Bilkent University
Department of Computer Engineering

CS492: Senior Design Project II

Detailed Design Report

Team ID: T2518

ReMind

Ahmet Yağız Sarıdoğan	22202072
Ayça Candan Ataç	22203501
Elif Ceren Çelik	22202534
Emine Noor	22201252
Irmak İmdat	22201570

Supervisor: Sinem Sav

Course Instructors: İlker Kurt, Mert Bıçakçı

12.03.2026

Table of Contents

1. Introduction

Dementia and Alzheimer's disease affect millions of patients and their caregivers worldwide, but most existing tools handle the problem in a fragmented way, either tracking location without any cognitive support or offering reminders without any safety mechanism. ReMind is a mobile health assistant designed to bring these together in a single, privacy-first application. This report documents the detailed design of the system.

1.1 Purpose of the System

Dementia and/or Alzheimer's disease create emotional challenges and practical difficulties that affect both patients, their family members, and their caregivers [1]. The diseases affect patients' spatial awareness, navigation abilities, and executive function, while simultaneously causing memory loss that occurs during daily activities. The symptoms necessitate ongoing supervision for patients, as they need protection from wandering, forgetfulness, and disorientation [2]. The need for patient safety supervision creates a conflict because it protects the patient but violates their independence and privacy. The constant need for supervision leads to anxiety and burnout among the caregivers [3].

ReMind is a mobile health assistant that will address these exact problems. It will offer various helpful features to achieve its goal of helping caregivers and protecting patients. For example, its location tracking feature will help the caregiver feel at ease whenever they are not with the patient. Its MoodAI will monitor the patient's data, alerting the caregiver of possible dangerous situations, such as confusion or distress. To help MoodAI make more accurate predictions, ReMind will be integrated with a smartwatch, which will enable it to collect more functional data from the user.

The proposed solution covers two main areas. First, on the patient side, ReMind provides safe-zone monitoring via geofencing, task and medication reminders, and an SOS mechanism that immediately alerts the caregiver. Second, on the caregiver side, ReMind allows monitoring of the patient's daily activity and location (only when outside a safe zone), sends reminders to support adherence, and sends alerts when MoodAI detects anomalies.

All features are implemented in a privacy-preserving way by on-device MoodAI, encrypted data storage, push notifications for reliable alert delivery, and explicit per-category consent collected at registration. Users are clearly shown what each piece of data is used for. The goal is a tool that helps patients maintain their daily routines safely, without feeling overwhelmed or pressured, while keeping caregivers informed and reassured.

1.2 Design Goals

The design goals of the ReMind platform are centered around usability, reliability, performance, security, privacy, scalability, and supportability.

1.2.1 Usability

The system needs to provide clean, immediate access to its features and be user-friendly, as the target users are patients and their caregivers. This requires an interface with clear, basic navigation, restrained color use, and a calm, non-overstimulating design.

- Ease of Navigation
    - The patient application should feature a clean interface that uses large buttons and minimal text usage.
    - The system should allow users to access the “Take Me Home” feature, reminders, and mood check-ins through an always-visible interface. This eliminates the need for complex menu navigation and is a simpler choice for patient comfort.
    - The caregiver interface should display information based on priority levels, starting with alerts (also sorted by priority) and then daily summaries.

- Accessibility
    - The system should align with WCAG 2.2 Level AA [12] accessibility principles, the criteria most relevant to the target user group to support users with visual, motor, and cognitive disabilities.
    - The system should use colors, text, and symbols for clear indications.
    - System designers should create mood check-ins and game interfaces in a way that minimizes user confusion and frustration (e.g., avoiding time limits and failure states).

- Enrollment & Guidance
    - Both patient and caregiver apps should include short enrollment flows that explain controls in plain language.
    - Tooltips and tutorials should clarify safe zone setup, alert logic, and device-sensor permissions. The patient and caregiver applications should begin with basic sections that describe their controls using clear terminology.

1.2.2 Reliability

The system must operate reliably under real-world conditions, which may include network fluctuations, battery limitations, and sensor signal interruptions.

- System Continuity
    - The system should support background monitoring of location and safe zone events using OS-level services, while respecting mobile operating system constraints on background execution and sensor access.
    - When watch data is unavailable, the system should fall back to phone sensors.
- Fail-Safe Alerting
    - The system needs to deliver safety alerts, and it requires network connectivity to function properly. The system stores outgoing notifications in a queue until delivery success is achieved, since the network might fail.
    - MoodAI should operate as a local anomaly detection system that functions independently from any network requirements.
- Resilience
    - The system should return to its safe operational state following crashes, while maintaining all current alerts and system logs.

1.2.3 Performance

Optimal performance is a necessity for the system, both for user safety and system reliability. Therefore, for better performance, the system should optimize its use of sensors, data processing, and alert transmission.

- Latency Requirements
    - The system should begin generating and sending safe zone breach alerts immediately through a process that should take no longer than five seconds upon detecting any breach.
    - The system should perform MoodAI anomaly inference within a reasonable time to generate real-time support cards.
- Sensor Efficiency
    - The system should adjust its background polling rates for GPS, heart rate, and movement data based on user behavior.
    - The system must perform low-frequency sampling during normal activities to conserve battery power, but switch to high-frequency sampling when it detects anomalies or when the user leaves the safe zone.

- Server Performance
    - The caregiver dashboard should display summary information immediately, with a refresh time of 3-5 seconds when showing data from previous months.

1.2.4. Security

The system must protect user data and prevent unauthorized access across all components.

- Authentication & Access Control
    - Role-based access control must ensure that caregivers can only access the data they are permitted to see.
    - Firebase Authentication [15] enforces email verification on every login. Therefore, unverified accounts cannot access patient data or generate link codes.
    - Patient-caregiver linking requires explicit patient approval via a 6-digit code. A caregiver gains no data access until the patient approves the link request.
- Encryption
    - All data stored in Firestore is encrypted at rest using AES-256 by Google Cloud infrastructure.
    - All communication between the app and Firebase [15] uses HTTPS/TLS 1.3, enforced automatically by the Firebase SDK.
    - Free-text mood notes will be encrypted at the application layer using AES-256 before being written to Firestore, with the encryption key stored in device secure storage.

1.2.5. Privacy

The system must handle patient data in a way that respects user autonomy, minimises data exposure, and ensures caregivers access only what is necessary for safe care.

- Data Minimisation
    - Health-related data (mood check-ins) is processed locally on-device where possible. Only the anomaly level is transmitted to the backend. Therefore, raw sensor data and model inputs never leave the device.
    - Location data is event-based only. Exact GPS coordinates are transmitted only on safe zone exit or SOS trigger. The system does not continuously track or store location data.
- Access Control & Consent
    - The system requires patient consent before any caregiver link is established, and patients can revoke caregiver access at any time from the settings screen.

1.2.6. Scalability

The system architecture needs to handle an expanding number of users, rising sensor data, and new AI functionality additions.

- User Scalability
    - The system should support hundreds of caregiver-patient pairs with no degradation in alert delivery speed or dashboard responsiveness.
    - Each caregiver may manage multiple patients without performance loss.
- Data Scalability
    - The cloud storage system should handle extended log retention periods, which include multiple months of patient data.
    - The system needs caregivers to access previous data through fast retrieval methods.
- Compute Scalability
    - The system should have a direct device execution of MoodAI fine-tuning and anomaly detection operations to minimize server processing needs.
- Feature Growth
    - The system design must support the addition of new cognitive games and advanced anomaly models through non-disruptive architectural changes.

1.2.7. Supportability

The system requires operational maintenance to enable developers to perform maintenance work and resolve problems without disrupting user access.

- Supportability
    - The system should support troubleshooting and maintenance without disrupting user access.
- Logging & Diagnostics
    - The system must store local logs without raw sensitive data, which include sensor data, alert generation conditions, and geofence entry events in secure storage.
    - With user consent, the system should provide anonymized telemetry to help resolve problems that affect battery life or cause abnormal app crashes and synchronization issues.
- Error Transparency
    - When failures or misconfigurations occur, the system should present clear, plain-language notifications with steps to restore functionality.
- Modular Architecture

- The system requires separate feature modules, which allow developers to make separate updates and bug fixes.

1.3 Definitions, Acronyms, and Abbreviations

Term	Definition
AES	Advanced Encryption Standard
AI	Artificial Intelligence
API	Application Programming Interface
BPM	Beats Per Minute
CI/CD	Continuous Integration / Continuous Delivery: automated build, test, and release pipeline
Firebase Cloud Messaging (FCM)	Google's push notification service used for caregiver alerts
Firebase	Google's mobile backend platform used for authentication, database (Firestore), and push notifications
Firestore	Google Cloud Firestore: the NoSQL document database used by ReMind for cloud data storage
Flutter	Google's open-source UI framework used to build the ReMind app for both Android and iOS from a single codebase
GDPR	General Data Protection Regulation
GPS	Global Positioning System
HR	Heart Rate
HRV	Heart Rate Variability
HTTPS	Hypertext Transfer Protocol Secure: HTTP over TLS; used for all communication between the app and backend
ISO/IEC 27001	International standard for information security management systems

JWT	JSON Web Token: a signed token issued by Firebase Authentication to verify user identity on each request
KVKK	Kişisel Verileri Koruma Kanunu: Turkish Personal Data Protection Law
ML	Machine Learning
mHealth	Mobile Health
MoodAI	The on-device anomaly detection module in ReMind
OS	Operating System
ReMind	The mobile health assistant described in this report
REST	Representational State Transfer: the architectural style used for ReMind's backend API
SDK	Software Development Kit: a set of tools and libraries used to build applications for a specific platform (e.g., Firebase SDK)
SOS	Emergency alert sent by the patient directly to the caregiver
TFLite	TensorFlow Lite: Google's lightweight ML framework used to run the MoodAI model on-device
TLS	Transport Layer Security
UI	User Interface
UX	User Experience
WCAG	Web Content Accessibility Guidelines

Table 1: Definitions, Acronyms, and Abbreviations

1.4 Overview

ReMind is a mobile health assistant that helps patients with Alzheimer's disease or other types of dementia and their caregivers. It manages daily activities, monitors patient safety and mood, and does not compromise patient privacy.

For patients, the system offers safe-zone monitoring using geofencing. It provides task and medication reminders, along with an SOS feature that alerts the caregiver immediately. For caregivers, it allows

them to monitor daily activity and location when the patient is outside a safe zone. It also tracks adherence to reminders and sends alerts when MoodAI detects any mood changes.

ReMind plans to improve the quality of life for both patients and caregivers. It aims to lower the supervision burden through smart monitoring and timely alerts, all while protecting the patient's privacy.

2. Current Software Architecture

Nowadays, ongoing supervision and passive reminders like phone calls or sticky notes are crucial components of dementia care. However, they do not properly respect the patient's independence or dignity and can exhaust carers.

Existing technology addresses parts of the issue, but none cover it entirely. AngelSense [4] and Jiobit [5] focus on GPS tracking, but they invade patient privacy with constant monitoring and do not track health or mood data. Medisafe [6] does a good job of reminding users to take their medication. However, it lacks location tracking and safety features. Constant Therapy [7] provides clinical-grade rehabilitation exercises, but it is expensive and does not have much use in daily life. The Apple Watch [8] has fall detection and heart rate monitoring. However, its interface is too complicated for dementia patients. It tracks either fully on or completely off, and there is no safe-zone logic.

What the current market lacks, which ReMind aims to fill, revolves around three main areas. First, a design approach that prioritizes location privacy. Existing GPS solutions continuously upload all location data, whereas ReMind only shares exact location information when the patient exits a safe zone. Second, mood and cognitive analysis. Many existing health applications do not provide a unified system for linking physiological data (e.g., heart rate and heart rate variability) with behavioral data in order to identify early signs of stress or anxiety before the situation escalates. Another issue is data security. Many health and wellness apps send users' raw physiological data (e.g., heart rate, sleep patterns) to online servers where it may be vulnerable to unauthorized access. By contrast, the ReMind application is designed to process sensitive physiological information directly on the user's device, while any data stored on remote cloud services is kept in encrypted form.

3. Proposed Software Architecture

3.1. Overview

ReMind has a modular client-server architecture made up of three layers: a mobile application layer, a backend service layer, and a data storage layer. The ReMind mobile application is built on Flutter [14] to support cross-platform apps that run on both Android and iOS devices. There are two main types of users in the mobile interface: patients and caregivers. Patients use a simplified interface that is easy to access, while caregivers have access to a dashboard where they see alerts, summaries of each patient, and tools to monitor patients.

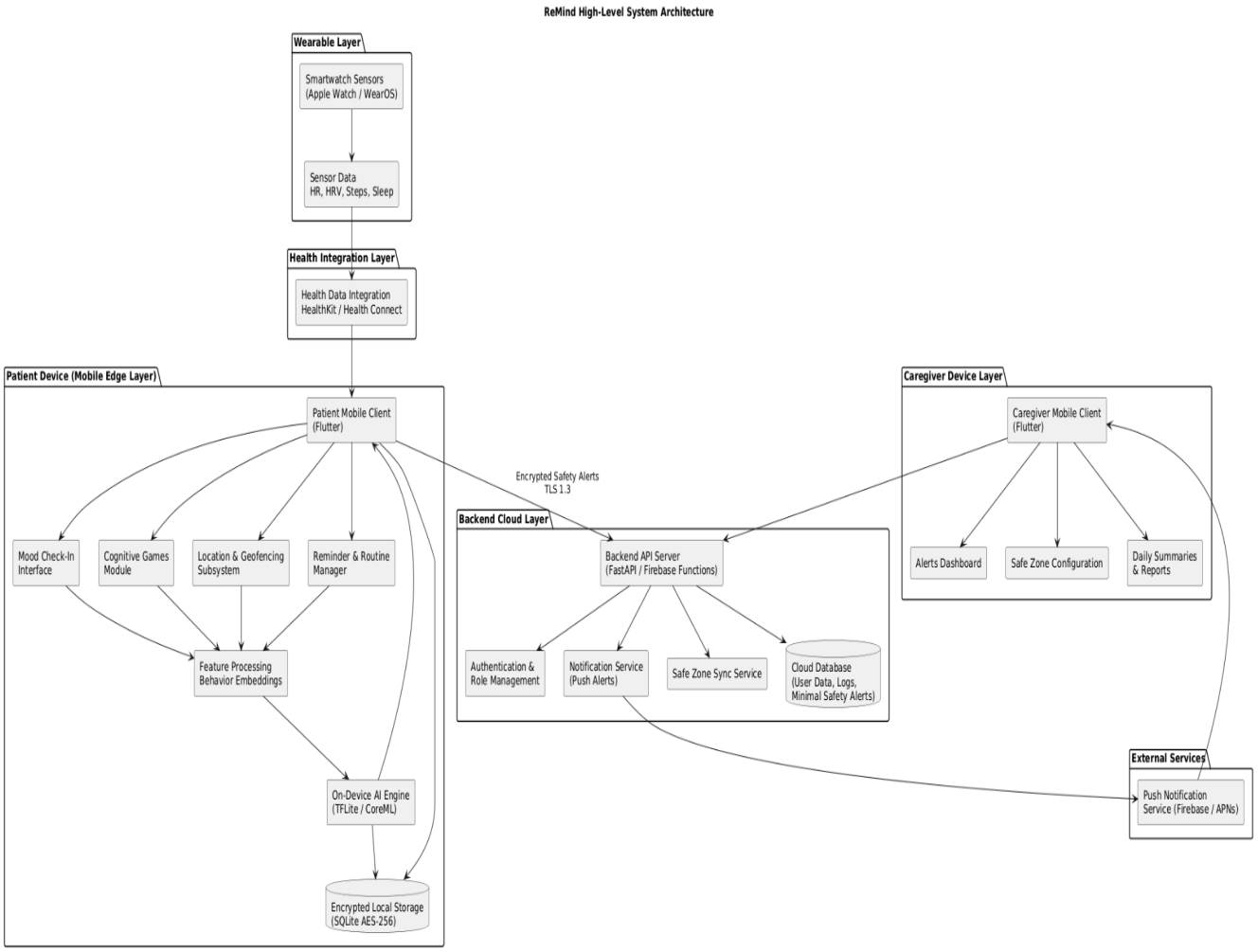
ReMind uses secure REST APIs to connect to the backend services that provide patient authentication, link patients and caregivers, create alerts, set up reminders, and sync data. The backend services are integrated with wearable devices by using platform health APIs from Apple HealthKit [8] and Google Health Connect [9] to access the data being collected as part of the patients' activity and health signals (e.g., steps and heart rate).

The standout component of the ReMind architecture is the MoodAI module which performs anomaly detection based on behavioral and sensor data collected on the patients' mobile device. The MoodAI module performs these calculations primarily on the mobile device to minimize privacy concerns and to reduce network dependability.

User account data, reminder-based scheduling information, alert logs, and system log files are retained in a secure cloud database as described in previous sections regarding the principles of privacy-first design.

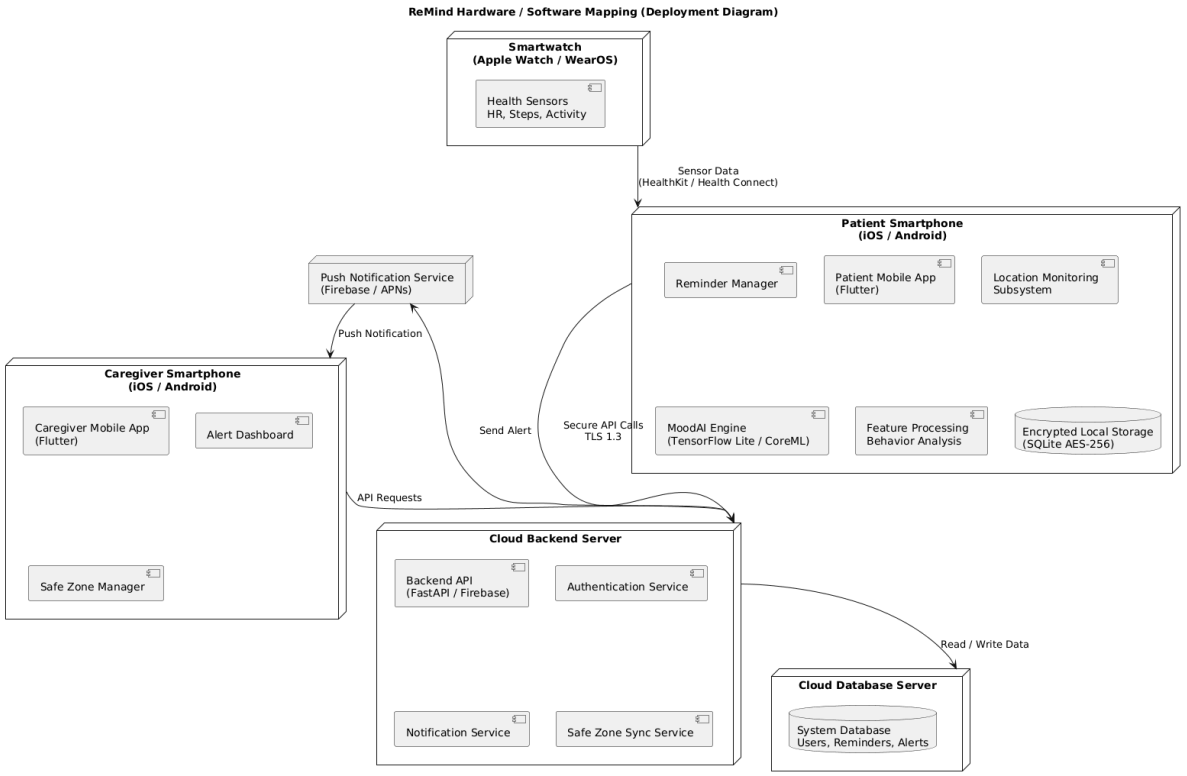
Above all, the architecture supports modularity, scalability, and privacy so that components of the system can change independently of one another while still maintaining secure and reliable communication between client mobile devices, backend service components (e.g., servers), and data storage infrastructure.

3.2. Subsystem Decomposition



Link: https://drive.google.com/file/d/1pYvri1RrXQ_CsyrKv3VIZgMAttMIaFwS/view?usp=drive_link

3.3. Hardware/Software Mapping



Link: https://drive.google.com/file/d/1tfVddxQ5M2vjqMuNV1FuRQOm5FVpRvbJ/view?usp=drive_link

The ReMind includes a multi-platform design for various hardware components while safely and successfully communicating between component parts.

The main client devices are patient and caregiver smartphones (Android and iOS), running on the Flutter [14] mobile application. These devices provide the same functionality across all platforms. Patient devices collect interaction data with the system, location data, and data from mood check-ins. Caregiver devices will primarily receive alerts from patient devices (for a particular medication) and display monitoring charts.

While smartphones are the main client devices for the ReMind system, smart watches and other wearable devices may also collect data about their use (for example, sensor data, such as heart rate, distance walked, number of steps, and activity data) through health APIs for platform integration, including Apple HealthKit [8] and Google Health Connect [9].

The cloud-based architecture at Firebase gives the ability to host back-end applications that support the patients' devices and the caregivers by providing server-based capabilities. The service is built on

top of Firebase along with the SDKs provided by them to implement the application logic, authentication, and sync the mobile app with the cloud-based data storage.

The app uses the Firebase services for user authentication, synchronizing reminder data, sending alert events and the linking between the patient and caregiver. Email and calendar notifications are sent to the user from Firebase Authentication and they can automatically manage how the data is shared and stay synchronized across multiple devices.

To help make sure that sensitive information about a user is not tampered with or seen by someone other than the intended recipient, all communication between Firebase service and mobile apps is established over a secure HTTPS connection implemented by Firebase.

ReMind utilizes Firebase Cloud Firestore as its structured cloud database to store persistent system data such as:

- User account information and authentication credentials
- The caregiver-patient relationship
- Reminder schedules and logs of adherence to those schedules
- Alert history and summary event logs

Primarily, all the sensitive health or behavioral information created by the sensors and mood-analysis functions of the mobile application will be processed locally on the mobile device. Only a summary of this information will then be transmitted back to a server which reduces the risk of violation of individual privacy and this limits the amount of data that must be stored.

There are functions within the ReMind system that occur directly on the mobile device vs the server. For example, the MoodAI anomaly detection module, which utilises lightweight machine learning frameworks is executed locally on your mobile device. This helps reduce latency, improve responsiveness and protect sensitive behavioral information by ensuring it remains with the user where possible.

3.4. Persistent Data Management

The data management plan of ReMind relies predominantly on the principles of privacy by design, which specify how data is stored, where it gets stored, and the retention period. The system keeps device-collected and cloud-stored data separate. All data collected from sensors within the device are processed fully on the device using the MoodAI module prior to any portion being transferred to the backend, maintaining the complete segregation of device-collected data and cloud based data. Raw

sensor signals retrieved from the device (HR, HRV etc.) will not be transmitted to the cloud, only processed summary events (summary alert logs, summary mood check-ins, etc.) are transmitted, which may be required by a caregiver to know how and/or what is happening with their patient. All transmitted data will be protected in transit by using TLS 1.3. The summary events will be encrypted with AES-256 when stored in the cloud. All other information relating to caregivers and/or patients that may be required for cross-device access and continuous interaction, will also be encrypted at AES-256 while being stored in the cloud. The cloud will have an adequate amount of storage for information that will be provided to caregivers, sufficient to allow for the comprehensive evaluation of historical trends, and will also have adequate space to allow for many caregiver-patient pairs to access the cloud at a rate of speed that does not degrade as the number of caregiver-patient pairs increase.

3.5. Access Control and Security

ReMind has two user roles, patient and caregiver, with strictly separated access controls, each of which can access only the data and functionality related to their tasks. Both the interface and API level have the same separation so that caregivers do not have access to a patient's actual sensor data, or cognitive game activities. Caregivers can only view a patient's alerts, reminder status, or location after the patient has left a designated safe zone. Authentication methods include email/password or email link passwordless verification. An email verification of previous registration is mandatory prior to any authenticated activity. Also, both parties to the patient/caregiver relationship must approve each other before either can link to the other. Either party can unlink their relationship with the other at any time. During the enrollment phase, the patient must provide consent by completing a consent process/step that states they approve of collecting their data, linking to the caregiver, and authorizing when the caregiver may obtain their location. All data sent between the mobile app and the backend service has been secured (using TLS 1.3), and all patient data stored on the server is encrypted (using AES-256). The system design was guided by the requirements of the GDPR [10] and KVKK [11]. All physical and technical security controls will be based on the principles of ISO/IEC 27001 [13].

4. Subsystem Services

4.1 Mobile Application Service

The ReMind mobile application is the main interface through which both patients and caregivers interact with the system. The mobile app is created with the Flutter [14] framework, creating a multi-platform alternative for both Android and iOS devices. This subsection houses all features

available in the system for end-users, by receiving user inputs, and sending those inputs to backend services while controlling communication between end-user devices and native components of the device. The app provides two distinct user interfaces, one for each of the two different user types (i.e., patient and caregiver). The two interfaces are designed to meet each user type's individual needs.

The mobile application utilized by patients has a simple and easy-to-use interface designed for daily assistance for alleviating daily medication, routine, mood and cognitive games, reminders as well as the "Take Me Home" functionality. The patient interface is designed for usability with large buttons, a limited number of navigation options, and distinct visual cues, which will help to serve individuals with declining abilities. The mobile app will also leverage device sensors and connect to rich data from wearable devices via health APIs (e.g., via Apple HealthKit or Google Health Connect). Each patient's activity will be measured by various types of activity and health metrics transmitted from those touch devices to provide behavioral data for further review and evaluation.

Caregivers utilize the application for tracking patient wellness. The caregiver interface provides users with capabilities to define safe zones, monitor alerts generated by the monitoring system, and retrieve summaries of patient activity and reminder compliance. Mobile applications perform secure REST API requests against backend services to synchronize reminders, update safe zone configurations and retrieve alert information. Mobile applications also communicate with components on the user's device (e.g., MoodAI, encrypted local storage) to process behaviours and preserve necessary data when network connectivity is poor.

4.2 Backend API and Data Management Service

The back-end API and data management services provide server-side infrastructure communicated to mobile applications and the persistent data store supported by the system. The back-end API and data management service are responsible for receiving requests from both the patient application and caregiver application, performing user authentication and synchronizing system data between all users including patient and caregiver users, as well as ensuring that all data associated with applications is stored reliably in a persistent data storage mechanism.

The back-end services are implemented using Firebase, which provides cloud-based functionality for mobile applications to perform operations such as registering new users, authenticating users, synchronizing reminders, configuring safe zones, and communicating alerts.

Another major function of this subsystem is managing user and role accounts. The back-end verifies the authentication credentials when users log on and creates user profiles for each patient and

caregiver. Additionally, this subsystem manages the relationship between a caregiver's account and a patient's account so that caregivers can only view data that is pertinent to the patients they are responsible for. There are comprehensive authentication and authorization mechanisms built within the system to ensure that any sensitive health or behavioral information about either the patient or caregiver is not accessible by anyone who is not authorized to see this data.

The backend subsystem features a data synchronization service that allows devices to keep the information on the cloud database in sync. Firebase is the repository for information needed by the system to generate reminders, store reminder response logs, define safe zones, and keep a history of alerts. All data inside the cloud database can be accessed by authorized users through an API. For example, when a caregiver changes a safe zone or sets up a reminder, the backend subsystems will handle the request and update its database accordingly. After updating the database, the backend syncs the configuration with the mobile device of the patient to ensure that the system behaves the same across all devices.

In addition to providing data storage and synchronization services, the backend subsystem of the system is responsible for processing and delivering notifications related to alarms or alerts. If a significant event occurs, such as violating the safe zone or an "anomaly score"/high score being detected by the MoodAI subsystem, then the mobile application will send a notification request to the backend server. The backend subsystem will process the request and then push the alert to the caregiver's mobile device. Caregivers will receive timely alerts through the backend subsystem whenever there is a potential safety issue.

4.3 Safe Zone & Location Monitoring Service

The Safe Zone and Location Monitoring Service is responsible for monitoring and responding to patient location according to their safe zones. Each patient may have up to three geofences entered by caregivers. These zones are synchronized with the patient device and stored in the backend. The application monitors zone boundaries using OS-level geofencing APIs while the patient is within a safe zone. This helps to protect the battery. While within the safe zone, ReMind does not gather exact GPS coordinates. Instead, the caregiver interface only shows the safe area where the patient is at that moment. In order to detect any boundary violations, the system switches to more frequent GPS tracking when the patient is outside of the safe zone. An alert is activated and a violation is confirmed only if the patient remains outside the boundary for at least 60 seconds or moves more than 50 meters beyond it. This is implemented to prevent false alarms caused by GPS fluctuations.

The service starts two simultaneous actions on notification of a breach. The "Take Me Home" function is activated by the patient's device. This feature allows you to navigate to the selected safe zone using a map. Additionally, an alert containing the patient's last known location, and a timestamp is sent to the caregiver. Until the patient returns to a safe area, the caregiver can monitor the patient's current location via their interface. After that, tracking automatically ends. When location tracking is enabled, the patient's screen provides a clear explanation. By doing this, the patient is informed that their caregiver is aware of their location. Even if the patient is in a secure area, the same location-sharing system is triggered if they press the SOS button.

4.4 Notification and Alert Service

The Notification and Alert Service delivers important safety and informational messages from the system to caregivers and patients. Even when the app is closed or running in the background, the service uses Firebase Cloud Messaging [15] to send push notifications to devices.

All outgoing alerts are stored locally on the sending device before they are sent. If the device has no network connection when an alert is created, the queue holds the message and tries to send it again once the connection returns. This applies to every type of alert, including safe zone breach alerts, MoodAI anomaly alerts, and SOS messages. SOS messages are especially important, so they are always prioritized in the queue and are never dropped or delayed for lower-priority messages.

When a patient's account is deleted, the service sends a final notification to all linked caregivers to inform them about the account removal. This makes sure caregivers do not monitor a connection that no longer exists.

4.5 MoodAI Module

The MoodAI module provides on-device anomaly detection within the patient-facing mobile application. The module’s purpose is to analyze patient behavioral and physiological signals in order to identify potential anomalies in the patient’s condition. The MoodAI module is able to receive information from mood check-in responses, sleep levels, physical activity levels, and phone usage. Additional signals, such as heart rate and heart rate variability, can be collected from a paired smartwatch via health-related APIs when available. When the smartwatch cannot supply data, the MoodAI module will still function by utilizing only phone-based signals. The MoodAI module implements lightweight machine learning algorithms using TensorFlow Lite and performs real-time inference on-device to process the received signals. The MoodAI model generates an anomaly score categorized as normal, moderate, or high.

When a moderate anomaly is detected, a patient-specific support card shall be made available to the patient. When a high anomaly is detected, an alert will be sent to the patient's caregiver. If a moderate anomaly persists beyond a configured time window, an additional alert will be generated and sent to the caregiver.

5. Test Cases

This section presents the test cases developed to validate the functionality and reliability of the ReMind system. The purpose of these tests is to confirm that all components of the ReMind system meet expectations in a variety of normal and extreme circumstances.

5.1 Functional Test Cases

Test Case ID	F001
Category	Patient Registration
Objective	Verify that patients can successfully register with ReMind.
Steps	1. Launch the ReMind mobile application 2. Select the option to Create a Patient Account 3. Fill out all of the required personal information 4. Accept the terms and conditions. 5. Submit the registration form. 6. Enter the verification code sent to the registered email address.
Expected Result	The system creates the patient account and the user is directed to the login page.
Date - Result	TBD

Table 2: Functional Test Case 1 (F001)

Test Case ID	F002
Category	Registration
Objective	Verify that the system rejects invalid email formats during registration.
Steps	1. Open the patient registration page.

	2. Enter an invalid email format in the email field. 3. Enter valid data in all other required fields. 4. Submit the registration form.
Expected Result	The system displays an inline error message indicating that the email format is invalid and that the account cannot be created.
Date - Result	TBD

Table 3: Functional Test Case 2 (F002)

Test Case ID	F003
Category	Registration
Objective	Verify that duplicate email registration is prevented.
Steps	1. Register a patient account using a valid email address. 2. Open the registration page again. 3. Enter the same email address used in step 1. 4. Submit the registration form.
Expected Result	The system rejects the second registration and displays a duplicate account error message.
Date - Result	TBD

Table 4: Functional Test Case 3 (F003)

Test Case ID	F004
Category	Caregiver Registration
Objective	Verify that a caregiver can successfully create an account.
Steps	1. Open the ReMind mobile application. 2. Select "Create Caregiver Account". 3. Enter required personal information (name, email, password). 4. Accept the terms and conditions. 5. Submit the registration form.

Expected Result	The caregiver account is successfully created, and the user is redirected to the login page.
Date - Result	TBD

Table 5: Functional Test Case 4 (F004)

Test Case ID	F005
Category	Login Authentication
Objective	Verify that registered users can log in successfully.
Steps	1. Open the ReMind mobile application. 2. Enter a valid registered email address. 3. Enter the correct password. 4. Select the Login button.
Expected Result	The user is authenticated and redirected to their dashboard within 2 seconds.
Date - Result	TBD

Table 6: Functional Test Case 5 (F005)

Test Case ID	F006
Category	Login Authentication
Objective	Verify that login fails when an incorrect password is entered.
Steps	1. Open the ReMind mobile application. 2. Enter a valid registered email address. 3. Enter an incorrect password. 4. Select the Login button.
Expected Result	Login is rejected and an error message is displayed. The user remains on the login page.
Date - Result	TBD

Table 7: Functional Test Case 6 (F006)

Test Case ID	F007
Category	Patient-Caregiver Linking
Objective	Verify that a caregiver can link to a patient account.
Steps	1. Log in as a caregiver. 2. Select "Link Patient" from the menu. 3. Enter the patient's linking code. 4. Send a linking request to the patient. 5. Confirm the linking request from the patient account.
Expected Result	The patient and caregiver accounts are successfully linked, and both accounts reflect the connection (caregiver dashboard shows patient information etc.).
Date - Result	TBD

Table 8: Functional Test Case 7 (F007)

Test Case ID	F008
Category	Patient-Caregiver Linking
Objective	Verify the system rejects invalid linking codes.
Steps	1. Log in as a caregiver. 2. Select "Link Patient" from the menu. 3. Enter an invalid linking code. 4. Submit the linking request.
Expected Result	The system displays an error message, and the linking request is not processed.
Date - Result	TBD

Table 9: Functional Test Case 8 (F008)

Test Case ID	F009
Category	Patient Dashboard
Objective	Verify that the Patient Dashboard displays all relevant information.
Steps	1. Log in as a patient.

	2. Open the Patient Dashboard.
Expected Result	The Patient Dashboard correctly displays upcoming reminders, the Mood Check-in button, the cognitive games section and navigation features.
Date - Result	TBD

Table 10: Functional Test Case 9 (F009)

Test Case ID	F010
Category	Caregiver Dashboard
Objective	Verify that the Caregiver Dashboard displays linked patient status.
Steps	1. Log in as a caregiver. 2. Open the Caregiver Dashboard.
Expected Result	The Caregiver Dashboard displays all linked patients and their information.
Date - Result	TBD

Table 11: Functional Test Case 10 (F010)

Test Case ID	F011
Category	Mood Check-in
Objective	Verify that patients can complete a Mood Check-in successfully.
Steps	1. Log in as a patient. 2. Select "Mood Check-in" from the Patient Dashboard. 3. Select an image representing the patient's current emotional state from the Mood Check-in interface. 4. Fill in the questionnaire about the patient’s mood. 5. Submit the Mood Check-in.
Expected Result	The patient’s response is processed by MoodAI, and the output is displayed to the caregiver.
Date - Result	TBD

Table 12: Functional Test Case 11 (F011)

Test Case ID	F012
Category	Mood Check-in
Objective	Verify that the system prevents submission of a Mood Check-in when no mood image is selected, or any required fields are missing.
Steps	1. Log in as a patient. 2. Open the Mood Check-in interface from the Patient Dashboard. 3. Attempt to submit the Mood Check-in without selecting a mood image or without filling in the required fields.
Expected Result	The system prevents submission of the Mood Check-in and displays a prompt for the patient to fill all required fields.
Date - Result	TBD

Table 13: Functional Test Case 12 (F012)

Test Case ID	F013
Category	Reminder System
Objective	Verify that caregivers/patients can create reminders.
Steps	1. Log in as a caregiver/patient. 2. Open the Reminder System from the dashboard. 3. Select "Create Reminder" and enter reminder details. 4. Save the reminder.
Expected Result	The reminder is saved and appears on the linked patient's and caregiver’s dashboards.
Date - Result	TBD

Table 14: Functional Test Case 13 (F013)

Test Case ID	F014
Category	Reminder System

Objective	Verify that reminder notifications are delivered to the patient and to the caregiver at the scheduled time.
Steps	1. Log in as a caregiver/patient. 2. Create a new reminder with a scheduled time. 3. Wait until the scheduled reminder time is reached.
Expected Result	The patient and the caregiver receive a reminder notification at the scheduled time.
Date - Result	TBD

Table 15: Functional Test Case 14 (F014)

Test Case ID	F015
Category	Safe Zone Management
Objective	Verify that a caregiver can create a Safe Zone.
Steps	1. Log in as a caregiver. 2. Open Safe Zone settings from the Caregiver Dashboard. 3. Select a circular location on the map. 4. Save the Safe Zone.
Expected Result	The Safe Zone is saved in the system and is visible in the caregiver's Safe Zone settings.
Date - Result	TBD

Table 16: Functional Test Case 15 (F015)

Test Case ID	F016
Category	Safe Zone Monitoring
Objective	Verify that the system detects when a patient exits the Safe Zone.
Steps	1. Set up a Safe Zone for a linked patient. 2. Move the patient’s device at least 50 meters beyond the boundary or keep the device outside of the Safe Zone for at least 60 seconds. 3. Allow the system to process GPS data.

Expected Result	The caregiver receives a Safe Zone exit alert within 5 seconds after the violation is confirmed, and the patient receives a prompt to navigate back into the Safe Zone.
Date - Result	TBD

Table 17: Functional Test Case 16 (F016)

Test Case ID	F017
Category	Safe Zone Monitoring
Objective	Verify that the Safe Zone exit alert is resolved when the patient re-enters the Safe Zone.
Steps	1. Set up a Safe Zone for a linked patient. 2. Trigger a Safe Zone exit alert by moving the patient's device outside the Safe Zone. 3. Move the patient's device back inside the Safe Zone boundary.
Expected Result	The Safe Zone exit alert is automatically resolved, and the caregiver is notified of the patient's return.
Date - Result	TBD

Table 18: Functional Test Case 17 (F017)

Test Case ID	F018
Category	Emergency Alert
Objective	Verify that the patient can trigger emergency alerts.
Steps	1. Log in as a patient. 2. Press the Emergency Alert button on the Patient Dashboard. 3. If the patient does not cancel the alert attempt within 5 seconds, the emergency notification is sent to the caregiver.
Expected Result	The caregiver receives an emergency notification within 3 seconds.
Date - Result	TBD

Table 19: Functional Test Case 18 (F018)

Test Case ID	F019
Category	Navigation Assistance
Objective	Verify that navigation guidance back to the Safe Zone is available when the patient is outside of it.
Steps	1. Move the patient's device outside the defined Safe Zone. 2. Select "Take Me Home" when the navigation notification appears.
Expected Result	Simple navigation instructions to the Safe Zone are shown to the patient.
Date - Result	TBD

Table 20: Functional Test Case 19 (F019)

Test Case ID	F020
Category	Patient-Caregiver Unlinking
Objective	Verify that a caregiver/patient can remove the patient-caregiver link.
Steps	1. Log in as a caregiver/patient. 2. Open the link settings in the dashboard. 3. Select the unlink option. 4. Confirm the unlinking action.
Expected Result	The link between the patient and caregiver is removed, and neither account reflects the connection (caregiver can not view the patient information anymore etc.).
Date - Result	TBD

Table 21: Functional Test Case 20 (F020)

Test Case ID	F021
Category	Safe Zone Privacy
Objective	Verify that caregivers cannot view the patient's exact location when the patient is inside the Safe Zone.

Steps	1. Log in as a caregiver. 2. Ensure the patient is located inside the defined Safe Zone. 3. Open the patient monitoring screen on the Caregiver Dashboard. 4. Observe the displayed patient location information.
Expected Result	The system only shows the "In Safe Zone" label for the patient's location.
Date - Result	TBD

Table 22: Functional Test Case 21 (F021)

Test Case ID	F022
Category	Emergency Alert
Objective	Verify that an SOS alert is not sent if the patient cancels the emergency alert.
Steps	1. Log in as a patient. 2. Press the Emergency Alert button on the Patient Dashboard. 3. Press the cancel alert button within 5 seconds.
Expected Result	No emergency alert is sent, and the caregiver receives no notification.
Date - Result	TBD

Table 23: Functional Test Case 22 (F022)

Test Case ID	F023
Category	Safe Zone Management
Objective	Verify that the caregiver cannot create more than three Safe Zones.
Steps	1. Log in as a caregiver. 2. Create three Safe Zones for a patient. 3. Attempt to create a fourth Safe Zone for the same patient.
Expected Result	The system does not create the fourth Safe Zone and displays a prompt saying there can be a maximum of three Safe Zones per patient.
Date - Result	TBD

Table 24: Functional Test Case 23 (F023)

Test Case ID	F024
Category	Reminder System
Objective	Verify that patient interaction with reminders (Done or Skipped) is correctly recorded in the system.
Steps	1. Log in as a caregiver and create a reminder for a patient. 2. Log in as the patient. 3. Wait until the reminder notification appears on the patient device. 4. Select Done or Skipped on the reminder notification. 5. Log in as the caregiver and check the reminder history.
Expected Result	The reminder interaction is correctly recorded and displayed in the caregiver account.
Date - Result	TBD

Table 25: Functional Test Case 24 (F024)

Test Case ID	F025
Category	Smartwatch Integration
Objective	Verify that a smartwatch can successfully pair with the ReMind mobile application and that the patient's physiological data can be retrieved successfully.
Steps	1. Navigate to the Device Pairing settings on the patient’s mobile phone. 2. Turn on the smartwatch and enable Bluetooth. 3. Select the smartwatch from the list of available devices. 4. Confirm the pairing request on both devices. 5. Open the ReMind mobile application. 6. Confirm the patient data is being captured by the ReMind application.
Expected Result	The smartwatch pairs successfully with the mobile application, and sensor data transmission becomes available.
Date - Result	TBD

Table 26: Functional Test Case 25 (F025)

Test Case ID	F026
Category	MoodAI Monitoring
Objective	Verify that the MoodAI system triggers a caregiver alert when the mood anomaly score exceeds the threshold.
Steps	1. Log in as a patient. 2. Ensure the smartwatch is paired and physiological data monitoring is active, and the Mood Check-in is available. 3. Use the system's test mode with simulated physiological sensor data and Mood Check-in answers representing a distress condition. 4. Allow the MoodAI module to process the incoming sensor data.
Expected Result	The MoodAI module detects the anomaly and sends an alert notification to the caregiver.
Date - Result	TBD

Table 27: Functional Test Case 26 (F026)

Test Case ID	F027
Category	Account Management
Objective	Verify that when a patient account is deleted, all linked caregivers receive a final notification and the caregiver-patient link is removed.
Steps	1. Log in as a patient and ensure a patient-caregiver link exists. 2. Delete the patient account through the application settings. 3. Observe the system behavior on caregiver devices.
Expected Result	All linked caregivers receive a final notification informing them that the patient account has been deleted. The caregiver-patient link is removed and the patient information no longer appears in the caregiver dashboard.
Date - Result	TBD

Table 28: Functional Test Case 27 (F027)

5.2 Non-Functional Test Cases

Test Case ID	NF001
Category	Performance
Objective	Verify that the system responds to user actions within an acceptable time.
Steps	1. Open the ReMind mobile application. 2. Submit a login request with valid credentials. 3. Measure the time from submission to dashboard load.
Expected Result	The system responds and loads the dashboard in under 3 seconds.
Date - Result	TBD

Table 29: Non-Functional Test Case 1 (NF001)

Test Case ID	NF002
Category	Performance
Objective	Verify that Safe Zone breach alerts are delivered within 5 seconds.
Steps	1. Configure a Safe Zone for a patient. 2. Move the patient's device outside the Safe Zone boundary. 3. Measure the time from the boundary crossing to the caregiver receiving the alert notification.
Expected Result	The caregiver receives the Safe Zone breach alert within 5 seconds.
Date - Result	TBD

Table 30: Non-Functional Test Case 2 (NF002)

Test Case ID	NF003
Category	Performance
Objective	Verify that MoodAI anomaly detection inference completes within an acceptable time.
Steps	1. Log in as a patient.

	2. Initiate a Mood Check-in by selecting a mood image from the interface. 3. Measure the time between image submission and the MoodAI result display.
Expected Result	The MoodAI inference completes and displays the result within 7 seconds.
Date - Result	TBD

Table 31: Non-Functional Test Case 3 (NF003)

Test Case ID	NF004
Category	Scalability
Objective	Verify that the system supports multiple concurrent users without degradation.
Steps	1. Simulate 100 concurrent users performing login and core actions using a load testing tool. 2. Monitor server response times and error rates during the load test.
Expected Result	The system maintains stable response times with no crashes or errors with 100 concurrent users.
Date - Result	TBD

Table 32: Non-Functional Test Case 4 (NF004)

Test Case ID	NF005
Category	Security
Objective	Verify that user credentials are securely stored in the database.
Steps	1. Open the database and inspect stored user records. 2. Verify that no plaintext passwords are present. 3. Confirm that a secure hashing algorithm is used.
Expected Result	All passwords are stored as hashed values; no plaintext credentials are present in the database.
Date - Result	TBD

Table 33: Non-Functional Test Case 5 (NF005)

Test Case ID	NF006
Category	Privacy/Security
Objective	Verify that patient location data is only accessible to authorised caregivers through backend access control.
Steps	1. Attempt to access a patient's location data using an account not linked to that patient through backend requests. 2. Observe the system response.
Expected Result	Access is denied and a permission error is returned. No location data is exposed.
Date - Result	TBD

Table 34: Non-Functional Test Case 6 (NF006)

Test Case ID	NF007
Category	Reliability
Objective	Verify that the system handles server interruptions without data loss.
Steps	1. Perform an active operation (for example, saving a reminder) in the application. 2. Simulate a server interruption mid-operation. 3. Restore the server and check the system state.
Expected Result	The system recovers without data loss and corruption.
Date - Result	TBD

Table 35: Non-Functional Test Case 7 (NF007)

Test Case ID	NF008
Category	Usability
Objective	Verify that the interface is intuitive and that users can complete key tasks without confusion.
Steps	1. Recruit elderly participants with dementia who are unfamiliar with the application.

	2. Ask participants to complete a set of core tasks (login, create a reminder, complete Mood Check-in). 3. Observe and record task completion rates and errors.
Expected Result	At least 80% of participants complete all tasks successfully without requiring assistance.
Date - Result	TBD

Table 36: Non-Functional Test Case 8 (NF008)

Test Case ID	NF009
Category	Compatibility
Objective	Verify that the application runs correctly on supported mobile platforms (Android and iOS).
Steps	1. Install the application on at least three different mobile devices across Android and iOS platforms with varying screen sizes and OS versions. 2. Execute core functions (login, dashboard, Safe Zone, Mood Check-in) on each device.
Expected Result	The application runs without errors, and all core features function correctly on all supported Android and iOS devices.
Date - Result	TBD

Table 37: Non-Functional Test Case 9 (NF009)

Test Case ID	NF010
Category	Security
Objective	Verify that all patient-related data stored on the device and server is encrypted using AES-256.
Steps	1. Access the local device storage and the server-side database after normal application use.

	2. Inspect stored records including Mood Check-in logs, location snapshots, and alert history. 3. Confirm that stored data is not in plaintext and that AES-256 encryption is applied.
Expected Result	All stored patient-related data is encrypted using AES-256. No patient data is present in plaintext in either device storage or the server database.
Date - Result	TBD

Table 38: Non-Functional Test Case 10 - NF010

Test Case ID	NF011
Category	Security
Objective	Verify that all communication between the mobile application and the backend uses TLS 1.3.
Steps	1. Run the application and perform actions that trigger network communication (login, Safe Zone alert, Mood Check-in submission). 2. Intercept and inspect outgoing network traffic using a network analysis tool. 3. Verify the protocol version (must be TLS 1.3) used for each connection.
Expected Result	All network communication between the application and the backend is encrypted using TLS 1.3. No data is transmitted over unencrypted connections.
Date - Result	TBD

Table 39: Non-Functional Test Case 11 - NF011

Test Case ID	NF012
Category	Privacy
Objective	Verify that raw physiological sensor data is processed on-device and is not transmitted to the backend.
Steps	1. Pair the smartwatch and ensure physiological data monitoring is active. 2. Monitor outgoing network traffic while the MoodAI module processes sensor data.

	3. Inspect backend logs to confirm no raw sensor values have been received.
Expected Result	Raw sensor data (HR, HRV, step counts) is not present in any outgoing network requests or backend storage. Only processed summary outputs from MoodAI are transmitted when caregiver alerts are required.
Date - Result	TBD

Table 40: Non-Functional Test Case 12 - NF012

6. Consideration of Various Factors in Engineering Design

6.1 Constraints

The influence that several engineering design factors, such as safety and security, had on the architectural and interface decisions was evaluated.

Factor	Effect Level (0–10)	Discussion
Public Safety	**9**	One important aspect of design is patient safety. ReMind uses geofencing, SOS alerts, and real-time notifications for caregivers to lower the chances of patients getting lost or staying in unsafe situations without help. Monitoring safe zones and quickly sending alerts are essential safety features of the system.
Security	**9**	ReMind handles sensitive information, including health-related data, location data, and caregiver-patient relationships. Security therefore significantly influenced the system's architecture. The system applies role-based access control between patients and caregivers, secure authentication mechanisms, and restricted data sharing policies. Wherever

Factor	Effect Level (0–10)	Discussion
		possible, data processing is performed on-device to minimize data exposure and reduce privacy risks.
Global Factors	**5**	Dementia affects populations across different countries, economic levels, and healthcare systems. ReMind is designed with an offline-first architecture and cross-platform mobile support to facilitate broader deployment. The use of Flutter [14] enables a single cross-platform codebase for Android and iOS devices, simplifying deployment and maintenance across different markets. Although the initial scope targets the Turkish market, the architecture allows for future internationalization and localization.
Cultural Factors	**6**	In Turkey, dementia care is often managed by family members rather than healthcare professionals. The system design reflects this by assuming a caregiver who is a family member or close acquaintance. The interface language and cognitive game content are designed to be culturally appropriate and easy to understand for the intended user population.
Social Factors	**8**	The system must accommodate users with declining cognitive and motor abilities. The mobile interface therefore emphasizes accessibility through large buttons, simple navigation, minimal time pressure, and clear alert prioritization. Caregiver notifications are designed to effectively communicate important information without overwhelming the user.
Environmental Factors	**3**	As a software-based mobile application, ReMind has a relatively limited environmental impact. Nevertheless, design decisions such as on-device processing, event-driven location sharing, and offline-first operation reduce energy consumption both on user devices and server infrastructure. Battery efficiency on the patient device is considered an explicit design constraint.
Economic Factors	**5**	The system is designed to remain financially accessible to families. ReMind minimizes operational costs by relying on lightweight mobile processing and low-cost cloud infrastructure. Additionally, using native

Factor	Effect Level (0–10)	Discussion
		mobile APIs to collect health data eliminates reliance on expensive third-party services.

Table 41: Effect of Engineering Design Factors on the ReMind system

Note: Factors rated 5 or higher had a noticeable influence on at least one architectural, interface, or policy decision in the ReMind system design.

6.1.1 Public Safety

Patient safety is the primary concern driving the design of ReMind. Dementia patients face major physical risks from wandering, disorientation, and undetected changes in their health status. Before any monitoring begins, ReMind makes sure that patients and caregivers are fully informed about what data is being collected, how it is used, and who can access it. Consent is taken per data category at registration and can be withdrawn at any time. Only the minimum data required for a safety outcome is collected. GPS is shared only when a patient leaves a safe zone. The MoodAI module helps in identifying early signs of deterioration, but all clinical decisions remain with caregivers and medical professionals.

6.1.2 Global Factors

Dementia affects populations across all income levels and geographic regions. The ReMind architecture takes this into account by not assuming that users have access to a high-bandwidth or consistently connected environment. The alert queue is offline-first, MoodAI inference is performed on-device, and local reminder storage can all function without a network connection. The ReMind application uses Flutter [14], which enables deployment to both Android and iOS operating systems using the same codebase. This helps reduce barriers to reaching diverse device ecosystems globally. Currently, the application is designed for use in Turkey, but it is architecturally compatible with international deployment with minimal modifications.

6.1.3 Cultural Factors

In Türkiye and many similar regions, dementia care is predominantly managed within the family. ReMind is designed with this in mind: the caregiver is assumed to be a family member or close acquaintance instead of a clinical professional because of the law. The interface language, alert

framing, and cognitive game content avoid as much as possible jargon. Future versions of ReMind will need to check cultural inclusivity more extensively, particularly if it is deployed to other regions with different norms around aging, privacy, law, and family roles.

6.1.4 Social Factors

Social factors influenced design decisions across both the patient and caregiver interfaces. The patient UI is designed for individuals with declining cognitive and motor abilities, such as: large buttons, minimal text, no time pressure, and no failure states. Mood check-ins done with image selection rather than text input. Cognitive games are non-competitive and have no leaderboards or timers. On the caregiver side, alerts are sorted according to severity to reduce cognitive load during stressful moments. ReMind avoids generating outputs that could be misread as a definitive clinical diagnosis, because a misinterpretation could have serious consequences for both the patient and their family.

6.1.5 Environmental Factors

ReMind is a software-only product with minimal direct environmental impact. The design choices for ReMind that minimize network traffic (on-device processing, event-driven location sharing, and offline-first queuing) also reduce the energy consumption of server-side infrastructure. The battery efficiency of the patient's device has been treated as a design constraint. For example, if the patient’s device had to continuously monitor GPS and synchronize information, the battery would be depleted too quickly to provide the patient with emergency access. The safe zone service uses low-frequency OS-level geofencing while the patient is inside the zone. When a breach is detected, it switches to higher-frequency sampling.

6.1.6 Economic Factors

ReMind is designed to be a free application, as adding financial cost to an already burdensome caregiving situation would be contrary to the project's goals. The backend runs on Firebase's free tier, which is enough for development. ReMind's system is designed to avoid all third-party paid APIs; map navigation uses native OS libraries, and wearable data is accessed through HealthKit and HealthConnect at no cost. At a larger scale, Firestore query volume and FCM delivery would require a cost review, but the system reduces unnecessary reads.

6.2 Standards

ReMind is built to follow GDPR [10] and KVKK [11] guidelines for all personal health data. This includes obtaining explicit informed consent before collecting data, allowing users to export their personal data as a PDF, and giving them the option to delete their account along with all related data.

All data in transit is protected by TLS 1.3, and data stored in Firestore is encrypted with AES-256. Sensitive data, like mood notes, is additionally encrypted at the application level, with keys stored in the device's secure storage.

The patient interface aims for WCAG 2.2 Level AA [12] compliance to assist users with cognitive and motor impairments. This includes proper color contrast ratios, no timed interactions in the client UI, and compatibility with iOS VoiceOver and Android TalkBack.

The team follows IEEE 730 software quality practices for testing documentation and test case traceability. The backend API design follows RESTful conventions. Firebase Security Rules [15] implement role-based access control to ensure that one patient’s data is not accessible to users who are not connected to them.

7. Teamwork Details

Each team member's technical abilities and interests were taken into consideration when assigning tasks for the project. This allowed everyone to contribute effectively to both CS491 and CS492. Major decisions, including architectural choices, privacy design, and changes in scope, were made collectively. Each member led one or more work packages, while also assisting in areas beyond their main responsibilities.

7.1 Contributing and Functioning Effectively on the Team

The team communicated regularly and divided the workload evenly throughout all project phases. Contributions included analysis, design, implementation, and documentation.

- Weekly team meetings were held throughout both semesters.
- Project reports, presentations, and the analysis and requirements document were prepared together. Each member wrote the parts most related to their work package and reviewed sections written by others.
- Ahmet Yağız Sarıdoğan led the mobile app development and deployment process. He built the main Flutter app structure and managed the CI/CD process.
- Ayça Candan Ataç was responsible for requirements and system design in CS491, and led the MoodAI dataset preparation, module design, and integration in CS492.
- Elif Ceren Çelik led the backend development and designed the system’s database structure. She implemented the Firebase-based backend infrastructure, configuring Firebase Authentication and Cloud Firestore to manage user accounts, reminders, alerts, and caregiver–patient relationships.

- Emine Noor led the literature review in CS491 and owned the privacy and security design throughout both semesters, including the consent model, Firestore Security Rules, and encryption architecture. When backend changes were needed to meet privacy requirements, she made those changes.
- Irmak İmdat led the project's gap analysis and testing strategy, and supported the backend by restructuring the app's architecture and database integrations. She also led the safe zone monitoring subsystem.

7.2 Helping to Create a Collaborative and Inclusive Environment

- The team worked hard to create a collaborative environment throughout the project. They set shared norms early on and applied them.
- At the beginning of each semester, they defined goals and scope together.
- Design decisions went through several rounds of group input before being finalized.
- The group used Google Drive for organized conversations about particular work packages and WhatsApp for everyday communication. This division allowed urgent messages to be kept visible without overpowering design discussions.

7.3 Taking Lead Roles and Sharing Leadership on the Team

Each member owned at least one subsystem. Leads made day-to-day decisions within their area but consulted the group before anything that touched other subsystems.

- The mobile application and deployment were managed by Ahmet Yağız Sarıdoğan. Early in CS491, he chose the Flutter project structure and state management approach, and he was in charge of the release pipeline. As features were developed, he collaborated directly with Elif Ceren on API contracts.
- Ayça Candan Ataç led the MoodAI module. From dataset preparation to integration, she led the entire process. She consulted with project advisor Sinem Sav about detection thresholds and escalation behavior during the design phase.
- Elif Ceren Çelik led the backend and database design. She defined the Firestore schema and made the core decisions on data synchronisation and notification delivery.
- Emine Noor led the design for privacy and security. She created the consent model, wrote the Firestore Security Rules, and established the encryption architecture. When backend changes were necessary to meet privacy requirements, she made those changes herself and consulted Sinem Sav on privacy decisions when needed.
- Irmak İmdat led the project coordination by conducting a gap analysis and an overall testing strategy. She also supported the backend by refactoring the monolithic codebase into a

modular ViewModel architecture, optimizing Firestore queries, and implementing comprehensive security rules.

8. Glossary

AI: Artificial Intelligence

API: Application Programming Interface

AES: Advanced Encryption Standard

BPM: Beats Per Minute

CI/CD: Continuous Integration / Continuous Delivery

FCM: Firebase Cloud Messaging

GDPR: General Data Protection Regulation

Geofencing: Virtual geographic boundary used to monitor device location

GPS: Global Positioning System

HCI: Human–Computer Interaction

HR: Heart Rate

HRV: Heart Rate Variability

HTTPS: Hypertext Transfer Protocol Secure

ISO/IEC 27001: International standard for information security management systems

ISMS: Information Security Management System

JWT: JSON Web Token

KVKK: Kişisel Verileri Koruma Kanunu (Turkish Personal Data Protection Law)

ML: Machine Learning

mHealth: Mobile Health

MoodAI: On-device anomaly detection module used in the ReMind system

NoSQL: Non-relational database model designed for flexible data storage

OS: Operating System

RBAC: Role-Based Access Control

ReMind: Mobile health assistant designed to support dementia patients and caregivers

REST: Representational State Transfer

SDK: Software Development Kit

SOS: Emergency distress signal requesting immediate assistance

TFLite: TensorFlow Lite machine learning framework for on-device inference

TLS: Transport Layer Security

UI: User Interface

UX: User Experience

WCAG: Web Content Accessibility Guidelines

Wi-Fi: Wireless Fidelity

9. References

[1] "Alzheimer’s Disease Facts and Figures," Alzheimer’s Association, 2024. [Online]. Available: https://www.alz.org/alzheimers-dementia/facts-figures.

[2] "World Alzheimer Report 2024: Global changes in attitudes to dementia," Alzheimer’s Disease International, 2024. [Online]. Available: https://www.alzint.org/resource/world-alzheimer-report-2024/.

[3] "2024 Alzheimer’s disease facts and figures," Alzheimer’s & Dementia, vol. 20, no. 5, 2024. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/38689398/.

[4] AngelSense, "GPS Tracker for Elderly with Dementia & Alzheimer’s," [Online]. Available: https://www.angelsense.com/gps-tracker-for-elderly/

[5] Jiobit, "Jiobit Location Monitor," [Online]. Available: https://www.jiobit.com/

[6] Medisafe, "Medisafe Pill & Med Reminder App," [Online]. Available: https://medisafeapp.com/

[7] Constant Therapy, "Constant Therapy – Brain Rehabilitation App," [Online]. Available: https://www.constanttherapy.com/

[8] Apple Inc., "HealthKit Developer Documentation," 2023. [Online]. Available: https://developer.apple.com/documentation/healthkit

[9] Google, "Health Connect API Documentation," 2023. [Online]. Available: https://developer.android.com/health-and-fitness/guides/health-connect

[10] European Parliament and Council, "Regulation (EU) 2016/679 (General Data Protection Regulation)," Official Journal of the European Union, Apr. 2016. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32016R0679

[11] T.C. Resmi Gazete, "Kişisel Verilerin Korunması Kanunu (KVKK), Kanun No. 6698," Apr. 2016. [Online]. Available: https://www.mevzuat.gov.tr/mevzuat?MevzuatNo=6698&MevzuatTur=1&MevzuatTertip=5

[12] W3C, "Web Content Accessibility Guidelines (WCAG) 2.2," Oct. 2023. [Online]. Available: https://www.w3.org/TR/WCAG22/

[13] ISO/IEC, "ISO/IEC 27001:2022, Information Security Management Systems," 2022. [Online]. Available: https://www.iso.org/standard/27001

[14] Google, "Flutter, Build apps for any screen," 2024. [Online]. Available: https://flutter.dev

[15] Google, "Firebase Documentation," 2024. [Online]. Available: https://firebase.google.com/docs