasyonu gibi temel zorluklarla karşılaşıldı. Buna rağmen, sistem başarılı bir şekilde duygu odaklı diyaloglar gerçekleştirmeyi başardı ve TÜBİTAK 2209-A kapsamında desteklenerek Teknofest 2025 Proje Sunum Aşaması'na yükseldi.

SenseAI, çok modlu yapay zekânın erişilebilir, duygusal zekâya sahip ruh sağlığı desteği sunmadaki potansiyelini ortaya koymaktadır — zihinsel iyi oluşun her zamankinden daha önemli olduğu günümüzde, daha hızlı, kişisel ve insan odaklı bir çözüm sunmaktadır.

**Anahtar kelimeler:** Çok modlu duygu tanıma, psikolojik sohbet botu, doğal dil işleme, giyilebilir cihazlar, ruh sağlığı yapay zekâsı, duyuşsal bilişim

# 1. Introduction

In recent years, there has been a rapid expansion in the development of digital mental health tools, particularly those powered by artificial intelligence. AI-driven systems now play an increasingly important role in analyzing and interpreting emotional states, offering the potential for more accessible, scalable, and personalized psychological support. With the advancement of wearable technologies and machine learning models, it is now possible to combine diverse data sources—including written text, vocal tone, facial expressions, and biometric signals such as heart rate variability—to gain a more holistic understanding of an individual's emotional well-being.

This project, SenseAI, aims to develop a psychology-focused chatbot that utilizes multimodal data to detect emotional states, deliver personalized feedback, and monitor users' emotional patterns over time. The chatbot integrates technologies such as transformer-based natural language processing (LLaMA), facial emotion recognition (DeepFace), speech-based emotion analysis, and physiological data interpretation from wearable devices.

The goal of this project is to explore how multimodal emotion analysis can be applied to create more effective and personalized digital mental health tools. This report outlines the system requirements, design decisions, implementation plan, and results obtained from developing and testing the SenseAI system.

## 1.1 Problem Statement

In modern life, many individuals face emotional distress and sudden psychological needs that remain unmet due to limited access to timely and personalized support. Existing digital tools often rely on single-modality inputs—typically text—which restricts their ability to understand the full emotional context. This leads to generic and emotionally disconnected responses. There is a clear need for intelligent systems that combine multiple modalities such as speech, facial expressions, and physiological data to offer more accurate, empathetic, and dynamic mental health support.

## 1.2 Background and Motivation

### 1.2.1 The Role of Chatbots in Mental Health Support

In recent years, digital tools have increasingly played a significant role in supporting mental health, with AI-powered chatbot systems emerging as key tools for addressing emotional needs. Mental health support chatbots provide users with a safe, non-judgmental environment to express their emotions and thoughts, offering a valuable space for self- reflection, stress management, and basic psychological support.

Research has shown that even text-based chatbots have a positive impact on users' mental well-being. For instance, applications like Wysa and Woebot employ Cognitive Behavioral Therapy (CBT) techniques to help users manage stress and anxiety effectively. However, these text-based systems have inherent limitations, as they rely solely on verbal input to interpret users' emotions and thoughts. This project, SenseAI, aims to overcome these limitations by incorporating multimodal data—such as voice tone, facial expressions, and biometric signals—to enable a more comprehensive emotional assessment, thus fostering more empathetic and accurate interactions.

The potential for chatbots to offer mental health support is immense, particularly as they can provide immediate assistance in times of need, help track emotional changes over time, and offer coping strategies that may reduce the burden on

traditional mental health services. By incorporating multimodal emotional analysis, SenseAI seeks to deliver a more human-like, empathetic interaction that surpasses the capabilities of standard text-based chatbot solutions.

### 1.2.2 Advancements in Multimodal Emotional Analysis

The field of emotion recognition has progressed significantly, moving beyond traditional text-based sentiment analysis to sophisticated multimodal approaches that integrate visual and physiological data. Multimodal emotional analysis enables more accurate predictions by combining inputs from various sources, such as voice tone, facial expressions, and heart rate variability (HRV).

Multimodal emotion recognition systems leverage physical and behavioral indicators—such as speech tone, facial cues, and biometric signals—to construct a nuanced understanding of users' emotional states. For instance, HRV is a valuable biometric indicator often associated with stress levels, while facial expression analysis can reveal emotions like happiness, sadness, or anger. By combining these modalities, multimodal systems achieve higher accuracy and allow for a deeper understanding of the user’s emotional context, ultimately enabling more empathetic responses in chatbot interactions. This project leverages these advancements to build a mental health chatbot that provides robust emotional support by integrating multimodal data sources to improve both the relevance and empathy of its responses.

### 1.3 Solution Statement

Our solution addresses the challenge of accurately detecting and responding to human emotions in conversational AI by integrating multimodal data sources—text, facial expressions, and biometric signals—into a unified framework. By leveraging advanced transformer models for natural language processing, convolutional neural networks for facial expression recognition, and time-series analysis for biometric data, SenseAI provides a nuanced understanding of the user's emotional state. This enables the chatbot to deliver empathetic and contextually appropriate responses, enhancing user experience in mental health support.

### 1.4 Contribution

This project presents SenseAI, a multimodal psychological assistant that analyzes text, voice, facial expressions, and heart rate data to assess emotional states and provide personalized support. Unlike traditional chatbots, it combines Transformer-based NLP with audio-visual and biometric inputs for more emotionally intelligent interactions. The system offers real-time, adaptive responses and demonstrates strong potential for both research and practical mental health applications.

## 2. Literature Search

### 2.1 Related Articles

Research in multimodal emotion recognition and AI-driven mental health support has expanded in recent years. This section reviews notable studies and advancements that provide a foundation for SenseAI’s approach to multimodal emotional analysis.

- **Multimodal Emotion Recognition Based on Facial Expressions, Speech, and EEG**

  Wang et al. (2022) propose a deep learning-based multimodal emotion recognition system called “Deep-Emotion,” which combines facial expressions, voice, and electroencephalogram (EEG) data for high accuracy in emotion recognition. This study highlights the potential of integrating physiological signals for robust emotional analysis, which informs SenseAI’s approach in utilizing biometric data alongside text and visual inputs to capture user emotions comprehensively.[1]

- **A Survey of Deep Learning-Based Multimodal Emotion Recognition: Speech, Text, and Face**

  Lian et al. (2023) examine various emotion recognition techniques across modalities like text, audio, and facial expressions. This survey categorizes deep learning models based on their performance in emotion recognition and offers a comparison of popular datasets. The findings emphasize the importance of choosing appropriate datasets and techniques for each modality, a principle applied in SenseAI’s model selection and data fusion methods.[2]

- **Multimodal Emotion Recognition Based on Facial Expressions, Speech, and Body Gestures**

  Yan et al. (2024) explore a three-modality system for emotion recognition, integrating facial expressions, speech, and body gestures to enhance recognition accuracy. This study demonstrates the benefits of analyzing non-verbal cues like gestures in emotional detection, underscoring the value of multimodal fusion. Although SenseAI does not use body gestures, this study supports the efficacy of combining visual and auditory data to achieve a more nuanced emotional understanding.[3]

- **Depression Detection Using Multimodal Analysis with Chatbot Support**

  Sharma et al. (2023) investigate an AI system for early depression detection through multimodal analysis, using text, audio, and image data. This approach aligns closely with SenseAI’s objectives, as both systems aim to provide real-time emotional support using AI-driven insights. The study emphasizes the role of multimodal data in detecting nuanced emotional states, reinforcing SenseAI’s focus on combining text, facial expressions, and biometric data for accurate emotional tracking.[4]

## 2.2 Similar Projects

Several existing projects have explored the use of AI-driven chatbots in mental health support, each with unique features and limitations. Reviewing these projects provides context for SenseAI's unique approach to multimodal integration.

### 2.2.1 Only Text-Based Chatbots

Most psychology-focused chatbots currently rely solely on text input to gauge user emotions and provide support. These include:

- **Woebot**: Uses Cognitive Behavioral Therapy (CBT) techniques to help users manage emotions. It focuses on emotional support but lacks multimodal analysis capabilities, relying only on text-based interactions.[5]
- **Wysa**: Offers tools for stress and anxiety management, including CBT and mindfulness exercises. The premium version provides access to therapists, though the

chatbot’s emotional understanding is limited to text.[6]

- **Youper**: Combines mood tracking with CBT techniques to enhance emotional awareness. It directs users to therapists when needed, although its capabilities are restricted to analyzing textual inputs.[7]
- **Headspace**: Primarily focused on stress and anxiety management, allowing users to connect with mental health professionals. While effective for guidance, it does not leverage multimodal data to improve emotional understanding.[8]

### 2.2.2 Ellie (SimSensei)

Developed at the University of Southern California and supported by DARPA, **Ellie** is an AI-driven system designed to assess mental health, primarily for military veterans.

It combines facial expression analysis and voice tone detection, offering a closer parallel to SenseAI’s multimodal approach. Ellie leverages tools like Stanford NLP, OpenFace, and OpenCV to analyze user emotions, with techniques like Pitch Tracking and Vocal Timbre Analysis to improve emotional accuracy. While Ellie focuses on veterans’ mental health assessment, its multimodal architecture validates the use of both facial and vocal data in emotion recognition.[9]

### 2.2.3 Replika AI

**Replika AI** provides a virtual companion for users, allowing communication through text and voice, though it does not utilize image or biometric data for emotion recognition. While Replika aims to fulfill companionship needs, it does not offer advanced emotional support features based on real-time multimodal inputs. Replika’s focus on companionship rather than psychological support sets it apart from SenseAI’s goal of delivering empathetic and clinically supportive interactions.[10]

# 3. Summary

## 3.1 Summary of Conceptual Solution

SenseAI is a multimodal psychological chatbot that integrates **textual**, **facial expression**, **biometric**, and **speech** data to perform robust emotion recognition. The system extracts emotional cues from each modality, assigns dynamic weights based on data availability and confidence, then fuses the results to infer the user's overall emotional state. This multimodal fusion improves accuracy and contextual awareness for more personalized and empathetic chatbot responses.

**Modalities and Processing**

| **Modality** | **Input Data** | **Processing Model** | **Weight in Fusion** |
|---|---|---|---|
| Text | User's typed or spoken text | Transformer-based NLP model | Dynamic, ~0.6-1.0* |
| Facial Expression | Uploaded images/videos of user face | CNN-based emotion classifier | 0 or 0.3 |
| Biometric Signals | Heart Rate Variability & Stress arrays | Signal processing + statistics | 0.1 or 0.2 |
| Speech | Recorded voice audio | Acoustic emotion recognition | 0 or 0.1 |

Text weight is adjusted dynamically depending on presence of face, bio, and voice inputs.

**Algorithmic Flow**

1. **Data Collection**:
   Receive user inputs including:
   - text (mandatory)
   - hr_array, stress_array (biometric arrays, mandatory)
   - multiple face images (optional)
   - voice audio file (optional)
2. **Individual Modal Analyses**:
   - **Text Analysis**: Transformer model classifies emotion from text.

- **Face Analysis**: For each image, CNN model outputs detected emotion. The most frequent valid detected emotion is selected.
- **Biometric Analysis**: HRV and stress data processed to determine bio-emotional state (e.g., calm, stressed).
- **Speech Analysis**: Acoustic features extracted and classified into emotion categories.

3. **Emotion Mapping**:
   Each detected emotion label is mapped to an integer value (e.g., angry=1, fear=2, disgust=3, sad=4, neutral=5, happy=7, surprised=8) to allow weighted numerical fusion.
4. **Dynamic Weight Assignment**:
   Weights are assigned depending on modality presence. This ensures total weights sum to 1, prioritizing text but enhancing it with other modalities when available.
5. **Fusion & Decision**:
   The final emotion score is computed as weighted sum. Rounded and clamped to valid emotion indices, then converted back to emotion labels.
6. **Response Generation**:
   The inferred overall emotion and input text are fed to the LLaMA-based chatbot to generate a psychologically informed response.

**Example Weights Calculation (if all modalities present):**

| **Modality** | **Weight** | **Emotion Value (e.g., happy=7)** | **Weighted Contribution** |
|---|---|---|---|
| Face | 0.3 | 7 | 2.1 |
| Voice | 0.1 | 5 | 0.5 |
| Bio | 0.1 | 5 | 0.5 |
| Text | 0.5 | 7 | 3.5 |
| **Total** | **1.0** | | **6.6** |

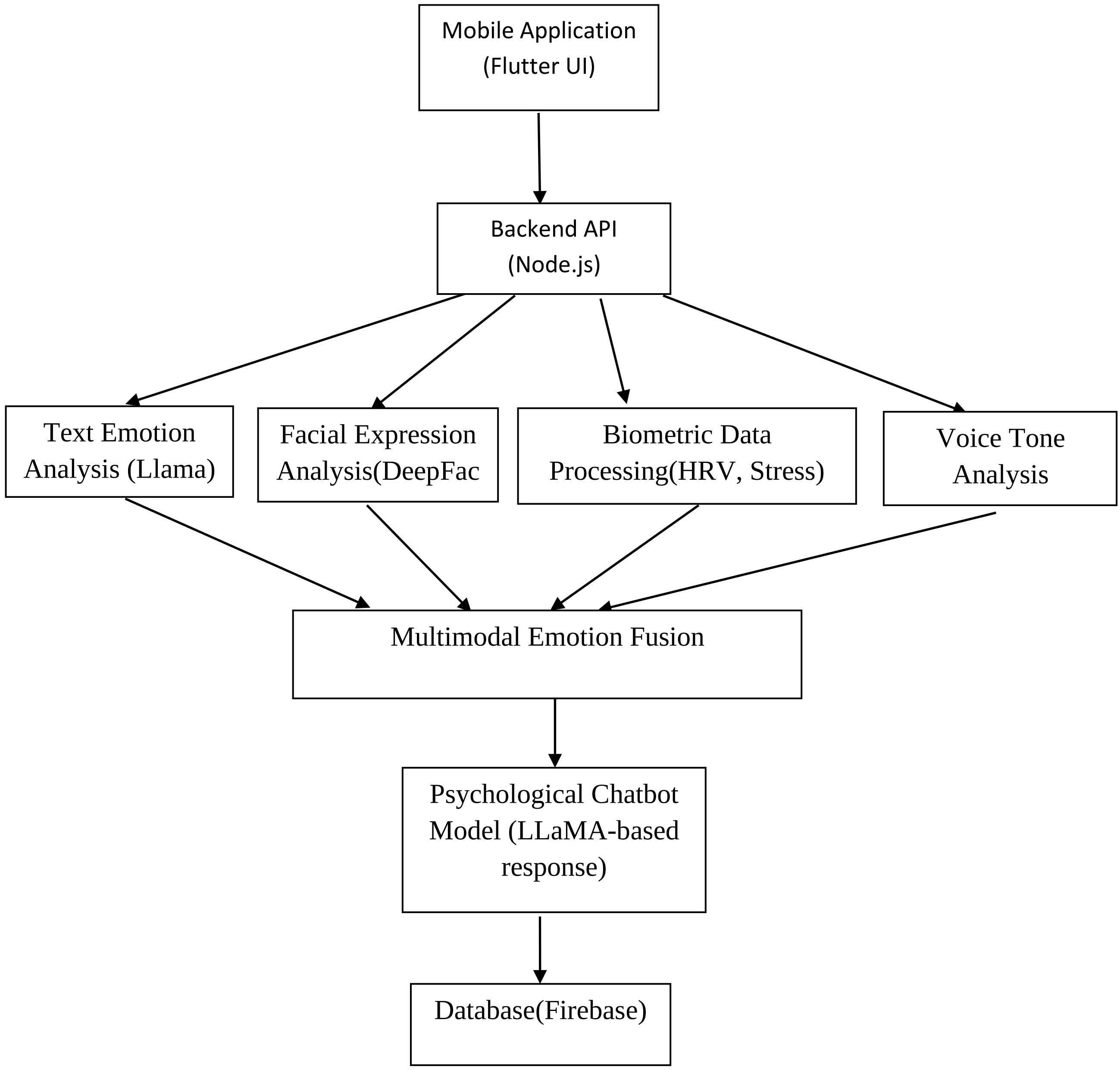
## 3.2 Technology Used

The SenseAI solution integrates multiple technologies to enable multimodal emotion recognition and a psychological chatbot. Below are the key technologies and components used:

| **Component** | **Technology / Tool** | **Description** |
|---|---|---|
| Backend API | Node.js (Express.js) | RESTful API development and system integration |

| | | |
|---|---|---|
| Text Emotion Analysis | Python, PyTorch, HuggingFace Transformers (LLaMA) | Transformer-based NLP model training and inference |
| Facial Expression Analysis | Python, DeepFace (CNN-based model) | Extracting emotions from facial expressions |
| Speech Emotion Recognition | Python, Librosa, PyTorch | Audio signal processing and emotion classification |
| Database | Firebase | Storing user data and emotion records |
| User Interface | Flutter | Mobile application development |
| Model Training & Deployment | PyTorch, Docker (Optional Kubernetes) | Model training, packaging, and scalable deployment |

**Block Diagram of SenseAI Solution**


Mobile Application (Flutter UI)
Backend API (Node.js)
Text Emotion Analysis (Llama)
Facial Expression Analysis(DeepFac
Biometric Data Processing(HRV, Stress)
Voice Tone Analysis
Multimodal Emotion Fusion
Psychological Chatbot Model (LLaMA-based response)
Database(Firebase)

# 4. Software Requirements Specification

## 4.1 Introduction

### 4.1.1 Purpose

The purpose of this Software Requirements Specification (SRS) is to provide a comprehensive overview of the functional and non-functional requirements for the SenseAI project. This document serves as a foundational guide for the design, development, and deployment of the application. It ensures all stakeholders have a shared understanding of the system's objectives and specifications while reducing ambiguity and promoting consistency during the project's lifecycle.

The primary audience for this SRS includes:

- **Development Team:** To implement the features and functionality outlined in the requirements.
- **Project Stakeholders:** To ensure the application aligns with business objectives and user needs.
- **Quality Assurance Team:** To validate that the system meets the specified requirements.
- **End-User Representatives:** To provide feedback and confirm the application addresses their expectations.
- **Future Developers/Maintainers:** To understand the original design and purpose for updates or maintenance.

This document is designed to be accessible and usefulfor both technical and non-technical stakeholders.

### 4.1.2 Scope of Project

The software product to be developed is SenseAI, a mobile psychology chatbot application designed to analyze emotional states and assist users in managing their mental well-being.

**What SenseAI will do:**

- **Camera Integration:** Analyze facial expressions to interpret the user’s emotionalstate.
- **Speech-to-Text (STT):** Convert user speech to text for processing by the AI.
- **Heart Rate Monitoring:** Retrieve BPM data from connected smart watches for additional mood analysis. (Zepp life will utilised to achieve

this)

- **Chatbot Support:** Provide AI-driven conversational support to guide users through mental wellness strategies.
- **Recommendations:** Offer personalized activities or content based on user mood and inputs.

**What SenseAI will not do:**

- SenseAI will not replace professional therapy sessions.

**Application Benefits and Goals:**

SenseAI is targeted at individuals seeking a convenient, AI-powered companion formental health support. It aims to:

- Improve accessibility to basic mental wellness tools.
- Provide emotional feedback and personalized suggestions.
- Leveragenon-invasive biometric data to enhance user experience.
- Deliver a user-friendly interface for seamless interaction.
- Ensure data security and user privacy.

### 4.1.3 Definitions, acronyms, and abbreviations

Table 1 Glossary of SRS

| Term | Description |
|---|---|
| CBT | Cognitive Behavioral Therapy |
| AI | Artificial Intelligence |
| NLP | Natural Language Processing |
| CNN | Convolutional Neural Network |
| DBMS | Database Management System |
| SRS | Software Requirements Specification |
| HA | High Availability |

### 4.1.4 Overview of Document

This SRS contains a comprehensive explanation of the system's functional and non-functional requirements and interfaces. It describes the features and capabilities SenseAI will provide, outlines system design constraints, assumptions, and dependencies to guide development.

This SRS is separated into different sections to increase readability, these sections include:

**Overall description (Section 2)**

A section for general description of the product and product perspective.

**Specific Requirements (Section 3)**

A section for describing both functional and non-functional requirements, design constraints, and software interfaces.

## 4.2 Overall Description

### 4.2.1 Product Perspective

This project aims to develop a multi-modal psychology chatbot that analyzes users' emotional states and contexts and helps people with a psychological guideline. This chatbot will perform emotional analysis using biometric data obtained from voice, facial expressions and wearable devices(e.g. smart watches) and provide personalized feedback. Users can use this application when they need instant support, check their old data and perform emotional state analysis. The application can be authenticated with the authentication API. Users' previous conversations will be deleted after a certain period of time in the database.

#### *4.2.1.1 System Interfaces*

**User Interface:** The interface through which users interact with the chatbot. This interface receives input from the user, such as voice, facial expressions, biometric data, and text input.

**Voice Recognition Interface:** Analyzes the user's voice commands and emotional tone.

**Image Processing Interface:** A system interface that analyzes the user's facial expressions.

**Biometric Data Interface:** An interface that analyzes the biometric data the user receives from wearable devices (e.g. smart watches).

**API Interfaces:** An interface that analyzes both historical and real time data from the user and provides instant feedback and suggestions by enabling the chatbot to integrate with external services (emotion analysis, biometric data collection, etc.). It is required for the client and server to communicate, allowing real-time data exchange and feedback. It ensures that requests sent by the client (user's device) are properly handled and processed by the server, which then returns appropriate responses to enrich the user experience.

**Database Interface:** An interface where user data is stored and queried.

#### *4.2.1.2 User interfaces*

- The interfaces in the system should be designed to be user-friendly, guiding and easy to use.
- The login screen should be large enough to enter the username and password and should be adaptable to different screens. There should be no elements that will make it difficult for the user to click on a field or impair the readability of theinterface.
- The buttons that the user will click on to select the desired data type on the homepage should be large enough and the fonts should be readable.
- The homescreen should provide users with easy access to the new chat, previous chats and the user page.
- The menu structure should be intuitive so that users can easily access the information they need. The main menu should provide users with quick access to the desired section.
- Warning messages should provide clear and direct information to the user.
- The interface should allow users to perform functions quickly and effectively.
- The steps required for any operation should be minimized. Unnecessary complexity should be avoided and a user should be able to easily use this application with an average of 5 minutes of training.
- The interfaces should be designed in a way that the user can easily remember how to use the application.
- The application shall provide support for Dark Mode on both Android and iOS platforms. Dark Mode should be available as an optional interface themefor users who prefer a darker, more visually comfortable interface. This feature will automatically respect the device's system-wide theme settings (if enabled) and allow the user to toggle between Light and Dark modes from within the app settings.
- The application shall ensure that the user interface (UI) remains consistent and responsive (e.g. enabling screen rotation) across different screen sizes, resolutions, and platforms (Android andiOS). The designwill be adaptive, ensuring that the app provides an optimaland consistent user experience regardless of whether the user is on a phone, tablet, or other device.

#### *4.2.1.3 Hardware interfaces*

The system will interact with several hardware components (cameras, microphones, biometric sensors, etc.) to collect data necessary for emotional analysis. The interfaces between the system (application) and these hardware components will be defined as follows:

1. *Camera Interface (for Facial Expression Analysis)*

○ **Functionality:** Captures users' facial expressions for emotion detection.

○ **Connection:** Uses the Android **Camera2 API** to access the camera hardware and process image data. The app will interface with image-processing libraries like OpenCV to analyze facial expressions.

○ **Interaction:** The system will request camera permissions from the user and activate the camera only with the user's consent.

2. *Microphone Interface (for Voice Emotion Detection)*

○ **Functionality:** Captures voice input for analyzing emotional tone and speech commands.

○ **Connection:** Utilizes Android's **AudioRecord** class or **MediaRecorder API** to access microphone data. The app will process the recorded audio using speech-to-text or emotion detection algorithms.

○ **Interaction:** The microphone will be activated only upon user permission, respecting privacy.

*3. Biometric Sensor Interface (for Wearable Device Data)*

○ **Functionality:** Collects biometric data such as heart rate and stress levels from wearable devices (e.g., smartwatches).

○ **Connection:** Communicates with wearable devices via Bluetooth or wireless protocols to retrieve data.

○ **Interaction:** Data will be fetched from the wearable device, with user consent for access to health-related data.

#### *4.2.1.4 Software Interfaces*

**Operating System (OS):** The application is built using the Flutter framework and relies on Android and iOS as the primary operating systems. The minimum required versions are Android 10 (API level 29) for Android devices and iOS 13.0 for iOS devices. These versions ensure that the application can access modern system APIs needed for biometric data, voice recognition, image processing, and other critical features. These OS versions also support improved privacy controls, security enhancements, and compatibility with Flutter plugins necessary for device functionalities.

**User Interface:** This is the interface where users interact with the chatbot. This interface receives various inputs from the user, such as voice, facial expressions, biometric data, and text input. It will be developed to have a user-friendly design, so that users can easily communicate with the chatbot.

**Voice Recognition Interface:** This is an interface that analyzes the user's voice commands and emotional tone. This interface ensures that voice commands are correctly perceived and that user emotions are extracted from the tone of voice. Real-time processing of the user's voice is a critical component for instant responses.

**Image Processing Interface:** This is a system interface that analyzes the user's facial expressions. This interface processes the necessary image data to determine the user's emotional state using face recognition and emotion analysis technologies. Real-time face recognition and emotion detection play an important role in the chatbot providing personalized feedback to the user.

**Biometric Data Interface:** An interface that analyzes biometric data received from wearable devices (e.g. smartwatches). This interface collects and interprets information such as heart rate, stress levels, and other health data. Integration of biometric data allows for more accurate feedback by providing a deeper understanding of the user's emotional state.

**API Interfaces:** An interface that analyzes data from users and provides feedback and recommendations. This interface is used to provide sentiment analysis, biometric data collection, and integration with other external services. API interfaces enhance the functionality of the chatbot, enriching the user experience. It is required for the client and server communication.

**Database Interface:** An interface where user data is stored and queried. This interface securely stores users' past sentiments, interactions, and other important information, and provides quick access when needed. The database interface is critical for managing and analyzing user data.

#### *4.2.1.5 Communication interfaces*

The system will use secure and efficient methods for communication between its components. Firebase will handle real-time data updates and ensure everything is synced quickly across devices. Authentication API will securely handle user sign-ins and Zepp Life API will help capture biometric data like heart rate. The system will also use RESTful APIs to make connecting the app to the backend simple and scalable.

The device should have access to a stable internet connection, Wi-Fi, or mobile data, to use all the functionalities.

#### *4.2.1.6 Memory constraints*

- The SenseAI app should require a minimum of 2 GB of RAM for optimal performance.
- Facial recognition and speech-to-text functionality require real-time processing, which can consume significant memory resources. While modern mobile devices have 4 GB or more of RAM, setting 2 GB as the minimum ensures compatibility with abroad range of devices.
- Users might run the app along side other applications. Allocating enough memory ensures smooth operation without crashes or significant lag.
- The apps hould require approximately 100 MB of available storage space for installation and operation.
- Previous queries will be kept in the database and will not be deleted unless the user is deleted. It will be used to analyze the query before and after.

#### *4.2.1.7 Operations*

**Regular Users:**

1. **Registration and Login:**

- Users can register and log in to the system.
- The system integrates with an external authentication API to secure user accounts.

2. **Data Input and Interaction:**

- Users can choose the type of query to send: text, image, audio, or biometric data.
- Users can initiate new queries at any time.
- Before starting the emotional analysis process, users select the query type.

3. **Profile and Data Management:**

- Users can update their registration information.
- Users can update their mood in their profile section.
- Users can access, search, rename, and delete previous queries or chats.

4. **Error Handling and Feedback:**

- Users will be notified about issues such as exceeding the character limit for text inputs or uploading unsupported file types.

5. **Access to System-Generated Insights:**

- The AI system analyzes input (text, image, audio, biometric data) and provides emotional feedback to users.
- Users can view historical emotional insights and feedback from the system.

6. **Query History:**

- Users can view their last query's access time and title.

7. **Data Privacy:**

- The application ensures compliance with privacy laws, securely storing and processing user data.

**AI System Responsibilities:**

1. Authenticate users via the external authentication API.
2. Analyze user inputs (text, image, audio, biometric data) to generate emotional insights.
3. Provide real-time emotional feedback based on user inputs.
4. Generate historical insights for users to access.
5. Notify users about errors, such as invalid data formats or network issues.
6. Support multilingual functionality in future versions.

**Administration:**

1. **User Management:**

- Admins can manage user accounts, including registration approval, deactivation, or deletion.

2. **Access to All Data:**

- Admins can access all users' historical data, chats, and other sensitive information (compliant with privacy laws).

3. **System Monitoring and Management:**

- Admins can monitor server health, query load, and user activity.

- Admins are responsible for system-level troubleshooting and resolving issues.
- Admins can back up and restore system data.
- Admins can add or remove users, reset user passwords, and modify user credentials.
- Admins can add new admin users or change admin credentials.

4. **Data Privacy and Compliance:**

- Admins ensure secure data management in compliance with regulations.

#### *4.2.1.8 Site Adaptation Requirements*

**Platform Support:** The application will be developed for both iOS and Android platforms, with Android as the primary target.

**User Customization Requirements**

1. Theme Selection: Users can toggle between Light and Dark Mode.
2. Profile Personalization: Users can update their name, picture
3. Personalized Feedback: Tailored emotional feedback based on input type.
4. Language Preference: Users can choose their preferred language.
5. Input Mode Preference: Users select their preferred input (text, voice, image, biometric).
6. Saved Query Names: Users can rename saved queries for easier tracking.

### 4.2.2 Product Functions

- The system will analyze the users' emotional state and provide real-time feedback.
- The system should comply with ethical standards while providing advice on mental health and should be sensitive to the needs of the users.[11][12]
- Users can log in to the system by registering.
- Users can send text, image, audio and biometric query.
- The user should be able to choose what type of data to send from the homepage.
- Users can start new query whenever they want and Access and delete old query from the user page.
- With the user-friendly interface, the user's interface will be easy to learn and remember with the use of the application.

- Wearable devices or existing health applications will be used for biometric data integration.

- The application will perform emotional analysis through multi-modal inputs such as voice, facial expressions, and biometric data collected from wearable devices like smartwatches.
- The system will analyze user responses using quantifiable metrics, such as numerical or statistical analysis, to derive insights about the mental state of the user. These insights will not be shared directly with the user but will inform the system's logic to provide personalized and contextually relevant responses back to the user.

### 4.2.3 User Characteristics

The intended users of SenseAI are primarily individuals seeking therapeutic assistance through a digital platform. Users are expected to have a general understanding of using mobile applications. No specific educational background is required; however, the app will be designed to users with diverse educational levels, from high school students to professionals.

Given the nature of SenseAI as a mental support app using AI, users may have varying levels of comfort with AI interactions. Some users may feel hesitant or cautious about AI’s ability to understand emotional context. Consequently, the app will prioritize transparent communication about data handling, privacy, and the limitations of AI responses to build user trust and enhance user comfort.

### 4.2.4 Constraints

**Hardware Constraints (HW)**

● 5 users supported simultaneously in the first stage: The system will support up to 5 concurrent users in the initial phase, and this number can be scaled as the hardware capacity improves.

● App size under 150 MB: The total size of the application should not exceed 150 MB to ensure it can be efficiently installed and run on various devices.

**Software Constraints (SW)**

● Text input character limit: The application will enforce a character limit on text inputs, with the exact limit determined during development. A warning will be shown for users who exceed the character limit.

● Video size and image resolution limits: The size and resolution of videos and images that users send will have specific limits, to be defined by the development team. Unsupported formats or sizes will trigger a warning message.

● 95% response time under 1 minute: The system is expected to process and return 95% of the responses to users within 1 minute.

● Previous query titles cannot exceed 100 characters: Users will not be able to set query titles exceeding 100 characters when renaming them.

**Operating System Constraints (OS)**

● Data privacy and compliance: User data must be stored and processed in compliance with data privacy laws, including the regulations specified for the target OS (e.g., Android and iOS).

**User Constraints**

● Data storage duration for previous chats: Users' previous AI chats will be stored in the database for a period to be determined by the development team, and users will be informed about this period.

● Unsupported features warning for users: Users who try to upload unsupported data (e.g., large video files or exceeding text input limits) will receive an appropriate warning message.

### 4.2.5 Assumptions and Dependencies

- **User permissions**: Permissions such as camera and microphone access are assumed to be allowed by the user.
- **User Device Specifications:** The characteristics ofthe devices that users use to access the software can affectthe performance of the application. For example, optimization requirements may arise for devices with lower hardware specifications. The mobile devices are assumed to have working camera microphone and touch screen. Also users are assumed to be familiar with basic

smart device interactions.

- **Network Conditions:** Users' internet connection speeds and network conditions can affect the performance of the software. Conditionssuch as longer response timesor data loss for low-bandwidth connections mayrequire reconsideration of the requirements.
- **Audio and Visual Quality:** The audio and video quality delivered to users can significantly impact the user experience. Inadequate audio or videoquality canmake communication difficult to understand and can reduce the effectiveness of the application. Therefore, the requirements should be updated to ensure high-quality audio and video streaming.

## 4.3 Requirements Specification

### 4.3.1 External Interface Requirements

#### *4.3.1.1 User interfaces*

This section will describe the logical characteristics of each interface in the system.
The section will include wireframes to better describe the interface.

##### *4.3.1.1.1 Sign Up*

On the Sign Up page, users will register to the system with an e-mail address and a password they will specify. E-mail and password are mandatory fields on this page. With ”Google” sign users can login with their google account. Users must click the "Sign Up" button to complete their registration to the system. (Figure 2)

##### *4.3.1.1.2 Login/Sign In*

The Login page is for users to access the system through their existing accounts. The required fields for this page are e-mail and password. With ”Google” sign users can login with their google account. After the fields are provided, users can click the “Log In” button to

access the system. With “Remember Me” button users can access their account without sign in process. (Figure 1)

##### *4.3.1.1.3 Chat History*

In this interface, users can access, delete and update their previous queries. Each query is kept with its date and title. Users can search for old queries with the search icon. They can add new queries with the "+" button. (Figure 5)

##### *4.3.1.1.4 Profile*

Users can update their mood in the profile section, see the last query access time and the title of the last query they talked about. They can start a new query with the "Start conversation" button. They can also access the settings page, chat section and profile page via the corresponding buttons. (Figure 6)

##### *4.3.1.1.5 Settings*

On this page, users can go to account settings, share the application URL, manage notifications, access privacy policy, terms C conditions and log out. They can go back to the previous page with the button on the top left. Users can access the chat section and profile page with the buttons below. (Figure 9)

##### *4.3.1.1.6 Admin*

When the admin clicks on the "Review User Record" button, they can search for a user by ID or Username with the search button in the window that opens. Users are listed according to the search. The admin selects the user to see the record. They can see ID, username, e- mail, last access date, query contents and log in/log out activities. They can see the queries of the users with the "query contents" button, the "log in/log activity history screen opens. They select a user for controlling from the list that appears. Admin can make the user an admin, block them or manage access. Admin can go to the settings page with the button on the top right. (Figure 7)

##### *4.3.1.1.7 Home Page*

Users can go to the previous query page with the "Chat history" button, write their own thoughts in the "My Inbox" section, access the settings page with the icon at the top left, and access the chat section and profile page with the buttons at the bottom. (Figure 3)

##### *4.3.1.1.8 Chat Page*

On this page, users send queries to the system and receive answers. Users can select the query type they will send with the paperclip icon at the top, and switch to dark mode with the icon next to it. They can go to the previous page. (Figure 4)

##### *4.3.1.1.9 Forgot Password*

If the user clicks on the "forgot password" button, the system directs them to a page where they will write the e-mail that will reset the password. If they enter a valid e-mail, the user should be directed to the page where they will determine the new password. They should click on the "Submit Password" button to save the new password. (Figure 8)

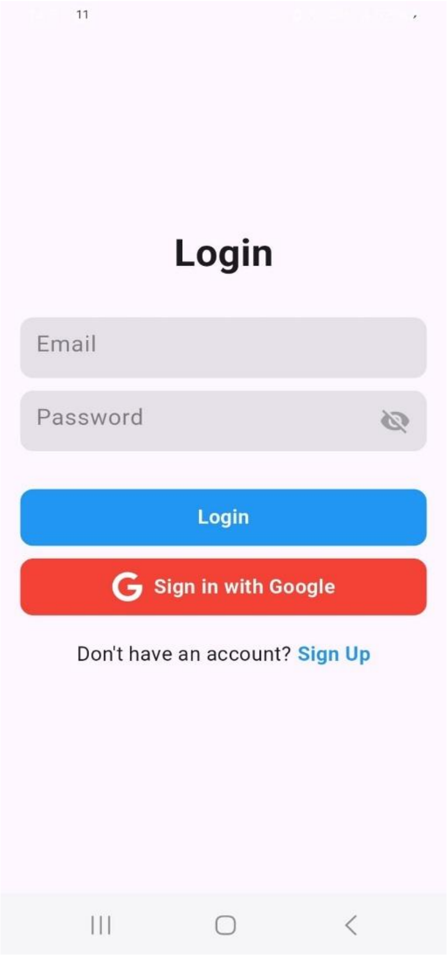


**Figure 1 Login page**

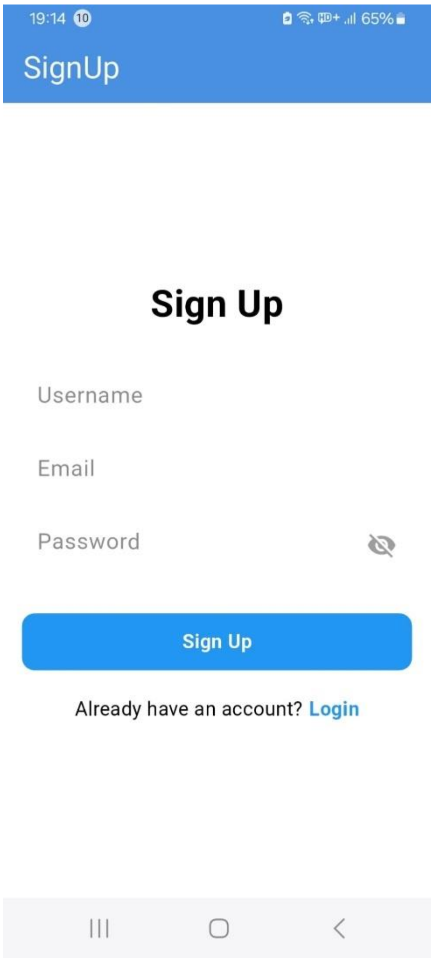


**Figure 2 Sign up page**

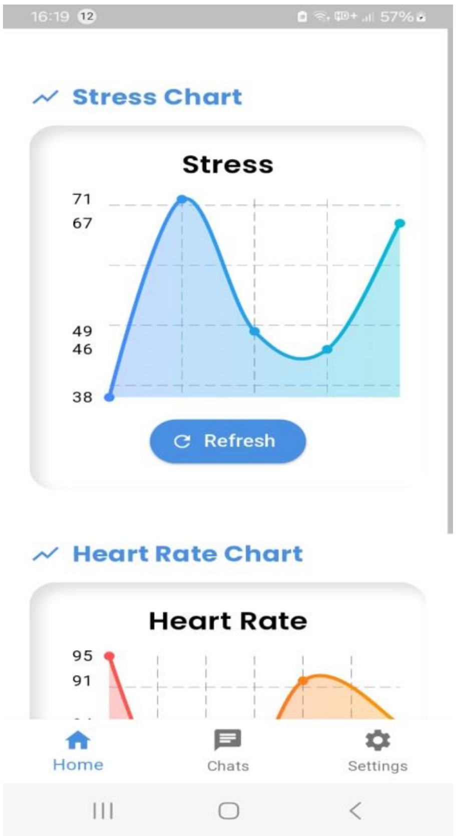
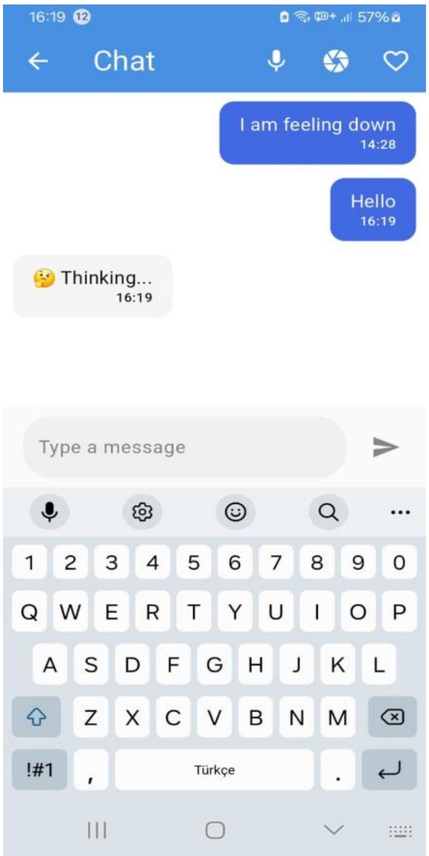


**Figure 3 Main page**

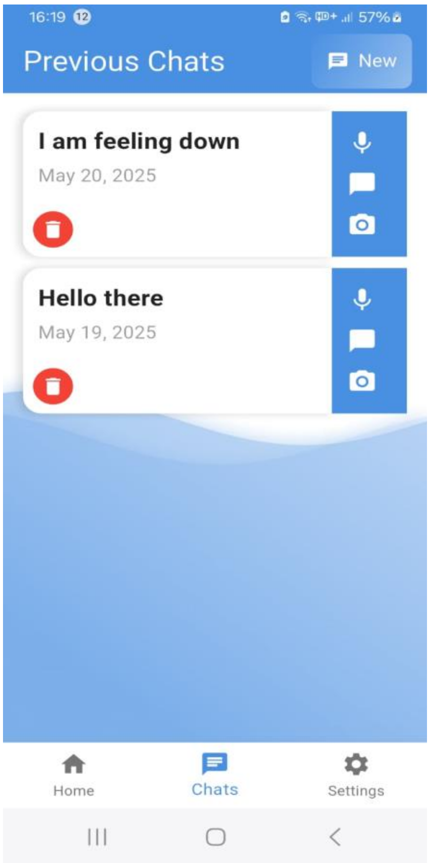


I am feeling down
14:28

Hello
16:19

🤔 Thinking...
16:19

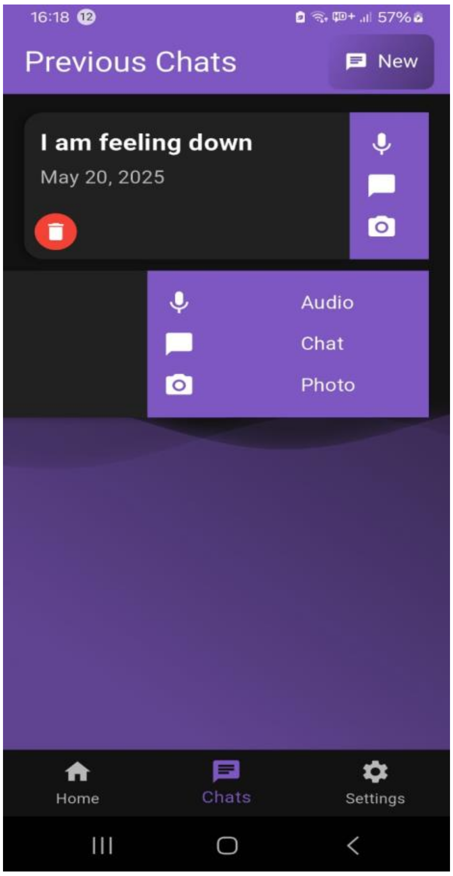
**Figure 4 AI chat page**



I am feeling down
May 20, 2025

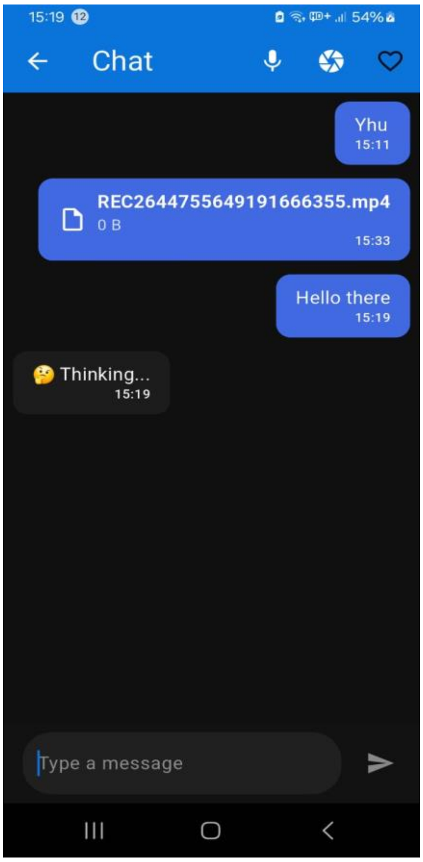
Hello there
May 19, 2025

**Figure 5 Chat history page**



I am feeling down
May 20, 2025

**Figure 6 Chat History Page Dark Mode**



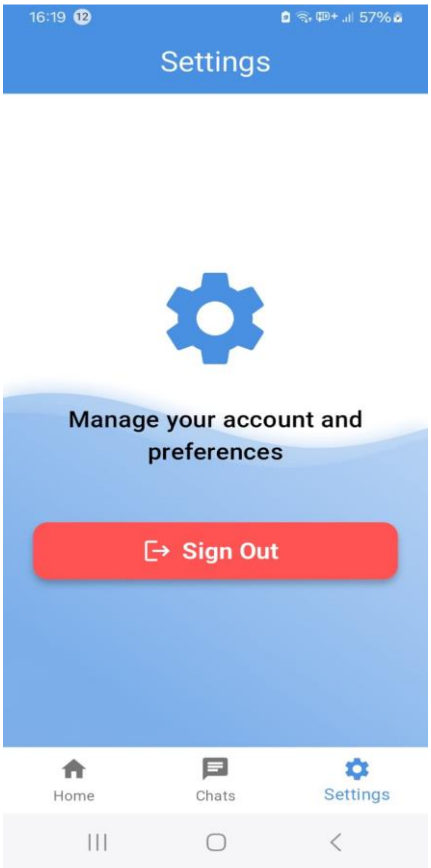
Yhu
15:11

REC2644755649191666355.mp4
0 B
15:33

Hello there
15:19

🤔 Thinking...
15:19

**Figure 7 AI chat page dark mode**



Manage your account and preferences

**Figure 8 Settings page**

#### *4.3.1.2 Hardware interfaces*

##### *4.3.1.2.1 Input Devices*

- Touch screen: SenseAI should support touch input on the mobile device’s touch screen For user interactions such as tapping and swiping.
- Microphone: The devices that users use should support microphone input for inputs sent via voice.
- Camera: Camera should be supported for inputs received from users’ facial expressions.
- Sensor: Smartwatch input should be supported for inputs received from users’ watches.

##### *4.3.1.2.2 Output Devices*

Display: SenseAI should be able to render the user interface on the mobile device's screen, utilizing the available screen resolution and aspect ratio.

##### *4.3.1.2.3 Network Connectivity*

SenseAI will require access to the device's internet connection to communicate with external services.

#### *4.3.1.3 Software Interfaces*

Software Interfaces will have the features mentioned in the 4. 2.1.4 Software Interfaces section.

#### *4.3.1.4 Communications Interfaces*

Communication Interface will have the features mentioned in the 4. 2.1.5 Communication Interfaces section.

### 4.3.2 Functional Requirements

**Users** *User Functional Requirements (User-FR)*

- User-FR-01: The system shall allow users to register and log in to the application.
- User-FR-02: Users shall be able to update and delete the information entered during registration.
- User-FR-03: The system shall enable users to send data via text, images, audio, and biometric data.
- User-FR-04: Users shall be able to choose the type of query (text, image, audio, or biometric data) from the homepage.
- User-FR-05: Users shall be able to start new AI chats and access or delete old chats

from their history page.

- User-FR-06: Users shall be able to search past queries by title and change the query name from history.
- User-FR-07: Users shall be able to choose between dark mode and light mode.
- User-FR-08: Users shall be able to subscribe to premium features or make in-app purchases for advanced functionalities.
- User-FR-09: Users shall be guided by the system to ensure images are taken in sufficient lighting and at appropriate angles for accurate analysis.
- User-FR-10: The system shall warn users when videos/photos exceed the size limit or when texts exceed the character limit.

**Administrators** *Admin Functional Requirements (Admin-FR)*

- Admin-FR-01: Administrators shall be able to monitor the system's compliance with ethical standards.
- Admin-FR-02: Administrators shall manage premium feature access and subscription statuses for users.
- Admin-FR-03: Administrators shall be able to track data quality issues and provide resolution guidelines to users.
- Admin-FR-04: Administrators shall oversee system updates to support additional languages and ensure accurate localization.

**System** ***System Functional Requirements (System-FR)***

- System-FR-01: The system shall analyze the user’s emotional state and provide feedback.
- System-FR-02: The system shall comply with ethical standards when providing mental health advice, ensuring sensitivity to users' needs.
- System-FR-03: Wearable devices or existing health applications shall be integrated for biometric data.
- System-FR-04: The system shall provide a user-friendly interface that is easy to learn, remember, and visually appealing.
- System-FR-05: The system shall clearly convey problems to users when errors occur and allow recovery.
- System-FR-06: The system shall support user authentication via an external authentication API to secure access to personal data and interactions.
- System-FR-07: The system shall initially support the English language, with future multilingual expansion potential.

***Non-Functional Requirements (NFR)***

- NFR-01: The system shall ensure high performance, with minimal response time for user queries.
- NFR-02: The system shall adhere to security standards to protect user data and privacy.
- NFR-03: The interface shall maintain visual consistency and responsiveness across multiple devices.
- NFR-04: The system shall be scalable to handle increased user traffic and data volume in the future.
- NFR-05: The system shall log errors and system performance metrics for analysis and troubleshooting.
- NFR-06: The system shall maintain a 99.9% uptime reliability rate.
- NFR-07: The application shall comply with applicable data protection regulations (e.g., GDPR, HIPAA).

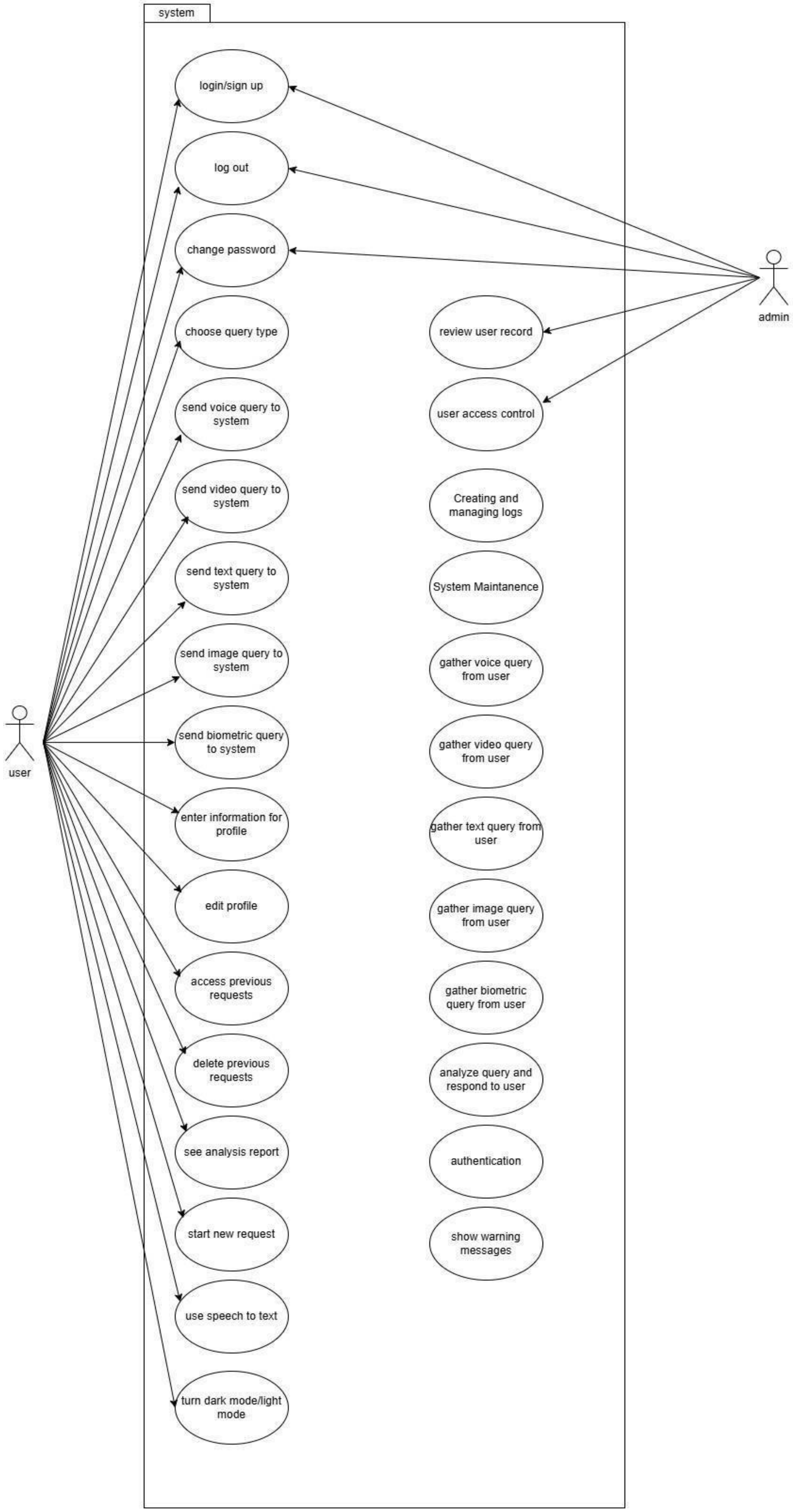

system
login/sign up
log out
change password
choose query type
send voice query to system
send video query to system
send text query to system
send image query to system
send biometric query to system
enter information for profile
edit profile
access previous requests
delete previous requests
see analysis report
start new request
use speech to text
turn dark mode/light mode
review user record
user access control
Creating and managing logs
System Maintanence
gather voice query from user
gather video query from user
gather text query from user
gather image query from user
gather biometric query from user
analyze query and respond to user
authentication
show warning messages
user
admin

**Figure 9 Use Case Diagram**

### *4.3.2.1 SenseAI Users*

#### *4.3.2.1.1 Logging In*

##### 4.3.2.1.1.1 Introduction/Purpose of Feature

The Logging In feature allows authorized users to securely access the SenseAI application and their personalized data. This feature serves as an entry point to ensure data privacy, personalized experience, and user accountability. The system should let users stay logged in once login is successfully completed.

##### 4.3.2.1.1.2 Stimulus/Response Sequence

1. System: Check if user logged in before
2. System: Direct to landing page if user logged in
3. System: Else, show login screen with text fields fore-mail and password
4. User: Fills in account info
5. User: Tap login
6. System: Check account credentials and direct user to landing page if correct
7. System: Else show error

##### 4.3.2.1.1.3 Associated Functional Requirements

REQ-1:Users should be able to login with their google accounts.

REQ-2: Users should be able to sign up by inputting their email and a password.

REQ-3: Users should be able to log out whenever they want

REQ-4: Logged in users should stay logged in unless they logged out (closing the app should not logout the users). The login screen should be skipped for already logged in users.

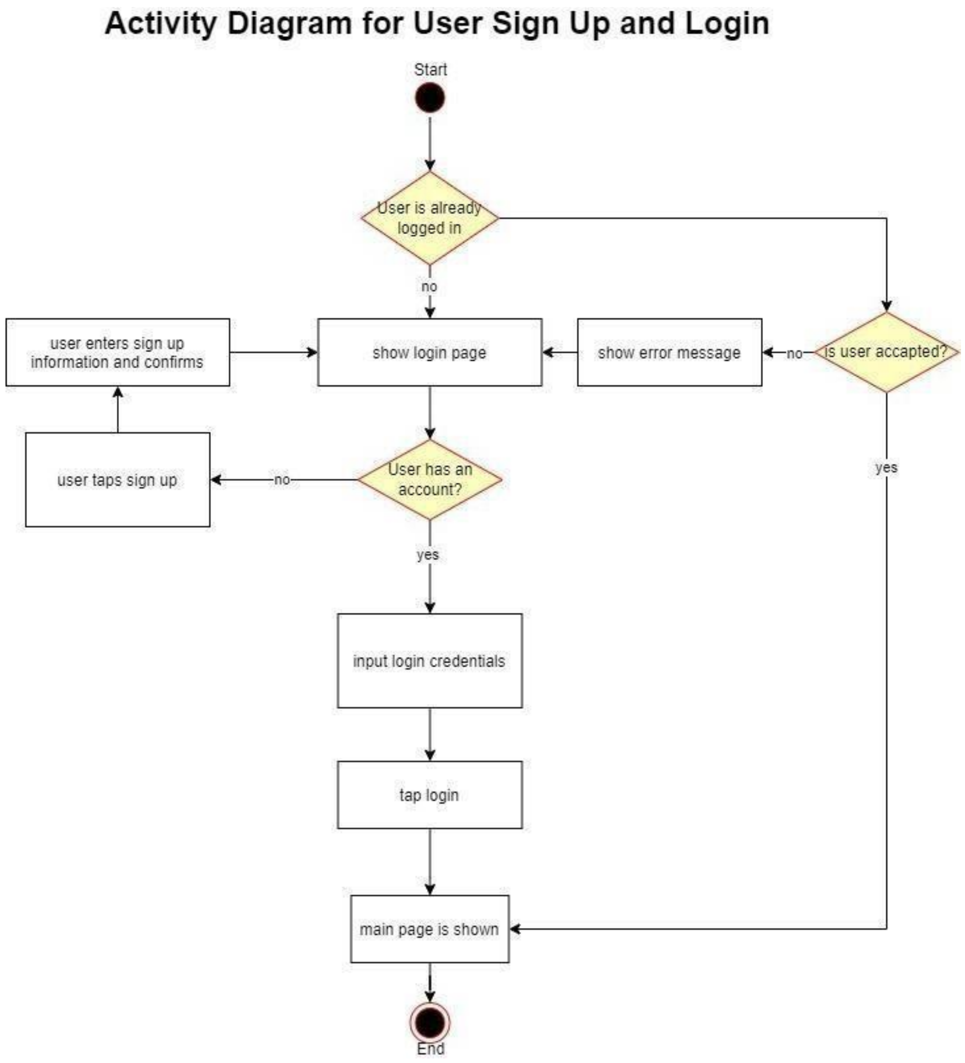


**Figure 10 Login/Sign Up Activity Diagram**

#### *4.3.2.1.2 Forgot Password*

##### 4.3.2.1.2.1 Introduction/Purpose of Feature

The system shall allow users to securely update their passwords to maintain account security and protect their personal information. This feature ensures that users can regularly update their credentials or recover access in case of a potential breach.

##### 4.3.2.1.2.2 Stimulus/Response Sequence

1. System: Display a "Change Password" option in the account settings.
2. User: Taps on the "Change Password" option.
3. System: Prompt the user to input their current password and the new password twice for confirmation.
4. User: Enters the current password and the new password.
5. System: Validates the current password and checks if the new password meets the required

security criteria (e.g., length, charactermix).

6. System: If validation passes, update the password and confirm the change to the user.
7. System: If validation fails, display an error message and prompt the user to try again.

##### 4.3.2.1.2.3 Associated Functional Requirements

REQ-1: The system must verify the user’s current password before allowing the password to be updated.

REQ-2: The new password must meet defined security standards, including minimum length, inclusion of special characters, and exclusion of easily guessable patterns.

REQ-3: The system should send a confirmation email or notification upon successful password change.

REQ-4: The feature must block repeated use of the last X passwords for additional security.

REQ-5: If the password change attempt fails multiple times, the system must temporarily lock the account to prevent unauthorized access.

#### *4.3.2.1.3 Sign Up*

##### 4.3.2.1.3.1 Introduction/Purpose of Feature

The system shall provide new users with a secure and user-friendly way to create an account. This feature ensures the onboarding process is seamless while verifying the user's identity and maintaining system integrity.

##### 4.3.2.1.3.2Stimulus/Response Sequence

1. System: Displaya"Sign Up" form with fields for required information(e.g., email, password, name).
2. User: Fills out the required information and submits the form.
3. System: Validate the input data for accuracy and completeness(e.g., validemail format, strong password).
4. System: Check if the email is already associated with an existing account.

5. System: If validation passes, create a new user account and send a confirmation email or

message to the provided address.

6. System: If validation fails, display an error message and prompt the user to correct the input.

#### 4.3.2.1.4 Associated Functional Requirements

REQ-1: The system must ensure that email addresses are unique and not already in use.

REQ-2: Passwords must meet security criteria, including minimum length, inclusion of special characters, and resistance to dictionary attacks.

REQ-3: The system should require users to verify their email address by sending a confirmation link or code.

REQ-4: The system must encrypt sensitive user data, such as passwords, during storage and transmission.

REQ-5: The system should provide error messages that guide users in correcting any issues with their input (e.g., “Email already in use” or “Password is too weak”).

REQ-6: The feature must comply with data privacy regulations by securely handling user-provided information.

REQ-7: Optional fields for user profile information (e.g., phone number, date of birth) should be clearly marked and not mandatory for account creation.

#### *4.3.2.1.5 Sending Text for Analysis*

##### 4.3.2.1.5.1 Introduction/Purpose of Feature

The system shall allow users to input text for emotion and sentiment analysis, enabling SenseAI to interpret emotional cues within user responses. By processing the input, SenseAI’s analysis module will detect underlying emotions (e.g., happiness, sadness, anger) and overall sentiment (positive, negative, neutral).

##### 4.3.2.1.5.2 Stimulus/Response Sequence

1. System: Show textbox for input
2. User: Enter input.
3. User: Taps "Send" button.
4. System: Validates text input and sends it to the model for analysis with other input types (if exists).

5. System: Displays loading indicator.
6. Model: Processes the input.
7. System: Receives analysis results, saves it to the database and the device.
8. System: Displays results to the user.

#### 4.3.2.1.5.3 Associated Functional Requirements

REQ-1: Chatting with ai should look and feel like the users are chatting with a friends on an app.

e.g. both input and output text should be visible as text bubbles and the screen should be vertically scrollable.

REQ-2: The device's keyboard should not obstruct the previous text while writing new inputs.

#### *4.3.2.1.5.4 Using Voice as Input*

##### 4.3.2.1.5.4.1 Introduction/Purpose of Feature

The system shall allow users to input audio by processing the audio through text-to-speech conversion. It shall also take audio input and analyze it to return a response to the user.

##### 4.3.2.1.5.4.2 Stimulus/Response Sequence

1. System: Show microphone icon for input
2. User: Taps the icon
3. User: Speaks to the microphone.
4. System: Detects speech and uses TTS (text to speech) and converts it to text input.
5. System: Validates input and sends it to the model for analysis with other input types(if exists).
6. System: Displays loading indicator.
7. Model: Processes the input.
8. System: Receives analysis results, saves it to the database and the device.
9. System: Displays results to the user.

##### 4.3.2.1.5.4.3 Associated Functional Requirements

REQ-1: Transcript of the user’s input should be added to the input textbox, ready to be sent.

#### *4.3.2.1.6 Sending Image for Analysis*

#### 4.3.2.1.6.1 Introduction/Purpose of Feature

The system shall allow users to input photos using their smart devices for analysis, enabling SenseAI to process the images and extract actionable insights, such as object recognition, emotional tone assessment, or visual patterns, to enhance the overall analysis process.

#### 4.3.2.1.6.2 Stimulus/Response Sequence

1. System: Show image icon for input
2. User: Taps the icon.
3. System: Enables the front camera for taking a photo
4. User: Takes a photo.
5. System: Validates input and sends it to the model for analysis with other input types (if exists).
6. System: Displays loading indicator.
7. Model: Processes the input.
8. System: Receives analysis results, saves it to the database and the device.
9. System: Displays results to the user.

#### 4.3.2.1.7 Associated Functional Requirements

REQ-1: Users should be able to take photos without exiting the app.

### *4.3.2.1.8 Sending Heartbeat Info for Analysis*

#### 4.3.2.1.8.1 Introduction/Purpose of Feature

The system shall enable SenseAI to receive and analyze users' heartbeat information to assess emotional or physical states. This analysis will take into account the user’s age, as age-related differences can impact heart rate norms and interpretations. By factoring in age, SenseAI can more accurately identify deviations from typical heartbeat patterns, helping to gauge stress, anxiety, or relaxation levels.

#### 4.3.2.1.8.2 Stimulus/Response Sequence

1. System: Display option for biometric data input.
2. User: Either check or uncheck the option
3. System: Check if the user has checked or unchecked the option for biometric data input. If checked, proceed; if unchecked, skip the biometric data retrieval step.

4. System: Check if Zepp Life is successfully connected.
5. System: Check if the required biometric data (BPM) is available in Zepp Life. If data is unavailable or incomplete, prompt the user to provide manual input or skip this step.
6. System: Access today's biometric information(BPM) of the user and send it to the model along with other input type (if exists)
7. System: Displays loading indicator.
8. Model: Processes the input.
9. System: Receives analysis results, saves it to the database and the device.
10. System: Displays results to the user.

#### 4.3.2.1.8.3 Associated Functional Requirements

REQ-1: The system should be able to access the latest heartbeat info from the user's Zepp Life account.

REQ-2: The users should be only able to use this input option if they have a Zepp Life account linked. If no account is linked the option should be grayed out. And tapping it should show a text bubble telling users they need to link their Zepp Life accounts.

REQ-3: If an error occurs while retrieving info or if no data exists, the app should inform the user about the error.

### *4.3.2.1.9 Viewing Previous Analysis (History)*

#### 4.3.2.1.9.1 Introduction/Purpose of Feature

The system shall provide users with access to a history feature, allowing them to view previous analyses and insights generated by SenseAI. This history will include past sentiment, emotion analyses from text, heartbeat, and image inputs, enabling users to track trends and changes overtime. By offering a historical view, SenseAI aims to support users in understanding their emotional patterns and progress

#### 4.3.2.1.9.2 Stimulus/Response sequence

1. System**:** Display option to view past analysis results
2. User: Selects the option to view previous analysis.
3. System: Checks if the user has any stored analysis results in the database or device.
4. If results are found, proceed.
5. If no results are found, display a message such as "No previous analysis available."

6. System: Retrieves and displays a list of previous analysis results.
7. User: Selects a specific past analysis entry to view detailed results.
8. System: Displays the selected analysis details, including insights, graphs, and any other relevant data.
9. User: Optionally, chooses to delete or update past analysis results (if allowed by the system).
10. System: If the user deletes or updates, confirms action and updates the database accordingly.
11. System: Displays the updated history of deletion.

###### 4.3.2.1.9.3 Associated Functional Requirements

REQ-1: History of previous interactions should be displayed with dates.

REQ-2: Users should be able to view the history chronologically ordered.

REQ-3: Users should be able to delete previous talks.

REQ-4: The system should provide a search or filter option, allowing users to find specific past interactions based on keywords or date ranges.

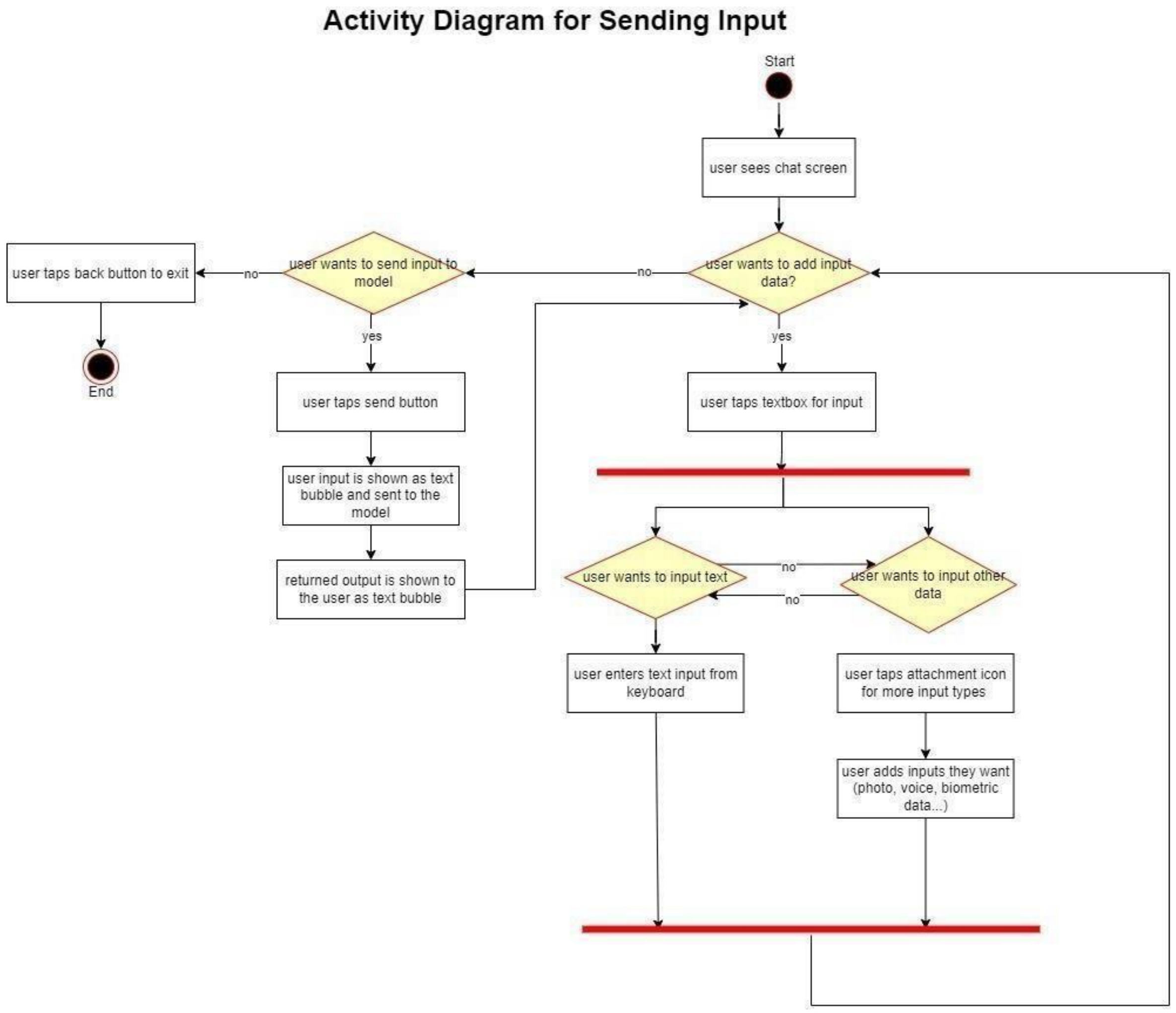

Activity Diagram for Sending Input
Start
user sees chat screen
user wants to add input data?
no
user wants to send input to model
no
user taps back button to exit
End
yes
user taps send button
user input is shown as text bubble and sent to the model
returned output is shown to the user as text bubble
yes
user taps textbox for input
user wants to input text
no
no
user wants to input other data
user enters text input from keyboard
user taps attachment icon for more input types
user adds inputs they want (photo, voice, biometric data...)

**Figure 11 Sending Input Activity Diagram**

### *4.3.2.2 Admin*

#### *4.3.2.2.1 Logging In*

The Logging In feature allows admins to securely access the SenseAI application and their personalized data. This feature serves as an entry point to ensure data privacy, personalized experience, and user accountability. The system should let users stay logged in once login is successfully completed.

#### **4.3.2.2.2 Stimulus/Response Sequence**

1. System: Check if user logged in before
2. System: Direct to landing page if user logged in
3. System: Else, show login screen with text fields for email and password
4. User: Fills in account info

5. User: Tap login
6. System: Check account credentials and direct user to landing page if the user is admin
7. System: Else show error

#### 4.3.2.2.3 Associated Functional Requirements

REQ-1: The system must securely authenticate the admin's credentials using industry-standard encryption methods.

REQ-2: The system should store a secure session token to keep the admin logged in until the session expires or the admin logs out manually.

REQ-3: If the user has previously logged in and their session is active, the system should skip the login screen and redirect them directly to the landing page.

REQ-4: Invalid login attempts should trigger appropriate error messages (e.g., "Invalid email or password") without exposing sensitive information.

REQ-5: The system must enforce a retry limit to prevent brute-force login attempts and temporarily lock the account after exceeding the limit.

REQ-6: The login feature must validate that the user has an admin role before granting access to the admin dashboard or personalized data.

REQ-7: A "Forgot Password" option should be available to help admins recover access in case of credential loss.

#### 4.3.2.2.4 Forgot Password

##### 4.3.2.2.4.1 Introduction/Purpose of Feature

The system shall allow users to securely update their passwords to maintain account security and protect their personal information. This feature ensures that users can regularly update their credentials or recover access in case of a potential breach.

##### 4.3.2.2.4.2Stimulus/Response Sequence

1. System: Display a "Change Password" option in the account settings.
2. User: Taps on the "Change Password" option.
3. System: Prompt the user to input their current password and the new password twice for confirmation.
4. User: Enters the current password and the new password.

5. System: Validates the current password and checks if the new password meets the required security criteria (e.g., length, charactermix).
6. System: If validation passes, update the password and confirm the change to theuser.
7. System: If validation fails, display an error message and prompt the user to try again.

#### 4.3.2.2.4.3Associated Functional Requirements

REQ-1: The system must verify the user’s current password before allowing the password to be updated.

REQ-2: The new password must meet defined security standards, including minimum length, inclusion of special characters, and exclusion of easily guessable patterns.

REQ-3: The system should send a confirmation email or notification upon successful password change.

REQ-4: The feature must block repeated use of the last X passwords for additional security. REQ-5: If the password change attempt fails multiple times, the system must temporarily lock the account to prevent unauthorized access.

### 4.3.2.2.5 Sign Up

#### 4.3.2.2.5.1 Introduction/Purpose of Feature

The system shall provide new users with a secure and user-friendly way to create an account. This feature ensures the onboarding process is seamless while verifying the user's identity and maintaining system integrity.

#### 4.3.2.2.5.2Stimulus/Response Sequence

1. System: Display a"Sign Up" form with fields for required information(e.g., email, password, name).
2. User: Fills out there quired information and submits the form.
3. System: Validate the input data for accuracy and completeness(e.g., valide mail format, strong password).
4. System: Check if the email is already associated with an existing account.
5. System: If validation passes, create a new user account and send a confirmation email or message to the provided address.
6. System: If validation fails, display an error message and prompt the user to correct the input.

#### 4.3.2.2.5.3Associated Functional Requirements

REQ-1: The system must ensure that email addresses are unique and not already in use.

REQ-2: Passwords must meet security criteria, including minimum length, inclusion of special characters, and resistance to dictionary attacks.

REQ-3: The system should require users to verify their email address by sending a confirmation link or code.

REQ-4: The system must encrypt sensitive user data, such as passwords, during storage and transmission.

REQ-5: The system should provide error messages that guide users in correcting any issues with their input (e.g., “Email already in use” or “Password is too weak”).

REQ-6: The feature must comply with data privacy regulations by securely handling user-provided information.

REQ-7: Optional fields for user profile information (e.g., phone number, date of birth) should be clearly marked and not mandatory for account creation.

### 4.3.2.2.6 Review User Record

#### 4.3.2.2.6.1 Introduction/Purpose of Feature

This feature enables admins to access and manage detailed records of individual users, including their activities, queries, and biometric or manual inputs. It supports efficient monitoring and accountability while ensuring user data privacy and system transparency.

#### 4.3.2.2.6.2 Stimulus/Response Sequence

1. System: Display a list of user records in the admin panel, with options to search and filter users.
2. Admin: Search for or filter the user list to locate a specific user ID,user name and select the user record.

3. System: Retrieve and display the selected user's record with the following sections:
    - Personal Information:
        - Username, email address, user ID, account status (active/inactive).
    - System Interaction Logs:
        - User’s login/logout actions, connection status.
        - The last activedate and time on the application.
    - Query History:
        - Achronological list of all queries submitted by the user:
            - Query content.
            - Submission date and time.
            - Results or error messages returned for the query.
            - Query status (completed, failed, pending).
    - Biometric Data:
        - Latest biometric data (e.g., BPM) if the user is connected to Zepp Life.
        - Connection status: "Connected" or "Not Connected."
    - Linked Services:
        - Status of connections with external services like Zepp Life.
4. System: Display a confirmation dialog foreach selected action.
5. Admin: Confirm or cancel the action.
6. System: Execute the action, display a record.

#### 4.3.2.2.6.3 Associated Functional Requirements

REQ-1: The system must provide an interface for searching, filtering, and viewing user records.

REQ-2: The user's query history must be displayed in detail, including query content, results, timestamps, and statuses.

REQ-3: Biometric data should only be displayed if the user is connected to an external service like Zepp Life.

REQ-4: All admin actions must be logged with action type, timestamp, and admin identity for auditing purposes.

REQ-5: User details can only be edited or deleted by authorized admins.

REQ-6: The query history must support search and filter functionality for efficient review. REQ-7: The system must maintain separate logs for user and admin activities.

#### 4.3.2.2.7 User Access Control

##### 4.3.2.2.7.1 Introduction/Purpose of Feature

This feature allows admins to manage and regulate user access within the system. It enables the assignment of roles, permission settings, and account status management, ensuring that users can only access the resources and actions appropriate to their role. This is critical for maintaining security, privacy, and operational efficiency within the application.

##### 4.3.2.2.7.2 Stimulus/Response Sequence

1. System: Search users with options to search, filter, and sort by roles, permissions, or activity status(active/inactive).
2. Admin: Select a specific user from the list to review or modify their access permissions. System: Retrieve and display the selected user’s access details.
3. Admin: Closes an account, changes access permissions, or changes the role of another account.
4. System: Display a confirmation dialog summarizing the changes.
5. Admin: Confirm or cancelthe action.
6. System: Execute the changes, update the user’s record in the database, and display a success or error message.

##### 4.3.2.2.7.3 Associated Functional Requirements

REQ-1 The system must provide a unified dashboard for searching, filtering, and managing user access.

REQ-2 Changes to roles, permissions, or account statuses must show their impact clearly before confirmation.

REQ-3 The system must maintain an audit log of all actions performed, including timestamps, changes, and admin details.

REQ-4 Account status changes should have safeguards (e.g., warnings) to prevent unintentional lockouts.

REQ-5 Permissions and roles should be manageable independently to ensure granular control over access.

### 4.3.3 Performance Requirements

- The application shall be available 24/7 to ensure accessibility at any time.
- 95% of system responses to user requests shall be completed within 1 minute to ensure timely feedback.
- The system shall initially support up to 5 simultaneous users, with scalability to support more users as the application matures and a more advanced hardware is provided.
- The system shall handle multimodal datatypes, including text, images, audio, and biometric data.
- Users who engage with audio and video functionalities shall use compatible devices to ensure smooth data transmission and processing.

### 4.3.4 Design constraints

The server's processor power, RAM and storage capacity should not negatively affect the performance of the Chatbot as the number of users increases. The device should support microphone, camera, smart watch, touchscreen features.

#### *4.3.4.1 Standards compliance*

##### 4.3.4.1.1 ***Material Design Guidelines***

Sense AI’s user interface will be designed in accordance with Google’s Material Design Principles[13] to provide a consistent, modern, and user-friendly experience. Material Design provides a comprehensive guide to visual design, motion, and interaction principles that enhance usability and add aesthetic appeal. This design approach makes it easy for users to interact naturally with the chatbot, while providing a professional and reliable interface standard.

### 4.3.5 Software system attributes

#### *4.3.5.1 Reliability*

A chatbot system that provides psychological support is critical in terms of reliability. The system must provide a sustainable service without losing users' past conversations and analyses. Therefore, regular data backup mechanisms will be implemented. In addition, continuous tests will be conducted to minimize system errors and provide uninterrupted service.

#### *4.3.5.2 Availability*

The system should be available 24/7 and include recovery and restart mechanisms for disaster recovery situations. High availability (HA) infrastructure will be used to minimize user waiting time.

#### *4.3.5.3 Security*

Users' personal data and chat content must be protected against both accidental and malicious access. Security measures will include:

- Use of reliable cryptographic algorithms (e.g. AES-256) to encrypt data,
- Two-factor authentication for user logins,
- Role-based access control to ensure only authorized individuals can access sensitive data,
- Regular verification processes to ensure data integrity.

#### *4.3.5.4 Maintainability*

The chatbot will be developed in a modular structure. Each function will be in a separate module and these modules can be updated or changed independently when necessary. Thanks to the object-oriented design, it will be easy to expand the chatbot by adding new features or improving existing features. The system should allow automatic updates to be installed remotely via the Google Play Store and Apple App Store without requiring user intervention. Updates should not disrupt the user experience and should ensure the integrity and security of the data. The system will apply Over-the-Air (OTA) updates to keep the system up to date via the Google Play Store and Apple App Store.

#### *4.3.5.5 Portability*

The mobile application that offers psychological support will be developed in a way that it can work seamlessly on different mobile devices and operating systems (iOS, Android). In this direction:

- Platform-dependent codes will be minimized,
- Tools that provide cross-platform compatibility will be used in the development process,
- It will be optimized to meet the requirements of mobile application stores,
- Its compatibility with various screen sizes and resolutions will be tested.
- This approach will ensure that the application offer saseamless user experience on different devices.

# 5. Software Design Description

## 5.1 Introduction

### 5.1.1 Purpose

The purpose of this Software Design Document (SDD) is to provide a comprehensive and detailed description of the architecture, design, and implementation approach for the SenseAI project. The SDD serves as a blueprint to ensure that all stakeholders have a clear understanding of the system’s structure, its components, and their interactions.

This document is intended for:

**Development Team:** To implement the features and functionality outlined in the requirements.

**Project Stakeholders:** To review and validate the proposed design.

**Quality Assurance Team:** To understand the system design for testing purposes.

**End-User Representatives:** To provide feedback and confirm the application addresses their expectations.

**Project Managers:** To ensure the design aligns with project objectives.

**Future Developers/Maintainers**: To understand the original design and purpose for updates or maintenance.

This document is designed to be accessible and useful for both technical and non-technical stakeholders.

### 5.1.2 Scope

This SDD outlines the design of the SenseAI system, which aims to facilitate mood tracking, data analysis, and session-based communication.

The document will cover:

- The overall system architecture.
- Detailed descriptions of system components and their interactions.
- Data models and storage mechanisms.
- Interfaces and dependencies.
- Non-functional requirements, such as performance and scalability.

The SDD will not include:

- Detailed implementation code.
- End-user documentation or training materials.

The intended audience includes developers, testers, project managers, and stakeholders who are involved in the project lifecycle.

### 5.1.3 Glossary

Table 2 Glossary of SDD

| Term | Description |
|---|---|
| AI | Artificial Intelligence Model |
| IOS | iPhone Operating System |
| IP | Internet Protocol Address |
| OS | Operating System |
| RAM | Random Access Memory |
| SSD | Solid State Disk |

### 5.1.4 Overview of document

This document details the design and development process of a software or system. Below are summaries of the topics covered in each section:

**Architecture (Section 2):** This section outlines the overall architecture of the system. It describes the architectural components, structures on the client and server sides, and how these components interact with each other. The main components of the system and their interconnections are detailed.

**User Interface (Section 3):** The user interface design is explained in this section. The design of different screens and the functionalities they provide are discussed. Interface components and the interaction methods used to improve user experience are described.

**High Level Design (Section 4):** This section provides a high-level design of the system. The structures and functions of the main modules are discussed in broad terms. The basic roles of each module, their functions, and the relationships between the modules are summarized.

**Low Level Design (Section 5):** This section delves into the detailed design of each module. It explains how each module operates, including the algorithms used and the data flows. Technical details of each module are provided.

**Database Design (Section 6):** This section explains the database design. It provides information on the database structure, relationships, tables, and how the database will be organized. Entity- relationship diagrams, tables, and their relationships are detailed.

## 5.2 Deployment diagram

***Clients***

**Mobile Phones / Tablets**

Minimum Requirements:

- **OS: Android 10.0 or iOS 13.0 and above**
- **RAM: 4 GB minimum (6 GB or more recommended for better performance)**
- **Storage: 200 MB of available space (additional space may be required for user data and media)**
- **Network: Active internet connection (Wi-Fi or 4G/5G recommended for reliable performance)**

***Application Server***

Minimum Requirements:

- **OS: Ubuntu 20.04 LTS or Windows 10/11 (Desktop versions)**
- **CPU: Dual-core processor (Intel i3/Ryzen 3 or higher)**
- **RAM: 8 GB minimum (16 GB recommended for smooth operation)**
- **Storage: 50 GB available space (SSD for better performance)**
- **Network: Standard internet connection (at least 20-50 Mbps)**

***Database Server***

Minimum Requirements:

- **OS: Ubuntu 20.04 LTS or Windows 10/11**
- **CPU: Dual-core processor (Intel i3/Ryzen 3 or higher)**

- **RAM: 8 GB minimum (16 GB recommended)**
- **Storage: 256 GB SSD (for fast read/write operations)**
- **Network: Standard internet connection (at least 20-50 Mbps).**

## 5.3 Architecture Design

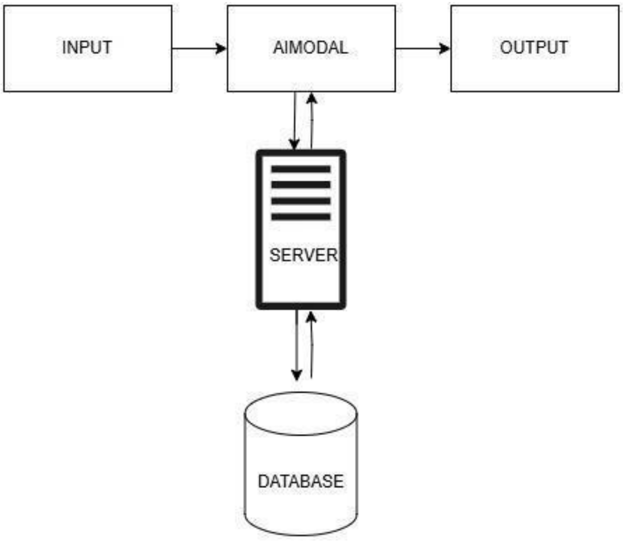
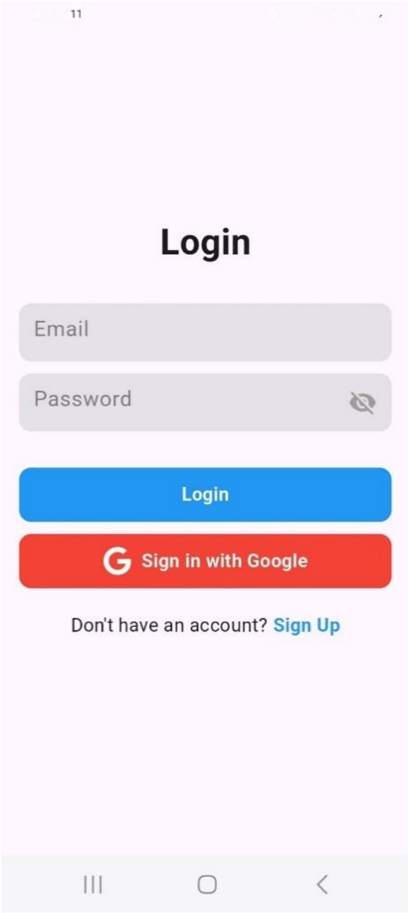


**Figure 12 Hardware**

### 5.3.1 User Interface

#### *5.3.1.1 Login Page*



Login

Email

Password

Login

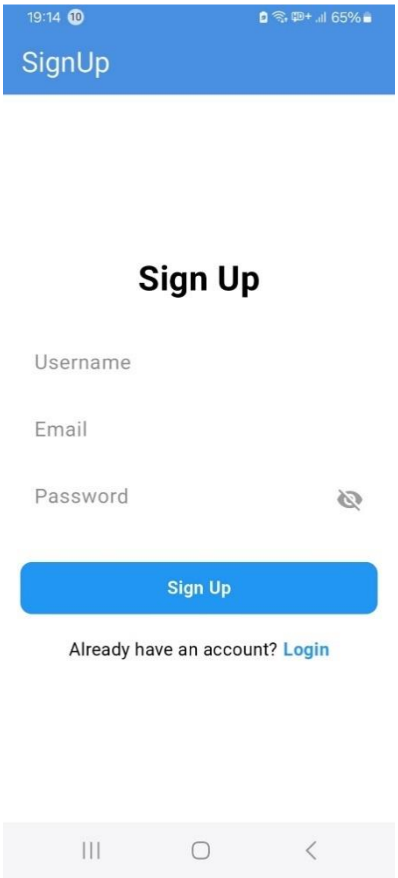
Sign in with Google

Don't have an account? Sign Up

**Figure 13 Login Page**

| Title | SenseAI |
|---|---|
| E-mail Field | Field where the user will enter his/her e-mail address. Accepts data only in e-mail format. |
| Password Field | The field where the user will enter the password. The characters are displayed hiddenly. |
| Remember Me Checkbox | Provides the option for the user to remember their login information. |
| Sign In Button | The main action button that allows the user to log in. |
| Google Login Icon | Icon that allows you to log in with a Google account. |
| Sign Up Link | Referral link for user to create account |
| Forgot Password Link | Allows the user to go to the reset screen when they forget their password. |

### *5.3.1.2* Sign up Page



19:14 10 65%

SignUp

Sign Up

Username

Email

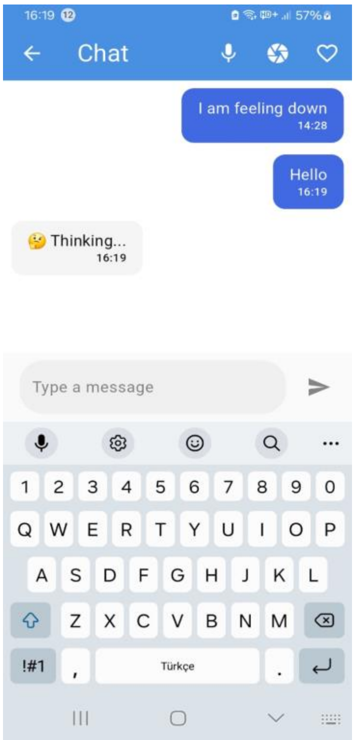
Password

Already have an account? Login

**Figure 14 Sign Up**

| Title | SenseAI |
|---|---|
| Username Field | Textbox for entering the user's username. |
| E-mail Field | Field where the user will enter his/her e-mail address. Accepts data only in e-mail format. |
| Password Field | The field where the user will enter the password. The characters are displayed hiddenly. |
| Checkbox | Checkbox labeled *"I have read and agree to the term of service"*. Users must check this box before signing up. |
| Sign Up Button | Main button for account creation, highlighted in purple. |
| Google Sign-Up Button | Secondary button with Google icon for signing up via Google authentication. |
| Sign In Link | Text link for existing users to navigate to the login page. Label: *"You already have an account? Sign in"* |

### *5.3.1.3 AI Chat Page*



I am feeling down
14:28

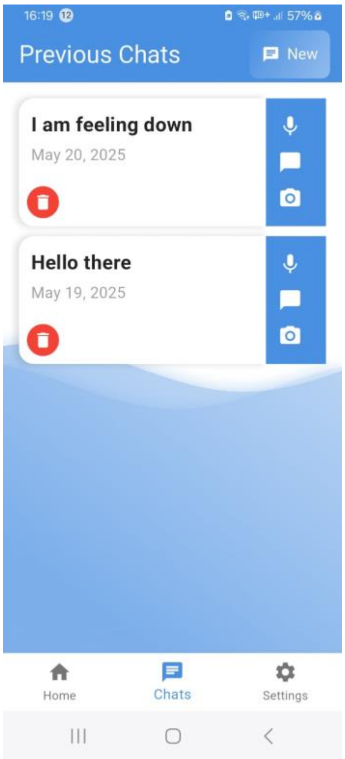
Hello
16:19

🤔 Thinking...
16:19

**Figure 15 AI Chat**

| | |
|---|---|
| Date and time | Shows last conversation time. |
| Paperclip Button | Shows the options for the type of data to be sent. |
| Moon Button | Button that changes dark mode/light mode. |
| Heart Rhythm Button | Chatbot allows biometric data collection. |
| Camera Button | Chatbot allows to get photo data. |
| Video Button | Chatbot allows to get video data. |
| Microphone Button | Receives voice data from the user for the text-to-speech feature. |
| Purple Speech Bubble | Shows the query sent by the user to the chatbot. |
| Gray Speech Bubble | Shows the answers the chatbot returns to the user. |
| Keyboard | Used to receive text data from the user. |
| Back Button | Redirects to the previous page. |

### *5.3.1.4 Chat History Page*



I am feeling down
May 20, 2025

Hello there
May 19, 2025

**Figure 16 Chat History**

| | |
|---|---|
| Title | SenseAI |
| Sub-Title | Recent chats |
| Title (in box) | Shows the title of the chat. |
| Date | Shows the chat’s date. |
| Pen Button | Edit the chat. |
| Bin Button | Delete the chat. |
| Search Button | Search the chat from the chats. |
| Sub-Title | Previous 7 days |
| Plus Button | Redirects to the chat page to add a new chat. |
| Back Button | Redirects to the previous page. |

### 5.3.1.5 *Main Page*

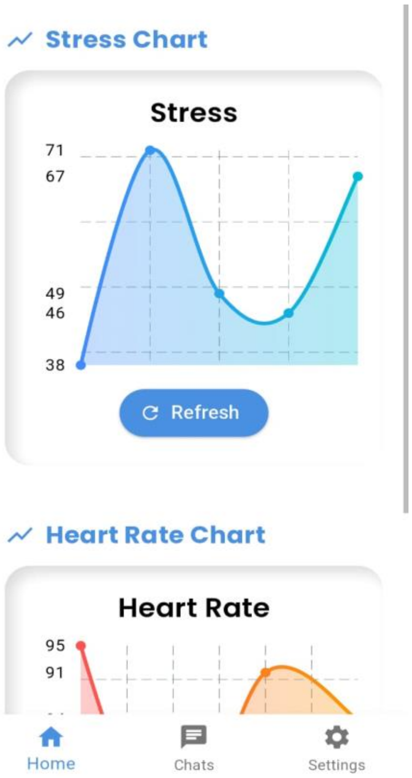
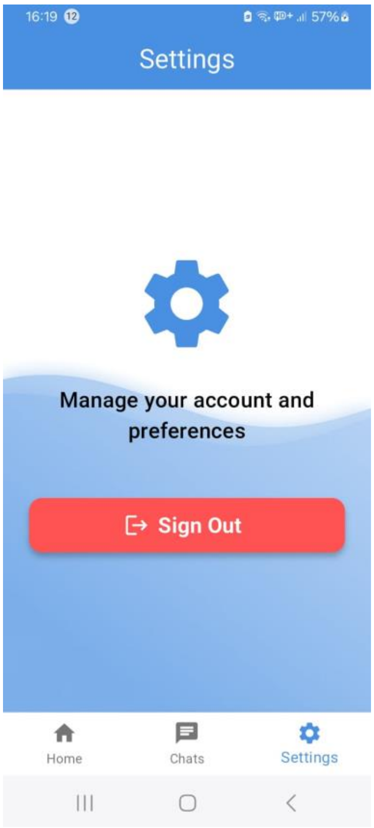


**Figure 17 Main Page**

| | |
|---|---|
| Title | SenseAI |
| Chat History Section | Button for accessing previous chat sessions. Label: *"Chats"*. |
| Stress Chart | Graph showing stress level taken from user |
| Heart Rate Chart | Graph showing heart rate taken from user via smart watches. |
| Home Button | Home icon for navigating to the main page. |
| Setting Button | Button for navigating to the settings page. |

### *5.3.1.6 Settings Page*



Manage your account and preferences

**Figure 18 Settings Page**

| | |
|---|---|
| Title | Settings |
| Chats Button | Button for accessing previous chat sessions. |
| Home Button | Home icon for navigating to the main page. |
| Setting Button | Button for navigating to the settings page. |
| Sign out Button | With this button users can log out the system. |

## 5.4 High Level Design

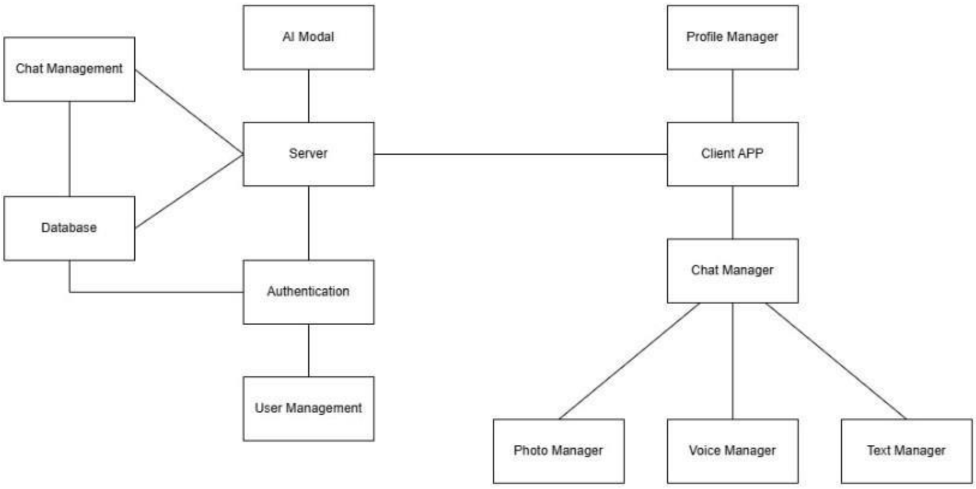


*Figure 19 Module Diagram*

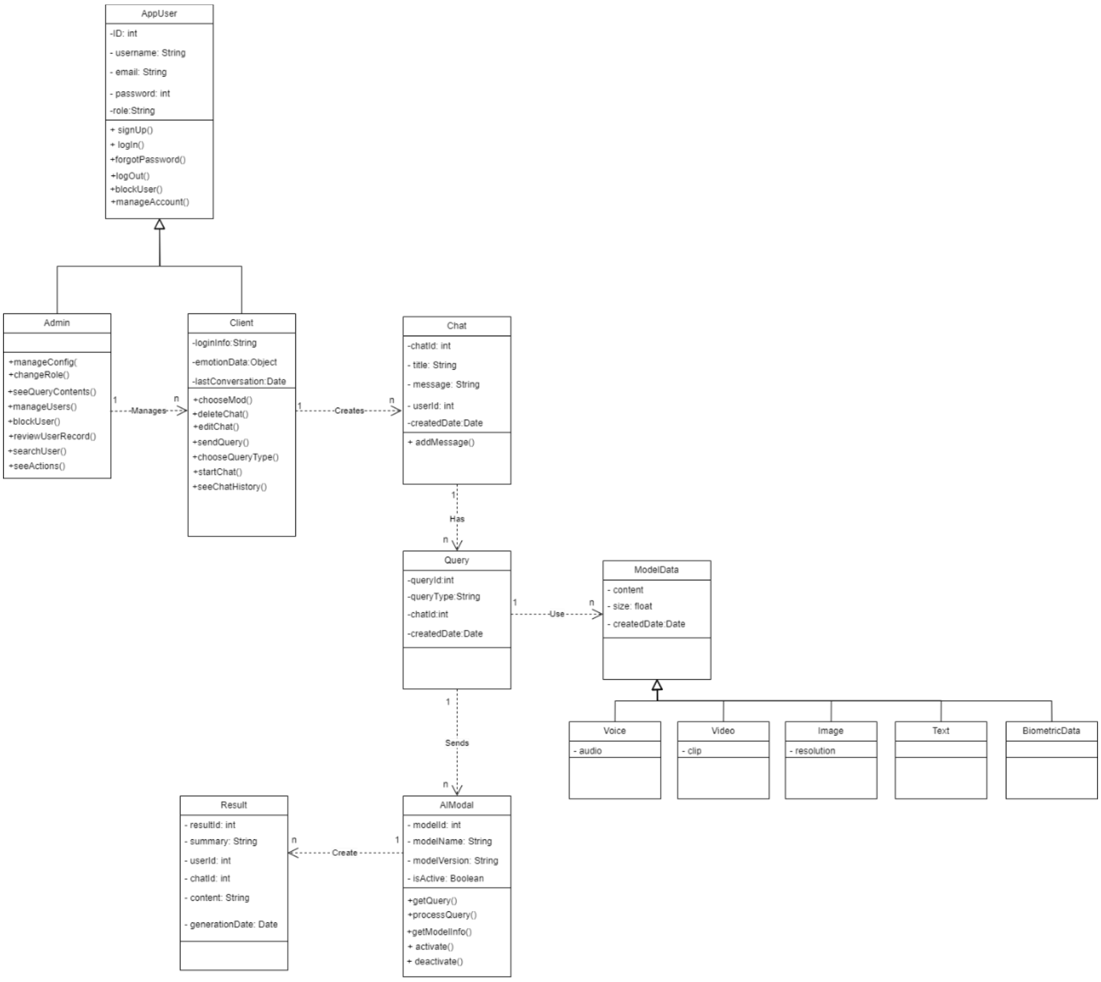


*Figure 20 Class Diagram*

### Server Module

The Server module provides the core backend infrastructure that runs the entire system. It handles HTTP requests and manages data communication between the client and server. Additionally, it provides real-time messaging services using technologies like socket.io. The server coordinates the overall operation of the application and works in integration with other modules to fulfill user requests. Data from users is processed through the server and appropriate responses are sent back to the user.

### AI Model Module:

The AIModel module is an artificial intelligence model that generates responses to user messages. This model works with a specific AI algorithm to analyze each incoming query and generate the most accurate response. AIModel can be customized based on the AI model used in the project and can continuously evolve based on user interactions. This module plays a critical role in ensuring that the conversation occurs in a meaningful and natural way.

### Chat Management Module

The Chat Management module is responsible for managing the chats that users have. This module ensures that chat histories are stored, messages are organized and sorted. Each message is stored with details like which user it belongs to and which chat session it is associated with. This module is critical for providing users with access to previous chats, ensuring a more comprehensive and consistent conversation history.

### Profile Management Module

The Profile Management module manages the editing and updating of user profiles. User information is stored and can be edited here. This module allows users to manage all information related to their accounts.

### Authentication Module

The Authentication module handles the authentication process of users. This module ensures a secure authentication process when users log in or create a new account. Typically, token-based authentication is used to manage each user's access level. The Authentication module is crucial for maintaining the security of the system.

**Client App Module**

The Client App module is the interface where the user interacts with the chatbot. This module typically works as a web or mobile application, enabling users to chat using text, voice, images, and other media types. The Client App captures the messages typed by users, sends them to the server, and displays the responses from the server. The user interface (UI) and user experience (UX) are optimized to provide a friendly and engaging conversation experience. In this module, users receive responses generated by the AIModel and continue their interactions with the chatbot.

**Chat Manager Module**

The Chat Manager module is responsible for handling the type of queries sent during a chat session. It allows users to select the type of query (such as text, photo, video, etc.) they want to send. This module ensures that the correct processing is applied based on the selected query type, enabling flexible interaction with the chatbot. This module is also responsible for displaying the output to the user.

## 5.5 Low Level Design

### 5.5.1 Authentication Module

This diagram explains the user's registration and login to the system.

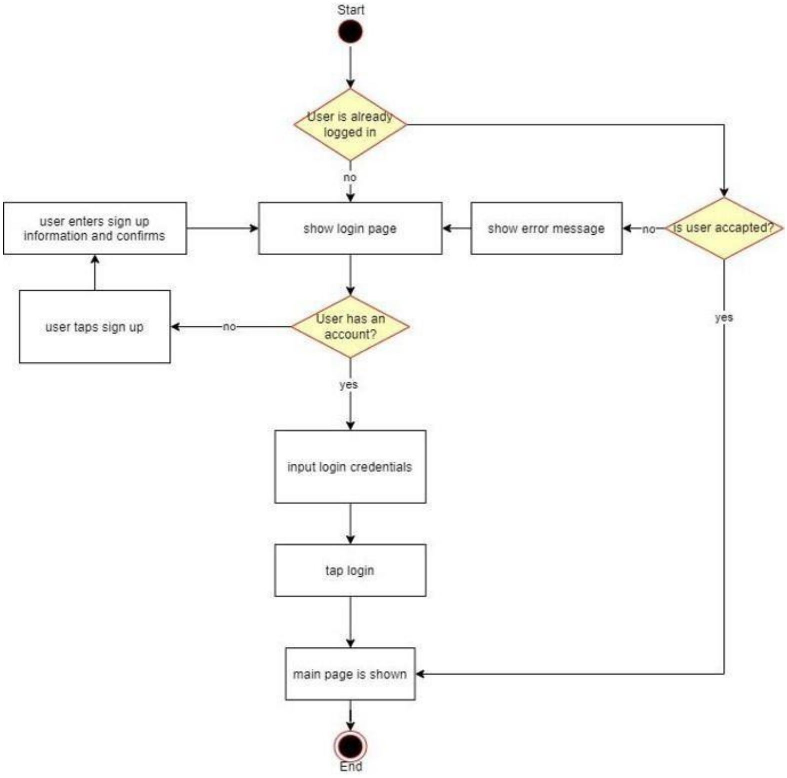


*Figure 21 Authentication Activity Diagram*

### 5.5.2 Chat Manager Module

It is a module where the type of query that the user will send is determined. The modules and steps selected by the user are shown in the diagrams below.

.

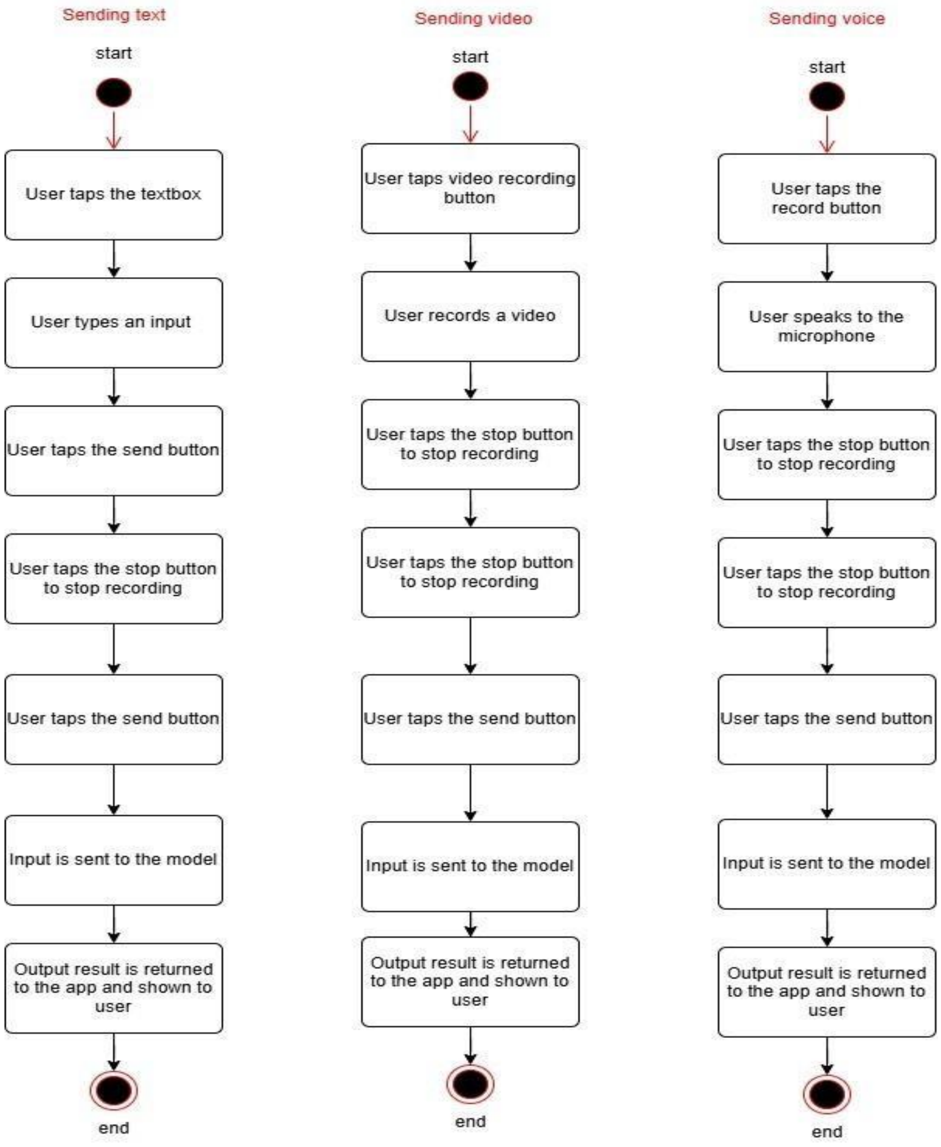

Sending text
start
User taps the textbox
User types an input
User taps the send button
User taps the stop button to stop recording
User taps the send button
Input is sent to the model
Output result is returned to the app and shown to user
end
Sending video
start
User taps video recording button
User records a video
User taps the stop button to stop recording
User taps the stop button to stop recording
User taps the send button
Input is sent to the model
Output result is returned to the app and shown to user
end
Sending voice
start
User taps the record button
User speaks to the microphone
User taps the stop button to stop recording
User taps the stop button to stop recording
User taps the send button
Input is sent to the model
Output result is returned to the app and shown to user
end

*Figure 22 Chat Activity Diagram*

### 5.5.3 Profile Management Module

The steps that users take to change their account properties are shown in the diagram below.

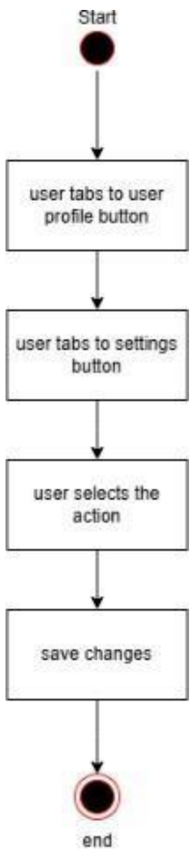

Start
user tabs to user profile button
user tabs to settings button
user selects the action
save changes
end

*Figure 23 Settings Activity Diagram*

## 5.6 Database Design

### 5.6.1 E-R Diagram

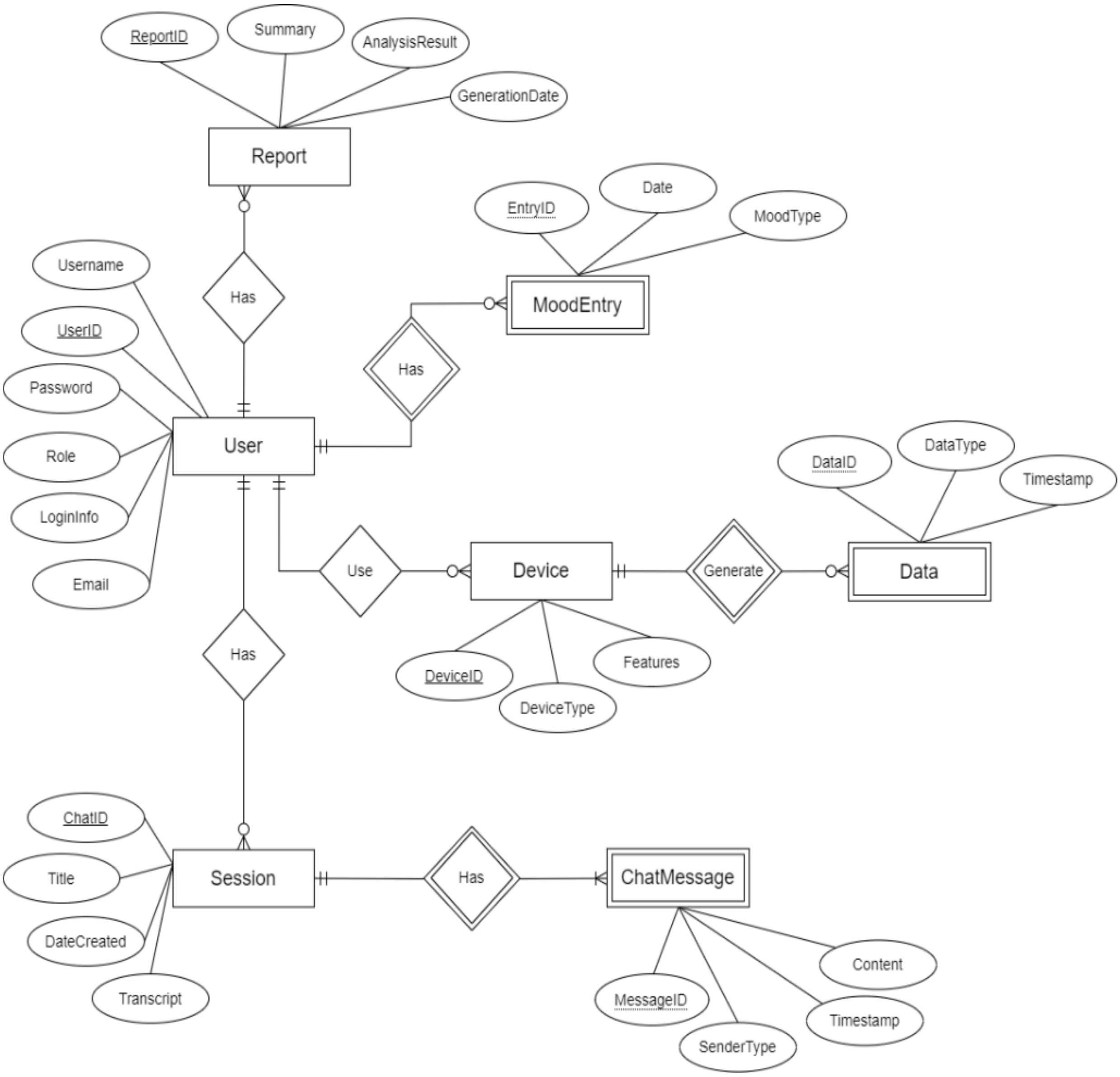


*Figure 24 ERD (Entity-Relationship)*

### 5.6.2 Tables

### User

**Attributes:**

1. **UserID**: Primary Key, Integer, Unique identifier for each user.
2. **Username**: String, the name chosen by the user.
3. **Password**: String, Encrypted password for the user.
4. **Role**: String, The role of the user (e.g., Admin, User).
5. **LoginInfo**: String, Information about the user's login records.
6. **Email**: String, User's email address.

### Report

**Attributes:**

1. **ReportID**: Primary Key, Integer, Unique identifier for each report.
2. **Summary**: String, A summary of the report.
3. **AnalysisResult**: String, the analysis result included in the report.
4. **GenerationDate**: Date, The date the report was generated.
5. **UserID**: Foreign Key, Integer, References the User table.

### MoodEntry

**Attributes:**

1. **EntryID**: Partial Key, Integer, Unique identifier for each mood entry.
2. **Date**: Date, the date the mood entry was recorded.
3. **MoodType**: String, The type of mood (e.g., Happy, Sad).
4. **UserID**: Foreign Key, Integer, References the User table.

**Session**

**Attributes:**

1. **ChatID**: Primary Key, Integer, Unique identifier for each session.
2. **Title**: String, The title of the session.
3. **DateCreated**: Date, the date the session was created.
4. **Transcript**: Text, The transcript of the session.
5. **UserID**: Foreign Key, Integer, References the User table.

**ChatMessage**

**Attributes:**

1. **MessageID**: Partial Key, Integer, Unique identifier for each chat message.
2. **Content**: Text, The content of the message.
3. **Timestamp**: DateTime, the time the message was sent.
4. **SenderType**: String, Type of sender (e.g., User, System).
5. **ChatID**: Foreign Key, Integer, References the Session table.

**Device**

**Attributes:**

1. **DeviceID**: Primary Key, Integer, Unique identifier for each device.
2. **DeviceType**: String, The type of device (e.g., Wearable, Smartphone).
3. **Features**: String, Features of the device.
4. **UserID**: Foreign Key, Integer, References the User table.

**Data**

**Attributes:**

1. **DataID**: Partial Key, Integer, Unique identifier for each data entry.
2. **DataType**: String, The type of data (e.g., Sensor Data, Analysis Data).
3. **Timestamp**: DateTime, the time the data was generated.
4. **DeviceID**: Foreign Key, Integer, References the Device table.

# 6. Test Plan

## 6.1 Introduction

### 6.1.1 Version Control

**Table 3 Version Control**

| Version No | Description of Changes | Date |
|---|---|---|
| 1.0 | First Version | March 16, 2025 |
| 1.1 | Second Version | April 20, 2025 |
| 1.2 | Third Version | May 18, 2025 |

### 6.1.2 Overview

This document outlines the test plan and design specification for SenseAI, which focuses on multimodal emotion recognition. The project utilizes NLP, computer vision, incorporating machine learning models to analyze text, visual, and audio input for accurate emotion classification. It includes a chatbot for psychological support and analysis reports for monitoring the mental states. This document ensures that fatures operate as intended, considering various scenarios. It details functional, non-functional, edge-case, API, admin, and security test specifications.

### 6.1.3 Scope

The test plan covers functional and non-functional testing of SenseAI. It references the SRS, SDD, and test cases for validation. Testing will include unit tests, integration tests, and system tests to ensure the application meets the design specifications.

### 6.1.4 References

[1] GitHub Repository: https://github.com/CankayaUniversity/ceng-407-408-2024-2025-SenseAI

[2] SenseAI Software Requirements Spesification v1.2

[3] SenseAI Software Design Description v1.2

### 6.1.5 Terminology

**Table 4 Terminology**

| Acronym | Definition |
|---|---|
| AI | Artificial Intelligence |
| TDS | Test Design Specification |
| TC_ID | Test Case Identifier |
| API | Application Programming Interface |
| TP | Test Plan |
| UI | User Interface |

## 6.2 Features to be tested

F1. User Login & Registration

F2. Chatbot (Text, Voice, Image, Heartbeat)

F3. Sentiment Analysis

F4. Admin functionalities (User blocking, Record viewing)

F5. API stability and concurrency

F6. User profile updates

F7. Error handling and edge case processing

F8. Security features (rate limiting, session expiry, password encryption)

### 6.2.1 User Login & Registration

This feature includes user authentication, account creation, password reset, and credential validation. It ensures only authorized users access the application and provides a secure onboarding process.

### 6.2.2 Chatbot

This covers the core interactive experience where users input queries via text, voice, image, and heart rate data. It validates that the chatbot receives, processes, and responds appropriately to different modalities.

### 6.2.3 Sentiment Analysis

This verifies the backend AI analysis of emotional state using input data, ensuring the system accurately interprets mood and context for analysis reports.

### 6.2.4 Admin Functionalities

This includes administrative access, user record viewing, access control, and blocking users. Tests will confirm appropriate role-based permissions and data handling.

### 6.2.5 API Stability and Concurrency

Validates that the REST API handles simultaneous requests, returns appropriate responses under load, and performs correctly without degradation

### 6.2.6 User Profile Updates

Tests for editing profile data such as mood, username, password, and profile visibility, verifying that data is saved and retrieved correctly.

### 6.2.7 Error Handling and Edge Case Processing

This ensures that the system gracefully handles invalid inputs, unsupported file types, and limits like input length, providing clear feedback to users.

### 6.2.8 Security Features

Includes validation of password encryption, rate limiting on failed logins, session timeout handling, and ensuring the AI doesn't leak sensitive user information.

## 6.3 Features not to be tested:

- Multilingual support (planned for future versions)
- Payment gateway integration (future enhancement)

## 6.4 Approach

The testing approach for SenseAI uses both black-box and white-box testing methods. Each feature will be tested against its requirements using user flows and edge cases. Unit tests will be executed using Flutter’s framework and QUnit for relevant logic components. Integration tests will verify component interaction. API tests using Postman will simulate typical and concurrent usage. Security and performance considerations will be addressed through manual checks and browser tools. Testing will be complete when all major functionalities meet expected outcomes and critical issues are resolved.

- Unit testing for each independent module
- Integration testing between system components
- System testing for complete workflow validation and edge case coverage
- Tools: Postman, JMeter, Chrome DevTools, Flutter test, Emulator
- Metrics: Response time, correctness, error handling, UI behavior

## 6.5 Item pass / fail criteria

### 6.5.1 Exit Criteria

- Feature functions as per SRS & SDD
- The system handles edge cases such as invalid input
- The system meets functional and non-functional requirements
- Emotion detection accuracy meets or exceeds the predefined benchmark

### 6.5.2 Fail Criteria

- Any deviation from expected output
- System responding slower than expected
- UI crash or logic error

## 6.6 Suspension Criteria and Resumption

- Testing will be suspended in the event of a system crash, downtime or discovery of severe bugs
- Testing may resume once the issue is resolved and retested successfully

## 6.7 Test Deliverables

- Test Cases
- Test Environment Setup
- Test Case Design
- Test Results
- Defect Reporting
- Re-testing

## 6.8 Testing Tasks

The testing tasks for the SenseAI project covers various phases in the testing process, ensuring a comprehensive and systematic approach. The main tasks are outlined below:

**Test Planning**

Developing a detailed test plan, outlining the scope, approach, and deliverables for testing the system functionality.

**Test Design**

Designing detailed test scenarios.

**Test Case Preparation**

Developing test cases based on the designed scenarios and the rules identified, ensuring coverage of all identified functionalities and scenarios.

**Test Execution**

Executing test cases as per the test plan. Performing black-box testing methods to validate the password changing functionality.

## 6.9 Environmental Needs

The test environment requires hardware configurations, Python 3 environment, relevant software libraries, and security measures to protect test facilities.

- Android/iOS Emulator
- Real mobile device
- Stable internet
- Backend server + database access
- Test accounts and Zepp Life credentials

## 6.10 Responsibilities

Testing responsibilities will be distributed among team members according to their respective areas of contribution.

- **Testing Manager**
  - Overall planning and coordination
  - Resource allocation
  - Scheduling
- **Tester**
  - Development of test cases
  - Documentation of test procedures
  - Test execution
  - Logging the results
  - Finding defects

## 6.11 Risks And Contingencies

| **Risk** | **Solution** |
|---|---|

| | |
|---|---|
| Limited access to labeled training data | Use publicly available datasets and semi-supervised labeling techniques. |
| Insufficient test data | Create synthetic test data |
| Potential limitations in computational resources | Schedule tests in off-peak hours and optimize model performance |
| Limited experience with QUnit | Allocate time for tester training and provide tutorials and documentation |
| AI model returns unpredictable results for unseen edge cases | Include diverse and adversarial inputs in testing scenarios |

## 6.12 Test design specifications

### 6.12.1 User Login & Registration

#### *6.12.1.1 Subfeatures to be tested*

- Valid login
- Invalid login
- Account creation
- Duplicate account prevention
- Password reset

#### *6.12.1.2 Subfeatures not to be tested*

- Multi-language login interface (future release)

#### *6.12.1.3 Approach*

Manual UI-based form submissions, validation against database records, and observing redirect behavior.

#### *6.12.1.4 Item Pass/Fail Criteria*

Pass if login/signup actions succeed with valid input and are blocked with invalid input.

#### *6.12.1.5 Environmental Needs*

- Test accounts
- Backend connectivity
- Browser access
- Database

#### *6.12.1.6 Test Cases*

Here list all the related test cases for this feature

**Table 5 Test Cases**

| TC ID | Requirements | Priority | Scenario Description |
|---|---|---|---|
| MO.LG.fn.01 | 3.1.1.2 | High | Login with valid credentials |
| MO.LG.fn.02 | 3.1.1.2 | High | Login with invalid credentials |
| MO.SU.fn.01 | 3.1.1.1 | High | Successful account creation |
| MO.SU.fn.02 | 3.1.1.1 | Medium | Attempting to register with existing email |

### 6.12.2 Chatbot

#### *6.12.2.1 Subfeatures to be tested*

- Text messaging
- Voice message
- Image upload
- Biometric input

#### *6.12.2.2 Subfeatures not to be tested*

- Support for non-human languages
- Unstructured symbolic inputs (e.g., emojis, ASCII art)

#### *6.12.2.3 Approach*

Simulate user input via various input channels.

#### *6.12.2.4 Item Pass/Fail Criteria*

- Pass: Message is accepted and AI responds
- Fail: No response, UI hangs, or invalid format accepted

#### *6.12.2.5 Environmental Needs*

- Microphone
- Camera
- Stable internet

#### *6.12.2.6 Test Cases*

Here list all the related test cases for this feature

**Table 6 Test Cases**

| TC ID | Requirements | Priority | Scenario Description |
|---|---|---|---|
| MO.CH.fn.01 | 3.1.1.8 | High | Send a text message to chatbot |
| MO.CH.fn.02 | 3.1.1.8 | Medium | Send an empty message |
| MO.AI.fn.01 | 3.2.1.5 | High | Send a voice message for analysis |
| MO.AI.fn.02 | 3.2.1.7 | Medium | Retrieve heart rate data |
| MO.AI.fn.03 | 3.2.1.6 | High | Send image for sentiment analysis |

### 6.12.3 Sentiment Analysis

#### *6.12.3.1 Subfeatures to be tested*

- Emotion classification

- Result generation
- Report logging

#### *6.12.3.2 Subfeatures not to be tested*

- Deep model training internals

#### *6.12.3.3 Approach*

Submit controlled inputs with known sentiments.

#### *6.12.3.4 Item Pass/Fail Criteria*

- Pass: Expected sentiment is returned
- Fail: Sentiment mismatches or analysis fails

#### *6.12.3.5 Environmental Needs*

- Test dataset with labeled sentiments

#### *6.12.3.6 Test Cases*

**Table 8 Test Cases**

| TC ID | Requirements | Priority | Scenario Description |
|---|---|---|---|
| MO.CH.fn.01 | 3.1.1.8 | High | Send a text message to chatbot |
| MO.AI.fn.01 | 3.2.1.5 | High | Send a voice message for analysis |
| MO.AI.fn.02 | 3.2.1.7 | Medium | Retrieve heart rate data |
| MO.AI.fn.03 | 3.2.1.6 | High | Send image for sentiment analysis |

### 6.12.4 Admin Functionalities

#### *6.12.4.1 Subfeatures to be tested*

- View records
- Block users

#### *6.12.4.2 Subfeatures not to be tested*

- Admin analytics dashboards

#### *6.12.4.3 Approach*

Login as admin and perform control actions.

#### *6.12.4.4 Item Pass/Fail Criteria*

- Pass: Actions succeed, users affected
- Fail: Permissions leak, incorrect records fetched

#### *6.12.4.5 Environmental Needs*

- Admin credentials
- User pool

#### *6.12.4.6 Test Cases*

**Table 9 Test Cases**

| TC ID | Requirements | Priority | Scenario Description |
|---|---|---|---|
| MO.ADM.fn.01 | 3.2.2.4 | High | View user records as admin |
| MO.ADM.fn.02 | 3.2.2.5 | Medium | Block a user |

### 6.12.5 API

#### *6.12.5.1 Subfeatures to be tested*

- Multiple request handling
- API status codes
- Performance under load

#### *6.12.5.2 Subfeatures not to be tested*

- Deprecated endpoints

#### *6.12.5.3 Approach*

Use Postman and browser DevTools with Promise()

#### *6.12.5.4 Item Pass/Fail Criteria*

- Pass: 200 OK, valid response under load
- Fail: 5XX errors, timeouts, long delays

#### *6.12.5.5 Environmental Needs*

- Postman
- Internet
- Backend logs

#### *6.12.5.6 Test Cases*

**Table 10 Test Cases**

| TC ID | Requirements | Priority | Scenario Description |
|---|---|---|---|
| MO.API.fn.01 | 3.1.4 | High | Validate AI model API response |
| MO.API.fn.02 | 3.1.4 | High | Handle concurrent API requests |
| MO.API.fn.03 | 3.1.4 | High | Return correct HTTP status codes |
| MO.API.fn.04 | 3.3 | High | Verify server response speed |
| MO.FR.fn.01 | 3.3 | High | Measure Promise() execution time |
| MO.PF.fn.01 | 3.3 | High | Measure AI response time under normal load |

### 6.12.6 User Profile Updates

#### *6.12.6.1 Subfeatures to be tested*

- Edit name
- Edit password
- Edit mood

#### *6.12.6.2 Subfeatures not to be tested*

- Profile picture change

#### *6.12.6.3 Approach*

Update profile and verify via GET.

#### *6.12.6.4 Item Pass/Fail Criteria*

- Pass: Data reflects change
- Fail: Errors in update, data inconsistency

#### *6.12.6.5 Environmental Needs*

- Active session
- Valid test user

#### *6.12.6.6 Test Cases*

**Table 11 Test Cases**

| TC ID | Requirements | Priority | Scenario Description |
|---|---|---|---|
| MO.PR.fn.01 | 3.1.1.4 | Medium | Update user profile information |

### 6.12.7 Error Handling and Edge Cases

#### *6.12.7.1 Subfeatures to be tested*

- Oversized input
- Unsupported formats
- No face images

#### *6.12.7.2 Subfeatures not to be tested*

- AI-generated feedback suggestion

#### *6.12.7.3 Approach*

Push input beyond limits, observe feedback.

#### *6.12.7.4 Item Pass/Fail Criteria*

- Pass: System handles gracefully
- Fail: Crash, freeze, or silent fail

#### *6.12.7.5 Environmental Needs*

- Test content for edge cases

#### *6.12.7.6 Test Cases*

**Table 12 Test Cases**

| TC ID | Requirements | Priority | Scenario Description |
|---|---|---|---|
| MO.ED.fn.01 | 3.1.1.8 | Medium | Input too long |
| MO.ED.fn.02 | 3.1.1.8 | Medium | Upload unsupported file type |
| MO.ED.fn.03 | 3.1.1.8 | Medium | No human face in image |
| MO.ED.fn.04 | 3.1.1.8 | Medium | Cartoon/animal/AI-generated face |
| MO.ED.fn.05 | 3.1.1.8 | Medium | Blurred or obstructed faces |
| MO.ED.fn.06 | 3.1.1.8 | Low | Multiple faces in an image |
| MO.ED.fn.07 | 3.1.1.8 | Low | Rotated/upside-down face images |
| MO.ED.fn.08 | 3.1.1.8 | Low | Face on paper or digital display |
| MO.FR.fn.02 | 3.2.1 | High | Measure chat history loading time |
| MO.FR.fn.03 | 3.1.1.8 | High | Upload image and video without delay |

Security Features

#### *6.12.7.7 Subfeatures to be tested*

- Login attempt limits
- Session expiration

- Password encryption
- Info leak

#### *6.12.7.8 Subfeatures not to be tested*

- Multi-factor authentication (future)

#### *6.12.7.9 Approach*

Simulate breaches and check defense.

#### *6.12.7.10 Item Pass/Fail Criteria*

- Pass: Data secure, user restricted on abuse
- Fail: Plain text passwords, session reuse, info leak

#### *6.12.7.11 Environmental Needs*

- Database
- Dev tools

#### *6.12.7.12 Test Cases*

**Table 13 Test Cases**

| TC ID | Requirements | Priority | Scenario Description |
|---|---|---|---|
| MO.SEC.fn.01 | 3.6.3 | High | Verify secure password storage |
| MO.SEC.fn.02 | 3.6.3 | High | Rate limit on failed login attempts |

# 7. Test Results

## 7.1 Individual Test Results

**Table 14 Individual Test Results**

| TC ID | Priority | Date Run | Run By | Result | Explanation |
|---|---|---|---|---|---|
| MO.SU.fn.01 | H | 20.04.2025 | Özge Alkan | Pass | |

| TC ID | Priority | Date Run | Run By | Result | Explanation |
|---|---|---|---|---|---|
| MO.LG.fn.02 | H | 20.04.2025 | Mustafa Tozman | Pass | |
| MO.LG.fn.01 | H | 20.04.2025 | Mustafa Tozman | Pass | |
| MO.CH.fn.01 | H | 20.04.2025 | Ece Heval Ünal | Pass | |
| MO.AI.fn.01 | H | 20.04.2025 | Ece Heval Ünal | Pass | |
| MO.AI.fn.03 | H | 20.04.2025 | Ece Heval Ünal | Pass | |
| MO.ADM.fn.01 | H | 23.04.2025 | Sadrettin Anıl Karaçay | Fail | This problem will be fixed in the next release |
| MO.API.fn.01 | H | 20.04.2025 | Zeliha Aybüke Baştürk | Pass | |
| MO.API.fn.02 | H | 20.04.2025 | Ece Heval Ünal | Pass | |
| MO.API.fn.03 | H | 20.04.2025 | Özge Alkan | Pass | |
| MO.API.fn.04 | H | 18.04.2025 | Özge Alkan | Pass | |
| MO.FR.fn.01 | H | 20.04.2025 | Zeliha Aybüke Baştürk | Pass | |
| MO.PF.fn.01 | H | 20.04.2025 | Özge Alkan | Pass | |
| MO.FR.fn.02 | H | 21.04.2025 | Mustafa Tozman | Pass | |
| MO.FR.fn.03 | H | 21.04.2025 | Mustafa Tozman | Pass | |

| TC ID | Priority | Date Run | Run By | Result | Explanation |
|---|---|---|---|---|---|
| MO.SEC.fn.01 | H | 18.04.2025 | Sadrettin Anıl Karaçay | Fail | This problem will be fixed in the next release |
| MO.SEC.fn.02 | H | 21.04.2025 | Mustafa Tozman | Fail | Token validation and security edge cases will be fully implemented |
| MO.SU.fn.02 | M | 20.04.2025 | Özge Alkan | Pass | |
| MO.CH.fn.02 | M | 18.04.2025 | Özge Alkan | Pass | |
| MO.AI.fn.02 | M | 23.04.2025 | Zeliha Aybüke Baştürk | Pass | |
| MO.ADM.fn.02 | M | 23.04.2025 | Sadrettin Anıl Karaçay | Fail | Role-based permission logic not finalized yet |
| MO.PR.fn.01 | M | 23.04.2025 | Sadrettin Anıl Karaçay | Fail | This problem will be fixed in the next release |
| MO.ED.fn.01 | M | 20.04.2025 | Özge Alkan | Pass | |
| MO.ED.fn.02 | M | 20.04.2025 | Zeliha Aybüke Baştürk | Pass | |
| MO.ED.fn.03 | M | 20.04.2025 | Zeliha Aybüke Baştürk | Pass | |
| MO.ED.fn.04 | M | 25.04.2025 | Ece Heval Ünal | Pass | |
| MO.ED.fn.05 | M | 25.04.2025 | Zeliha Aybüke Baştürk | Pass | |
| MO.ED.fn.06 | L | 25.04.2025 | Zeliha Aybüke Baştürk | Pass | |
| MO.ED.fn.07 | L | 25.04.2025 | Sadrettin Anıl Karaçay | Pass | |

| TC ID | Priority | Date Run | Run By | Result | Explanation |
|---|---|---|---|---|---|
| MO.ED.fn.08 | L | 25.04.2025 | Ece Heval Ünal | Pass | |

## 7.2 Summary of Test Results

**Table 15 Test Results**

| Priority | Number of TCs | Executed | Passed |
|---|---|---|---|
| H | 17 | 17 | 14 |
| M | 10 | 10 | 8 |
| L | 3 | 3 | 3 |
| **Total** | **30** | **30** | **25** |

## 7.3 Exit Criteria

**Table 7 Exit Criteria**

| Criteria | Met or Not |
|---|---|
| 100% of the test cases are executed | Y |
| 95% of the test cases passed | N |
| All High and Medium Priority test cases passed | N |

## 7.4 Known Problems

- Some admin and profile functionalities could not be fully validated due to the ongoing finalization of Firestore database structure and data consistency checks.
- There are test cases failed due to the absence of complete security policy enforcement and role-based access control. This is planned for completion in the next release.

## 7.5 Conclusion

All test cases were executed successfully. Despite a few non-critical failures related to database management and security edge cases, the core functionality of the SenseAI system is stable and operational.

The multimodal chatbot, emotion recognition pipeline, and user interface all performed as expected during tests. The system is ready to release and to be used in real-world scenarios, especially in a controlled or pilot phase. The failures do not impact primary user workflows, chatbot responsiveness, or multimodal analysis functionality, and they will be addressed in the next releases.

# 8. Conclusions

## Summary

In this project, we developed **SenseAI**, a multimodal artificial intelligence system aimed at providing psychologically informed support. The system integrates textual input, vocal tone, facial expression analysis, and biometric data to detect and interpret the user's emotional state, and then generate appropriate, supportive responses. We fine-tuned a LLaMA-based language model using a custom psychology-oriented dataset for emotion recognition. Additionally, components were developed to analyze facial expressions using DeepFace, voice-based emotional cues, and biometric data collected via Xiaomi Mi Band devices.

## What We Learned

- Multimodal approaches enable a richer and more accurate understanding of user emotions than unimodal systems.
- Transformer-based models, when fine-tuned on domain-specific datasets, can effectively generate empathetic responses.
- Integrating data from multiple modalities improves emotional context comprehension and leads to more human-like interaction.
- Detecting indirectly expressed emotions in textual input is crucial in psychological dialogue systems and mental health tools.

## Advantages

- A functional prototype has been built that demonstrates real-world applicability of multimodal AI in psychological support.
- The system not only detects emotion but also offers psychologically supportive, therapeutic-style responses.
- The modular and open-source-friendly design allows for future enhancements and extensions.

## Future Work and Open Problems

Looking ahead, there are several areas in which SenseAI can be expanded and improved:

- **Model Optimization**: We plan to continue training our current models to enhance both accuracy and inference speed. This includes experimenting with different fine-tuning strategies and loss functions specific to psychological dialogue.
- **Scaling Up**: One of our next goals is to train on larger models and more diverse datasets to improve generalization and handle more nuanced emotional expressions.
- **User Interface Enhancements**: The chatbot and reporting interfaces will be redesigned to provide a smoother, more intuitive user experience that aligns with modern UX principles.
- **Long-Term Context Retention**: We aim to implement a more robust memory system using an extended context window, allowing the chatbot to retain and refer to previous parts of the conversation for more coherent interactions.
- **Early Fusion Techniques**: We will explore early fusion strategies using the FLAVA model to better integrate vision, audio, and text modalities at an earlier stage of processing, which may improve multimodal alignment.
- **Database Challenges**: One open issue involves optimizing our backend database structure. The current implementation does not scale well with session-based storage of high-frequency biometric and interaction data. Addressing this will be critical for real-time and long-session usage.
- **Emotion Complexity**: Accurately identifying overlapping or ambivalent emotional states remains a technical and psychological challenge. More granular labeling and multi-label classification approaches will be investigated.

## Disadvantages

- Emotion misclassification may occur in ambiguous or highly contextual cases.
- The multimodal setup demands higher computational resources, limiting its performance on low-end or mobile devices.
- The language model’s generalization ability may be limited due to training on a domain-specific dataset.

## Acknowledgement

Acknowledge any individuals who have helped you during the course of the project, including your project advisor. If you have been supported by a company or a scholarship, then this should also be gratefully acknowledged.

We would like to express our deepest gratitude to our project advisors, Dr. Atila Bostan and Dr. İsmail Bora Çelikkale, for their invaluable guidance, mentorship, and encouragement throughout the development of SenseAI. Their insightful feedback and academic expertise were essential in shaping both the technical and conceptual direction of this project.

We are sincerely thankful to TÜBİTAK (The Scientific and Technological Research Council of Turkey) for their financial and institutional support, which played a critical role in enabling our research and development efforts.

We are also proud to share that our project passed the preliminary evaluation phase of TEKNOFEST, one of the most prestigious technology competitions in Turkey. This milestone further motivated our team and validated the relevance of our work in the field of AI and mental health.

Additionally, we acknowledge the open-source communities and tools that made this project possible, including **OpenAI**, **HuggingFace**, **Unsloth**, **DeepFace**, **Flutter**, and the developers behind **Xiaomi Mi Band** integrations.

Finally, we extend our heartfelt appreciation to every member of the SenseAI team. This project was built on the dedication, collaboration, and shared vision of a group of individuals committed to developing empathetic, intelligent systems for psychological well-being. Our team remained dedicated to exploring a subject we were genuinely passionate about. We believed in the potential impact of our work and pushed forward with determination. Through close collaboration, shared curiosity, and resilience, we were able to make meaningful progress and develop a project we are proud of. This journey not only challenged us technically, but also strengthened our commitment to creating technology that serves a greater human purpose.

# References

1. Wang, S., Zhang, Z., Chen, B., Zhang, Z., Fang, W., and Pan, J. (2022). "Multimodal Emotion Recognition Based on Facial Expressions, Speech, and EEG." *IEEE Transactions on Affective Computing*, vol. 13, no. 1, pp. 142-154. doi: 10.1109/TAFFC.2021.10026861.

2. Lian, H., Lu, C., Li, S., Zhao, Y., Tang, C., and Zong, Y. (2023). "A Survey of Deep Learning- Based Multimodal Emotion Recognition: Speech, Text, and Face." *Entropy*, vol. 25, no. 10, pp. 1440. doi: 10.3390/e25101440.

3. Yan, J., Li, P., Du, C., Zhu, K., Zhou, X., Liu, Y., and Wei, J. (2024). "Multimodal Emotion Recognition Based on Facial Expressions, Speech, and Body Gestures." *Electronics*, vol. 13, no. 18, p. 3756. doi: 10.3390/electronics13183756.

4. Sharma, A., Saxena, A., Kumar, A., and Singh, D. (2023). "Depression Detection Using Multimodal Analysis with Chatbot Support." *2023 IEEE International Conference on Computational Intelligence and Data Science (ICCIDS)*, Chennai, India, pp. 1-6. doi: 10.1109/ICCIDS56901.2023.10489080.
5. Woebot Health. "Woebot Health - Mental Health for the Digital Age." Available: https://woebothealth.com/. [Accessed: 24-November-2024].
6. Wysa. "Reports." Available: https://www.wysa.com/reports#6. [Accessed: 24 November-2024].
7. Youper. "AI Therapy." Available: https://www.youper.ai/. [Accessed: 24- November-2024].
8. Headspace for Work. "Headspace for Work." Available: https://organizations.headspace.com/. [Accessed: 24-November-2024].
9. News.com.au. "Meet Ellie, the robot therapist treating soldiers with PTSD." Available: https://www.news.com.au/technology/innovation/meet-ellie-the-robot-therapist- treating- soldiers-with-ptsd/news-story/0201fa7cf336c609182cffd637deef00. [Accessed: 24- November-2024].
10. Replika. "Replika - My AI Friend." Available: https://replika.ai/. [Accessed: 24-Oct- 2024].
11. American Psychiatric Association, Diagnostic and Statistical Manual of Mental Disorders, Fifth Edition (DSM-5). Arlington, VA: American Psychiatric Publishing, 2013.
12. American Psychological Association, Guidelines for the Practice of Telepsychology. [Online]. Available: https://www.apa.org/practice/guidelines/telepsychology, 2013.
13. Google. (2024). Material Design Guidelines. Retrieved from https://material.io/design