**Project Title**

Software Requirement Specifications

Bachelor of Science in Software Engineering

By

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Name** | **Registration #** | **Mobile #** | **E-Mail** |
| 1 | Ayesha Ishfaq | Fa’22/BSSE/166 | 0324-8529491 | Fa-22-bsse-166@lgu.edu.pk |
| 2 | Esha Farrukh | Fa’22/ BSSE/172 | 0300-1717607 | Fa-22-bsse-172@lgu.edu.pk |

**Supervised by Co-Supervised by**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Teacher Name Teacher Name

****

Department of Software Engineering

Lahore Garrison University

Lahore

**Table of Contents**

[**1. Introduction** 5](#_Toc214841392)

[1.1 Purpose 5](#_Toc214841393)

[1.2 Document Conventions 5](#_Toc214841394)

[1.3 Intended Audience and Reading Suggestions 6](#_Toc214841395)

[1.4 Product Scope 6](#_Toc214841396)

[**2. Overall Description** 6](#_Toc214841397)

[2.1 Product Perspective 6](#_Toc214841398)

[2.2 Product Functions 7](#_Toc214841399)

[2.3 User Classes and Characteristics 9](#_Toc214841400)

[**2.3.1 General Users (Developers, Tech Students, Researchers)** 9](#_Toc214841401)

[**2.3.2 Licensed Therapists** 10](#_Toc214841402)

[**2.3.3 System Administrators** 10](#_Toc214841403)

[2.4 Operating Environment 11](#_Toc214841404)

[2.5 Design and Implementation Constraints 12](#_Toc214841405)

[2.6 User Documentation 12](#_Toc214841406)

[2.7 Assumptions and Dependencies 12](#_Toc214841407)

[**3. External Interface Requirements** 13](#_Toc214841408)

[3.1 User Interfaces 13](#_Toc214841409)

[3.2 Hardware Interfaces 13](#_Toc214841410)

[3.3 Software Interfaces 14](#_Toc214841411)

[3.4 Communications Interfaces 16](#_Toc214841412)

[**4. System Features** 17](#_Toc214841413)

[4.1 AI Emotional Support Chat 17](#_Toc214841414)

[**4.1.1 Description and Priority** 17](#_Toc214841415)

[**4.1.2 Stimulus/Response Sequences** 17](#_Toc214841416)

[**4.1.3 Functional Requirements** 18](#_Toc214841417)

[4.2 Therapist Booking & Video Sessions 18](#_Toc214841418)

[**4.2.1 Description and Priority** 18](#_Toc214841419)

[**4.2.2 Stimulus/Response Sequences** 18](#_Toc214841420)

[**4.2.3 Functional Requirements** 18](#_Toc214841421)

[4.3 Mood Tracking & Analytics 19](#_Toc214841422)

[**4.3.1 Description and Priority** 19](#_Toc214841423)

[**4.3.2 Stimulus/Response Sequences** 19](#_Toc214841424)

[**4.3.3 Functional Requirements** 19](#_Toc214841425)

[4.4 User Authentication & Privacy Controls 19](#_Toc214841426)

[**4.4.1 Description and Priority** 19](#_Toc214841427)

[**4.4.2 Stimulus/Response Sequences** 19](#_Toc214841428)

[**4.4.3 Functional Requirements** 20](#_Toc214841429)

[4.5 Notification & Alerts System 20](#_Toc214841430)

[**4.5.1 Description and Priority** 20](#_Toc214841431)

[**4.5.2 Stimulus/Response Sequences** 20](#_Toc214841432)

[**4.5.3 Functional Requirements** 20](#_Toc214841433)

[4.6 User Profile Management 20](#_Toc214841434)

[**4.6.1 Description and Priority** 21](#_Toc214841435)

[**4.6.2 Stimulus/Response Sequences** 21](#_Toc214841436)

[**4.6.3 Functional Requirements** 21](#_Toc214841437)

[4.7 Therapist Dashboard (Web) 21](#_Toc214841438)

[**4.7.1 Description and Priority** 21](#_Toc214841439)

[**4.7.2 Stimulus/Response Sequences** 21](#_Toc214841440)

[**4.7.3 Functional Requirements** 21](#_Toc214841441)

[4.8 Admin Panel 21](#_Toc214841442)

[**4.8.1 Description and Priority** 22](#_Toc214841443)

[**4.8.2 Stimulus/Response Sequences** 22](#_Toc214841444)

[**4.8.3 Functional Requirements** 22](#_Toc214841445)

[4.9 Settings & Customization 22](#_Toc214841446)

[**4.9.1 Description and Priority** 22](#_Toc214841447)

[**4.9.2 Stimulus/Response Sequences** 22](#_Toc214841448)

[**4.9.3 Functional Requirements** 22](#_Toc214841449)

[**5. Other Nonfunctional Requirements** 22](#_Toc214841450)

[5.1 Performance Requirements 22](#_Toc214841451)

[5.2 Safety Requirements 23](#_Toc214841452)

[5.3 Security Requirements 23](#_Toc214841453)

[5.4 Software Quality Attributes 24](#_Toc214841454)

[5.5 Business Rules 26](#_Toc214841455)

[5.6 Detailed Subsystem Design 27](#_Toc214841456)

[**5.6.1 Chat & AI Processing Subsystem** 27](#_Toc214841457)

[**5.6.2 Mood Logging & Analytics Subsystem** 30](#_Toc214841458)

[**5.6.3 Therapist Booking & Session Subsystem** 31](#_Toc214841459)

[**5.6.4 Authentication & User Management Subsystem** 32](#_Toc214841460)

[**5.6.5 Notification Subsystem** 32](#_Toc214841461)

[**5.6.6 Database Management Subsystem** 33](#_Toc214841462)

[**5.6.7 System Architecture Subsystem** 34](#_Toc214841463)

[**5.6.8 Application Frontend Subsystem (Flutter)** 35](#_Toc214841464)

[**References** 35](#_Toc214841465)

# **1. Introduction**

## 1.1 Purpose

The purpose of this Software Requirements Specification (SRS) is to outline the requirements and objectives for MoodGenie, an AI-powered mental wellness application designed to support software developers, research scholars, and tech students who experience high stress and burnout due to demanding workloads and cognitive strain. The application integrates AI-driven emotional support, secure therapist access, and mood tracking analytics into one unified, privacy-first system to promote mental well-being within the technology community. Mental health has become a critical concern globally, with one in four individuals expected to experience a mental or neurological disorder in their lifetime [1]. Research also indicates that individuals in technical professions report higher rates of anxiety, depression, and work-related burnout compared to other fields [2]. Existing solutions such as Wysa, Woebot, and Youper provide partial support but lack personalization, therapist integration, and affordability for students [3, 4]. MoodGenie aims to overcome these limitations by delivering a cross-platform mobile solution built with Flutter, Node.js, Firebase, and OpenAI APIs, ensuring secure, accessible, and data-driven mental health support tailored to the tech community.

## 1.2 Document Conventions

This Software Requirements Specification (SRS) document follows the IEEE Standard 830-1998 guidelines for software documentation [5]. All requirements are written in clear, measurable, and verifiable terms. Functional requirements are labeled as FR-x, and non-functional requirements as NFR-x for easy traceability. Each requirement carries an assigned priority level—High (H), Medium (M), or Low (L)—to indicate its implementation order and significance.

Typographical conventions include:

* Bold text for section headings and major requirement titles.
* Italic text for emphasis or technical terms.
* Monospaced text (code format) for commands, API calls, or system responses.
* Requirements that are critical to the system’s success are marked as [Critical].
* Each requirement statement stands independently and does not inherit priority from higher-level sections.

## 1.3 Intended Audience and Reading Suggestions

This SRS is intended for all stakeholders involved in the development and deployment of MoodGenie, including project managers responsible for planning, developers implementing system features, UI/UX designers shaping the interface, QA testers validating functionality, backend engineers handling APIs and databases, external supervisors reviewing technical accuracy, and future documentation writers maintaining system updates. The document is organized into clear sections covering system overview, functional and non-functional requirements, system features, external interfaces, and constraints. Readers new to the project should begin with the Introduction and Overall Description, followed by Functional Requirements for implementation details, while testers and supervisors should focus on the detailed system requirements and acceptance criteria. This structured flow ensures that each stakeholder can easily locate the information most relevant to their role.

## 1.4 Product Scope

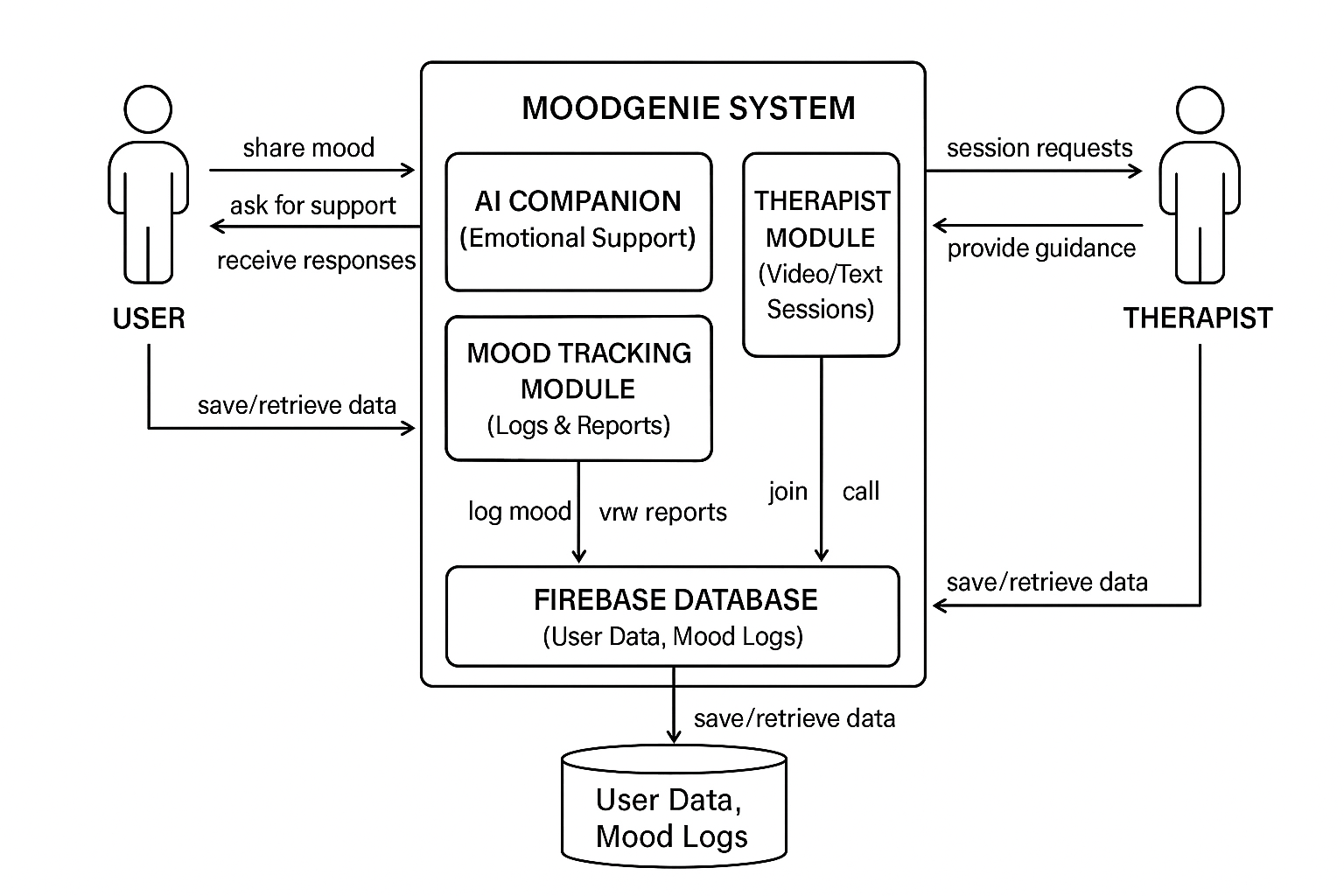
MoodGenie is an AI-powered mental wellness application designed to support software developers, tech students, and research scholars who frequently face stress, burnout, and cognitive overload. The purpose of the system is to provide an integrated platform offering 24/7 AI emotional support, secure therapist sessions, and mood tracking with personalized analytics, enabling users to manage their mental well-being more effectively [6]. The application contributes to accessible digital mental health care by combining AI-driven guidance with professional therapist interaction, supporting organizational and academic goals such as improved productivity, emotional stability, and healthier work environments. By aligning with global wellness strategies and Sustainable Development Goal 3 of promoting good health and well-being [7], MoodGenie aims to deliver a scalable, affordable, and privacy-first solution tailored to the unique stressors of the tech community.

# **2. Overall Description**

## 2.1 Product Perspective

MoodGenie is a new, standalone digital mental wellness application developed to address gaps in existing mental health solutions used by the tech community. Unlike current apps that focus individually on AI chatbots, therapy sessions, or mood tracking, MoodGenie integrates all three components into a single, unified platform [1, 2]. It does not replace any previous institutional system but instead introduces a comprehensive hybrid model that combines AI-driven emotional support with real-time therapist connectivity and structured mood analytics [3].

The system operates as an independent mobile application built on Flutter [4], supported by a backend infrastructure developed with Node.js, Express.js, Firebase, and OpenAI APIs [5]. MoodGenie communicates with external services through secure interfaces such as Firebase Authentication, Firestore Database, WebRTC for therapist video calls, and AI model APIs [6] for emotional analysis. These integrations allow the application to function smoothly while maintaining data privacy and scalability [7]. The following diagram illustrates the high-level structure and external interfaces of the MoodGenie system.



*Figure 1: High-Level System Architecture of the MoodGenie Application*

This perspective shows how MoodGenie functions as an integrated, modular system built to ensure seamless communication among components while delivering a user-centered mental health support experience.

## 2.2 Product Functions

MoodGenie provides a set of integrated mental-wellness features designed for software developers, tech students, and research scholars. The primary functions of the product include:

* **AI-Driven Emotional Support**

Allows users to interact with an empathetic AI companion for mood check-ins, stress management tips, journaling assistance, and conversational self-care guidance.

* **Therapist Access and Booking**

Enables users to book secure text or video sessions with licensed therapists, view therapist availability, and manage session history.

* **Mood Tracking**

Supports daily mood logging, capturing emotional state, triggers, activities, sleep quality, and stress levels.

* **Mood Analytics and Reports**

Generates weekly and monthly visual summaries to help users identify emotional patterns and behavioral trends.

* **User Authentication & Account Management**

Provides secure sign-up, login, password recovery, and profile management.

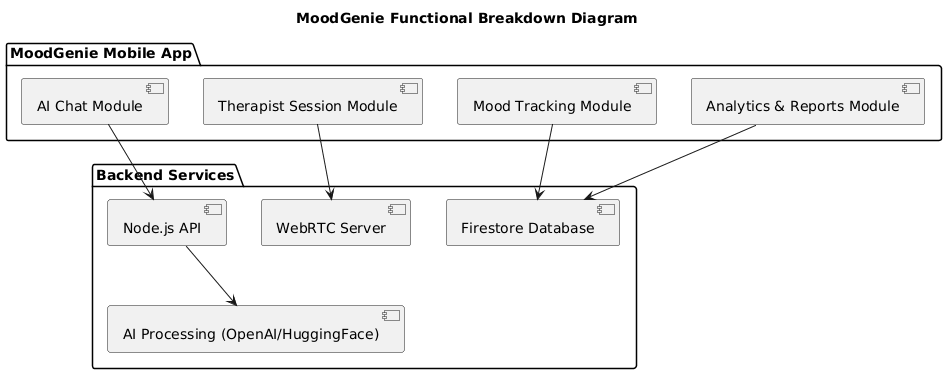
* **Notification and Reminder System**

Sends reminders for mood logging, upcoming therapy sessions, and self-care activities using Firebase Cloud Messaging.

* **Data Privacy and Consent Management**

Allows users to control access permissions, delete chat histories, and manage what data therapists can view.

* **Therapist Dashboard Access (Web)**
* Therapists can securely access user-consented mood logs, session notes, and schedule future sessions.
* The high-level functional architecture of MoodGenie is shown in Figure 2.
* It provides a clear overview of how the user interacts with the system’s core modules and how backend services support the core functionality.



*Figure 2: High-Level Functional Breakdown of MoodGenie*

## 2.3 User Classes and Characteristics

MoodGenie will be used by three primary user classes, each having different goals, permissions, and system interactions. These user groups vary in technical literacy, frequency of use, and security privileges.

### **2.3.1 General Users (Developers, Tech Students, Researchers)**

**Characteristics:**

* Basic to advanced technical literacy
* Use the app frequently (daily or weekly)
* Need emotional support, mood tracking, and therapy access
* Privacy-sensitive, require full control of data
* Largest and **most important** user class

**Functions They Use:**

* AI Chat Module
* Mood Tracking
* Analytics Dashboard
* Therapist Booking + Video/Text Sessions
* Profile & Consent Settings

### **2.3.2 Licensed Therapists**

**Characteristics:**

* Trained mental health professionals
* Access only when user consents
* Need to view mood reports, session history (limited), and conduct sessions
* Medium usage (sessions scheduled weekly/monthly)
* Higher security clearance

**Functions They Use:**

* Therapist Dashboard
* Appointment Management
* Secure WebRTC Session Panel
* Session Notes (only if user consents)

### **2.3.3 System Administrators**

**Characteristics:**

* High technical expertise
* Responsible for backend monitoring, therapist verification, and system security
* Low frequency of usage
* Highest privilege level

**Functions They Use:**

* Manage therapists
* View system analytics
* Maintain database rules
* Monitor API usage



## 2.4 Operating Environment

The MoodGenie system will operate in a modern mobile and cloud-based environment. The mobile application will run on both **Android (Android 10 and above)** and **iOS (iOS 13 and above)** devices equipped with minimum **2 GB RAM** [8], stable internet connectivity, and updated system libraries. The backend services will run on a **Node.js (v18+)** environment using **Express.js** frameworks, hosted on a cloud platform such as **Firebase** which provides Firestore Database, Authentication, and Firebase Cloud Messaging. The system will use **OpenAI APIs** or compatible AI models hosted on secure cloud servers for generating chatbot responses. Video and audio communication features will operate through **WebRTC** [9], requiring supported browsers and mobile OS-level permissions for microphone and camera. All system components must coexist with standard device services such as push notifications, secure storage, and background services without interfering with other applications.

## 2.5 Design and Implementation Constraints

The development of MoodGenie is subject to several design and implementation constraints. The system must be built using **Flutter** for cross-platform mobile development, which restricts UI components and performance optimizations to the Flutter framework [4]. The backend must use **Node.js and Express.js**, limiting the implementation to JavaScript or TypeScript environments. All data storage and authentication must rely on **Firebase services** [6], which imposes constraints on database structure, query performance, offline storage, and vendor-specific security rules. The application must comply with **privacy and security requirements**, including encrypted data transmission, secure handling of mental-health related information, and adherence to consent-based data sharing. Real-time audio and video sessions must use **WebRTC** [9], which requires device-level permissions and may face limitations on low-bandwidth networks. Integration with **OpenAI APIs or HuggingFace models** introduces constraints related to API rate limits [5], model response time, and token usage costs. The system must function on mobile devices with limited hardware resources, such as restricted memory, battery usage, and varying CPU performance. Additionally, the application must remain compatible with evolving OS policies for background services, notifications, and camera/microphone access.

## 2.6 User Documentation

The MoodGenie system will include essential user documentation to support onboarding, troubleshooting, and feature understanding. The delivered documentation will consist of an **in-app user guide**, **interactive onboarding tutorials**, and **FAQ/help-center content** accessible from the mobile application. A **web-based user manual** will also be provided for therapists and administrators, detailing account setup, session management, and dashboard usage. All documentation will follow standard digital help formats, including step-by-step instructions, annotated screenshots, and feature explanations. User support will be delivered in **English**, with the documentation optimized for mobile viewing and accessible through the application’s Help section.

## 2.7 Assumptions and Dependencies

The development of MoodGenie relies on several assumptions that may affect system requirements if they change. It is assumed that stable versions of **Flutter** [4], **Node.js** [10], **Firebase services** [6], and **OpenAI APIs** [5] will remain available throughout the project lifecycle. The system assumes consistent internet connectivity for AI chat responses, therapist sessions using WebRTC, and real-time database updates. It is also assumed that licensed therapists will have access to compatible web browsers and stable internet for using the therapist dashboard. The project depends on third-party components such as Firebase Authentication, Firestore Database, OpenAI APIs, and WebRTC libraries [9]. Any change in pricing, API policies, or service availability of these platforms may affect system functionality, cost, or design decisions. Additionally, the system assumes user devices will run on supported versions of **Android** or **iOS**, and that the backend server will operate in a secure cloud environment compatible with project requirements.

# **3. External Interface Requirements**

## 3.1 User Interfaces

The MoodGenie system will include three primary user interfaces: **Mobile Application Interface** for end-users, **Therapist Dashboard Interface** for licensed therapists, and an **Admin Panel** for system monitoring. The mobile app will use a clean, minimal, and intuitive layout following Material Design 3 standards to ensure usability on both Android and iOS devices. The interface will provide standard navigation elements such as a bottom navigation bar, floating action button for quick mood check-ins, and consistent screen layouts with uniform colors and typography. All screens will include standard elements such as a back button, error message display area, and help icon linking to in-app guidance.

The AI chat screen will support text input, quick action buttons, and message bubbles similar to modern messaging apps. The mood tracker interface will present graphs and mood logs in a structured layout using Flutter chart components. The therapist booking interface will include a calendar view, session selection buttons, and confirmation prompts. The therapist dashboard will provide a web-based UI with role-based access, a left-side navigation menu, session management screens, and consent-based user reports. Error messages will follow consistent formatting, stating the issue and recommended action. Accessibility considerations will include readable font sizes, contrast-compliant colors, and simple navigation structures.

## 3.2 Hardware Interfaces

MoodGenie will operate primarily on **mobile devices** and **web browsers**, requiring no specialized hardware. The system will interface with the following hardware components:

• **Smartphones and Tablets:**

Compatible with Android (v8.0 or higher) and iOS (v13 or higher). The app interacts with touchscreens, vibration motors (for notifications), microphone, and front-facing camera for WebRTC-based therapist video sessions.

• **Therapist Devices (Laptops/PCs):**

The therapist dashboard will run on standard desktop or laptop hardware supporting any modern browser with WebRTC capability.

• **Camera and Microphone Access:**

Required for secure video therapy calls. Interaction includes real-time media capture, encoding, and transmission using WebRTC.

• **Network Connectivity:**

Wi-Fi or mobile data is required to transmit chat messages, AI requests, therapist session streams, and mood log synchronization.

• **Push Notification Hardware:**

* Uses device-specific notification systems (Android FCM module and iOS APNs) to deliver alerts for sessions and reminders.
* No other direct hardware communication is needed apart from OS-managed device sensors and network interfaces.

## 3.3 Software Interfaces

MoodGenie interacts with several external software systems, APIs, and libraries essential for its operation:

• **Operating Systems:**

* Android (v8+), iOS (v13+).
* Web browsers (Chrome, Edge, Firefox, Safari) for therapist dashboard.

The application follows OS-level permissions for camera, microphone, notifications, and secure storage.

• **Backend Server (Node.js + Express):**

Provides RESTful APIs for authentication, mood logs, AI requests, session scheduling, and user profile management. Handles JSON-based requests and responses via HTTPS.

• **Firebase Services:**

* **Firebase Authentication:** For secure login via email, phone, or OAuth.
* **Firestore Database:** Stores user profiles, mood logs, therapist schedules, and chat summaries.
* **Firebase Cloud Messaging (FCM):** Sends push notifications.

Data is exchanged in structured JSON format.

• **OpenAI API:**

Used for generating empathetic chatbot responses. Inputs include user messages and mood context. Outputs are model-generated text responses, transmitted through secure HTTP requests.

• **HuggingFace Models:**

Used for sentiment analysis and risk detection. The app sends text samples to the model endpoint and receives classification results (positive, neutral, negative, risk flags).

• **WebRTC Module:**

Handles secure peer-to-peer video and audio streaming between user and therapist. Requires STUN/TURN servers to establish reliable communication.

• **Flutter Framework (Frontend):**

Uses Dart libraries for UI rendering, HTTP communication, charting packages for mood analytics, and local data storage.

• **Admin Panel Tools (Optional):**

May use Firebase Console for monitoring logs, analytics, and backend rules.

All interactions use secure HTTPS communication, follow JSON data structures, and comply with platform API requirements. No global data sharing is required beyond authenticated API transactions, and all stored data is encrypted in Firestore according to Firebase security rules.

## 3.4 Communications Interfaces

MoodGenie relies on secure, real-time, and reliable communication channels to support chat processing, data synchronization, and therapist video sessions. The communication requirements are as follows:

• **Network Protocols:**

The system uses **HTTPS** for all API communication between the mobile app, backend server, and Firebase services. WebRTC utilizes **SRTP** (Secure Real-Time Protocol) for encrypted audio and video transmission.

• **Message Formatting:**

All client–server interactions follow standardized **JSON** message formats for sending and receiving data such as chat queries, mood logs, authentication responses, and therapist schedules.

• **Real-Time Communication:**

WebRTC is used for peer-to-peer therapist video calls. It requires **STUN/TURN servers** to establish reliable connections, especially behind firewalls or NAT networks.

• **Data Synchronization:**

Firestore supports real-time data syncing for chat summaries, mood entries, session updates, and therapist availability. Changes are propagated instantly to users and therapists.

• **Push Notifications:**

Firebase Cloud Messaging (FCM) is used to deliver real-time alerts, including session reminders, mood check-in prompts, AI responses (if needed), and analytics notifications.

• **Security and Encryption:**

* All communications follow **TLS 1.2+ encryption**.
* WebRTC streams are protected using **DTLS-SRTP**.
* User authentication uses Firebase secure tokens.
* Sensitive messages such as therapy notes and mood analytics are encrypted at rest and in transit.

• **Supported Bandwidth:**

* Minimum 1–2 Mbps recommended for smooth video sessions.
* Standard mobile network speeds (3G/4G/5G or Wi-Fi) are sufficient for chat and mood tracking features.

• **Browser Compatibility:**

* The therapist dashboard supports WebRTC-enabled browsers, including Chrome, Firefox, and Safari.
* MoodGenie ensures reliable, secure, and privacy-focused communication across all modules, enabling seamless interaction between users, AI services, and therapists.

# **4. System Features**

## 4.1 AI Emotional Support Chat

### **4.1.1 Description and Priority**

This feature provides users with a 24/7 empathetic AI chatbot for emotional support, mood check-ins, journaling prompts, and coping strategies based on CBT and mindfulness.

Priority: **High**

### **4.1.2 Stimulus/Response Sequences**

* User opens the AI chat screen
* System loads previous chat history
* User sends a message
* System forwards message to OpenAI API
* AI generates a response
* App displays the response and logs mood indicators
* User selects a self-care activity
* System displays guided breathing or journaling prompts

### **4.1.3 Functional Requirements**

* **REQ-AI-1:** The system shall send user messages to the AI model and return an empathetic response.
* **REQ-AI-2:** The system shall maintain chat history for each user.
* **REQ-AI-3:** The system shall classify messages for sentiment and risk using HuggingFace models.
* **REQ-AI-4:** The system shall notify the user if the message indicates emotional distress.
* **REQ-AI-5:** The system shall gracefully handle API failures and display fallback responses.

## 4.2 Therapist Booking & Video Sessions

### **4.2.1 Description and Priority**

This feature allows users to book licensed therapists and conduct encrypted video or text sessions through WebRTC.

Priority: **High**

### **4.2.2 Stimulus/Response Sequences**

* **User selects a therapist**
* System displays available time slots
* User books a slot
* System sends confirmation and adds to therapist dashboard

• **At session time**

* System launches WebRTC call
* Both sides connect securely
* System logs session duration

### **4.2.3 Functional Requirements**

* **REQ-THER-1:** The system shall allow users to view therapist profiles and availability.
* **REQ-THER-2:** The system shall allow users to book, cancel, or reschedule sessions.
* **REQ-THER-3:** The system shall support encrypted real-time video calls via WebRTC.
* **REQ-THER-4:** The system shall notify both user and therapist before session start.
* **REQ-THER-5:** The system shall prevent access to video sessions without authentication.

## 4.3 Mood Tracking & Analytics

### **4.3.1 Description and Priority**

This feature logs daily emotional states and generates weekly and monthly analytics for stress trends.

Priority: **Medium–High**

### **4.3.2 Stimulus/Response Sequences**

* User logs mood
* System stores entry and updates charts
* User opens analytics
* System fetches emotion history
* System renders graphs and insights

### **4.3.3 Functional Requirements**

* **REQ-MOOD-1:** The system shall allow users to record daily mood with emojis, text, and optional notes.
* **REQ-MOOD-2:** The system shall generate visual charts for weekly and monthly trends.
* **REQ-MOOD-3:** The system shall store mood logs in Firestore.
* **REQ-MOOD-4:** The system shall allow exporting reports to PDF.
* **REQ-MOOD-5:** The system shall ensure no analytics is generated without valid data.

## 4.4 User Authentication & Privacy Controls

### **4.4.1 Description and Priority**

This feature ensures secure login, data encryption, and user control over privacy and consent.

Priority: **High**

### **4.4.2 Stimulus/Response Sequences**

* **User opens the app**
* System loads login screen
* User signs in
* System verifies credentials through Firebase Auth
* System redirects to dashboard

• **User manages privacy**

* System applies settings (delete data, restrict access)

### **4.4.3 Functional Requirements**

* **REQ-AUTH-1:** The system shall authenticate users using Firebase Authentication.
* **REQ-AUTH-2:** The system shall encrypt all sensitive data in transit (TLS 1.2+).
* **REQ-AUTH-3:** The system shall allow users to delete chat history and logs.
* **REQ-AUTH-4:** The system shall restrict therapist access unless consent is given.
* **REQ-AUTH-5:** The system shall log out users automatically after inactivity.

## 4.5 Notification & Alerts System

### **4.5.1 Description and Priority**

Sends reminders for mood logging, therapy sessions, and self-care tasks.

Priority: **Medium**

### **4.5.2 Stimulus/Response Sequences**

* System detects upcoming session → Sends reminder
* User misses mood log → Sends gentle alert

### **4.5.3 Functional Requirements**

* **REQ-NOTIF-1:** System shall send push notifications through FCM.
* **REQ-NOTIF-2:** System shall remind users of booked sessions.
* **REQ-NOTIF-3:** System shall notify about mood logging.
* **REQ-NOTIF-4:** System shall allow users to mute notifications.
* **REQ-NOTIF-5:** System shall show offline alerts when internet reconnects.

## 4.6 User Profile Management

### **4.6.1 Description and Priority**

Allows users to edit personal information and adjust wellness preferences.

Priority: **Medium**

### **4.6.2 Stimulus/Response Sequences**

* User opens profile → Updates data → System saves changes

### **4.6.3 Functional Requirements**

* **REQ-PROFILE-1:** System shall allow editing of profile information.
* **REQ-PROFILE-2:** System shall store profile data in Firestore.
* **REQ-PROFILE-3:** System shall allow users to set wellness goals.
* **REQ-PROFILE-4:** System shall sync changes across devices.

## 4.7 Therapist Dashboard (Web)

### **4.7.1 Description and Priority**

Web portal for therapists to manage sessions, view consented reports, and write notes.

Priority: **Medium–High**

### **4.7.2 Stimulus/Response Sequences**

* Therapist logs in → Views upcoming sessions
* Therapist opens patient → Views consent-based mood analytics

### **4.7.3 Functional Requirements**

* **REQ-TD-1:** System shall authenticate therapists separately.
* **REQ-TD-2:** System shall show booked sessions in calendar format.
* **REQ-TD-3:** System shall allow viewing analytics only if user consent exists.
* **REQ-TD-4:** System shall allow therapists to write session notes.
* **REQ-TD-5:** System shall support updating availability slots.

## 4.8 Admin Panel

### **4.8.1 Description and Priority**

Admin can manage therapists, users, and platform configuration.

Priority: **Low (Post-MVP)**

### **4.8.2 Stimulus/Response Sequences**

* Admin logs in → Views user statistics
* Admin verifies therapist documents

### **4.8.3 Functional Requirements**

* **REQ-ADMIN-1:** System shall authenticate admins with elevated privileges.
* **REQ-ADMIN-2:** System shall allow adding or disabling therapist accounts.
* **REQ-ADMIN-3:** System shall allow viewing platform usage analytics.
* **REQ-ADMIN-4:** System shall manage system-wide configuration.

## 4.9 Settings & Customization

### **4.9.1 Description and Priority**

User can customize appearance, AI tone, notification preferences.

Priority: **Low–Medium**

### **4.9.2 Stimulus/Response Sequences**

* User opens settings → Updates theme/tone → App applies instantly

### **4.9.3 Functional Requirements**

* **REQ-SET-1:** System shall allow theme switching (light/dark).
* **REQ-SET-2:** System shall allow setting AI conversational tone.
* **REQ-SET-3:** System shall allow notification customization.
* **REQ-SET-4:** System shall apply settings app-wide.

# **5. Other Nonfunctional Requirements**

## 5.1 Performance Requirements

* The AI chat feature shall respond to user messages within **3–5 seconds** under standard load.
* Mood logging and mood analytics shall update within **2 seconds** of user input.
* The dashboard must load within **3 seconds** after login.
* The app shall function smoothly on devices with at least **3 GB RAM** and **Android 8+ / iOS 12+**.
* WebRTC video call must maintain **minimum 720p quality**, with latency below **200ms** on a **5 Mbps+** internet connection.
* API calls to OpenAI and backend shall complete within **2–4 seconds** on average.
* Push notifications via Firebase Cloud Messaging shall be delivered within **2 seconds**.
* The system shall support:
* **10,000+ concurrent users (horizontal scaling)**
* **100 AI chat requests/second**
* **100 therapist sessions** running parallelly
* Offline mode shall allow the user to view previous chats and logs stored locally.
* All analytics charts must be generated within **4–6 seconds**.

## 5.2 Safety Requirements

* The system shall not provide harmful, triggering, or unsafe suggestions through the AI chatbot.
* High-risk messages (self-harm, panic, suicidal thoughts) must trigger **crisis alert responses** and guide users toward helplines.
* In case of video call interruption, the system must safely terminate the session without exposing private data.
* The app shall prevent users from deleting their account during an active therapy session.
* Emergency contacts and crisis instructions must be accessible from the sidebar.
* No PII (Personal Identifiable Information) shall appear automatically during calls.
* The AI model shall not store sensitive data; it shall only process the prompt temporarily.
* The app must prevent accidental sharing of data without user consent.
* Users must confirm twice before deleting chat history or mood logs.
* Therapist content must follow ethical guidelines for tele-mental-health.

## 5.3 Security Requirements

**Authentication**

* Firebase Authentication must be used for login (email/password, phone, or Google login).
* Accounts must lock after **5 failed attempts** (brute-force protection).
* Two-Factor Authentication (2FA) may be optionally added.

**Data Encryption**

* All communication must use **TLS 1.2+ HTTPS encryption**.
* Firestore must encrypt data at rest.
* WebRTC must use **DTLS + SRTP** for encrypted calls.
* User PII and chat history must be encrypted before saving.

**Access Control**

* Role-Based Access Control (RBAC):
* User
* Therapist
* Admin
* Therapists cannot access mood analytics without explicit user consent.
* Admins cannot read user chats (zero-knowledge design).

**API Security**

* All API keys must be stored in environment variables.
* Rate limiting must be applied to prevent abuse (max 10 requests/second per user).
* OpenAI API requests must not store identifying data.

**Privacy**

* Users must have the right to delete their data (“Right to be Forgotten”).
* The system must comply with **GDPR principles** for transparency and data minimization.
* Regular security audits must be performed every 3 months.

## 5.4 Software Quality Attributes

**Usability**

* The UI must be clean, simple, and beginner-friendly.
* The app must follow **Material Design 3** guidelines.
* Onboarding must be completed within **2 minutes**.
* Buttons, icons, and labels must be readable and accessible.
* Dark mode and light mode must be available.
* Critical actions must show confirmation prompts.

**Reliability**

* The application must achieve **99% uptime**.
* AI services must gracefully degrade if OpenAI API limit is reached.
* App shall auto-retry lost network requests.
* No data corruption shall occur during crashes.

**Availability**

* AI chat and mood logging must remain available offline using cached data.
* The backend shall support load balancing for high availability.
* Scheduled maintenance downtime must not exceed **2 hours per month**.

**Maintainability**

* Code must follow Clean Architecture and separation of concerns.
* Documentation for each module must be provided.
* Updating the AI model or backend must not require full app redesign.
* Modular structure (Flutter widgets, Node.js controllers) must allow easy extension.

**Portability**

* App must run on:
* Android 8 to latest
* iOS 12 to latest
* The backend must deploy smoothly on:
* Google Cloud
* AWS
* Local servers supporting Node.js

**Scalability**

* Horizontal scaling must be supported using Firebase auto-scaling.
* Node.js backend must support clustering.
* Database queries must be optimized to reduce read/write costs.

**Interoperability**

* The app must integrate smoothly with:
* Firebase Authentication
* Firestore
* OpenAI API
* HuggingFace models
* WebRTC
* Payment Gateway (future expansion)

**Testability**

* 80% of backend functions must have unit tests.
* Flutter widgets must include integration tests.
* The system must support automated testing pipelines using GitHub Actions.
* QA team must test correctness of mood analytics and therapist session flow.

## 5.5 Business Rules

* Only registered users can access the AI companion, mood tracking, and therapist booking features.
* Therapists must verify their identity and professional license before account approval.
* Therapists may access a user’s mood analytics only when the user provides explicit, time-bound consent.
* Admin users can manage therapist verification requests but cannot view user chats or mood logs.
* AI-generated responses must follow safe-content guidelines; the AI must not provide harmful clinical advice.
* Users must complete authentication before performing sensitive actions such as video calls, chat access, or report exports.
* Users can delete their chat history, but this action must require confirmation to prevent accidental deletion.
* Therapist sessions must be booked in available time slots only, and overlapping appointments are not allowed.
* The system must prevent therapists from initiating a session without the user’s approval.
* All financial payments (if added in future versions) must follow secure payment standards and must be logged for auditing.
* Crisis or high-risk messages must automatically trigger a predefined safety response and cannot be skipped by the AI module.
* Mood logs must be timestamped and cannot be modified after submission; only new entries may be added.
* Recommendations generated by the system must be non-clinical and must not replace professional diagnosis.
* Users must agree to the privacy policy before account creation, and withdrawal of consent will disable all system features.
* Role-based access control must ensure that each user only accesses features appropriate to their role (User, Therapist, Admin).

## 5.6 Detailed Subsystem Design

This section provides a detailed description of the major subsystems of the MoodGenie application. Each subsystem is broken down into its core responsibilities, internal components, and interactions with other parts of the system. The diagrams included within this section illustrate the system’s structure, behavior, and data flow across different stages of operation. The subsystem descriptions align with the system architecture defined earlier in this SRS.

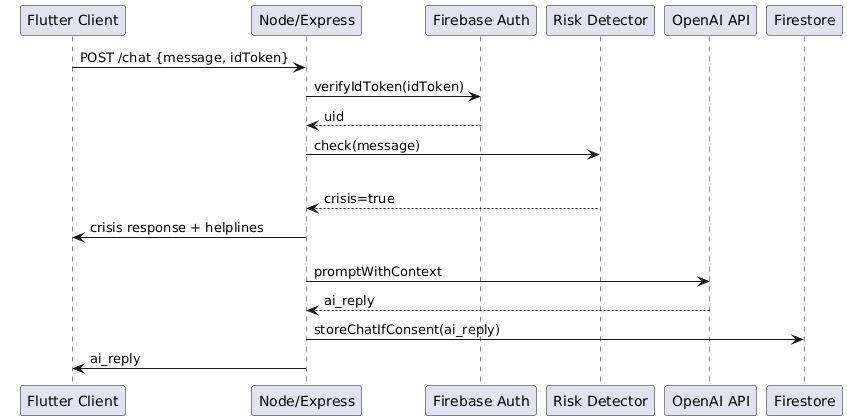
### **5.6.1 Chat & AI Processing Subsystem**

This subsystem manages all interactions between the user and the AI assistant. It receives user messages, performs safety checks, communicates with external AI models (OpenAI and HuggingFace), and stores chat logs depending on the user’s consent settings.

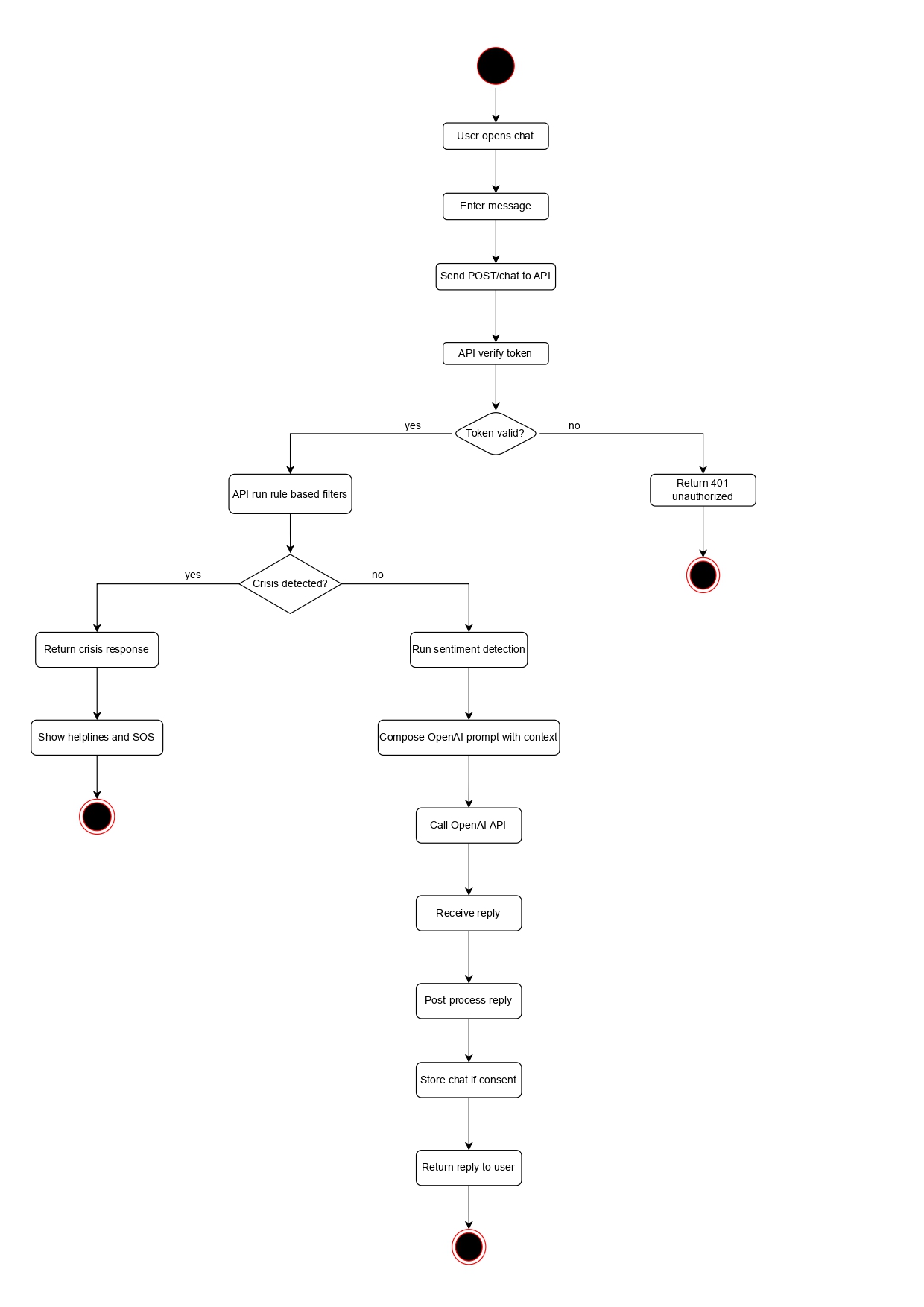
**Responsibilities**

* Receive chat input from the Flutter client
* Verify Firebase authentication token
* Execute rule-based filters and sentiment/risk detection
* Generate an AI response using OpenAI
* Store conversation logs (if consented)
* Return processed responses to the client

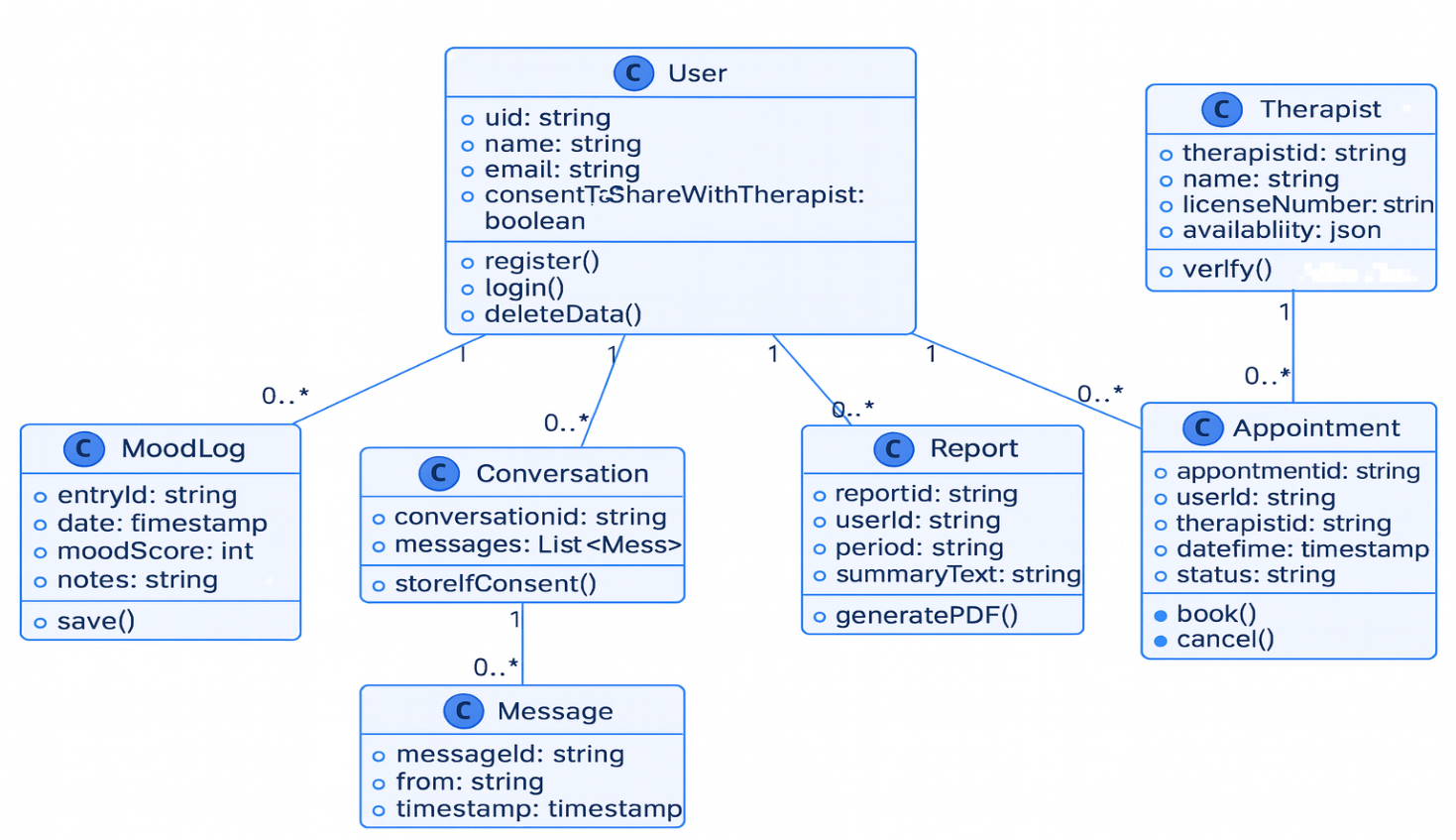
**Sequence Diagram – AI Chat Flow**



**Activity Diagram – Chat Processing**



**Class Diagram – Chat & Message Models**



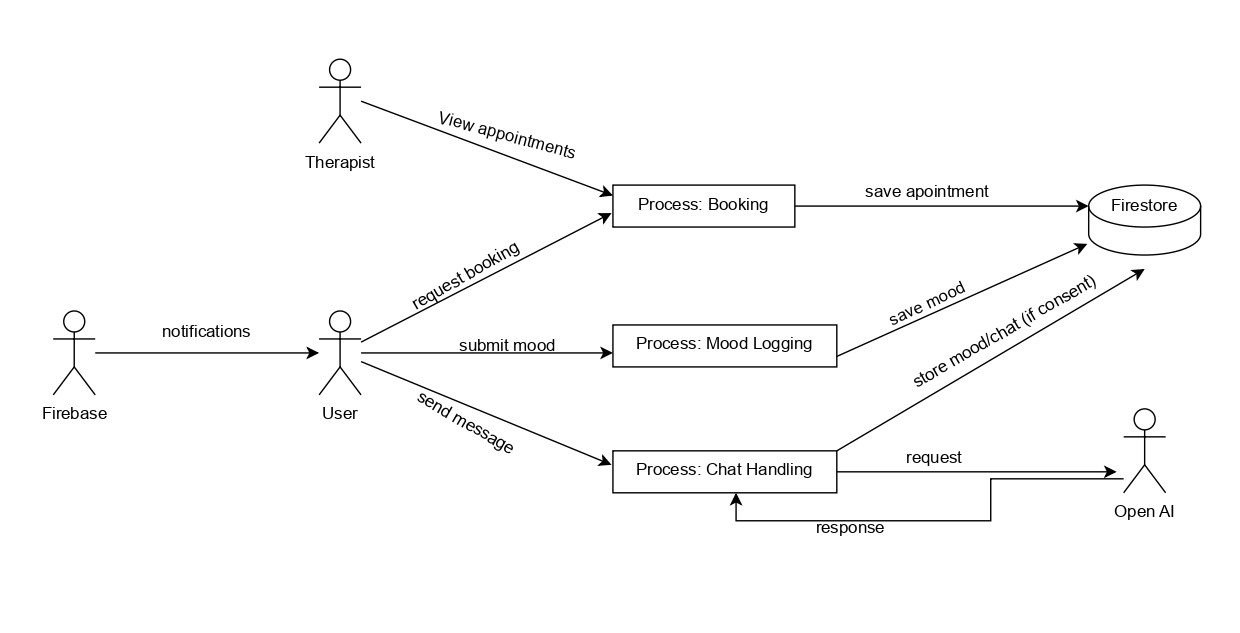
### **5.6.2 Mood Logging & Analytics Subsystem**

This subsystem handles the collection, storage, and processing of mood logs submitted by users. It also generates weekly and monthly analytical summaries through scheduled backend tasks.

**Responsibilities**

* Receive daily mood entries from user
* Save mood logs in Firestore subcollections
* Aggregate historical data for trends
* Generate insight reports (PDF/JSON)
* Support therapist access based on user consent

**DFD Level 1 – Mood Logging Flow**



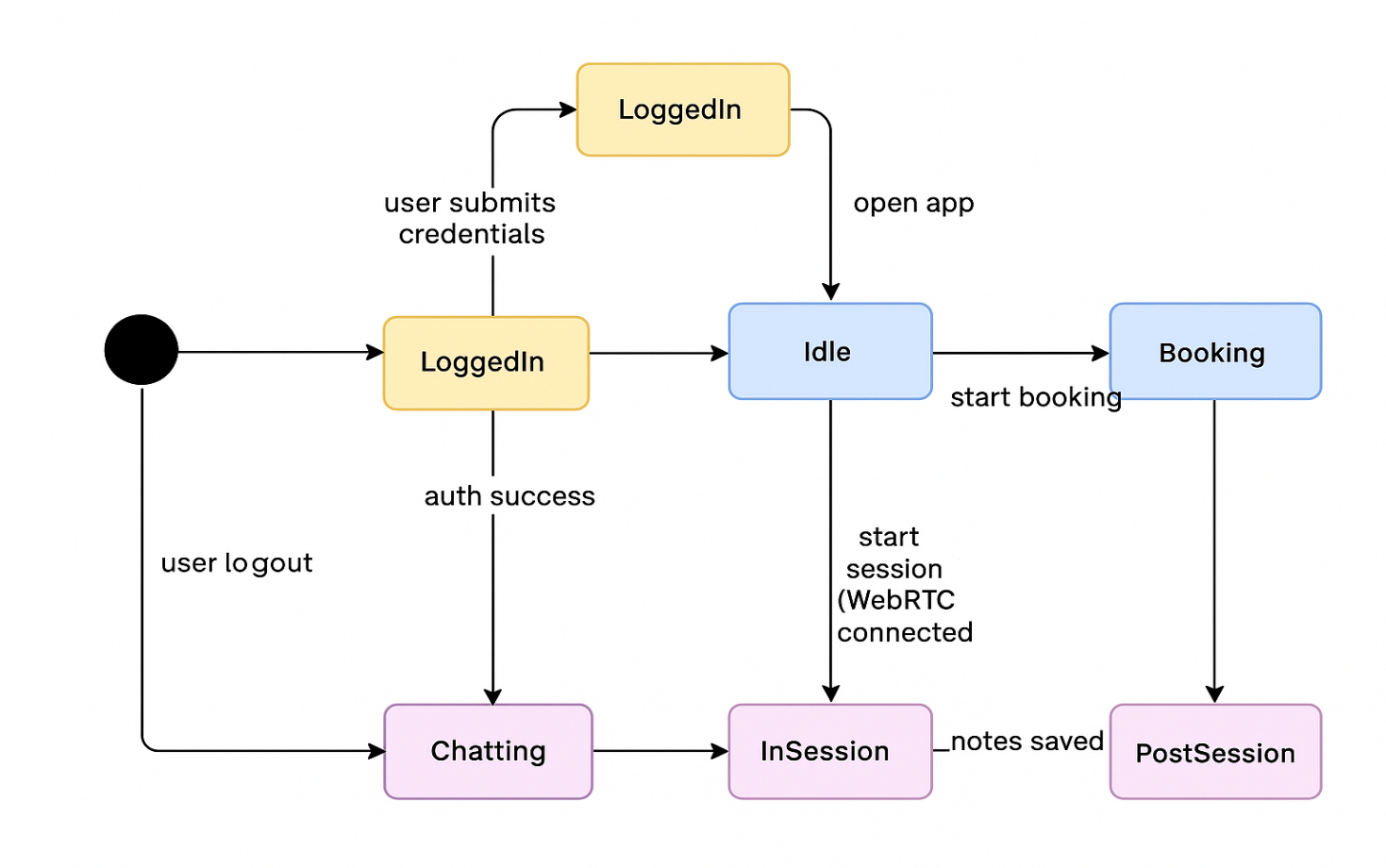
### **5.6.3 Therapist Booking & Session Subsystem**

This subsystem manages therapist availability, appointment scheduling, payment verification, and WebRTC signaling for secure video or text sessions.

**Responsibilities**

* Fetch therapist availability
* Handle booking requests and cancellations
* Manage payment verification
* Support WebRTC signaling (ICE, SDP exchange)
* Maintain session states and therapist notes

**State Machine – Session Lifecycle**



### **5.6.4 Authentication & User Management Subsystem**

Handles user account creation, authentication, profile management, and data-access permissions.

**Responsibilities**

* Firebase Authentication for signup/login
* Manage sessions and access tokens
* Control consent settings
* Manage profile attributes and preferences

### **5.6.5 Notification Subsystem**

This subsystem delivers reminders, therapist notifications, and system alerts through Firebase Cloud Messaging (FCM).

**Responsibilities**

* Register device tokens
* Deliver periodic reminders (mood check-ins, session alerts)
* Push appointment confirmations/cancellations
* Push crisis alerts if flagged

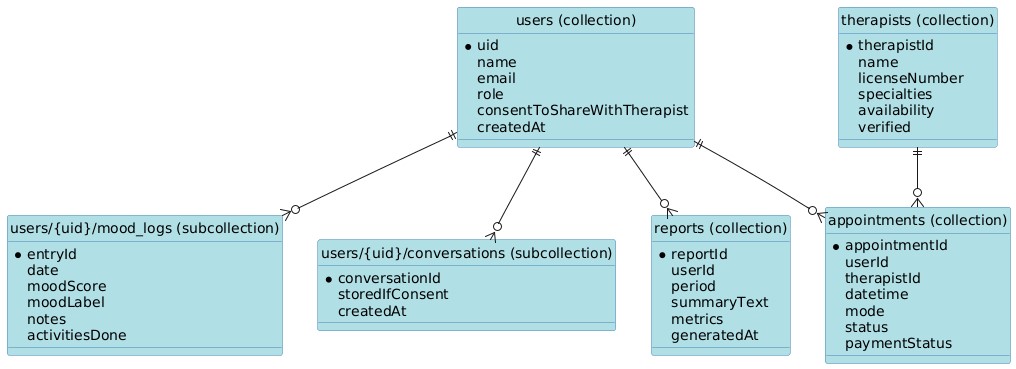
### **5.6.6 Database Management Subsystem**

This subsystem defines how structured and unstructured data is stored across Firestore collections and subcollections, ensuring consistency, privacy, and secure access.

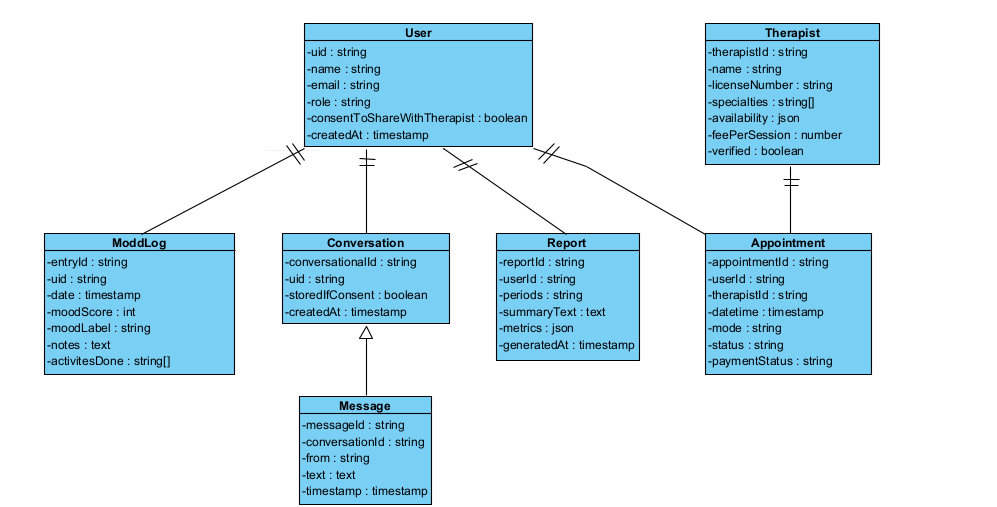
**Responsibilities**

* Define Firestore collections (users, therapists, appointments, reports, moods)
* Manage NoSQL document structure
* Enforce access rules via Firebase Security Rules
* Handle backup, retrieval, and deletion

**Database Diagram – Firestore Collections**



**ER Diagram – Full System Entities**



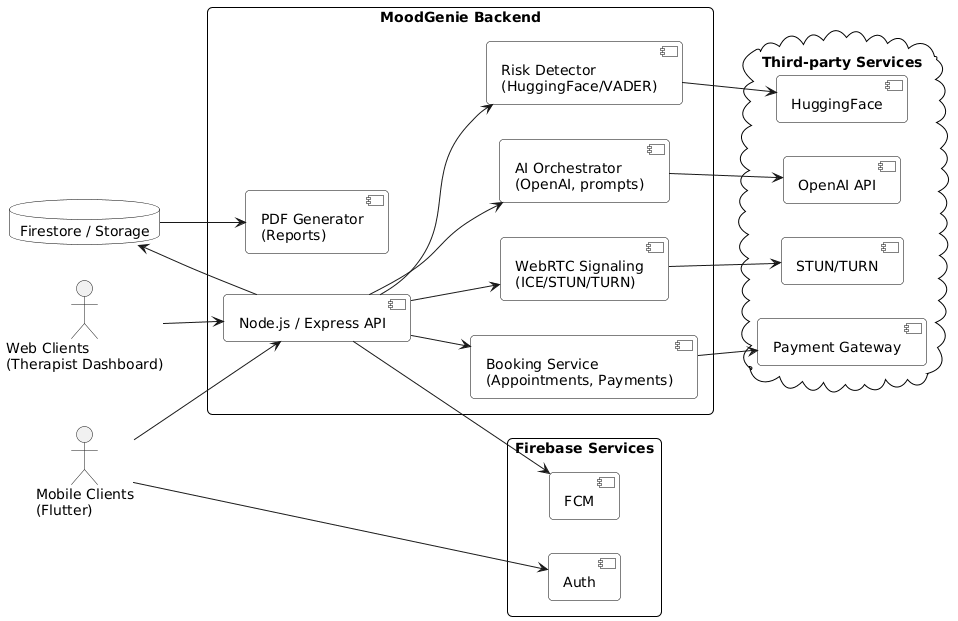
### **5.6.7 System Architecture Subsystem**

This subsystem outlines the complete structural layout of MoodGenie including all external services like Firebase, OpenAI, HuggingFace, and payment gateways.

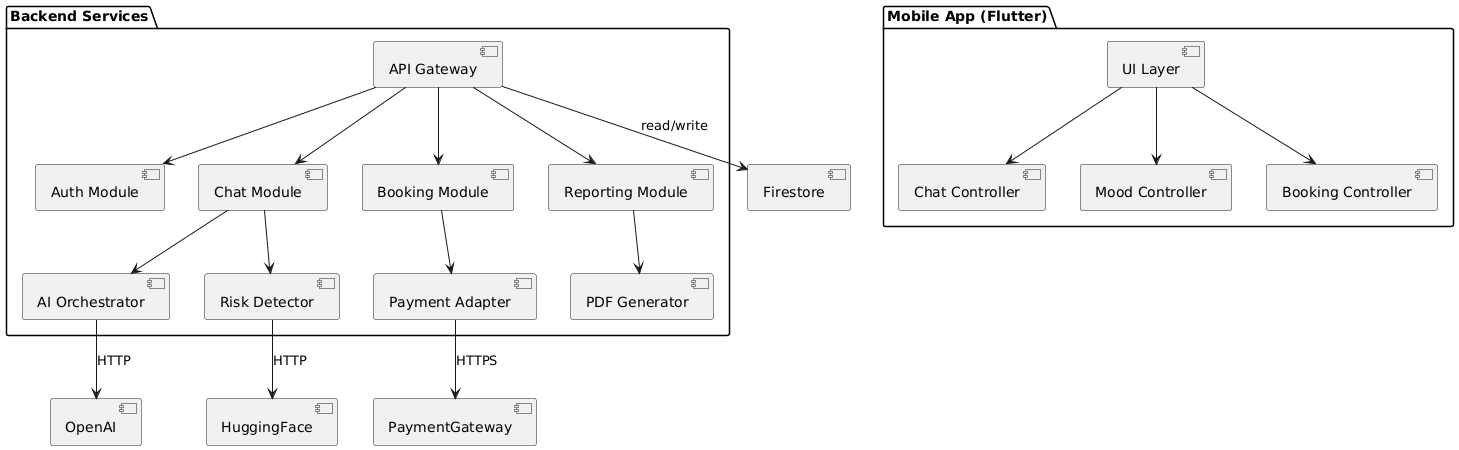
**Responsibilities**

* Define boundaries between client, backend, cloud services
* Manage communication protocols (HTTPS, WebRTC)
* Handle API routing using Express.js
* Manage external integrations

**Architectural Diagram – High-Level System Structure**



**Component Diagram – Backend & External Systems**



### **5.6.8 Application Frontend Subsystem (Flutter)**

Defines the UI/UX layers of the mobile client (Android/iOS) including screens, navigation, state management, and API communication.

**Responsibilities**

* Render UI screens (chat, mood logs, booking, reports)
* Manage state using provider/bloc (your choice)
* Interact with backend APIs
* Validate user input
* Provide accessibility-friendly UI

# **References**

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>