**Software Requirements Specification (SRS) Document**

Alethiosphere, Team 21, Anshul Krishnadas Bhagwat, Ishaan Romil, Manasi Mundada, Raaghav Doshi, Saketh Krishna Ayyagari

# Brief problem statement

AlethioSphere is an AI Avatar Video-Chatbot Application. Journalling is an effective, personal, and low-cost coping mechanism for stress, anxiety, and other mental health issues. Having an interactive AI video avatar to talk to can make journalling much easier by doing away with the format and formality of writing. For those who do not have access to adequate mental health support, this can serve as an intermediate and accessible mental health partner to organise one’s thoughts, and review past entries. The illusion of an understanding person to talk to is also somewhat soothing and calming.

An AI-driven video avatar that serves as a 24/7 mental health companion will be empathetic, interactive, and non-judgmental, helping students organize their thoughts, eliminate overthinking, and address illogical doubts. The avatar will not replace professional therapy but will act as a supportive companion that students can turn to whenever they need to talk or organize their thoughts.

# System requirements

UI/UX: Frontend: React based framework (most likely NextJs) with one or many of Material-UI, Tailwind CSS, Bootstrap CSS for frontend styling. ExpressJs framework for frontend to backend communication.

Backend:

Node.js with Express.js for API development, MongoDB for database storage, JWT-based authentication, RESTful API endpoints for user management, journal entries, and avatar customization, AES-256 for encrypting journal entries and sensitive data before storage.

Machine Learning:

*Output/Avatar*: Google Text-to-Speech (gTTS), Live2D to animate rigged characters, Rhubarb Lip Sync, Three.js as a web-based game engine, text2emotion (in future releases) for realism

# Users profile

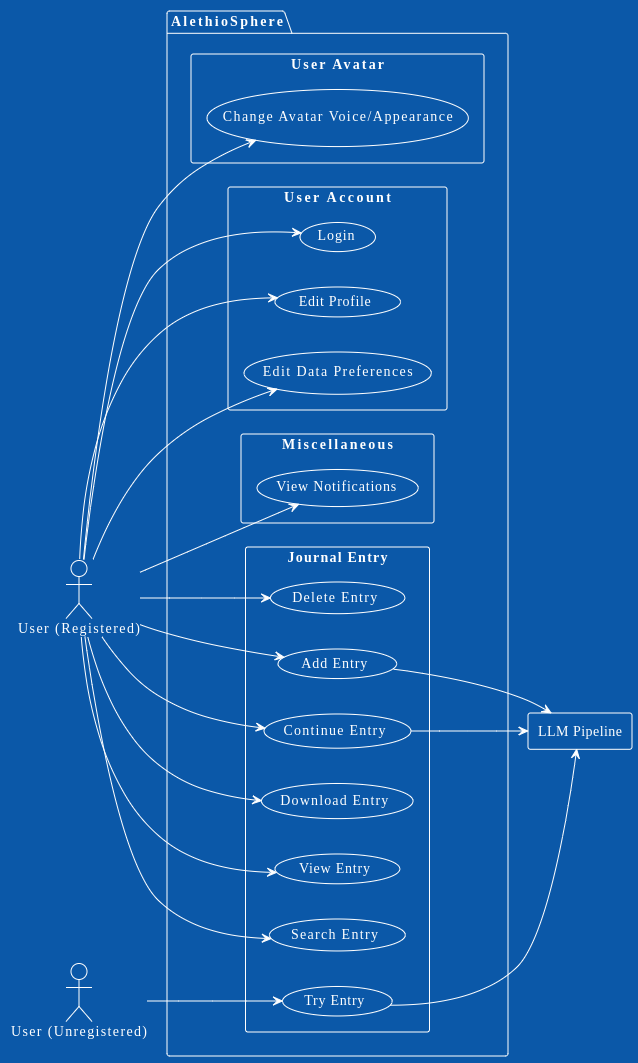
College students (ages 18-24) who experience stress related to exams, backlogs, assignments, financial pressures, and relationship issues. These students often struggle with overthinking, external validation, and the fear of being judged by peers. They find it difficult to open up about their feelings and seek help.

They are often adept with technology, although our application will be user friendly and not highly technical.

# Feature requirements (described using use cases)

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **User Case Name** | **Description** | **Release** |
| 1. | Delete Entry | Allows the user to delete a previously recorded journal entry, including text-input and speech-input text-transcripts | R1 |
| 2. | Add Entry (Start Video Chat + End Video Chat) | Allows the user to create a new journal entry by starting and stopping a video chat recording which then generates a text transcript of the video input and of the text input | R1 |
| 3. | Continue Entry (Add Entry with Context from previous) | Allows the user to start a new journal entry with the context (transcript) of a previous journal entry | R1 |
| 4. | Login | Allows users to access their accounts using authentication credentials or create an account by registering | R1 |
| 5. | Download Entry | Provides an option to download a transcript of a journal entry for offline use | R1 |
| 6. | Enter Text (During Video Chat) | Allows the user to give text-input and receive text-output from the model | R1 |
| 7. | View Entry | Enables users to view their previously recorded video journal entries | R1 |
| 8. | Search Entry | Allows users to search for specific journal entries based on keywords or filters | R1 |
| 9. | View Notifications | Displays system or user-generated notifications related to the journal entries | R1 |
| 10. | Change Avatar Voice/ Appearance | Provides options to modify the visual and auditory representation of the user’s avatar | R1 |
| 11. | Edit Profile | Allows users to update their personal information, such as name, email, and preferences | R1 |
| 12. | Edit Data Preferences | Allows users to configure data-sharing, privacy, and storage settings | R1 |
| 13. | Try Entry | Allows unregistered users to try out the application without storage capabilities | R1 |

# Use case diagram



# Use case description

|  |  |
| --- | --- |
| **Use Case Number:** | 1 |
| **Use Case Name:** | Delete Entry |
| **Overview:** | Allows the user to delete a previously recorded journal entry, including text-input and speech-input text transcripts. |
| **Actors:** | User |
| **Pre condition:** | 1. The user is logged into their account. 2. At least one journal entry exists in the user's journal. |
| **Main (Success) Flow:** | 1. The user navigates to the list of journal entries. 2. The user selects a specific journal entry to delete. 3. The system prompts the user to confirm the deletion. 4. The user confirms the deletion. 5. The system deletes the selected journal entry and updates the journal entry list. 6. The system displays a success message confirming the deletion. |
| **Alternate Flow** | (4a): The user cancels the deletion.   1. The journal entry remains unchanged. 2. The system returns to the journal entry list. |
| **Post**  **Condition:** | The selected journal entry is permanently deleted from the system. |

|  |  |
| --- | --- |
| **Use Case Number:** | 2 |
| **Use Case Name:** | Add Entry (Start Video Chat + End Video Chat) |
| **Overview:** | Allows the user to create a new journal entry by starting and stopping a video chat recording, which generates a text transcript of the video input and text input. |
| **Actors:** | User |
| **Pre condition:** | 1. The user is logged into their account. 2. The device has a functioning camera and microphone. |
| **Main (Success) Flow:** | 1. The user selects the option to create a new journal entry. 2. The system starts recording the video chat. 3. The user speaks or interacts (text) during the video chat. 4. The system responds by text or video as per the user’s input. 5. The user stops the recording. 6. The system processes the video chat and generates a text transcript. 7. The system saves the transcript as a new journal entry. 8. The system displays a success message confirming the creation and saving of the entry. |
| **Alternate Flow** | (5a): The user cancels the before stopping it.   1. The system prompts to confirm the cancellation 2. The user confirms cancellation. (If they do not confirm then the main flow continues.) 3. The recording is discarded, and no journal entry is created. |
| **Post Condition:** | A new journal entry, including the video chat transcript, is saved in the user's journal. |

|  |  |
| --- | --- |
| **Use Case Number:** | 3 |
| **Use Case Name:** | Continue Entry (Add Entry with Context from Previous Entry) |
| **Overview:** | Allows the user to start a new journal entry with the context (transcript) of a previous journal entry. |
| **Actors:** | User |
| **Pre condition:** | 1. The user is logged into their account. 2. At least one previous journal entry exists. |
| **Main (Success) Flow:** | 1. The user selects the option to create a new journal entry from a previous entry. 2. The system prompts the user to choose whether to continue from a previous entry. 3. The user selects 'Continue from Previous Entry.' 4. The system displays a list of previous journal entries. 5. The user selects a previous journal entry. 6. The system loads the transcript of the selected entry as context. 7. The user continues the journal entry. 8. The system saves the transcript as a new entry and appends the transcript of the previous one. 9. The system displays a success message confirming the creation of the entry. |
| **Alternate Flow** | (7a): The user cancels the before stopping it.   1. The system prompts to confirm the cancellation 2. The user confirms cancellation. (If they do not confirm then the main flow continues.) 3. The recording is discarded, and no journal entry is created.   (3a) The user selects ‘Start New Entry’   1. The system continues. (As per use case 2) |
| **Post Condition:** | A new journal entry is created with the context (transcript) of the selected previous entry. |

|  |  |
| --- | --- |
| **Use Case Number:** | 4 |
| **Use Case Name:** | Login |
| **Overview:** | Allows users to access their accounts using authentication credentials or create an account by registering. |
| **Actors:** | User |
| **Pre condition:** | The user has access to the login/registration page.    For login, the user has an existing account. |
| **Main (Success) Flow:** | 1. The user navigates to the login page. 2. The user enters their authentication credentials (username and password). 3. The system validates the credentials. 4. The system grants access to the user's account. 5. The system redirects the user to their journal dashboard. |
| **Alternate Flow** | (3a): The user does not have an account and selects 'Register.'   1. The system redirects the user to the registration page.  2. The user enters their registration details (e.g., name, email, password).  3. The system validates the registration details.  4. The system creates a new account and logs the user in.  5. The system redirects the user to their journal dashboard.    (3b): The user enters invalid credentials.   1. The system prompts to try again 2. If successfully entered after 3 tries', continue with main flow step 5 3. If unsuccessfully entering the credentials for more than 3 times (with appropriate warnings), the system asks to reset password through alternate means.   Post Condition: The system displays an error message and prompts the user to re-enter credentials. |
| **Post Condition:** | The user is logged into their account and can access their journal entries. |

|  |  |
| --- | --- |
| **Use Case Number:** | 5 |
| **Use Case Name:** | Download Entry |
| **Overview:** | Provides an option to download a transcript of a journal entry for offline use. |
| **Actors:** | User |
| **Pre condition:** | 1. The user is logged into their account. 2. At least one journal entry exists in the user's journal. |
| **Main (Success) Flow:** | 1. The user navigates to the list of journal entries. 2. The user selects a specific journal entry to download. 3. The system generates a downloadable transcript of the selected entry. 4. The user confirms the download. 5. The system provides the transcript in a downloadable format (e.g., PDF, TXT). 6. The user saves the transcript to their device. |
| **Alternate Flow** | (4a): The user cancels the download.   Post Condition: The transcript is not downloaded, and the system returns to the journal entry list. |
| **Post Condition:** | The selected journal entry transcript is downloaded and saved on the user's device. |

|  |  |
| --- | --- |
| **Use Case Number:** | 6 |
| **Use Case Name:** | Enter Text (During Video Chat) |
| **Overview:** | Allows the user to give text-input and receive text-output from the model during a video chat. |
| **Actors:** | User |
| **Pre condition:** | 1. The user is logged into their account. 2. The user is in an active video chat session. |
| **Main (Success) Flow:** | 1. The user selects the option to enter text during the video chat. 2. The system displays a text input field. 3. The user types their input and submits it. 4. The system processes the text input. 5. The system generates output, which is then converted to audio and presented via the avatar. 6. The system saves the text input and a transcription of the avatar’s output as part of the journal entry. |
| **Alternate Flow** | (3a): The user cancels the text input.   Post Condition: No text input or output is saved, and the video chat continues. |
| **Post Condition:** | The text input and output are saved as part of the journal entry. |

|  |  |
| --- | --- |
| **Use Case Number:** | 7 |
| **Use Case Name:** | View Entry |
| **Overview:** | Enables users to view their previously recorded video journal entries. |
| **Actors:** | User |
| **Pre condition:** | 1. The user is logged into their account. 2. At least one journal entry exists in the user's journal. |
| **Main (Success) Flow:** | 1. The user navigates to the list of journal entries. 2. The user selects a specific journal entry to view. 3. The system loads the transcript of the selected entry. 4. The user views the transcript. 5. The user closes the entry and returns to the journal entry list. |
| **Alternate Flow** | (3a): The selected entry fails to load.  Post Condition: The user closes the entry and returns to the journal entry list without having viewed the selected journal entry. |
| **Post Condition:** | The user successfully views the selected journal entry. |

|  |  |
| --- | --- |
| **Use Case Number:** | 8 |
| **Use Case Name:** | Search Entry |
| **Overview:** | Allows users to search for specific journal entries based on keywords or filters. |
| **Actors:** | User |
| **Pre condition:** | 1. The user is logged into their account. 2. At least one journal entry exists in the user's journal. |
| **Main (Success) Flow:** | 1. The user navigates to the search feature. 2. The user enters keywords or applies filters (e.g., date, tags). 3. The system processes the search query. 4. The system displays a list of matching journal entries. 5. The user selects an entry from the search results to view. |
| **Alternate Flow** | (3a): No entries match the search query.   Post Condition: The system displays a message indicating no results were found. |
| **Post Condition:** | The user successfully finds and views the desired journal entry. |

|  |  |
| --- | --- |
| **Use Case Number:** | 9 |
| **Use Case Name:** | View Notifications |
| **Overview:** | Displays system or user-generated notifications related to the journal entries. |
| **Actors:** | User |
| **Pre condition:** | 1. The user is logged into their account. 2. Notifications exist in the user's account. |
| **Main (Success) Flow:** | 1. The user navigates to the notifications section. 2. The system displays a list of notifications (e.g., reminders, updates, or alerts). 3. The user selects a notification to view details. 4. The system displays the details of the selected notification. 5. The user marks the notification as read or dismisses it. |
| **Alternate Flow** | (3a): No notifications are available.   Post Condition: The system displays a message indicating no notifications are available. |
| **Post Condition:** | The user views and interacts with the notifications. |

|  |  |
| --- | --- |
| **Use Case Number:** | 10 |
| **Use Case Name:** | Change Avatar Voice/Appearance |
| **Overview:** | Provides options to modify the visual and auditory representation of the user’s avatar. |
| **Actors:** | User |
| **Pre condition:** | 1. The user is logged into their account. 2. The user has an existing avatar. |
| **Main (Success) Flow:** | 1. The user navigates to the avatar settings. 2. The user selects the option to change the avatar's voice or appearance. 3. The system displays available voice and appearance options. 4. The user selects a new voice or appearance. 5. The system updates the avatar with the selected changes. 6. The system confirms the changes and displays the updated avatar. |
| **Alternate Flow** | (Step 3): The user cancels the changes.   Post Condition: The avatar remains unchanged. |
| **Post Condition:** | The avatar's voice and/or appearance is updated. |

|  |  |
| --- | --- |
| **Use Case Number:** | 11 |
| **Use Case Name:** | Edit Profile |
| **Overview:** | Allows users to update their personal information, such as name, email, and preferences. |
| **Actors:** | User |
| **Pre condition:** | The user is logged into their account. |
| **Main (Success) Flow:** | 1. The user navigates to the profile settings. 2. The user selects the option to edit their profile. 3. The system displays the current profile information. 4. The user updates the desired fields (e.g., name, email, preferences). 5. The user saves the changes. 6. The system validates and updates the profile information. 7. The system confirms the changes and displays the updated profile. |
| **Alternate Flow** | (5a): The user cancels the changes.   1. System prompts confirmation to cancel edits.    1. If yes then cancel the change and go back to profile page.    2. If no then stay on the editing pages with edits intact.   Post Condition: The profile remains unchanged. |
| **Post Condition:** | The user's profile information is updated. |

|  |  |
| --- | --- |
| **Use Case Number:** | 12 |
| **Use Case Name:** | Edit Data Preferences |
| **Overview:** | Allows users to configure data-sharing, privacy, and storage settings. |
| **Actors:** | User |
| **Pre condition:** | The user is logged into their account. |
| **Main (Success) Flow:** | 1. The user navigates to the data preferences in the settings menu.  2. The user selects the option to edit data preferences.  3. The system displays the current data-sharing, privacy, and storage settings.  4. The user updates the desired settings.  5. The user saves the changes.  6. The system validates and updates the data preferences.  7. The system confirms the changes and displays the updated settings. |
| **Alternate Flow** | (5a): The user cancels the changes.   1. Asks for confirmation of cancellation 2. If yes take them back to the settings page, If no, take them back to the editing section with all the preferences saved.   Post Condition: The data preferences remain unchanged. |
| **Post Condition:** | The user's data preferences are updated. |

|  |  |
| --- | --- |
| **Use Case Number:** | 13 |
| **Use Case Name:** | Try Entry |
| **Overview:** | Allows unregistered users to try out the application without storage capabilities without having to sign up or login. |
| **Actors:** | Unregistered User |
| **Pre condition:** | The user is not logged into an account. |
| **Main (Success) Flow:** | 1. The user selects the option to try the application. 2. The system provides a limited version of the application with basic functionality. 3. The user creates a journal entry (with text and/or video chat). 4. The system processes the entry but does not save it. 5. The user interacts with the application (e.g., views the transcript). 6. The user exits the trial mode. |
| **Alternate Flow** | (3a): The user attempts to save the entry.   1. After clicking on the save button, they will download their interaction with the avatar into their local storage. 2. Takes them back to the testing interface.   Post Condition: The system prompts the user to create an account to save the entry. |
| **Post Condition:** | The user experiences the application without creating an account or saving data. |