Mood Garden
Team 3
Bobby Barsh
Christian Madrid
Alejandra Juarez Carpio

Software Requirements Specification

Document

Version: 1.0 **Date: (01/30/2026)**

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) is to define the functional and non-functional requirements for the Mood Garden system. This document is intended for stakeholders, instructors, developers, and testers to understand what the system must do before implementation begins. The SRS serves as the official reference for system expectations and constraints.

1.2 Scope

Mood Garden is a software application designed to help users track, reflect on, and understand their emotional well-being over time. The system allows users to log mood entries, view trends, and receive basic mood-based feedback.

The system will:

- Allow users to input mood data manually
- Store mood entries securely
- Display mood history and trends visually
- Provide basic recommendations based on mood patterns

The system will not:

- Provide medical or clinical diagnoses
- Replace professional mental health services

The goal of Mood Garden is to promote self-awareness, emotional reflection, and healthy habits through simple and accessible technology.

1.3 Definitions, Acronyms, and Abbreviations.

SRS – Software Requirements Specification

UI – User Interface

User – Any individual who interacts with the Mood Garden system

Mood Entry – A user-submitted record describing emotional state

System – The Mood Garden application as a whole

1.4 References

- IEEE Std 830-1993 – IEEE Guide to Software Requirements Specifications
- Course syllabus and project guidelines provided by the instructor
- General software engineering best practices discussed in class materials

1.5 Overview

This SRS document is organized into sections describing the system from both a user and developer perspective.

- **Section 2** provides a high-level overview of system functionality and constraints for users and stakeholders.

- **Section 3** defines detailed system requirements intended for developers and testers.

2. The Overall Description

2.1 Product Perspective

Mood Garden is a standalone software application and does not depend on any external systems for core functionality. It is similar in concept to basic mood-tracking or journaling applications but focuses on simplicity and ease of use for students and general users.

The system operates as a self-contained platform where users interact directly with the application to record and review emotional data.

2.1.1 System Interfaces

Mood Garden does not interface with external enterprise systems. All system interactions occur between the user and the application itself.

2.1.2 Interfaces

The system provides a graphical user interface (GUI) that allows users to:

- Enter mood information
- Navigate mood history
- View charts or summaries of mood trends

The interface is designed to be intuitive and accessible, considering general usability and ADA guidelines where applicable.

2.1.3 Hardware Interfaces

The system has no direct hardware interface requirements. It operates on standard user devices such as desktops, laptops, tablets, or smartphones.

2.1.4 Software Interfaces

Mood Garden does not require mandatory integration with third-party software systems. Any software components used internally are implementation decisions and not requirements.

2.1.5 Communications Interfaces

The system communicates over standard internet protocols if deployed as a web-based application. No custom communication protocols are required.

2.1.6 Memory Constraints

There are no strict memory constraints. The system is expected to operate within the limits of typical consumer devices used by students.

2.1.7 Operations

Normal operation includes:

- User login (if applicable)
- Mood entry submission
- Viewing stored mood data

The system may perform unattended operations such as data storage and retrieval. Basic backup and recovery mechanisms are expected to protect stored data.

2.1.8 Site Adaptation Requirements

No site-specific adaptations are required. The system can be installed or accessed without modification to the user's environment.

2.2 Product Functions

Major functions of Mood Garden include:

- Recording mood entries
- Storing mood history
- Displaying mood trends over time
- Providing simple feedback based on mood data

2.3 User Characteristics

The intended users are students and general users with basic computer or mobile device experience. No advanced technical knowledge is required. The system is designed for ease of learning and minimal training.

2.4 Constraints

- Must comply with basic data privacy expectations
- Must protect user-entered data
- Must operate reliably on common consumer devices
- Must not provide medical advice

2.5 Assumptions and Dependencies

Assumptions:

- Users have access to a device with internet capability
- Users input mood data accurately

Dependencies:

- Availability of basic system resources (storage, processing)

2.6 Apportioning of Requirements.

Future versions may include:

- Advanced analytics
- Personalized recommendations
- Integration with external wellness tools

These features are not required for the initial version.

3. Specific Requirements

3.1 External Interfaces

- Input: User mood entries (text, selection, or scale)
- Output: Mood summaries, charts, and feedback messages

3.2 Functions

- **FR-1:** The system shall allow users to submit mood entries.
- **FR-2:** The system shall store mood entries securely.
- **FR-3:** The system shall allow users to view past mood entries.
- **FR-4:** The system shall generate mood trend summaries.
- **FR-5:** The system shall handle invalid or incomplete input gracefully.

3.3 Performance Requirements

- The system shall process user requests within 2 seconds under normal conditions.
- The system shall support multiple users without data loss.

3.4 Logical Database Requirements

The system shall store:

- User identifiers
- Mood entry data
- Time and date of entries

Data integrity and consistency must be maintained.

3.5 Design Constraints

The system must follow course guidelines and standard software engineering practices.

3.5.1 Standards Compliance

The system shall follow IEEE SRS documentation standards.

3.6 Software System Attributes

3.6.1 Reliability

The system shall operate without failure during normal use.

3.6.2 Availability

The system shall be available during typical user access times.

3.6.3 Security

The system shall protect user data from unauthorized access.

3.6.4 Maintainability

The system shall be easy to update and maintain.

3.6.5 Portability

The system shall be usable across multiple platforms where applicable.

ID	Characteristic	H/M/L	1	2	3	4	5	6	7	8	9	10	11	12
1	Correctness	H												
2	Efficiency	L												
3	Flexibility	L												
4	Integrity/Security	H												
5	Interoperability	H												
6	Maintainability	H												
7	Portability	L												
8	Reliability	M												
9	Reusability	L												
10	Testability	H												
11	Usability	H												
12	Availability	H												

3.7 Organizing the Specific Requirements

The detailed requirements in Section 3 are organized by functional modules. Each major subsystem of the application (e.g., authentication, data management, user interface, reporting) is presented in its own subsection. This structure allows stakeholders to easily locate requirements relevant to a specific feature and supports modular design and implementation.

3.7.1 User Management

Covers everything related to user accounts and identity.

- Account creation
- Login / logout
- Profile settings
- Data privacy preferences

3.7.2 Mood Input & Logging

Handles how users submit their emotional state.

- Mood entry form (text, sliders, emojis)
- Timestamping

- Storing entries
- Editing or deleting past entries

3.7.3 AI Mood Analysis Module

Describes how the system interprets user input.

- Sentiment analysis
- Classification of mood categories
- Trigger conditions for recommendations
- Handling ambiguous or unclear input

3.7.4 AI-Generated Supportive Messages

Defines how the system produces helpful responses.

- Message generation rules
- Tone and safety constraints
- Personalization logic
- Handling inappropriate or harmful input

3.7.5 Recommendation Engine

Covers how the system selects links or activities.

- Matching mood → recommended action
- Types of resources (breathing exercises, grounding techniques, journaling prompts, etc.)
- Ranking or prioritization logic
- Avoiding clinical or medical advice

3.7.6 User Dashboard

Everything the user sees after logging in.

- Mood history visualization
- Recent AI messages
- Trends or patterns
- Quick-access to recommended tools

3.7.7 Notifications & Reminders (Optional)

If your app sends nudges or reminders.

- Daily check-ins
- Mood-logging reminders
- Safety-aware messaging rules

3.7.8 System Administration

Internal tools for maintainers.

- Managing resource links
- Updating AI models or rules
- Viewing system logs

3.7.9 Non-Functional Requirements

Grouped separately for clarity.

- Performance
- Security
- Usability
- Portability
- Reliability

4. Change Management Process

Change Management Process (SRS Section)

The project will follow a structured change management process to ensure that all modifications to requirements are properly identified, evaluated, approved, and documented. This process prevents uncontrolled scope changes and ensures that all stakeholders have a shared understanding of updates to the SRS.

4.1 Change Request Submission

All proposed changes to requirements must be submitted formally in writing. Acceptable submission formats include:

- Email to the project team
- A written Change Request Form (CRF) stored in the project repository

Phone calls, verbal requests, or informal messages will not be accepted as valid change requests.

4.2 Logging the Change

Once submitted, the change request will be logged in the team's Change Log, which includes:

- Request ID
- Date submitted
- Requester name
- Description of the requested change
- Reason for the change

- Impacted sections of the SRS

This log is maintained by the Requirements Manager (or designated team member).

4.3 Evaluation

The project team will evaluate each change request based on:

- Impact on project scope
- Impact on timeline and milestones
- Technical feasibility
- Resource requirements
- Consistency with project goals

If needed, the team may request clarification from the customer before making a decision.

4.4 Approval Process

Changes are approved only through team consensus.
A change request is accepted if:

- A majority of the team agrees
- The change does not violate project constraints
- The customer confirms the need for the change

The customer **cannot** simply call and request new features without going through the formal submission and evaluation process.

4.5 Updating the SRS

If a change is approved:

- The SRS will be updated in the next revision cycle
- The updated version will include a new revision number and date
- The Change Log will be updated to reflect the approval and implementation
- All team members and the customer will be notified of the update

4.6 Version Control

All SRS revisions will be stored in a version-controlled repository (e.g., GitHub).
Each revision will include:

- Version number
- Summary of changes
- Author/editor
- Date of update

5. Document Approvals

Name:	Role:	Signature:	Date:
Bobby Barsh	AI/ML Engineer	Bobby Barsh	1/30/2026
Christian Madrid	Backend Developer	Christian Madrid	1/30/2026
Alejandra Juarez Carpio	Frontend Developer	Alejandra Juarez Carpio	1/30/2026

6. Supporting Information

Appendix supporting background information will be added once more implementation has been completed.