Mood Manager

AI-Based Personalized Mood Management System for Smart Homes

Hyeokjin Ma
Dept. of Information Systems
Hanyang University
Seoul, Republic of Korea
tema0311@hanyang.ac.kr

Hyunwoo Choi
Dept. of Information Systems
Hanyang University
Seoul, Republic of Korea
hhyyrr0713@hanyang.ac.kr

Junseong Ahn
Dept. of Information Systems
Hanyang University
Seoul, Republic of Korea
lljs1113@hanyang.ac.kr

Saeyeon Park
Dept. of Information Systems
Hanyang University.
Seoul, Republic of Korea
saeyeon0317@hanyang.ac.kr

Heejoo Chae
Dept. of Information Systems
Hanyang University
Seoul, Republic of Korea
heeju0203@hanyang.ac.kr

Abstract—**Smart homes today offer convenience and automation, yet their fragmented control of lighting, fragrance, and sound leaves the overall spatial experience disjointed and reactive. Existing systems focus on functional operation but lack the ability to anticipate user needs or create holistic, personalized environments. To address this gap, we present Mood Manager, an AI-driven virtual device that integrates ambient elements—light, scent, and audio—into a unified, context-aware system. Using few-shot prompting as its personalization engine, Mood Manager learns user preferences dynamically through real-time interactions, requiring no dedicated model retraining. By combining contextual awareness, feedback logging, and IoT-based prototyping, the system not only responds to explicit commands but also proactively curates atmospheres tailored to user states and environmental conditions. Over time, Mood Manager evolves into an autonomous "mood curator," transforming ordinary spaces into hyper-personalized thera-peutic environments. This approach demonstrates how hyper personalization, powered by lightweight AI techniques, can redefine smart home experiences from passive control toward active well-being enhancement.**

TABLE I: Role Assignments

Roles	Name	Task description and etc.
Software Developer (Full-stack)	Hyeokjin Ma	Manages the overall integration between front-end and back-end components to ensure stability, performance, and consistency of the system. Develops and maintains the data flow structure that connects the user interface, server processes, and AI modules. Oversees the planning and deployment of core features, identifies potential bottlenecks, and ensures that each module communicates effectively. Serves as a liaison across the team to coordinate development timelines, maintain data standards, and establish quality assurance procedures. Delivers architecture diagrams, integration reports, and performance improvement summaries.
Software Dveloper (Front-end)	Hyunwoo Choi	Designs and implements the overall user interface of the system with a focus on usability, clarity, and consistency. Develops interactive layouts that support mood recommendations, feedback collection, and device control in an intuitive way. Ensures accessibility and visual coherence across different plat-forms while continuously improving user experience through testing and iteration. Collaborates closely with the full-stack and back-end developers to maintain seamless communication between the interface and the system logic. Produces design documentation, component libraries, and user interaction reports.
Software Developer (Back-end)	Junseong Ahn	Leads the development and maintenance of the server-side architecture that manages data, security, and scalability. Designs the core database structure and develops application logic for user management, mood control, and feedback processing. Oversees data integration from internal and external sources to support personalized mood recommendations. Coordinates with AI developers to ensure stable data flow to learning modules and with the full-stack developer for API integrity and performance optimization. Produces backend specifications, operational guidelines, and system monitoring reports.
AI Developer	Saeyeon Park	Works on implementing few-shot prompting and adaptive reasoning using OpenAI-based models. Tunes retrieval-augmented generation (RAG) parameters to enhance relevance and continuity of responses. Converts multimodal insights—such as sensor readings, speech transcripts, and contextual cues—into structured JSON inputs for inference. Cooperates closely with the multimodal AI developer to integrate contextual information into the reasoning pipeline, and with back-end developers to align API communication and data formats. Produces experiment reports, RAG tuning summaries, and optimization plans for improved user adaptation.
AI Developer	Heejoo Chae	Focuses on integrating multimodal AI capabilities that combine biometric, voice, and environmental data from wearable devices into a cohesive user understanding model. Collects and preprocesses physiological signals, audio inputs, and contextual information to support adaptive personalization. Collaborates with back-end developers to manage real-time data flow and with the other AI developer to ensure that multimodal inputs are effectively interpreted within the recommendation system. Contributes to improving system awareness and context interpretation while maintaining stability and user privacy.

I. INTRODUCTION

A. Motivation

1) The Rise of Hyper-Personalization in AI

Artificial intelligence has transitioned from basic automation to highly adaptive and user-centric systems. Early applications of AI mainly focused on efficiency and scalability, offering generalized solutions designed for broad user groups. However, with the introduction of large language models (LLMs) and generative AI, the paradigm has shifted toward experiences that dynamically adapt to individuals. Users now expect technology not only to execute tasks but also to understand personal preferences, anticipate needs, and respond in intuitive, human-like ways. This shift marks the emergence of hyper-personalization — the ability to tailor services to the unique emotional and behavioral patterns of each user.

Among recent personalization techniques, few-shot prompting has become one of the most effective approaches for achieving real-time adaptation. Unlike traditional retraining or fine-tuning methods, few-shot prompting enables AI systems to learn user tendencies instantly through a small number of examples. This allows the model to refine its behavior continuously based on individual feedback without requiring extensive computation or model updates. As a result, AI evolves into a responsive companion capable of forming emotional resonance with users, delivering experiences that feel deeply personal and contextually aware — extending far beyond conventional automation.

2) The Current Gap in Smart Home Environments

a) Current State of Smart Home Systems

The global smart home market has expanded rapidly, offering consumers a wide range of connected devices that support automation, remote control, and energy management. From smart speakers and lighting systems to intelligent thermostats, these technologies have enhanced convenience and accessibility in daily life. However, most devices still operate independently, relying on user-initiated commands rather than seamless coordination between systems. Research indicates that while the adoption rate of smart devices is high, advanced features such as cross-device automation and contextual adaptation remain underutilized. More than half of users engage only with basic functions such as turning devices on or off, leaving advanced integrations unexplored. This reflects a fundamental gap between technological potential and the actual user experience—where automation exists, but intelligent interaction and emotional resonance do not.

b) Current Trends in the Home Fragrance Market

In parallel, the home fragrance market has shown remarkable growth, driven by consumer interest in wellness, self-care, and emotional comfort within living spaces. Scent has become an essential factor in shaping atmosphere and mood, complementing lighting and sound to create immersive environments. Recent developments include smart diffusers and app-controlled scent devices; however, these solutions largely remain standalone products with limited integration into broader smart home ecosystems. While lighting and audio technologies have achieved sophisticated automation and voice control, fragrance devices are still predominantly manual and lack context-aware functionality. This imbalance reveals an untapped opportunity—bridging the gap between emotional personalization and technological integration through intelligent, automated scent management.

c) Need for Integrated Multi-Sensory Control

To achieve a truly intelligent home environment, it is essential to technologically integrate lighting, sound, and fragrance into a single adaptive framework. Each sensory component influences human emotion in complementary ways: lighting affects focus and relaxation, sound shapes perception and mood, and fragrance modulates emotional comfort. By connecting these elements through AI-driven personalization and IoT synchronization, users can experience holistic, context-aware mood management rather than fragmented device control. This integration establishes the foundation for the proposed Mood Manager System, which coordinates sensory feedback and user interaction to deliver adaptive, emotionally intelligent smart home experiences.

3) Intelligent Mood Management System

Our project introduces the concept of the Mood Manager, a system designed to harmonize lighting, sound, and fragrance into a unified, intelligent experience. Unlike conventional smart home systems that require explicit commands, Mood Manager uses contextual cues and user feedback to infer preferences and deliver proactive adjustments. By employing few-shot prompting as its personalization engine, the system dynamically learns from minimal interaction data, adapting to individual users without the need for extensive retraining. This capability allows Mood Manager to evolve with each interaction, gradually curating an environment that feels increasingly attuned to the user's lifestyle and emotional needs.

The vision is to transform personal spaces into adaptive, therapeutic environments—"personalized therapy rooms" that respond seamlessly to mood and context. For example, on a rainy day, Mood Manager recalls the user's preferred warm lighting, calming fragrance, and soft background music, recreating a comforting atmosphere without requiring manual intervention. Ultimately, the project seeks to demonstrate how hyper-personalized AI, combined with IoT integration, can redefine smart homes as environments that enhance not only convenience but also emotional well-being and quality of life.

B. Problem Statement (Client's needs)

a. Fragmented Customization and Limited User Experience

Although smart home devices advertise convenience and personalization, the actual user experience remains fragmented. Lighting, fragrance, and sound—three core factors that shape the ambiance of a personal space—are often managed separately through individual applications or manual controls. According to market surveys, one of the top dissatisfaction factors among smart home users is the lack of seamless interoperability across devices. Users may configure settings individually but fail to fully leverage the advanced functions designed for cross-device scenarios. This leads to declining satisfaction despite increased device ownership. There is a clear need for improved customization and a differentiated user experience that can integrate these elements into a unified and meaningful interaction.

b. Static and Uniform Scenarios

Routine-based automation is a common feature in smart home systems, allowing users to schedule repetitive tasks such as turning on lights or playing background music at fixed times. While useful for simple patterns, such routines fall short in dynamic and unpredictable situations. For example, sudden changes in weather, irregular user moods, or unexpected guest visits are scenarios where rigid routines fail to adapt. Furthermore, current automation functions cannot flexibly manage complex, multi-sensory interactions across lighting, fragrance diffusion, and soundscapes. This limitation reveals the need for systems capable of real-time contextual awareness, leveraging AI and machine learning to adapt to diverse and complex living scenarios.

c. Overreliance on Passive and Manual Management

Despite their "smart" label, most home devices still require significant user intervention for configuration and operation. Users must decide how each device should behave in specific contexts, from adjusting an air conditioner's temperature to selecting fragrance intensity or curating playlists. Even when routines are available, users must manually design logical structures, which can lead to errors or inefficiencies. This overreliance on manual control burdens users with unnecessary complexity, keeping the focus on independent device operation rather than holistic optimization of the entire home environment.

d. Lack of Integrated Information and Orchestration

Smart devices individually generate rich data, such as energy consumption, device usage patterns, or even wellness-related metrics. However, this information is siloed within separate applications, requiring users to manually gather, analyze, and act on disparate data sources. For example, checking air quality data, lighting conditions, and ambient sound levels often involves switching between multiple apps without any overarching integration. The lack of information synthesis prevents users from receiving proactive and contextually relevant recommendations. Without an intelligent system to integrate and orchestrate these insights, the promise of a truly seamless smart home experience remains unfulfilled.

C. Research on Related Software

1. Home Assistant

Home Assistant is an open-source smart-home automation platform that allows users to connect and manage devices from multiple manufacturers in one interface. It supports automation through YAML scripts and plug-ins, enabling complex routines and data visualization dashboards. However, its automation remains rule-based rather than AI-adaptive, and it lacks mood-oriented personalization that learns user emotions or context dynamically.

2. openHAB (Open Home Automation Bus)

openHAB is a Java-based, vendor-neutral smart-home integration framework. It connects heterogeneous IoT devices through a modular "binding" system and provides local dashboards for user-defined automation rules. While it is powerful for interoperability, openHAB depends on manually defined logic and does not include AI-driven personalization or cross-sensory control such as lighting–sound–fragrance orchestration.

3. Philips Hue App + API

Philips Hue provides advanced smart-lighting control software that allows users to adjust brightness, color temperature, and dynamic scenes through a mobile app or API. It integrates with third-party platforms such as Spotify for light-music synchronization, creating partial multi-sensory experiences. Despite its mature lighting automation, Hue's software ecosystem lacks integration with fragrance or contextual AI personalization, limiting full environmental adaptation.

4. Pura Smart Fragrance App

Pura's mobile software manages Wi-Fi-connected fragrance diffusers, letting users schedule scents, monitor cartridge levels, and adjust intensity remotely. The app also connects Alexa and Google Home for voice-based control. Although Pura digitalizes scent management effectively, it operates as an isolated fragrance-control application and cannot coordinate scent with lighting or sound in an AI-driven manner.

5. Amazon Alexa Routines

Alexa Routines software enables users to automate tasks such as turning on lights or playing music based on time, sensor data, or voice triggers. It can execute multiple actions under a predefined "mood" routine (e.g., "Good Morning" or "Relax"). However, its logic is preprogrammed rather than contextual; it does not automatically adapt to environmental changes such as weather or user emotional state.

6. Google Home / Nest App

Google Home application serves as the control hub for devices powered by Google Assistant. It integrates voice commands, visual feedback, and multi-room media control, learning user habits to improve automation suggestions. While it demonstrates conversational intelligence, it mainly supports command execution and lacks true hyper-personalization or multi-sensory mood synthesis.

7. Recombee AI Recommendation Engine

   Recombee is a cloud-based recommendation-engine software that uses collaborative filtering and neural models to predict user preferences in real time. In the context of smart environments, its recommendation logic could inspire how the Mood Manager suggests optimal combinations of lighting, sound, and fragrance. Nevertheless, Recombee is oriented toward digital content recommendation rather than physical-environment orchestration.

8. Dynamic Yield Personalization Platform

   Dynamic Yield is an AI-driven personalization software that analyzes behavioral data to adapt content and optimize user experiences across digital interfaces. Its continuous learning framework and A/B testing tools illustrate how adaptive systems can refine personalization through feedback loops. Although designed for marketing applications, its approach parallels the Mood Manager's objective of refining environmental recommenddations based on user responses and context data.

II. Requirements

A. Sign up

Mood Manager requires essential personal and preference information to create a personalized smart-home profile. The system collects user credentials and baseline data to enable AI-driven customization from the first use.

1) Enter Phone Number

   The user must enter a valid mobile number, which is verified through an SMS authentication process. The verified phone number serves as a unique ID for future logins and device synchronization.

2) Create Password

   Users are required to create a password with at least eight characters, combining at least three of the following: uppercase letters, lowercase letters, numbers, and special characters. As users input their password, a real-time indicator displays its strength (Weak / Medium / Strong) to enhance security awareness.

3) Enter Name

   The user's name is collected and automatically set as the default nickname for their account. This name is also used by the AI for conversational interactions and natural responses.

4) Select Gender

   Gender selection is optional and used solely to improve fragrance and sound personalization (e.g., preferred scent tone, music genre). Users may skip this step if they prefer not to disclose the information.

5) Enter Birth Year

   The system requests only the birth year, not the full date of birth, to protect privacy while enabling broad personalization (e.g., age-based recommendation tendencies).

6) Agree to Data Usage & Privacy Policy

   Users must read and consent to the data usage policy, explaining how their environmental preferences and feedback logs are stored and used to improve AI personalization.

B. Sign In

Mood Manager provides flexible login options to ensure convenience and security for users.

1) Local Login through Mood Manager Account

   Users can log in by entering their registered phone number and password. If authentication is successful, a secure session token (JWT) is issued to maintain access throughout the session. In case of failure, a pop-up message such as "Invalid ID or password" is displayed with an option to reset the password.

2) Social Login via SNS Integration

   Mood Manager supports quick login through external services such as Google, KakaoTalk, or Apple ID. When users select a social login option, the system authenticates via the provider's OAuth 2.0 API and retrieves essential profile data (name, email, profile image). This data is automatically linked to the user's Mood Manager account.

3) Auto-Login and Session Management

   For convenience, the system offers an "Auto Login" toggle that securely stores encrypted session tokens on the device. When enabled, the user can access the system directly without repeated authentication, provided the session remains valid.

C. Device Registration

After signing in, users can register their smart devices and initialize personalization features in Mood Manager. Device registration serves as the foundation for AI-driven mood control, linking physical appliances with user preferences and environmental data. There are two stages: (1) registering connected IoT devices and (2) configuring personal and environmental settings for mood adaptation.

1) Device Registration

   Users can connect their smart devices to Mood Manager in two ways: manual device registration and automatic device discovery.

   a) Manual Registration via QR Code or Product Search

      Users can register devices by scanning the QR code printed on the product. Upon granting permission, the system automatically identifies the product model and connects it to the user's account. Alternatively, users may search by product name, serial number, or wireless protocol (Wi-Fi/Bluetooth).

   b) Automatic Device Discovery

      Mood Manager automatically detects compatible devices on the same local network using Wi-Fi or Bluetooth Low Energy (BLE) scanning. Detected devices are displayed in a list format, and users can select which ones to link to their personal environment.

   c) Device Grouping and Room Mapping

Once devices are registered, users can assign each device to a specific room or zone (e.g., bedroom, living room, study). This grouping enables synchronized mood control, allowing the AI to coordinate lighting, fragrance, and sound for each physical space.

2) Preference Survey

After device registration, users complete a short preference survey to initialize the AI's personalization engine. This survey helps the system understand basic mood-related tendencies and sensory preferences.

a) Lighting Preferences

Users select preferred brightness levels, color temperatures, and favorite tones (e.g., warm, cool, natural). The AI later uses this data to suggest appropriate lighting scenes based on time, weather, or activity.

b) Fragrance Preferences

Users choose from basic fragrance families such as floral, woody, citrus, or fresh. They can also indicate sensitivities or allergies (e.g., "avoid strong musk scents").

c) Sound Preferences

Users specify their preferred audio genres or soundscapes (e.g., lo-fi, jazz, ambient nature sounds) and desired volume levels during specific moods (e.g., focus, relaxation).

d) Mood Mapping

Users can connect moods (e.g., relaxed, focused, romantic) with their chosen lighting, fragrance, and sound combinations. These mappings act as initial training data for the AI's few-shot prompting model, allowing it to predict suitable combinations in new contexts.

3) Location Registration and Environmental Context

a) Location Setup

Users are prompted to register their primary location (home address or region) to allow Mood Manager to access relevant environmental data. Precise geolocation data is anonymized and stored securely.

b) External Environmental Integration

The system links with external APIs (e.g., weather, air quality, humidity, temperature) to determine environmental context in real time. This enables proactive mood adaptation, for example, activating a fresh citrus scent and bright lighting on humid days or lowering light intensity and playing warm sounds on rainy evenings.

c) Indoor Condition Synchronization

When available, indoor sensors (e.g., temperature, humidity, or brightness) are synchronized via IoT protocols such as Matter or MQTT, allowing fine-tuned environmental analysis.

4) AI Initialization and Verification

After device registration, preference input, and environmental setup are complete, Mood Manager's AI Personalization Engine is initialized. The system verifies all connected devices and settings through a brief diagnostic step, ensuring successful linkage and readiness for automatic mood control.

D. Mood Selection

The Mood Selection feature enables users to experience AI-driven, hyper-personalized environments through the integrated control of lighting, fragrance, and sound. Based on the user's registered devices, preferences, and contextual data, the system automatically recommends or adjusts the optimal "mood scene." Users can also manually modify each element through intuitive interfaces within the application.

1) Mood Recommendation and Scene Generation

a) Context-Aware Analysis

The system continuously monitors environmental variables such as time, weather, location, and recent user feedback logs. This contextual information is processed by the AI Personalization Engine, which uses a Few-shot Prompting mechanism to infer the most appropriate combination of light, sound, and fragrance for the current situation.

b) Mood Scene Suggestion

The AI generates one or more mood scene options (e.g., Relaxation, Focus, Romantic Evening) in JSON format, containing recommended values for brightness, fragrance intensity, and music volume. Users can preview and select a preferred mood scene or allow the system to apply it automatically.

c) Manual Adjustment and Override

Users can adjust each element of the mood scene individually (e.g., slightly dimmer lighting, lower fragrance intensity). All adjustments are recorded as user feedback to refine future recommendations.

2) Lighting Control

a) Integration with Smart Lighting Devices

Lighting control is implemented through protocols such as Matter or MQTT, enabling direct communication with smart bulbs or lighting panels (e.g., Philips Hue, Nanoleaf).

b) Brightness and Color Temperature Adjustment

Based on the selected mood, the system modifies brightness and color tone—for example:

- Relax Mode: 30–40% brightness, 2800–3000K warm tone
- Focus Mode: 70–80% brightness, 5000–6000K cool tone

c) Dynamic Lighting Effects

For certain moods, dynamic transitions (fade-in/fade-out or rhythmic lighting) are used to enhance immersion. These effects are generated in real time through the lighting control API.

3) Fragrance Control

a) Smart Diffuser Integration

The system communicates with Wi-Fi or BLE-enabled fragrance diffusers that support programmable intensity and duration.

b) Intensity and Duration Management

The AI determines the appropriate scent intensity (Level 1–5) and dispersal duration based on user context and previous preferences. For example, during a Relax mood, a citrus or woody scent may be diffused at moderate intensity for 30 minutes.

c) Safety and Refill Monitoring

The system includes safety guardrails such as preventing over-diffusion, enforcing cooldown intervals, and monitoring cartridge capacity. If refill levels are low, the user receives an in-app notification.

4) Sound Control

a) Music and Soundscape Selection

Mood Manager connects to streaming APIs (e.g., Spotify, YouTube Music) or local storage to play background sounds aligned with the selected mood. Example pairings include:

- Focus: Lo-fi beats, white noise
- Relax: Nature sounds, jazz
- Romantic: Soft instrumental music

b) Volume and Playback Scheduling

The system controls speaker volume through the IoT layer, maintaining balance with lighting and fragrance intensity. Playback automatically starts and stops according to mood duration or time of day.

5) AI-driven Multi-Sensory Synchronization

All three sensory components—light, sound, and fragrance—are orchestrated simultaneously by the IoT Orchestration Module. This module ensures that transitions occur smoothly and synchronously, creating a cohesive experience rather than separate device reactions. For instance, when the user switches from Focus to Relax, the lighting slowly dims while soft music and fragrance fade in together. The orchestration logic follows a transaction-based system that guarantees all connected devices receive the command successfully. If one device fails, the system automatically rolls back to the previous stable state to maintain harmony in the environment.

E. Chat-Based Feedback

The Chat-Based Feedback feature allows users to communicate naturally with the system after a mood scene has been applied. Through conversational feedback, users can express satisfaction, make adjustments, or request changes in real time. This interaction not only enhances user engagement but also serves as essential data for the AI Personalization Engine to refine future recommendations.

1) Chat Interface for Feedback Collection

a) Conversational Feedback Window

After a mood scene (lighting, fragrance, and sound) is applied, the system opens a chat interface with the AI assistant. The assistant asks short, context-aware questions such as:

- “Do you like this current mood setup?”
- “Would you prefer brighter lighting or a softer scent?”

b) User Responses

Users can respond freely in natural language (e.g., “Make it a bit warmer” or “The scent is too strong”). The system’s natural language processor (NLP) interprets the intent and modifies the settings in real time.

c) Quick Feedback Buttons

For quick interaction, users can also select from predefined feedback buttons such as satisfied, unsatisfied, or adjust. These simple responses are recorded as sentimental data for learning purposes.

2) Real-Time Adjustment and Data Logging

When the user provides feedback, the system immediately modifies corresponding IoT parameters through the orchestration module. For instance, “Make it a little brighter” increases lighting brightness, while “Reduce the scent” decreases diffuser intensity. Each feedback session is stored in the feedback log, including contextual data (time, environment, and device states) and final config-uration results. These logs are vectorized and stored in the embedding database, forming reference examples for Few-shot Prompting.

3) AI Learning and Personalization Update

The system periodically retrieves stored feedback data to enhance personalization. Positive or frequently accepted configurations are treated as “preferred examples,” guiding future inference in similar contexts. The AI dynamically updates user profiles based on this data, gradually shaping individual tendencies such as preferred lighting tone or scent strength. When conflicting prefe-rences occur, recency-based weighting ensures that the model adapts to the user’s current lifestyle rather than outdated data.

4) Emotion and Sentiment Recognition

The NLP module analyzes the emotional tone of chat inputs to detect user sentiment (positive, neutral, or negative). If stress or fatigue-related language is detected (e.g., “I’m tired,” “Too loud”), the system proactively adjusts to a Relax or Comfort mood scene. This sentiment-aware control enhances emotional intelligence and user satisfaction.

III. DEVELOPMENT ENVIRONMENT

A. Choice of software development platform

1) Development Platform

a) Web Platform

The web platform serves as the central foundation of the Mood Manager system, offering high accessibility, scalability, and compatibility across multiple operating systems. Through modern web technologies such as Next.js and TypeScript, developers can efficiently build responsive interfaces and maintain consistent user experiences on any device. The web platform supports seamless collaboration among team members using different operating systems including macOS, Windows, and Linux, as it only requires a web browser for development and testing. It also provides easy integration with cloud services such as Firebase Firestore for real-time data synchronization and AWS for hosting and deployment. Continuous updates in web technologies ensure compatibility with emerging standards and APIs, enabling long-term maintainability. For these reasons, the web platform is recognized as a flexible and future-proof environment that supports the core visualization and control components of the Mood Manager system.

b) Wear OS Platform

Wear OS, built on the Android ecosystem, is employed as the data collection platform for biometric and acoustic information in the Mood Manager project. It provides robust SDK support and native integration with key APIs such as Health Connect, AudioRecord, and Porcupine, enabling efficient access to physiological data and on-device voice recognition features. Development in Kotlin enhances code safety, readability, and interoperability with Android libraries while ensuring optimal performance. The platform's Foreground-Service feature allows continuous background operation for uninterrupted data capture, even when the watch display is inactive. Collected data are securely transmitted to Firebase Firestore for real-time synchronization with the cloud and web dashboard. As an Android-based system, Wear OS benefits from a large developer community, wide hardware availability, and strong compatibility with Google services. Due to these capabilities, Wear OS was selected as the optimal platform for reliable, real-time physiological data acquisition and contextual awareness within the Mood Manager ecosystem.

2) Language

a) Kotlin

Figure 1: Kotlin

Kotlin is a modern, statically typed programming language officially supported by Google for Android and Wear OS development. It enhances code safety through null safety features and concise syntax, reducing boilerplate compared to Java. Kotlin provides seamless interoperability with existing Android libraries, enabling efficient use of APIs such as Health Connect and AudioRecord. Its coroutine-based concurrency allows smooth background processing and real-time data handling, making it ideal for continuous biometric and audio data collection in wearable environments.

b) TypeScript

Figure 2: Typescript

TypeScript is a superset of JavaScript developed by Microsoft, introducing a static type system that improves code stability and readability. It catches errors at compile time, reducing runtime issues and enhancing maintainability, especially in large-scale projects. TypeScript's strong type inference and modular structure help developers define clear code intent and foster collaboration across teams. It preserves JavaScript's flexibility while providing the safety of typed languages, making it a robust choice for building reliable and scalable web applications.

c) Python

Figure 3: Python

Python is a high-level, general-purpose programming language widely used for backend development, data analysis, and artificial intelligence applications. Its simple syntax and extensive library support make it suitable for rapid prototyping and deployment of complex systems. Python's ecosystem includes frameworks and tools optimized for web APIs, machine learning, and data processing, allowing seamless integration with cloud environments. Its flexibility and readability facilitate efficient backend logic implementation for data-driven decision-making in the Mood Manager system.

3) Frameworks

a) Next.js

Figure 4: Next.js

Next.js is a React-based web framework that provides server-side rendering and static site generation for fast and scalable web applications. It supports efficient routing and strong TypeScript integration, improving performance and maintainability. The framework simplifies both frontend and backend development through built-in API routes and automatic optimization, making it ideal for real-time interactive systems. It also provides excellent developer experience with hot reloading and simple deployment options, ensuring rapid iteration and stable production performance.

b) FastAPI

Figure 5: FastAPI

FastAPI is a lightweight Python framework designed for building APIs quickly and efficiently. It uses Python type hints for automatic validation and documentation while supporting asynchronous processing for high-speed performance. The framework enables clean, readable, and easily extensible code, allowing developers to implement stable and responsive backend services. It is highly compatible with modern AI libraries and cloud environments, making it suitable for intelligent and data-driven applications.

c) Tailwind CSS

Figure 6: Tailwind CSS

Tailwind CSS is a utility-first framework that allows fast and consistent front-end design. It provides reusable style classes and encourages clean, responsive layouts without custom CSS. The framework ensures visual consistency, reduces repetitive styling tasks, and helps developers rapidly build modern and adaptive user interfaces. Tailwind also integrates seamlessly with component-based frameworks, enabling scalable and maintainable front-end design systems.

d) Android JetPack

Figure 7: Android Jetpack

Android Jetpack is a collection of libraries and components that simplifies Android app development. It provides modules such as Lifecycle, ViewModel and WorkManager that help manage data, background tasks and UI states efficiently. Jetpack offers a standardized architecture, improves app stability, and enhances productivity with built-in tools for Wear OS integration. Its modular structure supports cleaner code organization and long-term maintenance for complex Android applications.

4) Cloud Platform

a) Amazon Web Services(AWS)

Figure 8: Amazon Web Services(AWS)

AWS is a global cloud platform that provides hosting, storage and deployment services for modern applications. It offers high reliability, automatic scaling and strong security for both web and backend systems. AWS supports continuous integration and deployment pipelines, ensuring efficient updates and consistent performance across environments. Its broad service ecosystem allows flexible configuration and cost optimization, making it suitable for both development and large-scale production.

b) Firebase Firestore

Figure 9: Firebase Firestore

Firebase Firestore is a cloud-based NoSQL database that supports real-time synchronization and secure data storage. It allows fast data access and easy integration with mobile and web SDKs. Firestore automatically scales with usage and minimizes server management, enabling developers to focus on application functionality rather than infrastructure. Its real-time listener feature ensures that all connected clients instantly reflect changes, creating a seamless live data environment.

5) Cost Estimation

a) Hardware

The hardware cost consists of laptops used by five team members for development and testing. Each device is estimated at approximately ₩1,500,000, which provides sufficient performance for running Android Studio, Next.js, and cloud-based tools. Since the team requires multi-platform compatibility across macOS and Windows environments, the total hardware expense amounts to ₩7,500,000 in total.

b) Software

The software cost consists of cloud computing and AI model usage. For backend hosting and deployment, an AWS EC2 t3.xlarge instance is used, which provides 4 vCPUs, 16 GiB of memory, EBS-only storage, and network bandwidth up to 5 Gbps. The on-demand usage rate is USD 0.2816 per hour, approximately ₩370 per hour. Assuming a total usage of about 135 hours during the project development period, the total estimated cost for AWS is around ₩50,000. For AI inference, the project uses the OpenAI GPT-5 API, which is priced at USD 1.25 per one million input tokens and USD 10 per one million output tokens. The project is expected to use under one million input tokens and around three million output tokens, leading to an estimated total cost of USD 31.25, equivalent to approximately ₩41,000. Combining both components, the total estimated software cost is ₩91,000, covering cloud infrastructure and AI API usage throughout the development and testing phases.

B. Software in Use

1. Android Studio

Figure 10: Android Studio

Android Studio is the official integrated development environment for Android and Wear OS application development. It provides powerful tools for code editing, debugging, and performance profiling, allowing developers to build optimized and reliable mobile applications. The environment supports Kotlin and Java natively and integrates seamlessly with Android Jetpack, Health Connect, and other Google APIs. It also includes an advanced emulator for testing across various devices and OS versions. Android Studio's strong support for Gradle automation and version control improves productivity and ensures stable app deployment.

2. Visual Studio Code

Figure 11: Visual Studio Code

Visual Studio Code is a lightweight and highly extensible code editor developed by Microsoft. It supports multiple programming languages including TypeScript, Python, and Kotlin through a rich extension ecosystem. The built-in IntelliSense, debugging tools, and terminal features make it suitable for both frontend and backend development. It offers real-time collaboration via Live Share, allowing developers to work together effectively across different environments. VS Code's flexibility and integration with GitHub and Docker make it an ideal all-purpose tool for modern software projects.

3. GitHub

Figure 12: GitHub

GitHub is a cloud-based version control and collaboration platform built on Git. It enables developers to manage code efficiently, track revisions, and collaborate across teams through pull requests and branches. Its project management tools, including issues and boards, help maintain workflow organization. GitHub also provides CI/CD pipeline integration and security scanning features for automated testing and code quality assurance. The platform's global community and open-source ecosystem promote collaboration and innovation across diverse development teams.

4. Figma

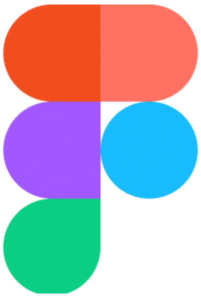

Figure 13: Figma

Figma is a web-based design and prototyping tool used to create user interfaces and interactive prototypes. It supports real-time collaboration, allowing multiple designers and developers to work simultaneously. Its reusable component system and responsive design capabilities streamline the UI/UX workflow. Figma's browser-based nature eliminates installation issues and ensures cross-platform accessibility. The tool's integration with design systems and plugins enhances team efficiency and visual consistency across products.

5. Notion

Figure 14: Notion

Notion is an all-in-one workspace that combines documentation, task management, and project collaboration. It allows teams to organize notes, schedules, and resources in a centralized environment. Customizable templates and database features make it adaptable to different workflows. Notion supports real-time collaboration and version tracking, reducing communication gaps between developers and designers. Its flexibility and minimal interface help maintain clear documentation throughout all stages of project development.

6. PostgreSQL

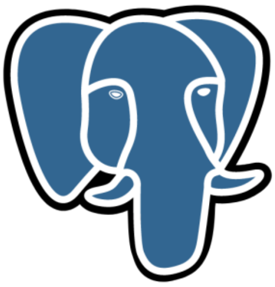

Figure 15: PostgreSQL

PostgreSQL is an open-source relational database management system known for its robustness and compliance with SQL standards. It provides advanced features such as transactions, indexing, and JSON support for hybrid data structures. The database ensures high data integrity and fault tolerance, making it suitable for mission-critical systems. PostgreSQL's scalability and strong community support enable continuous improvements and long-term stability. Its compatibility with cloud services and ORMs makes it a preferred choice for modern data-driven applications.

7. Docker

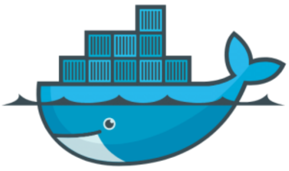

Figure 16: Docker

Docker is a containerization platform that enables applications to run consistently across different environments. It packages code and dependencies into lightweight containers that improve scalability and deployment speed. Docker eliminates environment inconsistencies, simplifying development and testing. Its support for microservices architecture allows modular and efficient system design. The platform's integration with orchestration tools like Kubernetes makes it a standard for modern cloud-native applications.

8. ChatGPT

Figure 17: ChatGPT

ChatGPT is an advanced conversational AI model developed by OpenAI, accessible through the OpenAI API for integration into software systems. It provides natural language understanding and generation capabilities that assist developers in automating documentation, debugging, and code generation tasks. Through the API, ChatGPT can be embedded directly into applications, enabling intelligent interactions such as question answering, summarization, and contextual reasoning. It supports multiple programming languages and frameworks, making it highly adaptable to various development environments. The API's scalability and pay-per-use pricing structure allow flexible usage without the need for dedicated infrastructure. By leveraging the GPT-5 architecture, ChatGPT delivers reliable, context-aware responses that improve development efficiency and enhance user experience in interactive systems.

9. Gemini

Figure 18: Gemini

Gemini is Google's advanced AI model designed for reasoning, coding assistance, and data analysis. It provides multimodal capabilities, allowing text, image, and structured data processing within a single framework. The model assists developers with intelligent suggestions, code generation, and system optimization. Gemini's adaptability and integration with Google's ecosystem make it a valuable tool for research and advanced development workflows.

IV. SPECIFICATION

A) Sign Up Page

1) ID Registration

Users create a user ID with 5 to 20 letters and numbers only. A Check ID button confirms if the ID is already taken and shows a clear message. If the ID is available the screen shows a simple success mark and locks the field to avoid accidental changes. A Change button unlocks it again before final submission if needed. Short helper text below the field explains the rule and gives one example format.

2) Password Creation

Users enter a password and it is hidden with star symbols by default. A Show button reveals the password and hides it again when tapped a second time or after a short timeout. A simple strength guide under the field shows needs more characters or looks good based on length and variety. A Confirm Password field prevents typos and shows a message if the two values do not match. Basic tips appear under the field such as use at least eight characters.

3) Phone Number Verification

Users enter a mobile phone number and tap Send code to receive a six digit code by text message. A small timer shows how long the code stays valid and a Resend button appears after a short wait. Users type the code and see a clear success or try again message. If the number is already registered the page suggests going to login or resetting the password. Input errors explain what went wrong such as invalid number or expired code.

4) Personal Information Input

Users provide full name birth year and gender with gender being optional. Birth year is used instead of full birthdate to reduce sensitive data collection. Short notes explain that this information personalizes recommenddations. All values can be edited later in My Page. Required fields are marked and the Next button stays disabled until they are filled.

5) SNS Sign Up

Users tap a social sign up button and are guided through the provider screen where they confirm basic profile sharing. After returning to the app the screen shows the imported name and profile image and asks the user to review missing fields such as phone number or birth year. If the social account matches an existing user ID or phone number the app shows a simple message and connects the accounts after user confirmation. If the phone number is not yet verified the user completes the same code verification step before finishing. A success message appears and the app moves to the login page automatically.

6) Registration Completion

A brief review step shows the entered information except the password so users can check it once more. Users agree to the terms and privacy policy by ticking checkboxes and tap Create account. A success message appears and the app moves to the login page automatically. If the network fails the page keeps the inputs and shows a Try again button. Clear guidance is shown for any error so users can finish without confusion.

B) Sign In Page

1) Local Sign In

a) Success

Users enter their ID and password, which are hidden with star symbols and can be briefly shown by tapping the Show button. After pressing Sign In, if the information matches, the app opens the main page and shows a short welcome message with the user's name. The login time is recorded, and the app remembers the last visited screen for smoother access next time.

b) Wrong Password

If the entered password is incorrect, the field border turns red, and a clear message appears asking the user to try again. The ID stays on the screen so users only reenter the password. A small hint message appears if Caps Lock is on, and the Forgot Password link guides users directly to reset. The screen limits repeated failed attempts and shows a short wait message before retrying.

c) Password Change

When selecting Forgot Password, users input their phone number and receive a six-digit code via text message. A timer shows the remaining time to enter the code, and users can resend it after a short delay. Once verified, the user creates a new password in two fields, both hidden with stars, and a success message appears after saving. The system then redirects users to the login page with the ID prefilled for convenience.

2) SNS Sign In

Users can sign in with social accounts such as Google, Naver, or Kakao by tapping the provider button. After confirming account access on the provider's page, users return to the app where basic information like name and email are automatically filled. If the account is new, the app asks for additional fields such as phone number or age before completion. When sign-in succeeds, the app moves to the main page and shows a welcome notification.

C) Registration Page

1) Register

Users start device registration by selecting the QR registration option from the main menu. The system requests camera permission and opens a scanner view with a visible square frame in the center. When the QR code is detected, the code value appears briefly for confirmation before registration begins. A short loading message is shown while the app checks code validity and connects to the server. The user can cancel at any time by pressing the Back button or closing the scanner window.

2) Pre-Survey

After scanning, a short pre-survey appears to understand the user's preferences, such as scent strength, lighting color, and preferred mood type. Each question is shown on a single screen with multiple-choice options for quick input. A Skip button allows users to proceed without answering, ensuring registration continues smoothly. The survey results are saved for later use in personalization but are not required for device setup. A short message at the end explains that preferences can be changed anytime in settings.

3) Device Name Setup

If the scanned QR code is invalid, unreadable, or belongs to an unsupported device, the system shows a clear message such as Invalid QR Code or Cannot recognize this device. The user can retry scanning immediately by pressing Try Again or cancel and return to the previous screen. If the camera is blocked or permission is denied, the system displays a short guide to enable camera access. When repeated failures occur, a Contact Support button appears with a link to troubleshooting instructions.

4) Registration Failed

Once the QR code is recognized successfully, the user is prompted to assign a name to the registered device. A short placeholder example (e.g., Living Room Diffuser) appears inside the field, and a Confirm button activates only after the name is entered. A short helper message explains that names help users identify their devices easily in the app. If the same name already exists, a message asks the user to choose another one. After confirmation, the device name is saved, and the screen proceeds to the success page.

5) Registration Success

After completing all steps, the app displays a large success icon and a message confirming that the device has been successfully registered. The new device automatically appears in the main device list, and a short message such as Device successfully connected is shown. The user can press Go to Home to view and control the device immediately. A tutorial button is also available to guide new users through basic functions if they wish to learn more.

D) Main Page

1) Mood Selection

The top area of the screen shows circular icons, each representing a mood such as Relax, Focus, or Energy. Users swipe left or right to browse available moods, and the selected mood enlarges slightly with its name displayed at the center. Once a mood is selected, the lower section automatically updates to show detailed control panels in rectangular cards for fragrance, sound, lighting, and duration. Each mood stores its own preset of values so users can switch instantly without reconfiguring settings. A small edit icon beside each mood allows quick access to modification or duplication.

2) Add New Mood

a) Name Setup

Users can create a new mood by tapping the Add Mood button and entering a name in a text field, with examples like Morning Focus or Sleep Mode shown as placeholders. A Save button activates once a name is entered. The app prevents duplicate names and notifies the user if the same mood name already exists.

b) Fragrance Setup

A list of available oils is displayed with sliders to adjust blend ratios. Users can combine up to three oils, and a preview bar shows the total balance visually. When the user finishes adjusting, the selected combination is saved as part of the mood profile automatically.

c) Sound Setup

Users choose ambient sounds or playlists such as Rain, Jazz, or Lo-fi. A small play icon lets them preview before saving, and volume can be adjusted with a horizontal slider. A short description or tag (e.g., Calm, Focused, Refreshing) appears next to each playlist for easy selection.

d) Lighting Setup

A color wheel or temperature bar allows users to choose light color and warmth. A live preview updates instantly to show changes in tone. The user can also tap the Auto Sync button to match lighting with the chosen mood type automatically.

e) Duration Setup

A time bar lets users set how long the mood remains active, from 5 minutes up to 2 hours. A "Loop" option allows repeating the mood automatically until stopped. A countdown timer appears under the bar once the mood starts to help users track remaining time.

3) Edit Current Mood

a) Fragrance Adjustment

Users can fine-tune fragrance ratios or overall intensity using sliders labeled per oil type. A live indicator shows changes immediately in percentage. The Save Changes button becomes active once any adjustment is made, ensuring that settings are not lost accidentally.

b) Sound Adjustment

Users adjust the sound type or switch between playlists, controlling both volume and mix balance with slider bars. A small play button previews the change before saving. The system remembers the last used volume level and restores it automatically the next time the same mood is activated.

c) Lighting Adjustment

Lighting color and brightness can be modified with a dual-bar control for color temperature and intensity. The device light updates instantly to match selections.

A Reset to Default button restores the original lighting preset if the user wants to undo changes.

d) Duration Adjustment

A duration bar lets users extend or shorten the current mood session in real time. When adjusted, the remaining time updates dynamically at the bottom of the screen. If the duration ends, the app displays a small notification with options to restart or switch to another mood.

4) AI-Based Recommended Mood

An AI suggestion card appears near the top of the page showing "Recommended Mood for Now." It updates automatically based on user behavior, biometric data, and recent mood selections. The user can apply the recommendation instantly by tapping "Apply Now" or view details explaining why the mood was suggested. A refresh icon generates a new recommendation if the user wants an alternative.

5) Mood Routine Management

This section allows users to schedule mood changes automatically throughout the day. A simple list view displays entries such as 08:00–Relax Mode, 14:00–Focus Mode, and 22:00–Sleep Mode. Users tap Add Routine to create new ones, selecting a start time, end time, and mood. Each routine has an on/off toggle for quick control and can be reordered with drag-and-drop. Notifications appear five minutes before each scheduled change to remind users.

6) Manual Control Panel

Below all mood cards, a quick control bar allows users to temporarily adjust fragrance strength, sound volume, or light brightness without editing a full mood preset. These manual adjustments reset automatically when a new mood is selected, keeping presets consistent. A reset icon restores all values to their saved state.

E) Setting Page

1) Device Management

a) Device Registration

Users can register a new device by selecting "Add Device," which opens the QR scanner screen. When the QR code is scanned successfully, a confirmation message appears and the device is added to the list. If the scan fails, the app provides retry and manual entry options to complete registration.

b) Device Deletion

Users can remove a registered device by selecting it from the list and pressing "Delete." A confirmation popup appears asking, "Are you sure you want to remove this device?" Once confirmed, the device disappears from the list and its data is cleared from the app.

c) Device Name Edit

Each registered device shows an Edit icon next to its name. Tapping this allows users to type a new name, which updates immediately after saving. If the same name already exists, the app shows a message prompting users to choose another name.

d) Pre-Survey

Users can retake the short preference survey to update personal data such as scent, lighting, or sound preferences. The results automatically refresh AI recommendations and saved presets. A Skip option is available for users who wish to maintain their previous preferences.

2) Customer Support

A "Contact Support" section provides quick access to customer assistance. Users can choose between FAQ, Live Chat, or Email Inquiry options. Common issues are listed with step-by-step guidance, and a direct "Send Inquiry" button opens an input form for detailed questions. Submitted inquiries show a confirmation message and expected response time.

3) Privacy Policy

The Privacy Policy section displays how user data such as account information, preferences, and device activity is collected and used. Users can scroll through the document or tap "Download PDF" for offline viewing. A summary at the top highlights key items like data storage period and user rights for transparency.

4) Notification Settings

Users can manage all app notifications from this section, including alerts for device updates, AI recommendations, and routine reminders. Each option has an On/Off toggle, and changes are saved instantly. A "Do Not Disturb" mode allows silencing all notifications during selected hours.

5) App Version and Updates

This section shows the current app version and provides a "Check for Updates" button. When an update is available, users can tap "Update Now" to open the app store directly. The system also notifies users automatically when a new version is detected.

F) My Page

1) Profile Information Edit

a) Password Change

Users can change their password by entering their current password and then typing a new one in two matching fields. The password input is hidden with star symbols and can be revealed temporarily with a Show button. A short guide below the field reminds users to include at least eight characters with a mix of letters and numbers. After saving, a success message appears, and the user is returned to the My Page screen.

b) Birth Year / Gender Edit

Users can update their birth year and gender information from this section. A simple dropdown or number input is provided for selecting a new birth year, and gender can be changed between Male, Female, or Prefer not to say. After pressing Save, the system confirms the update with a short message such as “Profile updated successfully.”

2) SNS Account Connection

Users can link or unlink their social accounts such as Google, Apple, or Kakao from this section. Each connected account is shown with a small icon and connection status. When linking a new account, a short provider screen appears for approval before completing the connection. Unlinking requires confirmation to prevent accidental disconnection, and the app ensures users can still sign in locally if SNS login is removed.

3) Logout

The Logout button is placed at the bottom of the screen and signs the user out of the current device immediately. A short popup appears asking, “Do you want to log out?” before the process begins. After confirming, the app returns to the sign-in page and clears saved session data for security.

4) Account Deletion

Users can permanently delete their account by selecting “Delete Account” and entering their password for confirmation. A short warning message explains that all personal data, preferences, and connected devices will be removed. After confirming twice, the system processes deletion and shows a “Account successfully deleted” message before returning to the welcome screen.

5) Data Backup and Restore

This section allows users to back up their mood settings, device data, and AI preferences to the cloud. A “Back Up Now” button saves the current configuration, while a “Restore Data” button reloads previously saved data to the same or a new device. The backup date and file size are displayed below for easy reference.