

AI EMOTIONAL VENTING & FEEDBACK APPLICATION (VENTI AI)

Ranjan Khadka

University of Minnesota

Math, Science and Technology Department

Software Construction : SE2300

Silvia Preston

Jan 20, 2026

Figure 1:Data Flow Diagram	4
Figure 2: Component Diagram	5
Figure 3: Gnatt Chart	7

AI EMOTIONAL VENTING & FEEDBACK APPLICATION (VENTIAI)

Project Selection

The software project I selected for this assignment is VentiAI which is an AI-powered emotional venting and feedback application. The purpose of this system is to provide users with a safe, judgment-free digital space where they can express their frustrations, stress, or emotional concerns. Users can freely type or speak their thoughts, and the AI system will analyze the message and respond with empathetic feedback, emotional validation, and gentle coping suggestions. The application focuses on listening and reflection rather than problem-solving alone.

What makes this project unique is its emphasis on emotional feedback rather than productivity or task completion. Unlike standard chatbots or journaling apps, VentiAI is designed specifically for emotional release and self-reflection. The system is intentionally limited in scope and positioned as a supportive tool rather than a replacement for professional mental health services, making it ethical, and realistic for an academic software engineering project.

Requirements Gathering

Functional Requirements

1. The system shall allow users to enter text-based or speech based emotional vents.
2. The system shall analyze user input to determine emotional tone and sentiment.
3. The system shall generate empathetic and supportive feedback responses.
4. The system shall provide reflective statements summarizing the user's emotions.

5. The system shall offer optional coping suggestions (e.g., breathing, reframing).
6. The system shall show a prompt stating that the app is not a medical or therapy service.

Non-Functional Requirements

1. The interface must be simple, calming, and easy to navigate.
2. The system should generate feedback within two seconds of user input.
3. User data must persist between application sessions.
4. The system must use modular components for future AI improvements.
5. Vent data must be stored locally and protected from unauthorized access.
6. The system must avoid harmful, judgmental, or clinical language.

Architecture Design

High-Level Architecture

VentiAI will use a layered architecture consisting of the following components:

1. Presentation Layer: User interface for vent input, AI responses, and history viewing.
2. Application Logic Layer: Controls sentiment analysis, response generation, and validation.
3. AI Processing Layer: Performs natural language processing and emotional tone analysis.
4. Data Layer: Manages local storage of vent entries and AI responses.

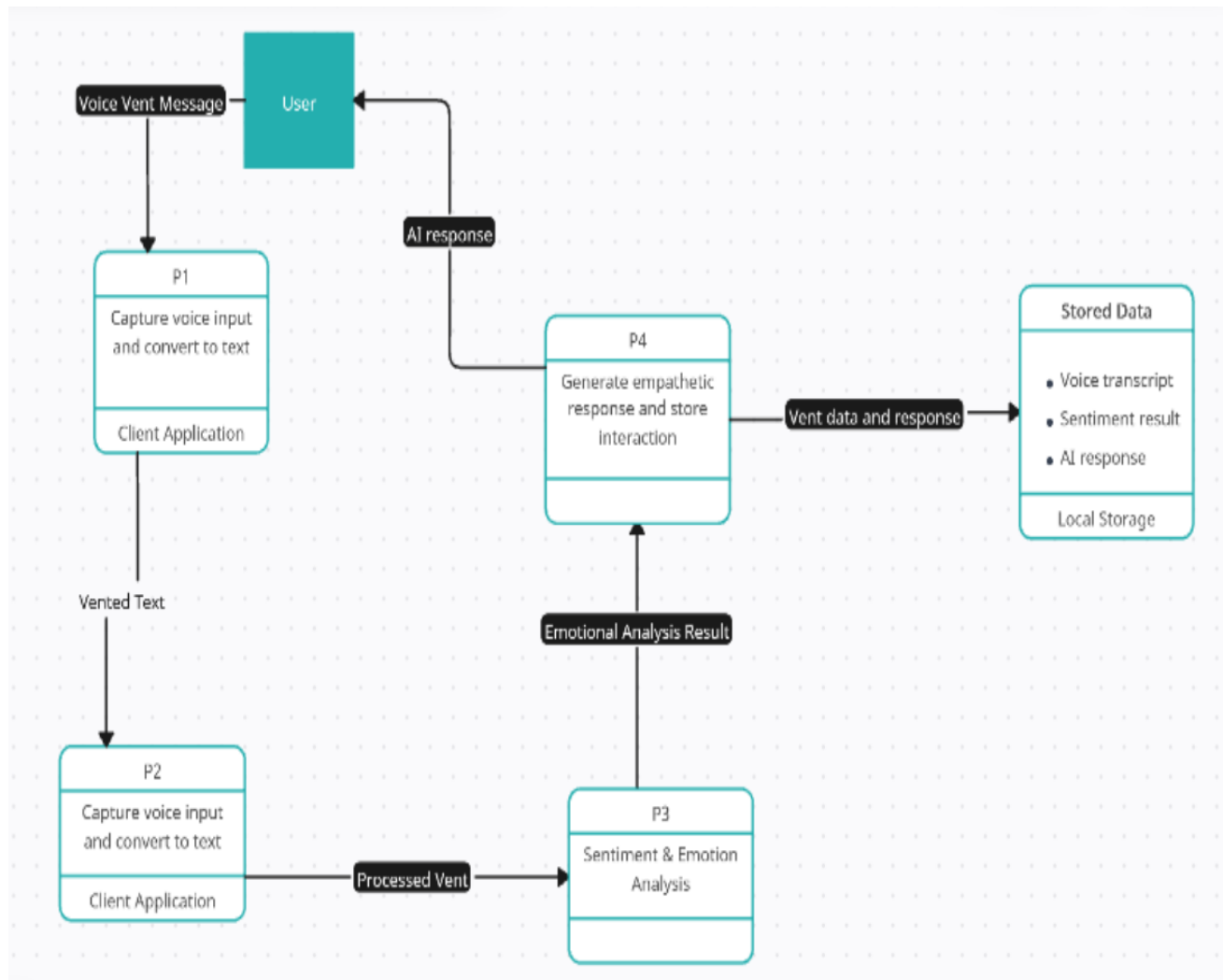


Figure 1: Data Flow Diagram

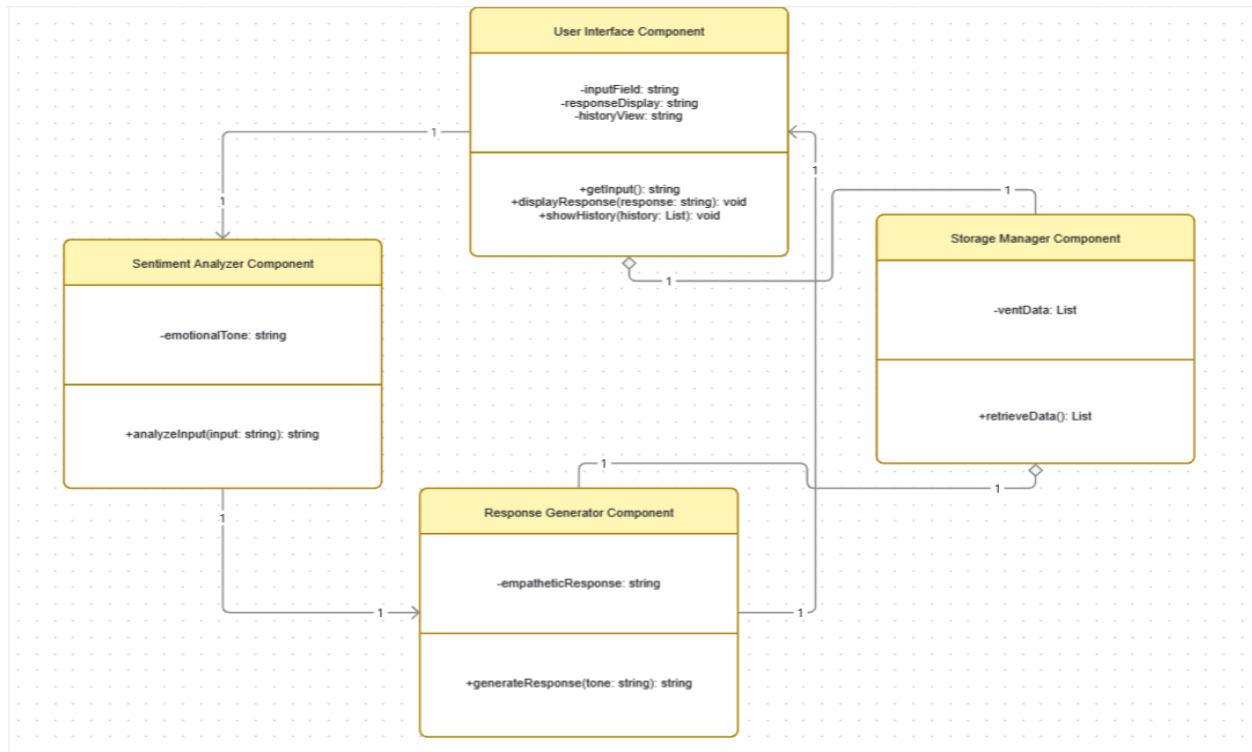


Figure 2: Component Diagram

- User Interface Component: Input field, response display, history view
- Sentiment Analyzer Component: Determines emotional tone
- Response Generator Component: Produces empathetic feedback
- Storage Manager Component: Saves and retrieves vent data

The architecture is scalable and allows future features such as personalization, mood tracking, or cloud synchronization.

Detailed Planning

Milestones and Timeline

Week	Milestone
Week 1	Requirements analysis and ethical design review
Week 2	Architecture design and UI prototyping
Week 3	AI logic and core feature implementation
Week 4	Testing, validation, and documentation

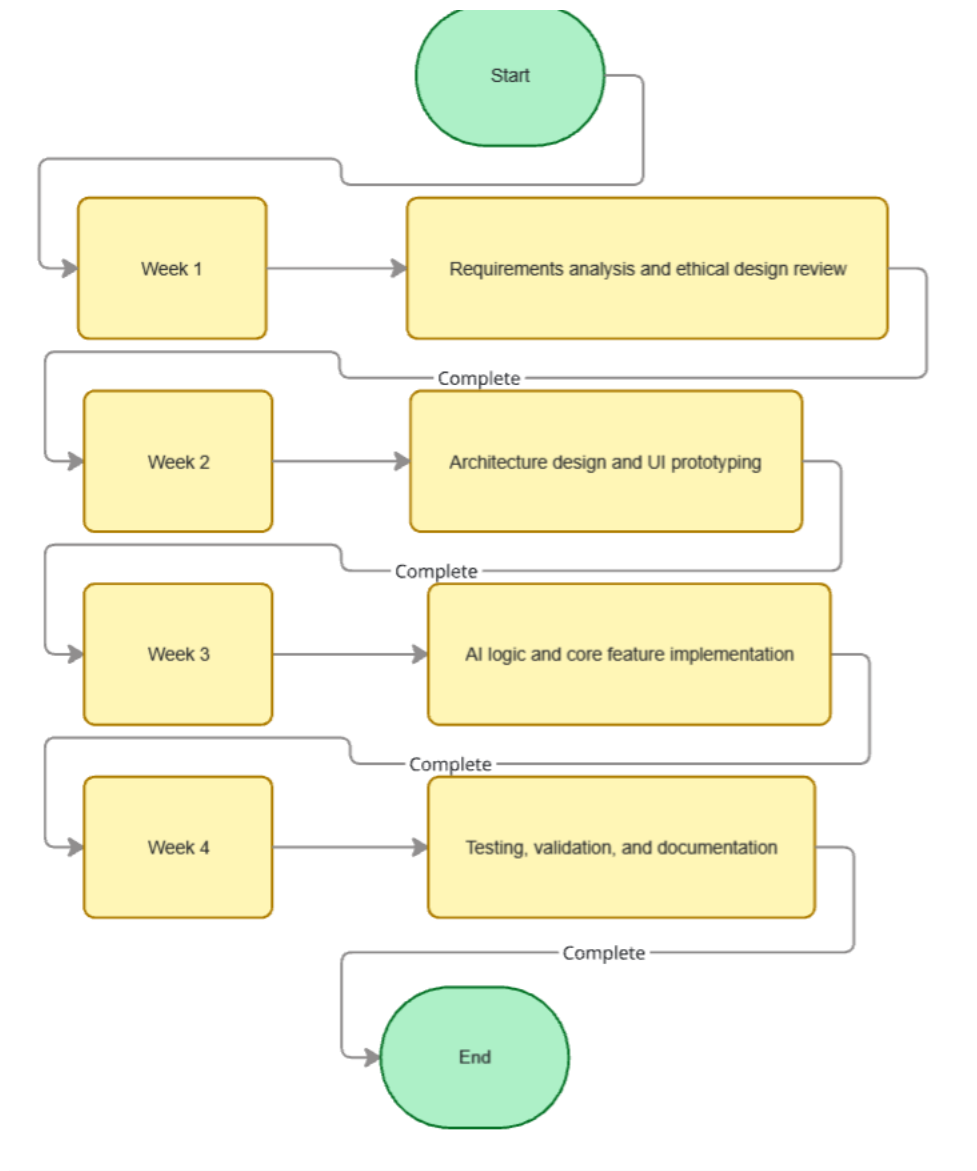


Figure 3: Gantt Chart

Resource Allocation

- Developer: 1 (Ranjan Khadka)
- Tools: VS Code, GitHub, project planning software

Risk Management

- Risk: AI responses perceived as insensitive
 - *Mitigation:* Using predefined empathetic and kind templates
- Risk: Data privacy concerns
 - *Mitigation:* Store data locally and encryption for cloud transmission

Proposed Implementation

The project will be implemented using JavaScript(along with HTML, CSS) with the use of libraries like TensorFlow.JS along with integration with OpenAI API or LangChain.js and rapid development capabilities . Basic natural language processing can be implemented using rule-based sentiment analysis or lightweight TTS libraries. The user interface will be developed using React.js, Vue.js, or vanilla HTML/CSS/JS and most of the backend will be made using node.js

Programming tradition will include meaningful variable names, modular functions, consistent formatting, and clear inline comments. Version control will be managed through Git to ensure traceability and organized development

Testing and Validation

Testing will ensure the application meets both functional and ethical requirements. Unit testing will validate sentiment detection and response generation logic. Integration testing will confirm smooth communication between the UI, AI logic, and storage components. Manual testing will focus on usability and emotional tone appropriateness.

Sample Test Cases

1. Vent Submission Test: Verify user input is accepted and processed.

2. Sentiment Analysis Test: Confirm emotional tone is identified correctly.
3. Response Quality Test: Ensure responses are empathetic and non-judgmental.
4. Data Persistence Test: Restart the application and confirm history remains.
5. Disclaimer Test: Verify prompt is displayed consistently.

Documentation

Comprehensive documentation will be written and will include:

- User Manual: Instructions on how to install, launch, vent, view responses, and access history.
- Technical Documentation: System architecture, AI logic flow, and data structures.
- Ethical Guidelines: Explanation of system limitations and responsible use.

All documentation will be written in clear, accessible language to support both users and future developers.

Conclusion

VentiAI will show the practical application of software engineering principles along with addressing modern, real-world need. By combining careful requirements analysis, ethical AI design, structured architecture, and thorough testing, this project will presents a realistic and ethical approach to building an AI-assisted emotional support application.