Software Design Specification

SentiCare

Project Code: EWZ473254

Internal Advisor: Dr. Saad Razzaq **Saad...**

External Advisor: Dr. Mohsin Atta

Project Manager: Dr. Muhammad Ilyas

Project Team: Esha Gulzar (BSCS51F22S047)
Wajeeha Ijaz (BSCS51F22S032)
Muhammad Zain Ul Abidin (BSCS51F21S054)

Submission Date: 15/12/2025

_____

Project Manager's Signature

Document Information

Category	Information
Customer	Department of Computer Science, University of Sargodha
Project	SentiCare
Document	Software Design Specification
Document Version	1.0
Identifier	<PGBH01-2025-DS>
Status	Draft
Author(s)	Esha Gulzar (BSCS51F22S047) Wajeeha Ijaz (BSCS51F22S032) Muhammad Zain ul Abidin (BSCS51F21S054)
Approver(s)	Dr. Muhammad Ilyas
Issue Date	Sept. 15, 2025
Document Location	
Distribution	1. Dr. Saad Razzaq 2. Dr. Muhammad Ilyas 3. Dr. Mohsin Atta

Definition of Terms, Acronyms and Abbreviations

Term	Description
STT/TTS	Speech-to-text/ Text-to-Speech
RL	Reinforcement Learning
NLU	Natural Language Understanding
MFCC’s	Mel-Frequency Cepstral Coefficients
BERT	Bidirectional Encoder Representations
PPO	Proximal Policy Optimization
API	Application Programming Interface
CBT	Cognitive Behavioral Therapy

Table of Contents

1. Introduction

1.1 Purpose of Document

This Software Requirements Specification document will outline SentiCare, a bilingual chatbot for accessible psychological assistance to users. Developers, project managers and other stakeholders participating in the platform's development will be the target audience for this document. The design follows an **Object-Oriented Design (OOD) methodology**, using UML diagrams such as class diagrams, component diagrams, sequence diagrams, and ER models.

1.2 Project Overview

SentiCare is an AI-powered bilingual (English/Urdu) mental health chatbot designed to provide accessible, stigma-free emotional support through intelligent conversation. The platform will operate as a 24/7 accessible system that preserves user privacy while delivering personalized mental health assistance.
The interaction flow will begin when users engage through voice input via a web-based interface. The system will capture raw audio and perform parallel dual-channel analysis: (1) Voice Biomarker extraction will process the raw audio to extract acoustic features using MFCC (Mel-Frequency Cepstral Coefficients), analyzing speech patterns, pitch variations, and tonal characteristics to detect underlying emotional states, (2) simultaneously, the same raw audio will be converted to text using STT (Speech-to-Text) technology for linguistic analysis.

The system will conduct structured mental health assessments using dialogue-based questionnaires. The questionnaires will guide users through self-evaluation protocols. The transcribed text will process through a Natural Language Understanding (NLU) pipeline powered by mBERT (multilingual BERT), extracting user intent, sentiment polarity, and language identification. Both the extracted voice biomarker features and NLU outputs (intent and sentiment) will converge at the Emotion Analyzer, which performs multimodal fusion to generate comprehensive emotional assessments including specific emotion labels and quantified intensity scores. These integrated outputs will be sent to the Conversation Manager, which will manage conversation history, user context, and emotional trajectories indexed by conversation ID and user ID.

Based on the detected emotional state and user intent, the CBT Template Manager will select the most appropriate intervention from a library of 30-45 structured Cognitive Behavioral Therapy (CBT) templates. A Reinforcement Learning mechanism using Proximal Policy Optimization (PPO) will optimize template selection by learning from conversation outcomes and user engagement patterns. The RL Optimizer will continuously refine response strategies to maximize therapeutic effectiveness, creating a feedback loop that personalizes interventions over time. The outputs will include evidence-based therapeutic techniques, coping strategies, and behavioral exercises tailored to the user's emotional state.

Finally, generated responses from the Response Generator will convert to natural-sounding speech through TTS (Text-to-Speech) technology. The system will enable users to receive empathetic, contextually appropriate support in their preferred language (English/Urdu) through an intuitive and voice-driven conversational experience.

1.3 Scope

Using SentiCare, Users will be able to easily assess their mental health via dialogue-based assessment. System will make use of AI-powered features and provide responses like therapies and exercises. The system will use Cognitive Behavioral therapy (CBT) techniques and user's emotional progress over time to provide responses.
It will not replace professional psychiatrists, prescribe medications or clinical diagnoses. It will be used as a proactive approach just before appointment. It will not share user's data with third parties. It will not provide medical advice beyond emotional support.

2. Design Considerations

2.1 Assumptions and Dependencies

The design of SentiCare assumes that all core components defined in the SRS (such as STT, TTS, mBERT-based NLU, voice biomarker extraction module, multimodal Emotion Analyzer, and PPO-based RL optimization) will remain stable and compatible with the chosen technology stack. It is also assumed that required third-party tools (Python libraries, ML frameworks, cloud hosting, and database systems) will continue supporting the APIs and performance levels. Such tools are needed for real-time processing.

SentiCare depends on the availability of a reliable microphone interface in the browser for capturing raw audio, sufficient bandwidth for transmitting audio data, and consistent network connectivity for parallel STT and voice biomarker processing as well as model inference. Additionally, the modular design assumes that future model upgrades (e.g., switching STT engine, retraining NLU models, or updating emotion fusion algorithms) can be integrated without major architectural changes. These assumptions guide the system structure and ensure that components remain loosely coupled, scalable, and maintainable.

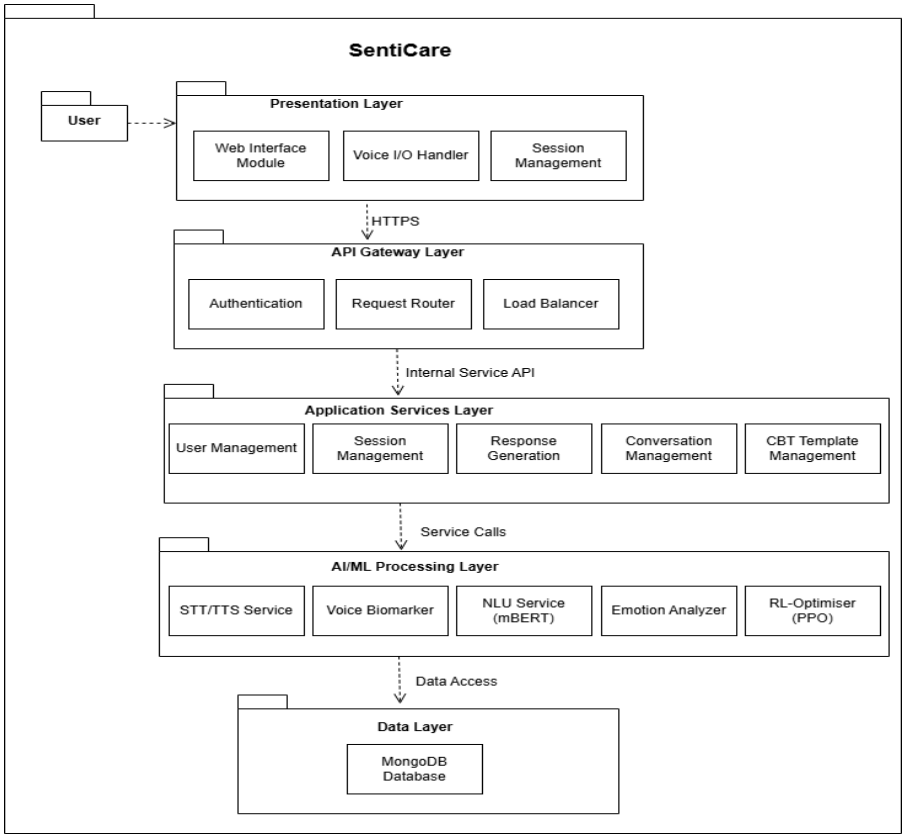
2.2 Risks and Volatile Areas

Risk	Impact	Mitigation
Changes in ML model performance	Incorrect emotion detection	Allow model via modular design
STT accuracy variability	Wrong transcript →wrong therapy	Use confidence threshold
Biomarker misclassification	Misleading emotional analysis	Ensemble scores from text + audio
RL policy instability	Wrong template selection	Reward normalization & PPO clipping
Data privacy issues	Legal exposure	End-to-end encryption + anonymization

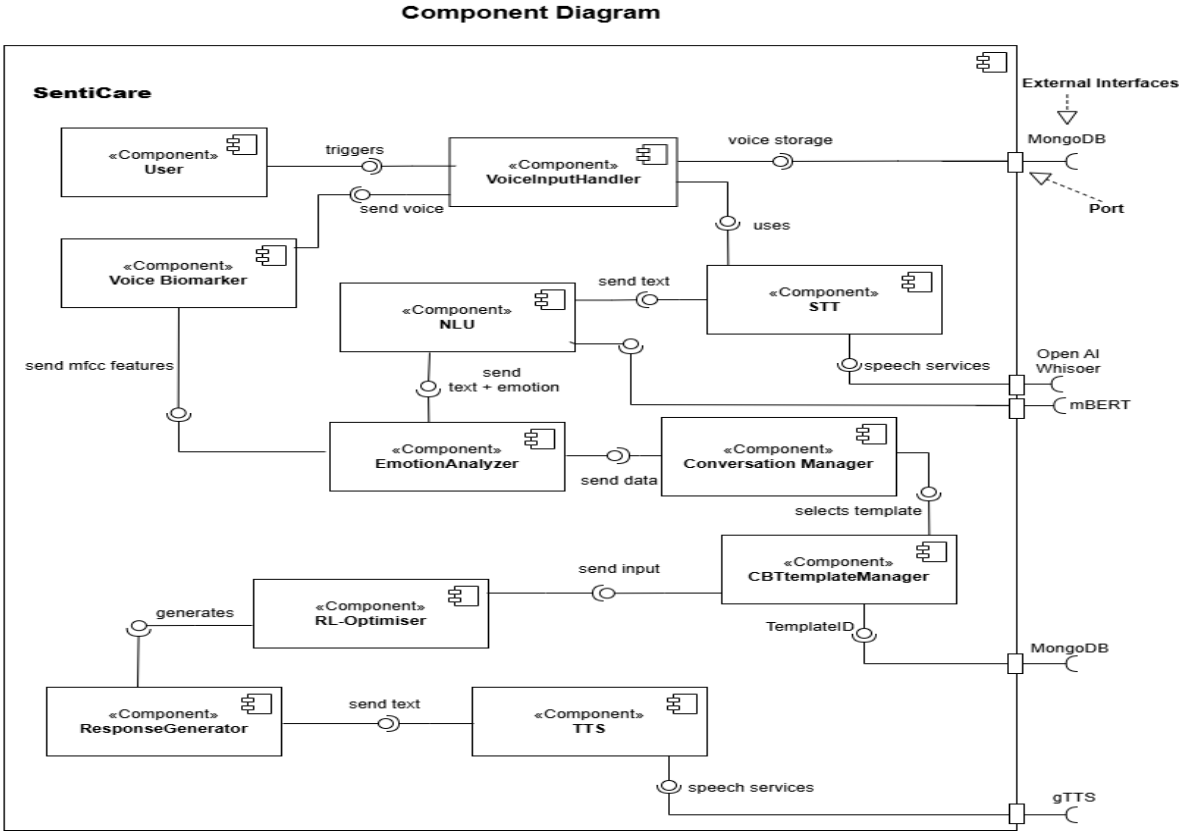
3 System Architecture

3.1 System Level Architecture

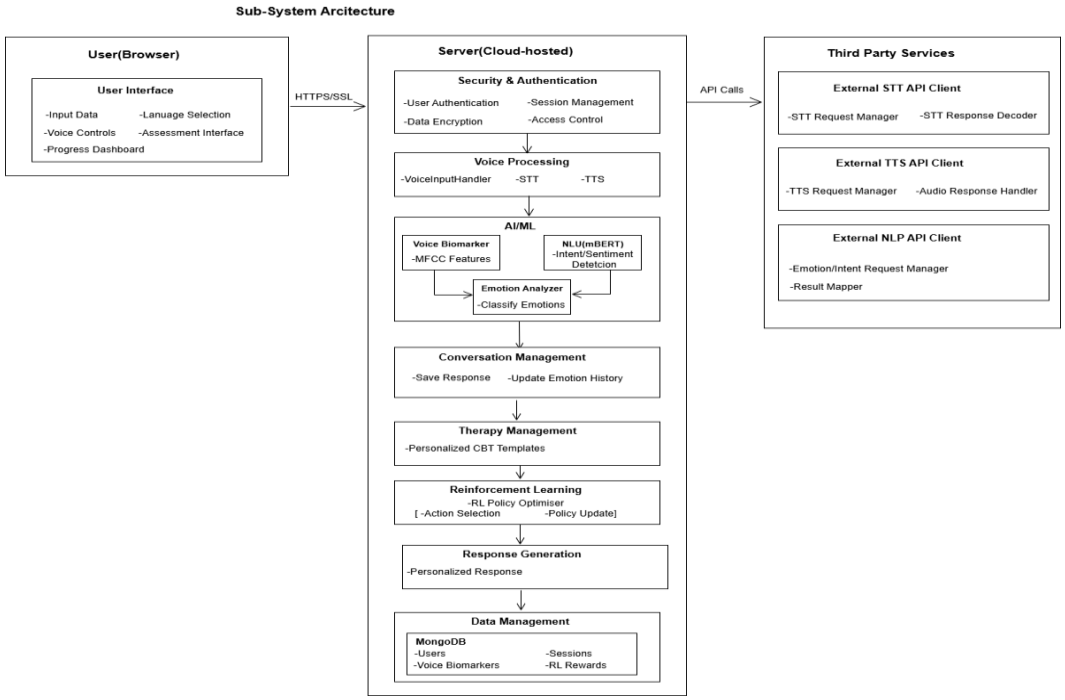
Package Diagram



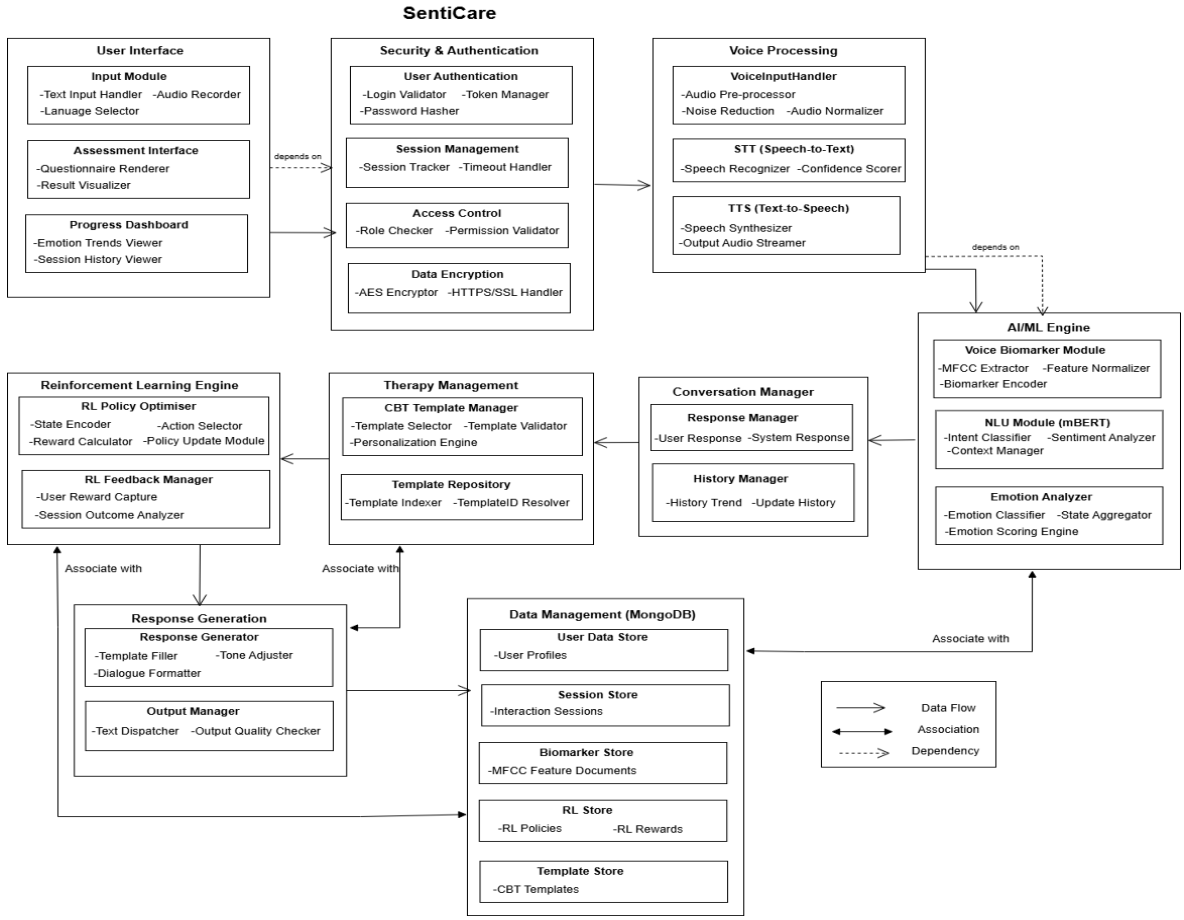
Component Diagram:



3.2 Sub-System / Component / Module Level Architecture



3.3 Sub-Component / Sub-Module Level Architecture (1…n)



4 Design Strategies

4.1 Strategy 1—Modular, service-oriented architecture (logical microservices)

Split the system into well-defined modules/services (Authentication, Voice Input, STT, Voice Biomarkers, NLU, Emotion Analyzer, CBT Template Manager, RL Optimizer, Response Generator, TTS, Data Store). Each module has a narrow responsibility and communicates via clear interfaces (REST/gRPC/messages).

Reasoning / Goals mapped:

− System requires cloud backend, STT/TTS, RL, and mBERT — each is complex and benefits from separation.

− Enables independent development, testing, deployment and scaling of heavy components (e.g., NLU and RL on GPU nodes).

Trade-offs:

− Easier extension and reuse; teams can replace a module (e.g., swap STT provider) with minimal impact.

− Fault isolation (one service down doesn't take whole system).

− Slight latency overhead from inter-service communication.

Impact on system design:

− High-level architecture: API gateway → microservices → data stores.

− Use containerization (Docker) and orchestration (Kubernetes or managed services) for deployment.

− Define clear service contracts (OpenAPI proto/gRPC).

− For development, provide local mock services for heavy modules (mBERT, RL) to speed testing.

For extension & reuse: New therapies or analytics services can be added as separate services. Existing modules become reusable libraries or microservices.

4.2 Strategy 2 — Hybrid emotion detection (feature fusion of voice biomarkers + NLU)

Fuse acoustic features (MFCC, pitch, prosody) from Voice Biomarkers with textual features (mBERT embeddings, sentiment scores) in the Emotion Analyzer to produce a single, robust emotion label.

Reasoning / Goals mapped:

System specifies dual-channel analysis — fusion improves detection accuracy vs single modal approaches.

Trade-offs:

− Higher accuracy and robustness.

− Allows richer analytics (correlate voice vs text signals).

− More complex preprocessing and model design (data alignment, temporal synchronization).

− More compute and storage for storing features**.**

Impact on system design / data management:

− Emotion Analyzer must accept multimodal inputs and maintain time-aligned windows.

− Storage: feature store (or collection of time-stamped records) for voice features and transcripts for each session.

− Synchronization: use session timestamps to align audio frames with transcripts.

− Concurrency: feature extraction pipelines can be parallelized (audio and STT in parallel), but fusion needs synchronized inputs.

4.3 Strategy 3 — Reinforcement learning (PPO) as a personalization layer on top of template library

Use PPO to optimize selection/ordering/phrasing of CBT templates and micro-interventions based on rewards (user feedback signals, sentiment improvement, session completion).

Reasoning / Goals mapped:

System requires RL to continuously refine responses for therapeutic effectiveness. RL personalizes interventions based on user reactions and progress.

Trade-offs:

− Personalization + potential to improve outcomes over time.

− RL needs careful reward design and safety constraints (avoid harmful suggestions).

− Training/online updates are complex and require offline evaluation and human oversight.

Impact on architecture & data:

− RL service must receive state (emotion, history, template features) and produce actions (template + parameters).

− Logging: store state-action-reward tuples (RL-Rewards table) with user consent for training.

− Versioning: maintain policy model versions and an audit trail.

− Safety: add a rule-based guard (safe policy or filter) so RL cannot propose unsafe content.

4.4 Strategy 4 — Language-first / voice-first user interface with graceful fallback

Primary UX is voice-driven (voice capture → STT → conversation), with a web UI that also supports keyboard/text interaction and visual progress dashboards.

Reasoning / Goals mapped:

System emphasizes voice input and bilingual support (English/Urdu) and 24/7 accessibility. Voice-first reduces friction and increases accessibility.

Trade-offs:

− Natural and accessible interactions; good for users preferring spoken interaction.

− Requires robust STT/TTS and network quality for good UX.

− Designing conversational flows for both voice and text adds development effort.

Impact on structure & data:

− UI module implements microphone permissions, real-time audio streaming to backend, and a visual transcript.

− Include UI state machine: listening → processing → response → follow-up.

− For offline/poor-network, offer a local client-side recording upload fallback or text input.

Practical UI paradigms:

− Provide clear affordances (microphone button, language switch, progress indicator).

− Show transcript + suggested CBT exercises visually after TTS to help literacy and review.

4.5 Strategy 5 — Privacy-by-design and strict data governance

Encrypt data at rest and in transit, pseudonymize stored user identifiers (replace user's data with artificial ones), collect only required data, provide consent workflows, and implement access controls and audit logs.

Reasoning / Goals mapped:

System states to not share user's data with third parties and GDPR compliance; mental health data is sensitive.

Trade-offs:

− Builds trust and meets legal/ethical obligations.

− Adds complexity to implementation (key management, access roles).

− May limit certain analytics unless explicitly consented.

Impact on architecture & data management:

− Use store for encrypted data, session data, RL logs (consent required).

− Anonymize data before using it for model training unless explicit consent is saved.

− Use field-level encryption for sensitive features (e.g., raw audio).

− Keep audit trail for all data accesses and changes.

Concurrency & distribution implications:

− Ensure that services that handle decryption run in trusted secure zones; tokenized access for others.

− Use short-lived tokens and role-based access to prevent leaks.

4.6 Strategy 6 — Reuse and component libraries (templates, models, utilities)

Represent CBT templates, NLU preprocessing pipelines, and visualization widgets as versioned reusable libraries. Package commonly used code (audio preprocessing, feature extraction, DB access) as shared modules.

Reasoning / Goals mapped:

System includes a CBT library (30–45 templates); packaging these increases reuse and allows consistent updates.

Trade-offs:

− Faster development and consistency across services.

− Need governance for library changes and versioning to avoid breaking consumers.

Impact on architecture:

− Maintain internal package registry (pip/npm) for shared libraries.

− Versioned template store (so historical sessions reference the exact template text used).

− Use semantic versioning for library changes.

Extension: New CBT techniques (mindfulness modules, psychoeducation) can be added as new template packages without changing core RL/response code.

4.7 Strategy 7— MFCC-Centric Data Model & Persistence (MongoDB)

The system stores each session as a stream of events in MongoDB, and keeps all MFCC feature vectors in a dedicated MongoDB Feature Collection. Only MFCCs (not raw audio) are saved, making storage efficient and directly usable by the RL engine.

Reasoning / Why MongoDB + MFCC store

− MongoDB handles semi-structured event data well and allows fast inserts for session events.

− MFCC vectors fit naturally as documents (e.g., {sessionId, timestamp, mfcc:[…]}) and are easy to query by time for RL training.

− Storing MFCCs instead of raw audio reduces storage and speeds up RL experiments.

Trade-offs

Pros: lightweight storage, fast reads for RL, easy horizontal scaling through MongoDB sharding.

Cons: No raw audio means MFCC extraction cannot be corrected later; must ensure extraction accuracy.

Impact on system design

− A MongoDB SessionEvents collection stores chronological events (AudioCaptured, MFCCExtracted, RLActionTaken).

− A MongoDB MFCCFeatures collection stores MFCC vectors indexed by sessionId + timestamp.

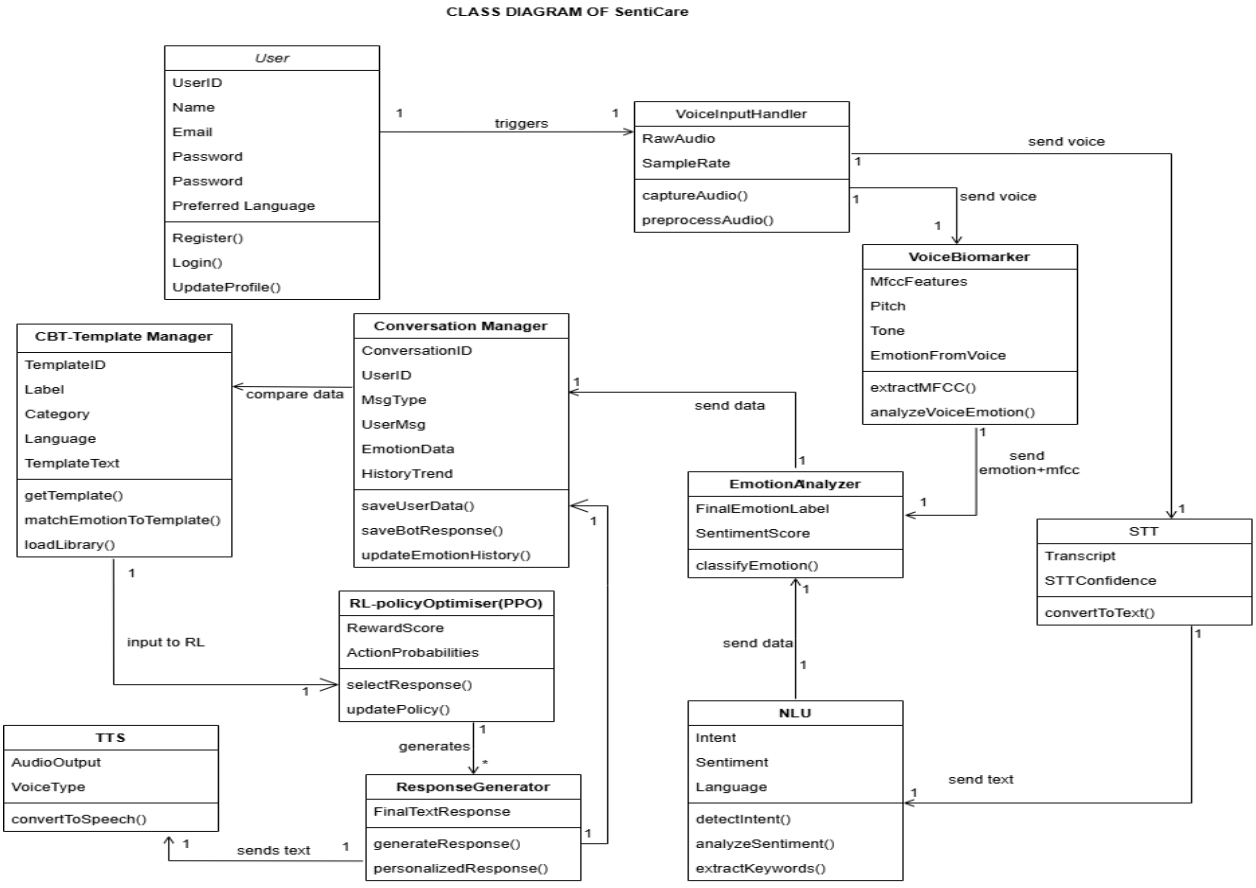
− RL module reads MFCC sequences directly from MongoDB to compute rewards and update policies.

Concurrency considerations

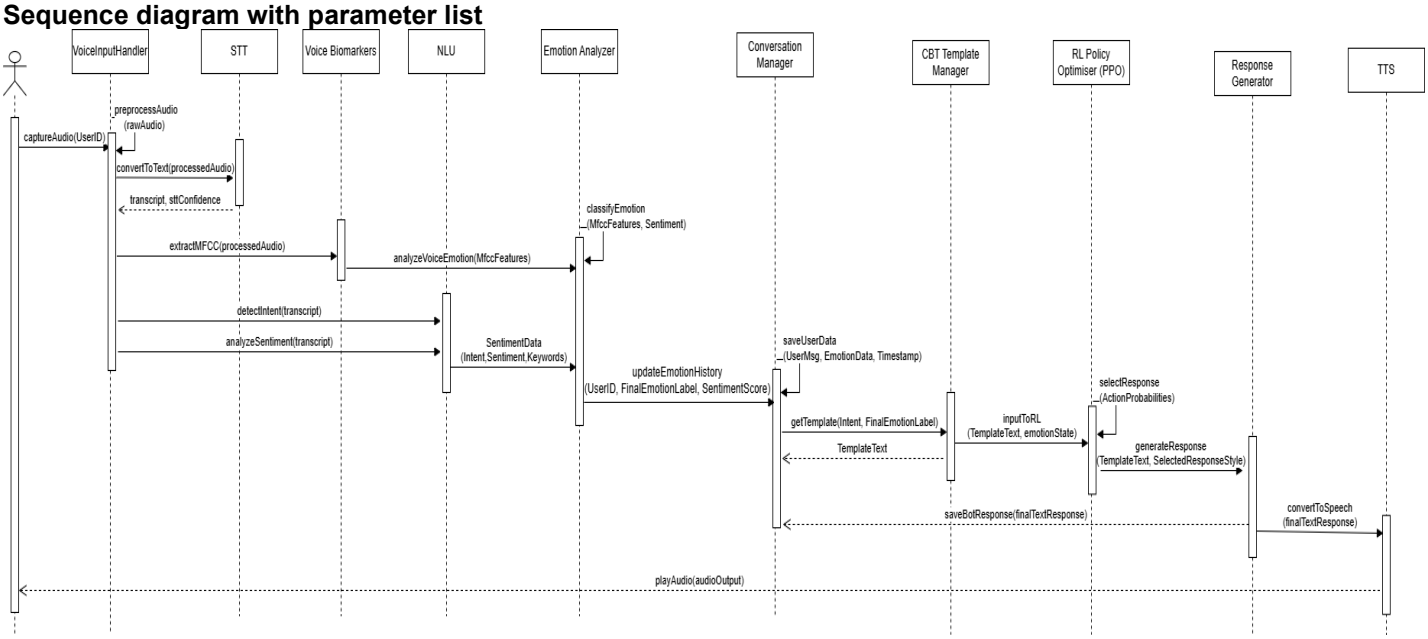
Apply optimistic concurrency when updating any derived session state (e.g., emotional progress) to avoid conflicts.

5 Detailed System Design

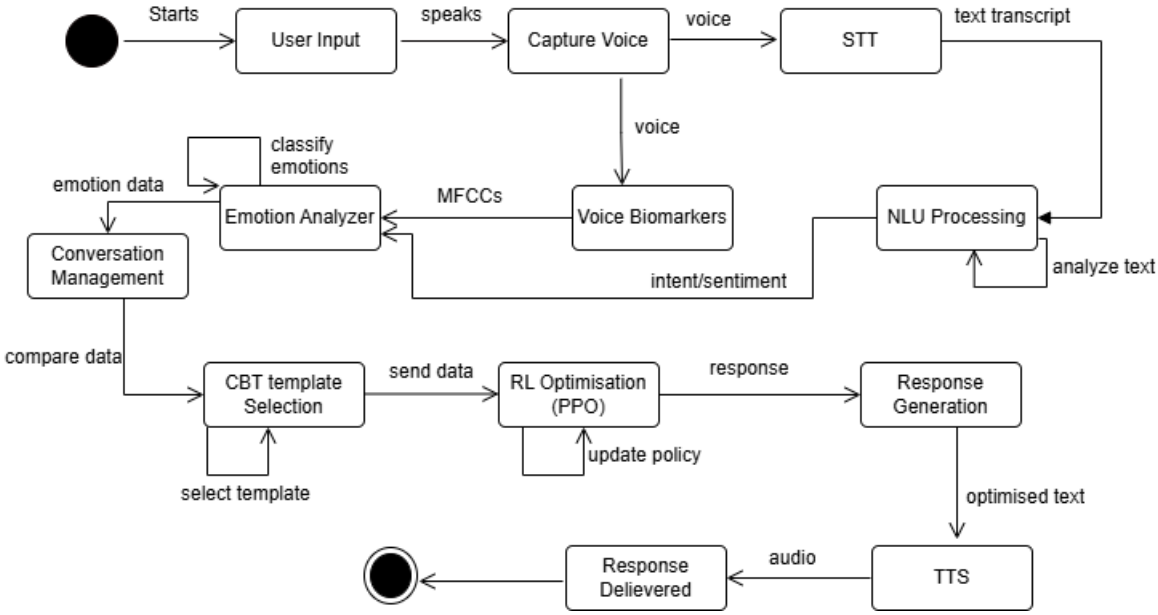
Class Diagram



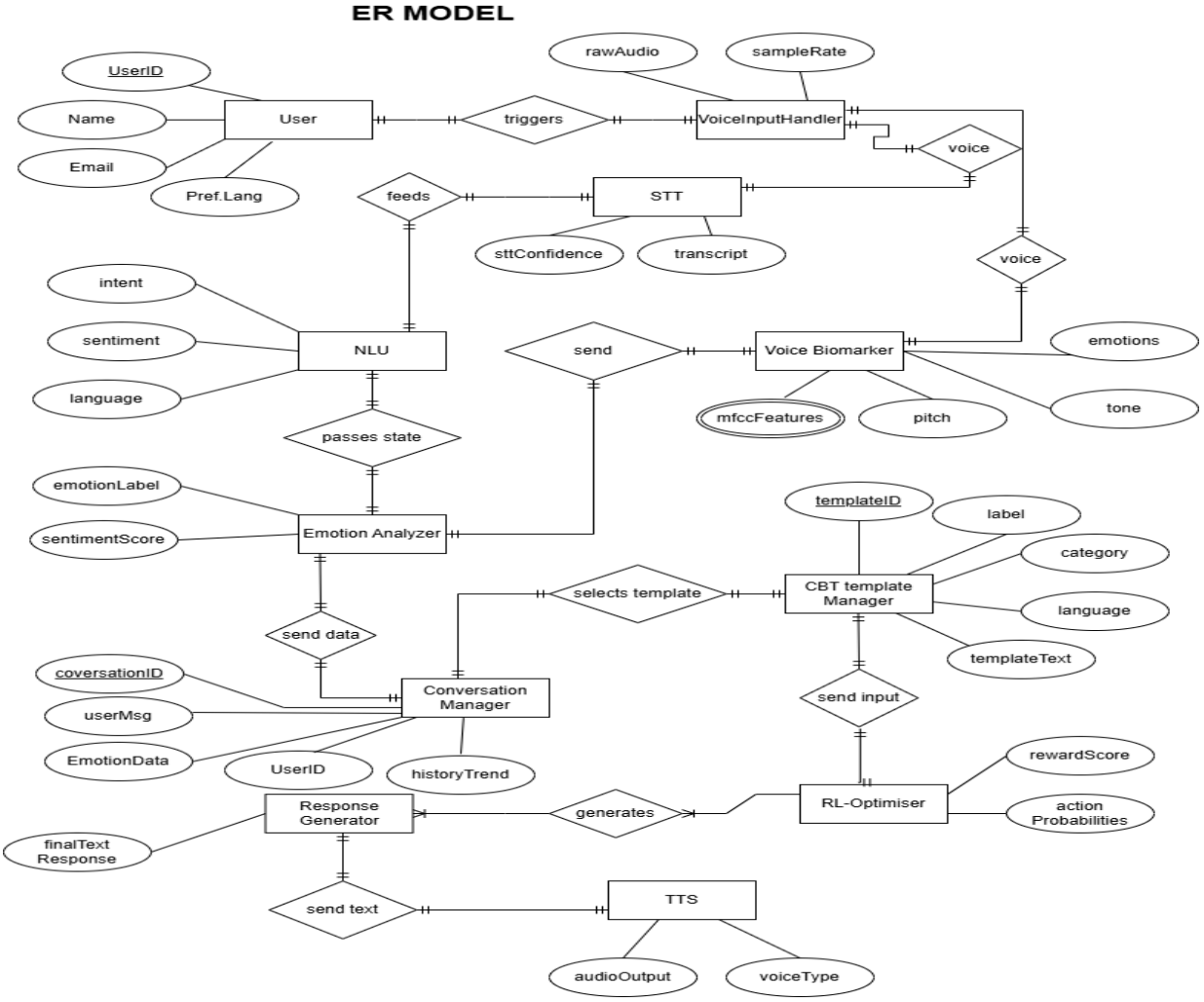
Sequence diagram with parameter list



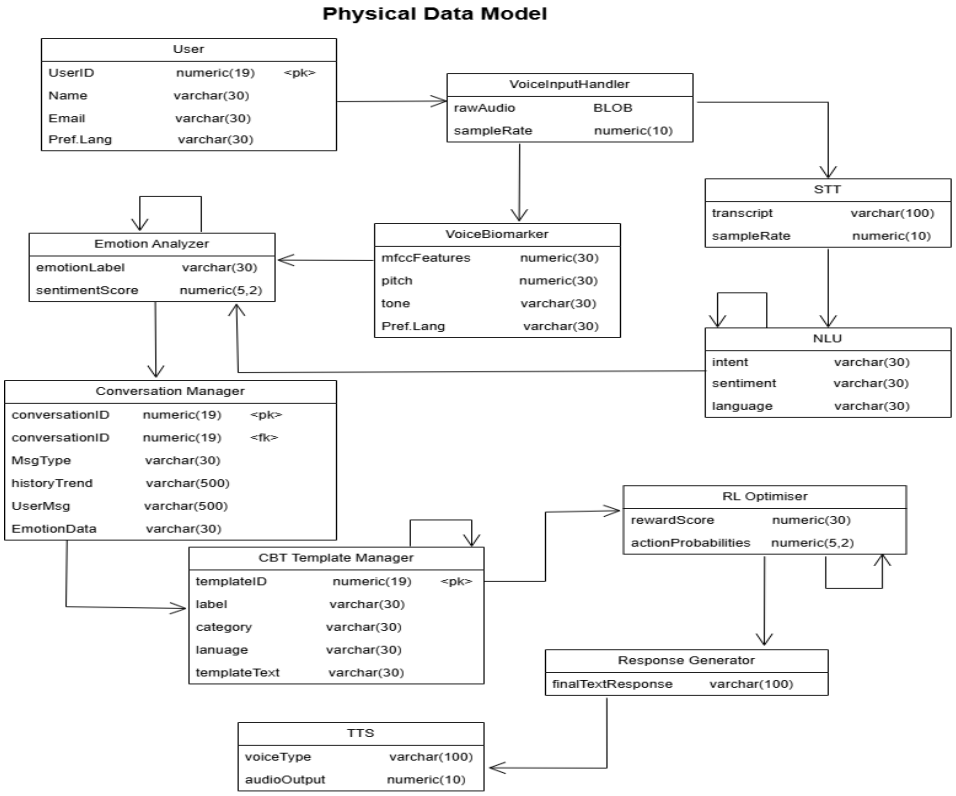
State Transition Diagram



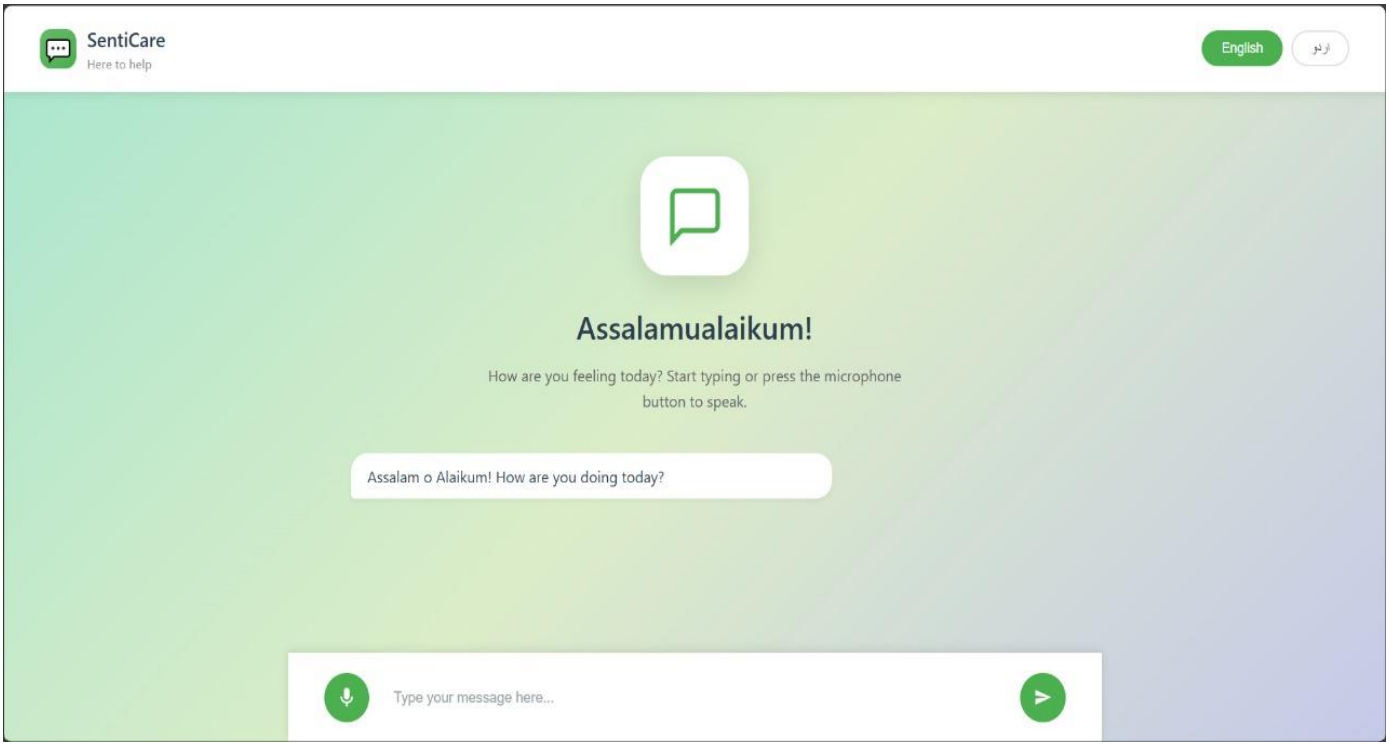
Logical data model (E/R model)

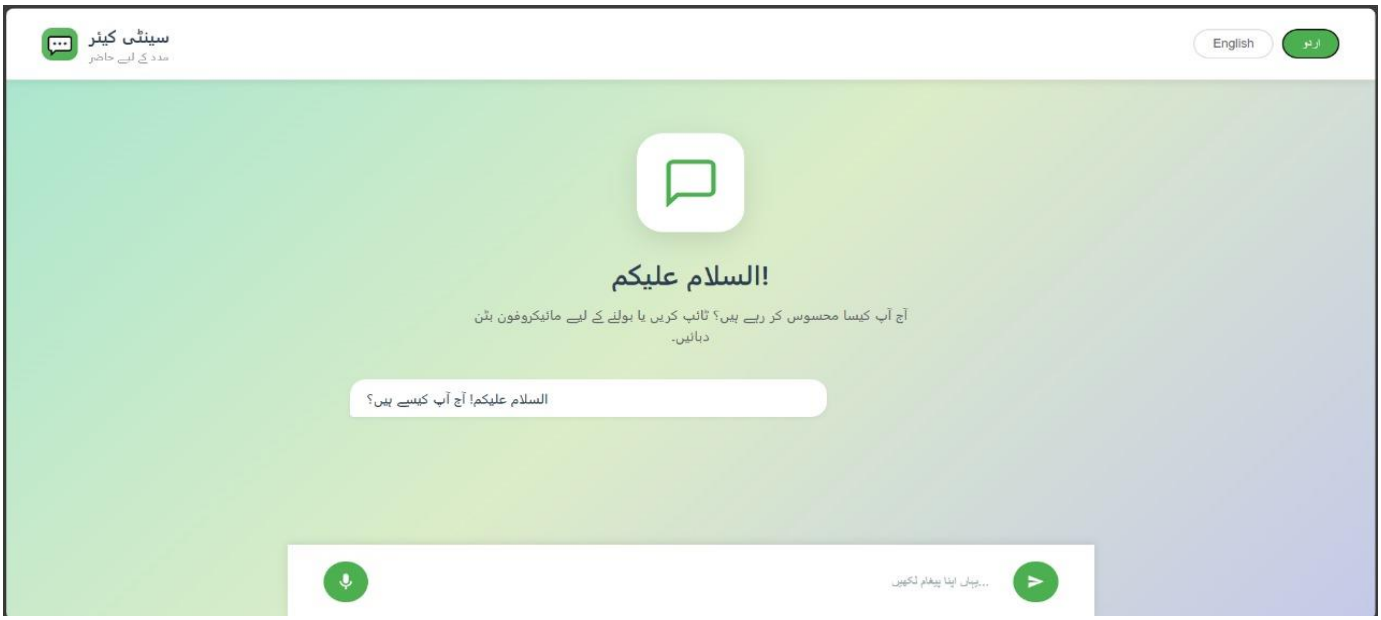


Physical data models



Detailed GUIs





6 References

Ref. No.	Document Title	Date of Release/ Publication	Document Source
PGBH01-2025-Proposal	Project Proposal	Oct 6, 2025	https://github.com/sheikh-zain786/SentiCare-Capstone/tree/main/proposal
PGBH01-2025-FS	Software Requirements Specifications	Oct 20, 2025	https://github.com/sheikh-zain786/SentiCare-Capstone/tree/main/SentiCare%20SRS

7 Appendices

1. UI Mockups: Visuals of key interfaces (Homepage, Language Selection, Voice Input Screen, Assessment Questionnaire, CBT Exercise Interface, Progress Dashboard).

2.Database-Schema: Tables for Users, Sessions, Assessment-Scores, Voice-Biomarkers, CBT-Templates, RL-Rewards, and Crisis-Detection-Logs.

3. API Documentation: Key API endpoints (User Authentication, Voice Upload, STT Processing, Emotion Detection, CBT Template Retrieval, Session Storage, TTS Generation).

4. Compliance:

- Data Privacy: GDPR compliance, data protection laws, user consent protocols

-Ethical Guidelines: Mental health data handling, crisis referral procedures, anonymization standards

5. CBT Template Library: Complete collection of 30-45 structured CBT templates in English and Urdu with clinical references.

6. Evaluation Metrics: Success measurements (User engagement rate, sentiment improvement scores, technique effectiveness by disorder type, RL convergence metrics).