**Document Control**

* **Document Title:** Requirements Specification: Chat History Management Module
* **Project:** AI Chatbot (Web & Mobile)
* **Module:** Chat History Management
* **Author:** Trung Ngo, PM
* **Version:** 1.0
* **Date:** 2025-04-10
* **Status:** Draft

**1. Introduction**

This document details the User Requirements (UR) and System Requirements (SR) for the Chat History Management module of the AI Chatbot application. This module enables Registered Users to view, search, and manage their past interactions with AI Characters across both web (React/Node.js) and mobile (Flutter) platforms. These requirements serve as the definitive source of truth for development, testing, and product management related to chat history functionality.

**2. Definitions and Acronyms**

* **AI:** Artificial Intelligence
* **API:** Application Programming Interface
* **BA:** Business Analyst
* **Character:** An AI entity within the application with which the User interacts.
* **Chat History:** The collection of past Chat Sessions recorded for a specific Registered User.
* **Chat Session:** A single, distinct conversation thread between a Registered User and a Character, potentially spanning multiple interactions over time until explicitly ended or implicitly timed out (details TBD in design).
* **DB:** Database
* **JWT:** JSON Web Token (Example authentication mechanism)
* **Registered User:** An individual who has successfully created an account and logged into the application.
* **SR-F:** Functional System Requirement
* **SR-NF:** Non-Functional System Requirement
* **UI:** User Interface
* **UR:** User Requirement

**3. User Requirements (UR)**

This section outlines the specific goals and tasks Registered Users need to accomplish related to their chat history. These requirements are written from the user's perspective.

*(Applying Checklist 0a: User Requirements)*

| ID | User Requirement Statement | Rationale / Benefit | Testable |
| --- | --- | --- | --- |
| UR-CHM-001 | As a Registered User, I want to view a list of my past chat sessions | so that I can easily see which Characters I've talked to and when. | Yes |
| UR-CHM-002 | As a Registered User, I want the list of past chat sessions to show identifying information for each session (e.g., Character, date) | so that I can quickly distinguish between different conversations without opening them. | Yes |
| UR-CHM-003 | As a Registered User, I want to open and view the full message history of a specific past chat session | so that I can review the details and context of that specific conversation. | Yes |
| UR-CHM-004 | As a Registered User, I want to easily navigate through long chat histories within a selected session | so that I can efficiently review lengthy conversations without excessive scrolling. | Yes |
| UR-CHM-005 | As a Registered User, I want to search my entire chat history using keywords | so that I can quickly locate specific information or past conversations by topic. | Yes |
| UR-CHM-006 | As a Registered User, I want to see which chat sessions match my keyword search | so that I can easily identify and select the relevant conversation(s). | Yes |
| UR-CHM-007 | As a Registered User, I want to delete a specific chat session from my history | so that I can remove conversations I no longer need or wish to keep for privacy. | Yes |
| UR-CHM-008 | As a Registered User, I want to be asked for confirmation before deleting a chat session | so that I don't accidentally delete important conversations. | Yes |
| UR-CHM-009 | As a Registered User, I want the option to delete my *entire* chat history | so that I can clear all my past conversation data if desired for privacy or reset. | Yes |
| UR-CHM-010 | As a Registered User, I want to be asked for explicit, strong confirmation before deleting my *entire* chat history | so that I fully understand the irreversible consequences of this action. | Yes |
| UR-CHM-011 | As a Registered User, I want to access my chat history consistently across both web and mobile platforms | so that my conversation history is synchronized regardless of the device I use. | Yes |
| UR-CHM-012 | As a Registered User, I want to see a clear indication if I have no chat history yet | so that I understand the section isn't broken and know I haven't chatted yet. | Yes |

**4. System Requirements (SR)**

This section details the specific functional capabilities the system must provide and the non-functional qualities it must possess to meet the User Requirements and overall business goals for Chat History Management.

*(Applying Checklist 0b: System Requirements)*

**4.1. Functional Requirements (SR-F)**

These requirements describe *what the system must do*.

| ID | System Requirement Statement | Related UR | Verifiable/Testable |
| --- | --- | --- | --- |
| SR-F-CHM-001 | The system **shall** securely store all messages exchanged between a Registered User and a Character within a defined Chat Session. | UR-CHM-001 | Yes |
| SR-F-CHM-002 | The system **shall** associate stored messages with the correct Registered User ID, Character ID, and a unique Chat Session ID. | UR-CHM-001 | Yes |
| SR-F-CHM-003 | The system **shall** store a timestamp (UTC) for each message within a Chat Session. | UR-CHM-001 | Yes |
| SR-F-CHM-004 | The system **shall** provide an API endpoint to retrieve a paginated list of Chat Sessions for the authenticated Registered User. | UR-CHM-001 | Yes |
| SR-F-CHM-005 | The system **shall** order the retrieved list of Chat Sessions chronologically, with the most recently active session appearing first by default. | UR-CHM-001 | Yes |
| SR-F-CHM-006 | The system **shall** include the following metadata for each session in the list view API response: Chat Session ID, Character ID, Character Name, Character Avatar/Image URL, Timestamp of the last message, and a short preview snippet of the last message content. | UR-CHM-002 | Yes |
| SR-F-CHM-007 | The system **shall** provide an API endpoint to retrieve the complete message history (User prompts, Character responses, timestamps) for a specified Chat Session ID, belonging to the authenticated Registered User. | UR-CHM-003 | Yes |
| SR-F-CHM-008 | The system **shall** implement pagination or equivalent mechanism (e.g., infinite scroll support) for retrieving messages within a single, potentially long, Chat Session via the API. | UR-CHM-004 | Yes |
| SR-F-CHM-009 | The system **shall** provide an API endpoint to search across all Chat Sessions belonging to the authenticated Registered User based on provided keywords. | UR-CHM-005 | Yes |
| SR-F-CHM-010 | The system **shall** ensure the search function queries the content of both User messages and Character responses within the user's history. | UR-CHM-005 | Yes |
| SR-F-CHM-011 | The system **shall** return a list of Chat Sessions that contain matches for the search keywords, including sufficient metadata to identify the sessions (as per SR-F-CHM-006). | UR-CHM-006 | Yes |
| SR-F-CHM-012 | The system **shall** provide an API endpoint to permanently delete a specific Chat Session identified by its ID, belonging to the authenticated Registered User. | UR-CHM-007 | Yes |
| SR-F-CHM-013 | The system **shall** implement a confirmation step within the UI (Web and Mobile) before invoking the single session deletion API call. | UR-CHM-008 | Yes |
| SR-F-CHM-014 | The system **shall** provide an API endpoint to permanently delete *all* Chat Sessions associated with the authenticated Registered User. | UR-CHM-009 | Yes |
| SR-F-CHM-015 | The system **shall** implement a distinct, high-friction confirmation step within the UI (e.g., requiring typed confirmation) before invoking the delete-all-history API call. | UR-CHM-010 | Yes |
| SR-F-CHM-016 | The system **shall** ensure that deleting a Chat Session or the entire history permanently removes the associated message data from primary accessible storage. | UR-CHM-007, 009 | Yes |
| SR-F-CHM-017 | The system **shall** restrict access to any chat history data (list retrieval, session retrieval, search, deletion) strictly to the Registered User who owns that history. | UR-CHM-001-011 | Yes |
| SR-F-CHM-018 | The system **shall** ensure chat history data is synchronized and accessible consistently via both Web (React/Node.js) and Mobile (Flutter) interfaces for the same Registered User. | UR-CHM-011 | Yes |
| SR-F-CHM-019 | The system **shall** return an appropriate state or empty list (distinct from an error) via the API when a Registered User requests their chat history list but none exists. | UR-CHM-012 | Yes |
| SR-F-CHM-020 | The system **shall** handle potential errors gracefully during history retrieval, search, or deletion (e.g., network issues, database errors) and provide informative error codes/messages via the API. | N/A | Yes |

**4.2. Non-Functional Requirements (SR-NF)**

These requirements describe *qualities and constraints* of the system.

| ID | System Requirement Statement | Category | Measurable/Testable |
| --- | --- | --- | --- |
| SR-NF-CHM-001 | The API endpoint for retrieving the first page (e.g., 20 sessions) of the chat history list **shall** respond within 1.5 seconds under 95th percentile load. | Performance | Yes (Measure API response time under simulated load) |
| SR-NF-CHM-002 | The API endpoint for retrieving a full chat session (up to 200 messages) **shall** respond within 2 seconds under 95th percentile load. | Performance | Yes (Measure API response time under simulated load) |
| SR-NF-CHM-003 | The API endpoint for chat history search **shall** respond within 4 seconds for users with up to 20,000 stored messages under 95th percentile load. | Performance | Yes (Measure API response time with large test data sets under load) |
| SR-NF-CHM-004 | Deletion of a single Chat Session **shall** be logically reflected in the user's history view within 1 second after successful API confirmation. | Performance | Yes (Measure time from API success response to UI update/data unavailability) |
| SR-NF-CHM-005 | All chat history data stored at rest in the primary database **shall** be encrypted using AES-256 encryption or equivalent industry standard. | Security | Yes (Verify database configuration and encryption settings) |
| SR-NF-CHM-006 | All API endpoints related to chat history **shall** require valid authentication (e.g., JWT Bearer token) and authorization confirming the user's identity and ownership of the requested data. | Security | Yes (Test endpoints without/with invalid/with valid tokens from different users) |
| SR-NF-CHM-007 | The chat history storage mechanism (database schema, infrastructure) **shall** be designed to scale horizontally to support at least 1 million concurrent users and 1 billion stored messages. | Scalability | Yes (Architectural review, load testing against projected scale, database scaling plan verification) |
| SR-NF-CHM-008 | The system **shall** maintain audit logs for chat history deletion events (single session and bulk delete), recording user ID, timestamp, and type of deletion. | Auditing | Yes (Verify log creation and content upon deletion actions) |
| SR-NF-CHM-009 | The chat history feature UI on both web and mobile platforms **shall** adhere to the established application-wide style guide and usability standards. | Usability | Yes (UI/UX review, usability testing) |
| SR-NF-CHM-010 | Error messages presented to the user related to chat history operations (e.g., load failure, delete failure) **shall** be clear, concise, and avoid technical jargon. | Usability | Yes (Review error message content during testing) |
| SR-NF-CHM-011 | The system **shall** ensure data consistency for chat history across geographically distributed data centers, if applicable (details TBD based on infrastructure). | Reliability | Yes (Test data replication delays and consistency checks in a distributed environment, if applicable) |

**End of Requirements Specification: Chat History Management Module (v1.0)**