# Software Requirements Specification (SRS) Document

| | |
|---|---|
| ***Project number*** | Team 49[1] |
| ***Project Title*** | Wave Diaries |
| ***Document*** | Software Requirements Specification (SRS) Document |
| ***Creation date*** | February 15, 2025 |
| ***Created By*** | Raunak Seksaria (2023113019) |
| ***Client*** | Mr. Jazeel Jabbar, HopeLog(Startup) |

## Brief problem statement

The problem focuses on aiding people suffering from mental health issues such as depression, stress, and anxiety. The core objective is to build an AI-based model, integrated with a speech-to-speech module which can track the user's mood, and empathetically respond to alleviate the distress that the user might be facing. The model's context-aware therapeutic responses can act as an alternative to treatment by professional psychologists where money, time and opportunity can all be constraints for patients. It can also act complementary to the treatment and help the user tackle issues on-the-go in real-time, while the patient's personal therapist can review these conversations. The problem requires a personalized, sensitive solution committed to the needs of the user with very little margin of error available.

## System requirements

The system requirements for our solution is as follows:

1. A Trello Board has been created for task management.
2. V S Code is the primary editor which we shall be using.
3. MERN stack-based web-app
    a. MongoDB
    b. Express.js
    c. React (might use Next.js or Remix: react based Frameworks(undecided))
        i. Figma
        ii. Material UI/ Tailwind
    d. Node.js
    e. Associated Tools:
        i. Typescript
        ii. ESLint
        iii. Postman
        iv. JWT
4. GEN AI technologies (including for Automatic speech-to-text):
    a. LangChain: Whisper
    b. Groq
    c. Gemini
    d. GPT 4o

[1] Aditya Gaur (2023101052), Manit Roy (2023113022), Raunak Seksaria (2023113019), Shivam Gupta (2023101062)

    e. Smallest.ai
    f. Hugging Face
5. Python and python libraries required:
    a. PyTorch/ TensorFlow
    b. NumPy
    c. Pandas
    d. NLTK
    e. SpaCy
    f. PyAudio
6. Deployment:
    a. StreamLit
    b. Docker
    c. AWS

## Users profile

***Primary Target Audience:*** 13–19-year-old (Teenagers)

***Mode of usage:*** Web-app

***Technical Literacy (familiarity with software):*** High

Teenagers or adolescents preparing for competitive examinations, those dealing with college applications, and even those in school often face significant stress and anxiety due to intense competition. Some teenagers also grow up in a toxic environment, one which they have little control of, and these are more susceptible to such issues. Teenagers suffering from faltering relationships, or socially and morally depressed teenagers can end up using our application. This age group is particularly vulnerable to mental health challenges, both during and after puberty, as they navigate personal life complexities alongside academic pressures. As they mature, their ability to manage these challenges evolves, and we need to keep note of this while designing the application.

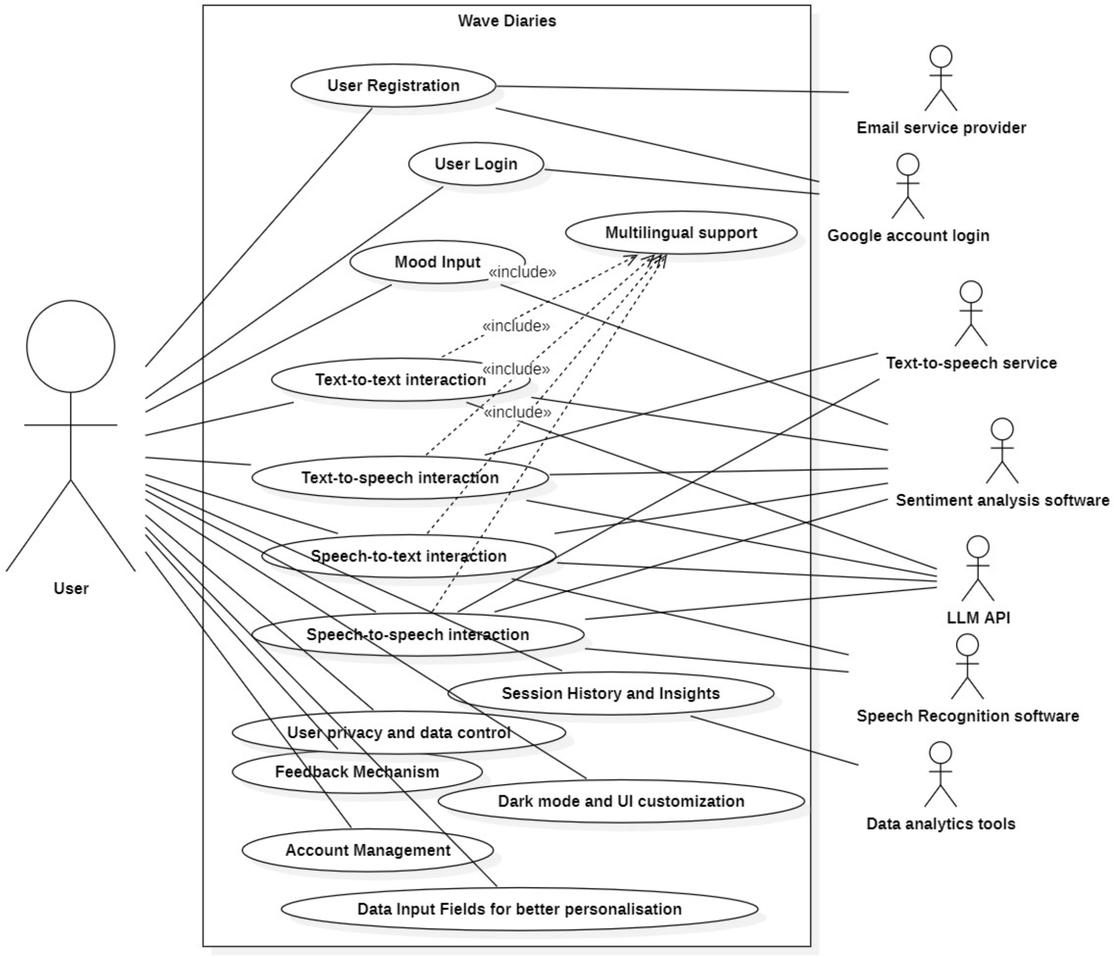
The responses of the model are intended to cater to these users' specific needs, personalized for our primary target audience. We note the general technical literacy and an appeal for aesthetic UIs that teenagers have and intend to design our web app as one that caters to their specific demands.

## Feature requirements (described using use cases)

| No. | Use Case Name | Description | Release |
|---|---|---|---|
| **1** | User Registration | Enables a new user to create an account by providing required details (e.g., email, password, and basic profile information). The system validates the information and sends a verification email to complete the registration. | R1 |
| **2** | User Login | Allows a registered user to log in securely by entering their credentials. The system verifies the provided details and | R1 |

| | | | |
|---|---|---|---|
| | | grants access to the personalized dashboard upon successful authentication. | |
| **3** | Mood Input | Allows the user to input their current mood in terms of presets. This helps personalise the responses better, and aids the system in its sentiment analysis. | R1 |
| **4** | Speech-to-Speech Interaction | Permits users to interact with the system using voice. The system converts spoken input to text, processes it, and then outputs a voice response, providing a natural conversational experience. Includes multilingual support | R1 |
| **5** | Text-to-speech interaction | Permits users to interact with the system using text. The system converts spoken input to text, processes it, and then outputs a voice response, providing a natural conversational experience. Includes multilingual support. | R1 |
| **6** | Text-to-text interaction | Permits users to interact with the system using text. The system converts spoken input to text, processes it, and then outputs a text response, providing a chat-like experience. Includes multilingual support. | R1 |
| **7** | Speech-to-text interaction | Permits users to interact with the system using voice. The system converts spoken input to text, processes it, and then outputs a text response, providing a chat-like experience. Includes multilingual support. | R1 |
| **8** | Session History and Insights | Allows users to view past interactions and mood trends through saved session logs. This feature helps users reflect on their progress and gain insights into their emotional patterns over time. Can be extremely useful to mental health support professionals to track their patient's trends. | R1 |
| **9** | Feedback Mechanism | Enables users to rate their experience and submit feedback about the system's interactions. The collected feedback supports continuous improvement of the AI and overall user experience. | R2 |
| **10** | Dark Mode and UI Customization | Allows users to customize the application's appearance by selecting themes (such as dark mode). These preferences are saved in the user profile to ensure a personalized and consistent experience across sessions. | R2 |
| **11** | Account Management | Provides users with the ability to update profile details, change passwords, or delete their account. The system securely processes these updates while ensuring data integrity and privacy. | R1 |
| **12** | User Privacy and Data Control | Lets users manage their personal data by reviewing and, if desired, deleting stored conversation logs and other personal information. This ensures that users have full control over their data in compliance with privacy standards. | R2 |
| **13** | Data Input Fields for better personalisation | The system dynamically adapts to deliver a seamless personalized and localized experience. | R2 |

## Use case diagram


Wave Diaries
User Registration
User Login
Multilingual support
Mood Input
«include»
«include»
Text-to-text interaction
«include»
«include»
Text-to-speech interaction
Speech-to-text interaction
Speech-to-speech interaction
Session History and Insights
User privacy and data control
Feedback Mechanism
Dark mode and UI customization
Account Management
Data Input Fields for better personalisation
User
Email service provider
Google account login
Text-to-speech service
Sentiment analysis software
LLM API
Speech Recognition software
Data analytics tools

## Use case description

***Delete all the blue text and fill-in the template before adding this to your repository or turning it in to your instructor.***

| | |
|---|---|
| **Use Case Number:** | UC-XX (Replace XX with a number) |
| **Use Case Name:** | Enter the name of Use Case |
| **Overview:** | Describe the purpose of the Use Case and give a 1-2 line description. This could be the same as the description provided in feature requirements section. |
| **Actors:** | List all actors that participate in this Use Case. |
| **Pre condition:** | Enter the condition that must be true before the main flow is executed. |

| | |
|---|---|
| **Flow:** | Main (success) Flow: Steps should be numbered. |
| | Alternate Flows: Include the post condition for each alternate flow if different from the main flow. |
| **Post Condition:** | Enter the condition that must be true when the main flow is completed. |

### *UC-1*

| | |
|---|---|
| **Use Case Number:** | UC-01 |
| **Use Case Name:** | User registration |
| **Overview:** | Enables a new user to create an account by providing required details (e.g., email, password, and basic profile information). The system validates the information and sends a verification email to complete the registration. |
| **Actors:** | User, Email service Provider, Google account login |
| **Pre condition:** | • The system is up and running.<br>• The user reaches the landing page or the registration page in the webapp. |
| **Flow:** | Main (success) Flow:<br>1. The user inputs required information (name, email, password, date of birth, etc.).<br>2. The user reviews and accepts the privacy policy and consent forms for data usage and mental health support.<br>3. The user submits the data entered, which is then sent to our system.<br>4. The system checks if the email or username already exists.<br>5. The system validates that mandatory fields are filled and comply with format requirements (valid email format, strong password).<br>6. The system creates a new account in the database.<br>7. The system sends a verification email or code to the user using our email service provider<br>8. The user enters the validation code from the verification message to verify themselves.<br>9. Registration Complete: The system displays a success message and redirects the user to the login page. |
| | Alternate Flow:<br>*a) System crash<br>1. The system halts normal processing and displays a dedicated error notification page to the user, indicating that a system error has occurred.<br>2. The error page provides instructions to refresh the page, try again later, or contact support with a reference code for troubleshooting.<br>3. The system logs detailed crash information (including context and error codes) for troubleshooting and recovery purposes.<br>Postconditions**:**<br>• The user is redirected to a safe error state and informed about the system crash.<br>• Any unsaved session data may be lost, and the user is advised to restart the session after the system is restored to normal operation.<br>1a) User selects ‘Login with Google’ option<br>1.<br>4a) Duplicate Username: The user’s chosen username is already registered. |

| | |
|---|---|
| | 1. The system displays an error message indicating that the username is already in use.<br>2. The user is prompted to enter different credentials.<br>3. Goes back to step 3 in main flow<br>Post Condition:<br>Registration is not completed until a unique username is provided.<br>3a)<br>a. Condition: Mandatory fields are missing, or the provided data fails format checks (e.g., invalid email).<br>b. Flow:<br>i. The system highlights the invalid fields and provides instructions for correction (if any).<br>ii. The user updates the information and resubmits.<br>c. Post Condition: Registration remains incomplete until valid data is provided.<br>1. AF3: User Declines Consent<br>d. Condition: The user does not agree to the privacy policy or data usage terms.<br>e. Flow:<br>i. The system notifies the user that consent is required to create an account.<br>ii. The user can either accept the terms or exit the process.<br>f. Post Condition: Registration is terminated if the user does not provide consent. |
| **Post Condition:** | A new user account is successfully created and verified. The user can now log in to the AI-based mental health application and access personalized support, mood tracking, and other therapeutic features under secure and privacy-compliant conditions. |

### *UC-2*

| | |
|---|---|
| **Use Case Number:** | UC-02 |
| **Use Case Name:** | User Login , Google Account Login |
| **Overview:** | Allows users to login into their respective accounts to access the features of this AI driven mental health web application. |
| **Actors:** | User, Google Account Login |
| **Pre condition:** | The user has an existing account in the system.<br>The system is online and capable of processing authentication requests. |
| **Flow:** | Main (success) Flow:<br>1. The user opens the application, and the login page is displayed.<br>2. The user enters a valid username and password.<br>3. The user clicks the "Login" button.<br>4. The system checks the credentials provided against the stored user database.<br>5. If the credentials match, the system creates a session for the user.<br>6. The system redirects the user to the home page, indicating a successful login. |
| | Alternate Flow: |

| | |
|---|---|
| | 2a) Invalid Credentials<br>o **Condition:** The user enters an incorrect username or password.<br>o **Flow:**<br>♣ The system displays an error message indicating invalid credentials.<br>♣ The user is prompted to re-enter valid credentials or reset the password.<br>o **Post Condition (for this alternate flow):** The user remains on the login page until valid credentials are provided or they choose to exit.<br>• **AF2: System Unavailable**<br>o **Condition:** The system or authentication service is temporarily offline.<br>o **Flow:**<br>♣ The system displays a notification that the service is currently unavailable.<br>♣ The user may attempt to log in again later.<br>o **Post Condition (for this alternate flow):** No session is created. The user remains logged out. |
| **Post Condition:** | Upon successful login, the user is authenticated and gains access to the system's exclusive features and benefits. |

### *UC-3*

| | |
|---|---|
| **Use Case Number:** | UC-03 |
| **Use Case Name:** | Mood Input |
| **Overview:** | Allows the user to input their current mood in terms of presets. This helps personalise the responses better and aids the system in its sentiment analysis. |
| **Actors:** | User, Sentiment Analysis Software |
| **Pre condition:** | • The user must have an existing account.<br>• The user is logged in.<br>• The user should be in mood input page. |
| **Flow:** | Main (success) Flow:<br>1. Choosing Emotion: User can choose between a set of predefined emotions.<br>2. Tracking: User's emotion is kept track of on a regular basis and for a given timeframe user can see how their emotions have fluctuated. |
| | Alternate Flow:<br>*a) System Crash<br>1. The system should be able to show detailed crash information with error codes and instructions to refresh the page and try again later.<br>Postconditions: The user is redirected to a safe error state and the user will be advised to restart the session. |

| | |
|---|---|
| | 2a) Database Error 1. The system might be unable to access the database in which the user's whole mood history is stored, and it should be displayed to user that the database is down, wait for some time for it to respond.<br>Postconditions: The user will be redirected to a database error page and will to told that user mood history couldn't be fetched. |
| **Post Condition:** | User can document their emotions, and for a given timeframe user can see how their emotions have fluctuated |

### *UC-4*

| | |
|---|---|
| **Use Case Number:** | UC-04 |
| **Use Case Name:** | Speech to Speech |
| **Overview:** | This use case enables a registered user to interact with an intelligent bot via voice. The system captures the user's spoken input, converts it to text using speech recognition, processes the text with a large language model (LLM), and then converts the LLM's response back into speech—all in the language chosen by the user. |
| **Actors:** | User, LLM API, Text to Speech, Speech to Text |
| **Pre condition:** | 1. The user has an existing account and is logged in.<br>2. The system is connected to the necessary external APIs for speech-to-text, LLM processing, and text-to-speech. |
| **Flow:** | - Main (success) Flow:<br>**1. Choose Language:** The user selects their preferred language from a list of supported options.<br>2. **Record Voice:** The system prompts the user to choose an audio input device (if multiple devices are available) and to grant the necessary permission. The user records their voice input.<br>**3. Speech-to-Text Conversion:** The system sends the recorded audio in small, parallelized chunks to a speech recognition service. The service converts the audio to text, supporting languages beyond English.<br>**4. Interacting with the LLM:** The converted text is forwarded to the LLM, which processes the input and generates a response. The response is formulated in the language initially selected by the user.<br>**5. Text-to-Speech Conversion:** The system converts the LLM's text response back into speech. The spoken response is delivered to the user. |
| | Alternate Flow:<br>*a) System Crash<br>1. The system halts normal processing and displays a dedicated error notification page to the user, indicating that a system error has occurred.<br>2. The error page provides instructions to refresh the page, try again later, or contact support with a reference code for troubleshooting.<br>3. The system logs detailed crash information (including context and error codes) for troubleshooting and recovery purposes.<br>Postconditions**:**<br>- The user is redirected to a safe error state and informed about the system crash. |

| | |
|---|---|
| | • Any unsaved session data may be lost, and the user is advised to restart the session after the system is restored to normal operation.<br>**3a, 4a, 5a) API Errors (Speech-to-Text, LLM, or Text-to-Speech):**<br>1. If the system fails to communicate with any API due to network issues or service downtime:<br>o If the network connection is lost, a network error message is displayed.<br>o If the network is available but a specific service is down, an error message indicates which service is unavailable.<br>7. **Postcondition:** The system halts further processing and informs the user until the issue is resolved.<br>**2a) Microphone Not Detected:**<br>1. The system lists all available audio capture devices and prompts the user to select one.<br>2. If the desired device is not listed, the user is advised to restart the system and update their audio drivers.<br>**Postcondition:** The system proceeds only after a valid device is selected; otherwise, it remains in an error state.<br>**2b) Microphone Permission Denied:**<br>1. If the user denies microphone permission, the system displays a “Microphone Access Denied” error.<br>**Postcondition:** The operation is paused, awaiting permission or a user-initiated abort.<br>**2c) Microphone Malfunction:**<br>1. If no audio input is detected even after a valid device is selected and permission is granted, the system assumes the microphone is malfunctioning.<br>**Postcondition:** The system displays a “Microphone Not Functioning” error and suggests checking the hardware or contacting support. |
| **Post Condition:** | • The user's voice input is successfully processed, and an audio response is generated in the selected language.<br>• The conversation state is updated for future interactions.<br>• In the event of an error, the system remains in a safe state, and the user receives clear instructions on how to proceed. |
| | |

### *UC-5*

| | |
|---|---|
| **Use Case Number:** | UC-05 |
| **Use Case Name:** | Text-to-Speech |
| **Overview:** | This use case allows a registered user to interact with the bot by typing their input while receiving audio responses generated by converting text from the LLM into speech. |
| **Actors:** | User, LLM API, Text to Speech |

| | |
|---|---|
| **Pre condition:** | - The user has an existing account and is logged in.<br>- The system's LLM and text-to-speech (TTS) services are available.<br>- The UI supports text input, and the device has audio output capabilities. |
| **Flow:** | Main (success) Flow:<br>**Main Flow:**<br>1. **Choose Language: The** user selects their preferred language for the response.<br>2. **Enter Text:** The user types their query or message into the interface.<br>3. **Process Input with LLM:** The system sends the text input to the LLM. The LLM generates a text response in the chosen language.<br>4. **Text-to-Speech Conversion:** The text response is converted into speech by the TTS service.<br>5. **Play Audio Response:** The system plays the generated audio response to the user. |
| | Alternate Flow:<br>*a) System Crash<br>4. The system halts normal processing and displays a dedicated error notification page to the user, indicating that a system error has occurred.<br>5. The error page provides instructions to refresh the page, try again later, or contact support with a reference code for troubleshooting.<br>6. The system logs detailed crash information (including context and error codes) for troubleshooting and recovery purposes.<br>Postconditions**:**<br>- The user is redirected to a safe error state and informed about the system crash.<br>- Any unsaved session data may be lost, and the user is advised to restart the session after the system is restored to normal operation.<br>**4a) TTS Service Error:**<br>1. If the TTS service fails, the system displays the text response along with an error message indicating the audio conversion issue.<br>**Postconditions:** The user receives the text response and may choose to retry the TTS conversion or continue with the text response.<br>**3a, 4b) Network/API Issues:**<br>1. If a network or API error occurs during LLM processing or TTS conversion, the system alerts the user and suggests retrying.<br>**Postconditions:** The system maintains a safe state and holds the current transaction until the network or API issue is resolved. |
| **Post Condition:** | - The user's text input is successfully processed.<br>- An audio response is generated and played in the user's selected language. |

### *UC-6*

| | |
|---|---|
| **Use Case Number:** | UC-06 |
| **Use Case Name:** | Text-to-Text |
| **Overview:** | This use case allows a registered user to interact with the bot by typing their queries and receiving text-based responses. |
| **Actors:** | User, LLM API |

| | |
|---|---|
| **Pre condition:** | • The user has an existing account and is logged in.<br>• The system's LLM is available and connected.<br>• The user interface supports text input and output. |
| **Flow:** | Main (success) Flow:<br>1. **Choose Language:** The user selects their preferred language from a list of supported options.<br>2. **Enter Text:** The user types of their query or message into the interface.<br>3. **Process Input with LLM:** The system sends the text input to the LLM. The LLM processes the input and generates a text response in the selected language.<br>4. **Display Response:** The system displays the text response on the screen. |
| | Alternate Flow:<br>*a) System Crash<br>7. The system halts normal processing and displays a dedicated error notification page to the user, indicating that a system error has occurred.<br>8. The error page provides instructions to refresh the page, try again later, or contact support with a reference code for troubleshooting.<br>9. The system logs detailed crash information (including context and error codes) for troubleshooting and recovery purposes.<br>Postconditions**:**<br>• The user is redirected to a safe error state and informed about the system crash.<br>• Any unsaved session data may be lost, and the user is advised to restart the session after the system is restored to normal operation.<br>**3a) LLM or System Error:**<br>1. If the LLM fails to process the request, the system displays an error message with instructions to retry.<br>**Postconditions:** The system remains in a stable state, and the user can re-enter the text without data loss.<br>**3b) Network/API Issues:**<br>If a network or API error occurs, the system alerts the user and provides options to retry or check the connection.<br>• **Postconditions:** No processing is completed, and the system remains safe and consistent until connectivity is restored. |
| **Post Condition:** | The user's text input is successfully processed, and the text response is displayed in the selected language. |

### *UC-7*

| | |
|---|---|
| **Use Case Number:** | UC-07 |
| **Use Case Name:** | Speech-to-Text |
| **Overview:** | This use case allows a registered user to interact with the bot by speaking, with the system converting the spoken input to text for display or further processing. |
| **Actors:** | User, LLM API, Speech to Text |
| **Pre condition:** | • The user has an existing account and is logged in. |

| | |
|---|---|
| | • A functional microphone is connected, detected, and permissions are granted.<br>• The speech-to-text (STT) API is available. |
| **Flow:** | Main (success) Flow:<br>1. **Choose Language:** The user selects the language for both speech input and text output.<br>2. **Record Voice:** The system verifies the presence of an audio input device and prompts the user for microphone access if necessary. The user records their voice input.<br>3. **Speech-to-Text Conversion:** The recorded audio is sent in small, parallelized chunks to the STT service. The STT service converts the audio to text, supporting multiple languages.<br>4. **Display Text Response:** The converted text is displayed to the user for confirmation or further processing. |
| | Alternate Flow:<br>*a) System Crash<br>10. The system halts normal processing and displays a dedicated error notification page to the user, indicating that a system error has occurred.<br>11. The error page provides instructions to refresh the page, try again later, or contact support with a reference code for troubleshooting.<br>12. The system logs detailed crash information (including context and error codes) for troubleshooting and recovery purposes.<br>Postconditions**:**<br>• The user is redirected to a safe error state and informed about the system crash.<br>• Any unsaved session data may be lost, and the user is advised to restart the session after the system is restored to normal operation.<br>**2a) Microphone Issues:**<br>1. If no microphone is detected, the system lists available audio devices and prompts the user to select one or update drivers if necessary.<br>**Postconditions:** The system does not proceed with speech-to-text conversion until a functional audio input device is confirmed.<br>**2b) Permission Denied:**<br>1. If the user denies microphone access, the system shows a “Microphone Access Denied” error message.<br>**Postconditions:** The system halts further voice input processing and remains in a safe state until permission is granted.<br>**3a) STT API Failure:**<br>1. If the STT service fails due to network issues or service downtime, the system displays an error message with retry options.<br>**Postconditions:** The system remains stable, and the user is prompted to reinitiate voice recording once the STT service is restored. |
| **Post Condition:** | The spoken input is successfully converted to text and displayed for the user. |

### *UC-8*

| | |
|---|---|
| **Use Case Number:** | UC-08 |
| **Use Case Name:** | Session History and Insights |

| | |
|---|---|
| **Overview:** | Allows users to view past interactions and mood trends through saved session logs. This feature helps users reflect on their progress and gain insights into their emotional patterns over time. Can be extremely useful to mental health support professionals to track their patient’s trends. |
| **Actors:** | User, Data Analytics Tool |
| **Pre condition:** | • User is logged into the system,<br>• The Data Analytics Tool is operational and properly connected to the system. |
| **Flow:** | Main (success) Flow:<br>1. Access Request: The user selects the "Session History and Insights" option from their dashboard.<br>2. Data Retrieval: The system retrieves the user's session logs and associated mood data from the database.<br>3. Data Processing: The system sends the retrieved data to the Data Analytics Tool, which processes the information to generate visualizations like charts, graphs etc, and summary trends.<br>4. Display Results: The system displays the processed session history and insights to the user.<br>5. User Interaction: The user can filter or sort the displayed data (e.g., by date range or mood category) to gain specific insights. |
| | Alternate Flow:<br>*a) System Crash<br>1. The system halts normal processing and displays a dedicated error notification page to the user, indicating that a system error has occurred.<br>2. The error page provides instructions to refresh the page, try again later, or contact support with a reference code for troubleshooting.<br>3. The system logs detailed crash information (including context and error codes) for troubleshooting and recovery purposes.<br>Postconditions**:**<br>• The user is redirected to a safe error state and informed about the system crash.<br>• Any unsaved session data may be lost, and the user is advised to restart the session after the system is restored to normal operation.<br>2a) No Session Data Available<br>1. The system detects that insufficient session logs exist for the user requesting to see this page.<br>2. The system displays a message informing the user that no session data is available<br>3. The system also displays a button that takes the user to a new session, where mood tracking can be done.<br>Postconditions:<br>• The user is notified that there is no available session data.<br>• The user is prompted to initiate new conversations and sessions.<br>3a) Data Analytics Tool Unavailable<br>1. The Data Analytics Tool fails to process the data or times out.<br>2. The system captures the error and transitions to the error handling routine for this alternate flow.<br>3. An error message is displayed to the user indicating that detailed insights are temporarily unavailable.<br>Postconditions**:**<br>The user is informed that the session insights could not be generated due to a temporary unavailability of the Data Analytics Tool. |

| | |
|---|---|
| **Post Condition:** | The user successfully views an interactive display of their session history and insights, enabling reflection on their emotional progress. |

### UC-9

| | |
|---|---|
| **Use Case Number:** | UC-09 |
| **Use Case Name:** | Feedback Mechanism |
| **Overview:** | Enables users to rate their experience and submit feedback about the system’s interactions. The collected feedback supports continuous improvement of the AI and overall user experience. |
| **Actors:** | User |
| **Pre condition:** | The user is logged into the system. |
| **Flow:** | Main (success) Flow:<br>1. Access Feedback Section: The user navigates to the "Feedback" section.<br>2. Input Feedback: The user selects a rating and optionally enters detailed comments in the provided text box.<br>3. Submit Feedback: The user submits the feedback by clicking the "Submit" button.<br>4. Confirmation Display: The system processes the input, stores the feedback, and displays a confirmation message to the user. |
| | Alternate Flow:<br>*a) System Crash<br>1. The system halts normal processing and displays a dedicated error notification page to the user, indicating that a system error has occurred.<br>2. The error page provides instructions to refresh the page, try again later, or contact support with a reference code for troubleshooting.<br>3. The system logs detailed crash information (including context and error codes) for troubleshooting and recovery purposes.<br>Postconditions:<br>• The user is redirected to a safe error state and informed about the system crash.<br>• Any unsaved feedback data may be lost, and the user is advised to restart the feedback submission process after the system is restored to normal operation.<br>2a) Incomplete or Invalid Feedback<br>1. The system detects that the submitted feedback is incomplete or contains invalid entries (e.g., no rating selected).<br>2. The system displays an error message prompting the user to complete all required fields.<br>3. The user corrects the feedback and resubmits the form.<br>Postconditions:<br>The user is notified that the feedback submission is incomplete or invalid.<br>4a) Feedback Submission Error<br>1. The system encounters an error while saving the feedback to the database.<br>2. An error message is displayed to the user indicating that the feedback submission failed and suggests trying again later.<br>Postconditions:<br>• The user is informed that the feedback could not be recorded due to a temporary issue<br>• The system logs the error for further investigation. |
| **Post Condition:** | The user successfully submits feedback and receives a confirmation message that their input has been recorded, contributing to the system's continuous improvement. |

### UC-10

| | |
|---|---|
| **Use Case Number:** | UC-10 |
| **Use Case Name:** | Toggle Dark Mode |
| **Overview:** | This use case describes how a user toggles the dark mode setting on the home page of the application. Dark mode provides an alternative UI theme to reduce eye strain and create a more comfortable viewing experience, which is necessarily important in a sensitive mental health environment. |
| **Actors:** | User |
| **Pre condition:** | The user is logged into the application and is on the home page.<br>The dark mode toggle button is visible and accessible at the top of the home page (on a navbar). |
| **Flow:** | Main (success) Flow:<br>1. Display of Home Page: The home page loads with the dark mode toggle button prominently displayed.<br>2. User Clicks Dark Mode Toggle: The user clicks the dark mode button to switch the UI theme.<br>3. System Processes Request: The system registers the toggle action and initiates the theme switch.<br>4. Theme Change Execution: The system applies the dark mode theme across the application interface.<br>5. Visual Confirmation: The user sees the UI updated to dark mode, confirming that the toggle was successful |
| | Alternate Flow:<br>• **AF1: Toggle from Dark Mode to Light Mode**<br>o **Condition:** The user clicks the dark mode button while the application is already in dark mode.<br>o **Flow:**<br>♣ The system registers the toggle action as a request to switch back to light mode.<br>♣ The system reverts the UI to the default light theme.<br>♣ The user’s preference is updated and persisted as "light mode."<br>o **Post Condition:** The application now displays the light theme, and the toggle reflects the change.<br>• **AF2: Theme Application Failure**<br>o **Condition:** The system encounters an error when attempting to apply the dark mode theme.<br>o **Flow:**<br>♣ The system displays an error notification indicating that the theme change could not be applied.<br>♣ The UI remains in the current mode (either dark or light).<br>♣ The system logs the error for further investigation.<br>o **Post Condition:** The user is informed of the issue, and no preference is updated until the error is resolved. |

| | |
|---|---|
| **Post Condition:** | • Upon successful completion, the user’s interface is updated to the selected theme (dark or light), and the chosen preference is saved for future sessions.<br>• The user experience is enhanced by offering a customizable, eye-friendly display that aligns with personal preferences and supports mental well-being. |

### UC-11

| | |
|---|---|
| **Use Case Number:** | UC-11 |
| **Use Case Name:** | User Account Management |
| **Overview:** | This use case describes how a logged-in user manages their account information within the AI-based mental health application. In addition to updating basic personal details, the user can modify their preferences—such as mood tracking settings (e.g., frequency of mood updates, notification preferences, and data sharing settings)—to tailor the application to their mental health needs. The process emphasizes data integrity, privacy, and user control over sensitive information. |
| **Actors:** | User |
| **Pre condition:** | The user is authenticated (logged in) and has an existing account in the system.<br>The account management interface is accessible from the home page.<br>All necessary security protocols and data protection measures are in place. |
| **Flow:** | Main (success) Flow:<br>1. **Access Account Management**<br>a. The user navigates to the “Account Management” section from the application’s home/dashboard.<br>2. **Display Current Account Details and Preferences**<br>a. The system retrieves and displays the user’s current account information (e.g., username ,email) along with configurable preferences, including mood tracking settings (e.g., mood tracking frequency, notification options, and data sharing preferences).<br>3. **Initiate Update Process**<br>a. The user selects an “Edit” option to update specific details such as contact information or mood tracking preferences.<br>4. **Present Editable Form**<br>a. The system displays an editable form pre-populated with the current account data and mood tracking settings.<br>5. **User Makes Changes**<br>a. The user updates one or more fields—such as changing email, updating password, or modifying mood tracking preferences (for instance, opting for daily or weekly mood reminders, enabling/disabling detailed mood analytics, or setting thresholds for notifications).<br>6. **Validate Input Data**<br>a. The system validates the entered data (e.g., checks for valid email format, password strength, and acceptable values for mood tracking frequency and notifications).<br>7. **Save Updated Account Details**<br>a. If validation succeeds, the system updates the account information in the secure database and logs the changes for auditing purposes.<br>8. **Confirm Update** |

| | |
|---|---|
| | a. The system provides a confirmation message to the user indicating that the account information and preferences have been successfully updated. |
| | Alternate Flow:<br>• **AF1: Invalid Data Entry**<br>o **Condition:** The user provides invalid or improperly formatted data (e.g., invalid email, or unacceptable mood tracking frequency).<br>o **Flow:**<br>♣ The system highlights the erroneous fields and displays context-specific error messages.<br>♣ The user corrects the data and resubmits the update.<br>o **Post Condition:** The update is not saved until all data meets validation requirements.<br>• **AF2: Update Cancellation**<br>o **Condition:** The user decides to cancel the update process.<br>o **Flow:**<br>♣ The user clicks a “Cancel” button or navigates away from the account management page.<br>♣ The system discards any unsaved changes and returns to the account overview.<br>o **Post Condition:** No changes are applied; the account remains as previously configured.<br>• **AF3: System Error or Database Failure**<br>o **Condition:** The system encounters an error (e.g., database connectivity issues) while saving the update.<br>o **Flow:**<br>♣ The system displays an error message advising the user to try again later.<br>♣ The system logs the error for further investigation.<br>o **Post Condition:** The account information remains unchanged until the error is resolved.<br>• **AF4: Unauthorized Sensitive Change**<br>o **Condition:** The user attempts to update sensitive fields (e.g., changing the password) without proper re-authentication.<br>o **Flow:**<br>♣ The system prompts the user to re-enter their current password or complete a secondary verification step.<br>♣ Upon successful re-authentication, the system proceeds with the update.<br>o **Post Condition:** Sensitive changes are applied only after confirming the user’s identity. |
| **Post Condition:** | • The user’s account information and mood tracking preferences are updated securely and are reflected in subsequent sessions.<br>• All modifications are logged for audit and security purposes, ensuring transparency and compliance with data protection and privacy guidelines. |

## UC-12

| | |
|---|---|
| **Use Case Number:** | UC-12 |
| **Use Case Name:** | |
| **Overview:** | |
| **Actors:** | |
| **Pre condition:** | |
| **Flow:** | Main (success) Flow: |
| | Alternate Flow: |
| **Post Condition:** | |

## UC-13

| | |
|---|---|
| **Use Case Number:** | UC-13 |
| **Use Case Name:** | Data Input Fields for better personalisation |
| **Overview:** | The system dynamically adapts to deliver a seamless personalized and localized experience. |
| **Actors:** | User |
| **Pre condition:** | • The system is up and running.<br>• The LLM API is operational and properly connected.<br>• The user navigates to the personalisation settings or data input page from their dashboard. |
| **Flow:** | Main (success) Flow:<br>1. The system displays various data input fields (e.g., preferred language, interests, location, personal information, relationship types, etc.) that are used to tailor the user experience.<br>2. The user fills in the input fields with the required details.<br>3. The user submits the information to the system.<br>4. The system sends the provided data to the database for processing and personalization logic for future chats.<br>5. The system displays a confirmation message indicating that the personalisation settings have been successfully updated. |
| | Alternate Flow:<br>*a) System Crash<br>1. The system halts normal processing and displays a dedicated error notification page to the user, indicating that a system error has occurred.<br>2. The error page provides instructions to refresh the page, try again later, or contact support with a reference code for troubleshooting.<br>3. The system logs detailed crash information (including context and error codes) for troubleshooting and recovery purposes.<br>Postconditions**:**<br>• The user is redirected to a safe error state and informed about the system crash.<br>• Any unsaved session data may be lost, and the user is advised to restart the session after the system is restored to normal operation.<br>2a) Invalid Input Data: The system detects that one or more input fields contain invalid data or are incomplete. |

| | |
|---|---|
| | 1. The system highlights the invalid fields and displays error messages with instructions for correction.<br>2. The user updates the information and resubmits the form.<br>Post Conditions:<br>• The user is clearly notified that the input data is invalid.<br>• The data update process remains incomplete until valid data is provided.<br>4a) Network error<br>1. Error message to try again later is displayed<br>2. User is redirected to dashboard<br>Postconditions:<br>Error displayed, user redirected.<br>4b) Database error<br>1. Error message shown for cannot connect to database, try again later.<br>2. User is redirected to dashboard<br>Postconditions:<br>Error displayed, user redirected. |
| **Post Condition:** | The system successfully applies the user's input data to update the personalisation settings, delivering a localized and tailored user experience. |