Bilkent University
Department of Computer Engineering

CS492: Senior Design Project II

Design Project Final 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

04.05.2026

Table of Contents

1. Introduction

Dementia and Alzheimer's disease affect more than 55 million people worldwide today, and projections suggest this number will rise to 78 million within the next decade [1]. These conditions create substantial emotional challenges and practical difficulties not only for the patients themselves but also for their family members and 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 designed to address these problems by combining safety monitoring, daily routine assistance, and cognitive support into a single privacy-first ecosystem. Unlike existing market solutions, which tend to focus on a single capability such as GPS tracking (AngelSense, Jiobit) or medication reminders (Medisafe), ReMind unifies these capabilities under one platform whose central design principle is to share the minimum necessary information at the right time. The application supports two user roles, Patient and Caregiver, each with a tailored interface, and exchanges sensitive data only when a safety condition has been triggered.

The patient-facing application provides a calm, accessible interface featuring large buttons, minimal text, and predictable navigation. Patients can access medication and routine reminders, take short "Mood Check-ins," play non-competitive cognitive games, request "Take Me Home" navigation, and trigger an SOS alert when they feel unsafe. The caregiver-facing application provides a dashboard that surfaces alerts in priority order, summarises recent reminders and mood trends, allows up to three safe zones to be configured per patient, and displays the patient's exact GPS location only after a safe-zone exit has been confirmed.

Behind the user-facing features, ReMind incorporates an on-device anomaly-detection module called MoodAI. The module analyses behavioural signals (mood check-ins, phone usage patterns, and physical activity) to compute a daily anomaly score and surfaces support cards or escalates alerts to the caregiver according to configurable thresholds. The model is designed for on-device deployment via TensorFlow Lite; smartwatch and HealthKit integration is planned for a follow-on phase.

The whole system is implemented in Flutter for cross-platform support on Android and iOS, and is backed by Firebase Authentication, Cloud Firestore, and Firebase Cloud Messaging. All cloud-stored data is encrypted at rest through Firestore's AES-256 encryption feature, all communication uses TLS 1.3. The privacy design was informed by GDPR and KVKK principles. As a prototype, full regulatory compliance is out of scope for this project.

This final report documents the as-built ReMind system. Section 2 explains the functional and non-functional requirements that guided implementation. Section 3 explains the final architecture and design including subsystem decomposition, hardware/software mapping, persistent data management, and access control. Section 4 describes how each subsystem was implemented in code. Section 5 presents the suite of functional and non-functional test cases used to validate the system. Section 6 details the maintenance plan. Section 7 reflects on the various engineering, ethical, and societal factors considered in the design. Sections 8 through 11 conclude the report with future work, a user manual, a glossary, and references.

2. Requirements Details

This section enumerates the requirements that the ReMind system was built to satisfy. The functional requirements describe what the system does for its users, organised by feature area. The non-functional requirements describe quality attributes (usability, reliability, performance, security, privacy, scalability, and supportability) that influenced the architecture, the user interface, and the operational policies of the application.

2.1. Functional Requirements

2.1.1. User and Account Management

- **FR-UM-01 User Roles.** The system supports two primary user roles: Patient and Caregiver. Each role has its own onboarding flow, dashboard, and permitted actions.
- **FR-UM-02 Registration and Authentication.** Both Patient and Caregiver can register, log in, and log out using email-based authentication, either with a username/password combination or via a passwordless email verification link delivered through Firebase Authentication.
- **FR-UM-03 Account Verification.** The system verifies that the email address provided at registration belongs to the user before allowing any authenticated action by sending a verification link or six-digit code.
- **FR-UM-04 Caregiver-Patient Linking.** The system links a Caregiver and a Patient profile only after both parties explicitly approve the linking request. Linking is initiated via a six-digit code generated by the Patient's device.
- **FR-UM-05 Link Revocation.** Either the Patient or the Caregiver can terminate an existing link at any time from the Settings screen. After unlinking, all linked-data access is revoked and a final notification is sent to the other party.

2.1.2. Location Monitoring and Safe Zones

- **FR-LOC-01 Safe Zone Definition.** A Caregiver may set up and modify up to three circular geofences ("safe zones") per Patient using a map-based interface.
- **FR-LOC-02 Safe Zone Limits.** While the Patient is inside a safe zone, the application uses OS-level geofencing to monitor zone membership, conserving battery and avoiding continuous GPS sampling.
- **FR-LOC-03 Safe Zone Violation Detection.** Geofencing detects when the Patient enters or exits a safe zone. A violation is confirmed only after the Patient has remained outside the boundary for at least 60 seconds or has moved more than 50 metres beyond it, in order to suppress false alarms caused by GPS jitter.
- **FR-LOC-04 Safe Zone Exit Alert.** When a violation is confirmed, the Caregiver receives a high-priority alert that includes the reason, last known location, and timestamp.
- **FR-LOC-05 Current Location Display.** The Caregiver interface shows whether the Patient is inside or outside a safe zone. The exact GPS coordinates of the Patient become visible to the Caregiver only when the Patient is outside a safe zone.

- **FR-LOC-06 Location Monitoring Indicator.** The Patient application displays a clear, persistent visual indicator whenever active location tracking is enabled, in keeping with the principle of monitoring transparency.

2.1.3. "Take Me Home" Navigation

- **FR-NAV-01 Take Me Home Feature.** The Patient application provides a "Take Me Home" feature that, when activated, guides the Patient back to a previously selected home location using map-based navigation with simple, easy-to-understand visual instructions.
- **FR-NAV-02 Foreground Requirement.** Navigation operates while the Patient application is in the foreground. Background behaviour conforms to the mobile operating system's background-execution restrictions.

2.1.4. Reminders, Medication, and Daily Routines

- **FR-REM-01 Schedule Definition.** Both Patients and Caregivers can create and edit medication schedules and daily-routine reminders for the Patient through the application.
- **FR-REM-02 Reminder Delivery.** The Patient application delivers reminders at the scheduled time via local notifications and in-app prompts, even if the device is offline.
- **FR-REM-03 Reminder Interaction.** For each reminder, the Patient can respond with one of three options: "Done", "Skipped", or "Snoozed".
- **FR-REM-04 Snooze Behavior.** When a reminder is snoozed, the system automatically reschedules it after a short configurable delay.
- **FR-REM-05 Adherence Logging.** The system logs every reminder response, including timestamp and response type, so that Caregivers can review adherence trends.

2.1.5. Mood Check-ins and Cognitive Support

- **FR-MOOD-01 Mood Check-in Prompts.** The Patient application periodically prompts the Patient to complete a Mood Check-in. Each check-in is designed to be completed in roughly 10-15 seconds using simple questions and image selection.
- **FR-MOOD-02 Mood Timeline Storage.** Mood Check-in results are stored locally per Patient, including timestamp and the Patient's responses, and form the input feed for both the Caregiver mood timeline and the MoodAI module.
- **FR-MOOD-03 Mood Trend Visualization.** The Caregiver interface allows mood trends to be reviewed over time, with daily and weekly summaries.
- **FR-COG-01 Cognitive Games Library.** The Patient application offers a library of memory and cognitive games that are calming and non-competitive.
- **FR-COG-02 Non-Competitive Design.** Games explicitly avoid competitive elements: there are no leaderboards, no time limits, and no failure states.

2.1.6. Sensor and Smartwatch Integration

- **FR-SEN-01 Smartwatch-Phone Pairing (planned).** Smartwatch pairing is designed so that a Patient's watch can pair with the application using the same authenticated account. This feature is planned for a follow-on phase.

- **FR-SEN-02 Supported Wearable Platforms (planned).** The design targets Apple Watch via HealthKit on iOS and Wear OS smartwatches via Health Connect on Android. Integration is planned for a follow-on phase.
- **FR-SEN-03 Phone-Only Mode.** The current implementation operates in phone-only mode, using on-board sensors for activity data. Smartwatch fallback notification is planned for when wearable integration is added.
- **FR-SEN-04 Periodic Vital and Activity Collection (planned).** When smartwatch integration is added in a follow-on phase, the system will periodically collect basic vital and activity data (heart rate, heart-rate variability, and step count) at sampling intervals tuned to balance freshness against battery and privacy constraints.

2.1.7. MoodAI Anomaly Detection and Support Cards

- **FR-AI-01 MoodAI Module.** The Patient application includes a MoodAI module that aggregates the Patient's behavioural data into a daily feature row covering phone-usage statistics (unlock count, screen-on time, evening-usage ratio), physical-activity counts, and mood check-in responses (OASIS image selection mapped to valence and arousal), and computes a daily anomaly score using a pretrained FT-Transformer with a self-supervised masked-reconstruction objective. The model is small enough (≈160k parameters) to run on-device through TensorFlow Lite (LiteRT).
- **FR-AI-02 Input Adaptation by Mode.** The MoodAI module adapts its feature set depending on the available sensors. It does not assume that all signals are present and the schema reserves structural placeholders for sleep, heart-rate, and richer mood signals so that the same model can ingest smartwatch data when it becomes available without retraining the architecture. The model currently operates in phone-only mode.
- **FR-AI-03 Personalised Anomaly Classification.** After a fourteen-day per-Patient baseline-calibration window, the daily anomaly score is converted to a personal z-score and classified into three levels: NORMAL, MEDIUM (gradual behavioural decline detected over multiple days), and HIGH (significant sudden behavioural change, enhanced $z \geq 2$). The system's behaviour adapts to the level.
- **FR-AI-04 Support Card Triggering.** When MoodAI classifies the current day as MEDIUM, a support card is shown on the Patient's phone and the system follows the Patient's response.
- **FR-AI-05 Support Card Options.** Support cards offer at minimum the following responses: "I'm okay", "Alert my caregiver", and "Take me home".
- **FR-AI-06 Escalation of Persistent or HIGH Anomalies.** If a MEDIUM classification persists past its configured time window, or if MoodAI classifies a day as HIGH, the system escalates by generating a high-priority alert to the Caregiver. Each escalation includes the human-readable rationale produced by the explanation system.

2.1.8. Alerts and Caregiver Notification Management

- **FR-ALR-01 High-Priority Alert Conditions.** The system generates a high-priority alert to the Caregiver when the Patient exits a configured safe zone, when MoodAI detects a high anomaly, when a moderate anomaly persists beyond a configured duration, or when the Patient explicitly triggers SOS / "Alert my caregiver".

- **FR-ALR-02 Alert Content.** Each high-priority alert contains at minimum a human-understandable reason, the Patient's last known location, and a timestamp.
- **FR-ALR-03 Alert Delivery Channels.** Push notifications via Firebase Cloud Messaging are the default and primary delivery channel, with optional SMS or email channels available depending on platform.
- **FR-ALR-04 Alert History View.** The Caregiver interface provides an alert history view per Patient, sortable by time and severity.
- **FR-ALR-05 Notification Preferences.** Caregivers may configure quiet hours, preferred alert channels, and the sensitivity level for MoodAI-generated alerts.

2.1.9. Privacy, Consent, and Data Governance

- **FR-PRV-01 Onboarding Consent Flow.** During onboarding, an explicit per-category consent flow lets the Patient approve linking, approve collection and limited sharing of location and health data, and configure when and how location is shared.
- **FR-PRV-02 Privacy & Data Use Page.** A dedicated "Privacy & Data Use" page describes the data categories collected, how each category is used, and which data is shared with the Caregiver and under what conditions.
- **FR-PRV-03 Minimal Server Storage.** Only essential summary events are stored on the server. Raw sensor data and detailed behaviour logs remain on the device.
- **FR-PRV-04 Monitoring Awareness.** Clear visual indicators are displayed whenever monitoring or data collection is active.
- **FR-PRV-05 Account and Data Deletion.** Patients may request deletion of their account and all associated cloud-stored data, subject to legal retention requirements.
- **FR-PRV-06 Data Export for Clinical Use.** The system can export a PDF summary covering alert history, reminder adherence, and mood trends for a chosen time range, suitable for sharing with a clinician.

2.2. Non-functional Requirements

2.2.1. Usability

ReMind serves an unusually broad range of cognitive and physical abilities. The Patient interface therefore uses large buttons, restrained colour palettes, high contrast, and minimal text. Critical features (Take Me Home, reminders, mood check-ins, and SOS) are reachable from the home screen without menu navigation. The Caregiver dashboard surfaces information by priority: alerts first, then today's summary, then weekly trends. A short onboarding flow with tooltips explains safe-zone setup, alert logic, and sensor permissions in plain language. The system meets WCAG 2.2 Level AA, never relies on colour alone to convey information, and avoids time-pressured interactions.

2.2.2. Reliability

The system continues to operate during network fluctuations. Background tasks for location monitoring do not require user intervention. The MoodAI module performs inference locally and is therefore independent of network availability. After a crash, the application returns to a safe operational state with all current alerts and logs preserved.

2.2.3. Performance

Safe-zone-breach alerts are generated and dispatched within five seconds of detection. The MoodAI module produces an anomaly inference within one second so that support cards remain perceptibly real-time. Background polling of GPS, heart rate, and movement is adaptive: low-frequency during normal activity to preserve battery, high-frequency when an anomaly is detected or after a safe-zone breach. The Caregiver dashboard renders the latest summary view within three to five seconds even when displaying months of historical data.

2.2.4. Security

Role-based access control ensures Caregivers only see data they are explicitly permitted to see. Firebase Authentication enforces email verification on every login. Patient-Caregiver linking requires explicit Patient approval via a six-digit code; until that approval, no patient data is exposed. All data stored in Firestore is encrypted at rest with AES-256, and all communication between the application and Firebase uses TLS 1.3. Free-text mood notes are stored in Firestore under the standard AES-256 at-rest encryption provided by Google Cloud infrastructure.

2.2.5. Privacy

The system applies data-minimisation aggressively. Mood-check-in inputs and physiological readings are processed locally; only the resulting anomaly level is transmitted. Location data is event-based: exact GPS coordinates leave the device only on a safe-zone exit or an SOS trigger. Patients consent per category at registration and may withdraw consent at any time from the Settings screen, at which point cloud data is purged according to the documented retention policy.

2.2.6. Scalability

The cloud architecture is designed to support thousands of caregiver-patient pairs without degradation in alert delivery latency or dashboard responsiveness, and a single Caregiver may manage multiple Patients. Cloud storage handles months of retained summary data, and indexed Firestore queries provide constant-time access to historical trends. The MoodAI module runs entirely on the device, so compute costs scale with users rather than with the central infrastructure. New cognitive games and improved anomaly models can be added without requiring schema or contract changes.

2.2.7. Supportability

Each subsystem is packaged as an independent module so that bug-fixes and feature updates can be released without disturbing other parts of the application. The system maintains local diagnostic logs (without raw sensitive data) for crash analysis. Anonymised telemetry (opt-in only) is uploaded to support performance debugging. When a configuration problem occurs (e.g. missing sensor permission), the application surfaces a plain-language notification with restoration steps and informs the Caregiver if the missing permission affects safety functions.

3. Final Architecture and Design Details

3.1. Overview

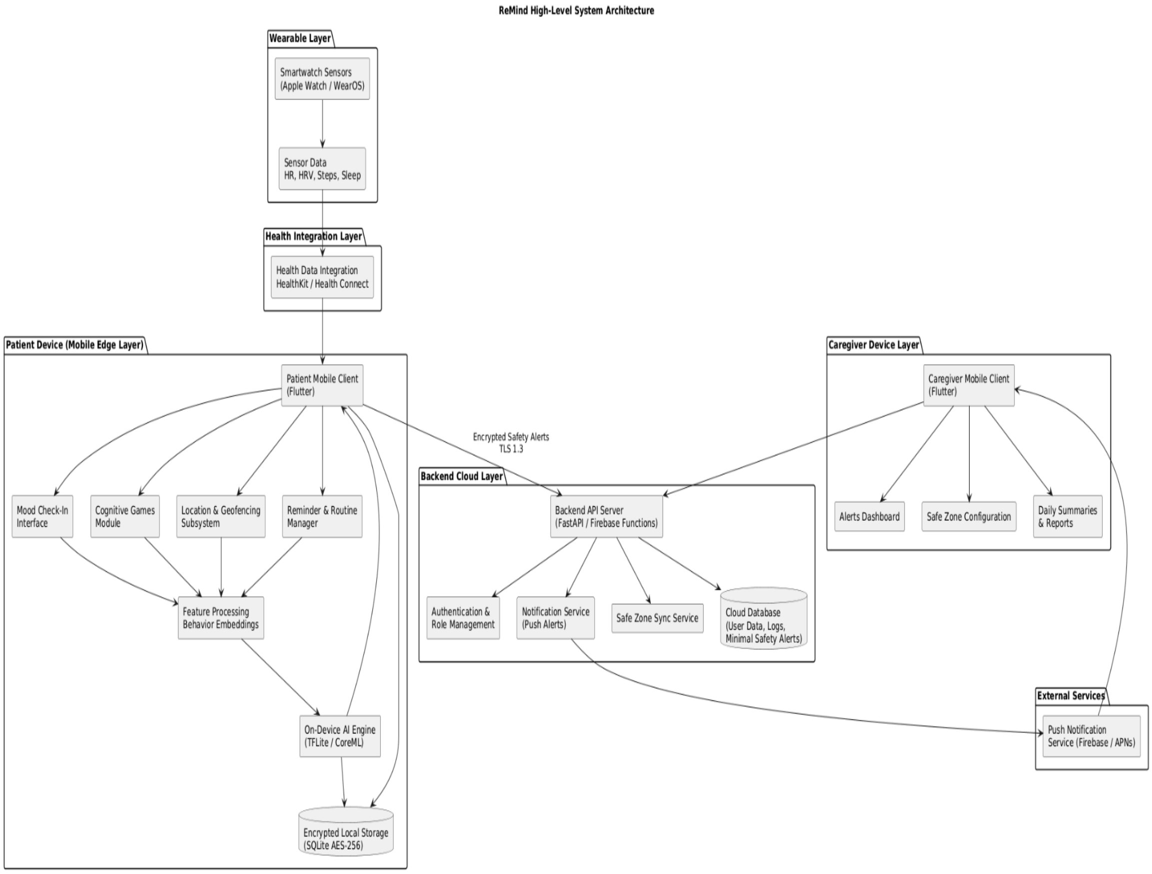
ReMind has a modular three-tier client-server architecture composed of (i) a mobile application layer, (ii) a backend service layer, and (iii) a data storage layer. The mobile application is built on Flutter so that a single Dart codebase can produce native binaries for both Android and iOS without sacrificing platform-specific UX expectations. Within the application there are two distinct user experiences: a simplified Patient interface optimised for accessibility, and a Caregiver dashboard that surfaces alerts, summaries, and configuration tools in priority order.

The backend layer is implemented on Firebase. Firebase Authentication validates user identity, Cloud Firestore is the system of record for user accounts, caregiver-patient relationships, reminder schedules and adherence logs, alert history, and summary event logs, and Firebase Cloud Messaging dispatches push notifications to the Caregiver when an alert needs to be raised. The architecture is designed so that wearable data would be read through Apple HealthKit on iOS and Google Health Connect on Android directly inside the Patient device. Wearable integration is planned for a follow-on phase.

The MoodAI module is the architectural feature that most distinguishes ReMind. It runs locally on the Patient device using TensorFlow Lite; the model is fine-tuned to the Patient's own behavioural baseline. By performing inference on the device, the design eliminates a class of privacy risks that plague cloud-based health AI products and makes the anomaly-detection pipeline robust to intermittent connectivity. Only the resulting anomaly level (normal / moderate / high) and any triggered alert events leave the device.

All inter-tier communication is mediated by REST APIs over HTTPS / TLS 1.3, enforced automatically by the Firebase SDK. Above all, the architecture supports modularity, scalability, and privacy: components evolve independently while continuing to communicate securely between client devices, Firebase services, and persistent storage.

3.2. Subsystem Decomposition



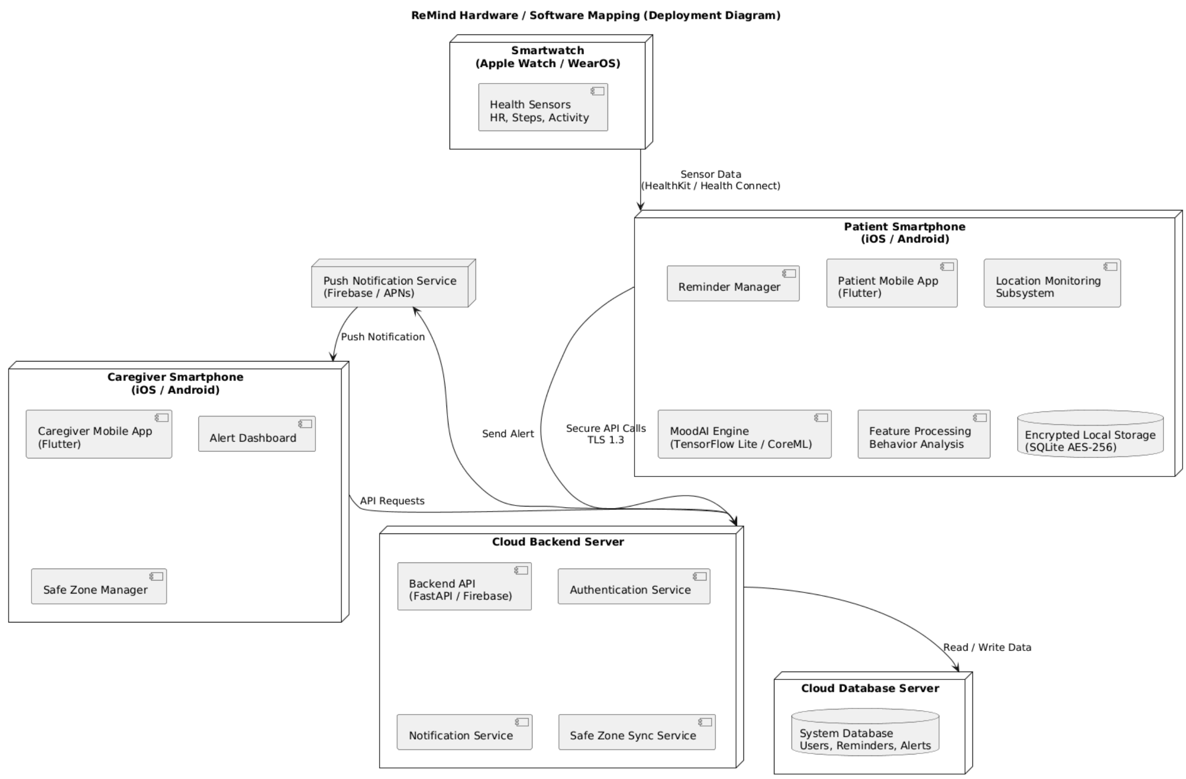
Link: https://drive.google.com/file/d/1pYvri1RrXQ_CsyrKv3VIZgMAttMIaFwS/view?usp=drive_link

ReMind is decomposed into five subsystems, each with a clearly delineated responsibility and a stable interface to the others:

- **Mobile Application Service.** The Flutter application that presents the Patient and Caregiver interfaces. It receives user input, presents notifications, manages local persistence, and forwards events to the backend through the Firebase SDK.
- **Backend API and Data Management Service.** The Firebase-hosted services that authenticate users, persist data in Cloud Firestore, broadcast push notifications, and enforce Security Rules for role-based access.
- **Safe Zone & Location Monitoring Service.** OS-level geofencing for safe-zone membership tracking inside the Patient app, plus the alerting and tracking pipeline that activates when a violation is confirmed.
- **Notification and Alert Service.** The fan-in component for all events that may produce a Caregiver-facing alert (safe-zone breach, MoodAI anomaly, SOS, system events). Provides queueing, prioritisation, and delivery via Firebase Cloud Messaging.
- **MoodAI Module.** The behavioural anomaly-detection pipeline built on an FT-Transformer that has been pretrained with a self-supervised masked-reconstruction objective on the StudentLife behavioural-sensing dataset. MoodAI consumes a daily feature row built from phone-usage statistics (unlock count, screen time, evening-usage ratio), physical activity (daily active-reading

count), and mood check-in responses (OASIS image selection mapped to valence and arousal). After fourteen days of personal-baseline calibration, each day is classified as NORMAL, MEDIUM, or HIGH by combining the model's reconstruction-based anomaly score with per-feature personal z-scores. The model is small (≈160k parameters) and on-device-aware so that production deployment via TensorFlow Lite (LiteRT) is straightforward.

3.3. Hardware/Software Mapping



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

The Patient and Caregiver smartphones (Android and iOS) are the principal client devices.They run the Flutter mobile application, which interacts with native sensors. HealthKit / Health Connect integration and on-device TensorFlow Lite inference are planned for a follow-on phase. Patient devices contribute interaction data, location data, and mood check-in inputs; Caregiver devices primarily render alerts and summary charts.

Smartwatches (Apple Watch, Wear OS) are planned as auxiliary sensor sources in a follow-on phase. The current implementation uses phone-only mode with on-board sensors.

Firebase provides the cloud-side execution environment. Firebase Authentication, Cloud Firestore, and Firebase Cloud Messaging are consumed through the Firebase SDK. There is no custom server image; all backend logic that the team owns is expressed as Cloud Functions and as Firestore Security Rules. All connections are protected by TLS 1.3 enforced by the SDK.

3.4. Persistent Data Management

Data management strictly follows privacy by design: data is split between device-local storage and cloud-resident storage, and the boundary between them is intentional. Raw sensor signals, raw mood-check-in inputs, the daily MoodAI feature row, and the per-Patient anomaly baseline are kept on the device. Only summary events are written to Firestore: alert classification (NORMAL / MEDIUM / HIGH), reminder responses, location snapshots taken during a safe-zone breach, alert-trigger metadata.

In transit, summary events are protected by TLS 1.3. At rest, Firestore-stored data is encrypted with AES-256 by Google Cloud infrastructure. Free-text mood notes are stored in Firestore under the standard AES-256 at-rest encryption provided by Google Cloud infrastructure. The cloud has enough storage for the multi-month historical trends that Caregivers expect, and Firestore's indexing keeps query latency essentially constant as the number of caregiver-patient pairs grows.

3.5. Access Control and Security

ReMind enforces strict separation between the Patient role and the Caregiver role at both the interface layer and the API layer. A Caregiver cannot read a Patient's raw sensor data, mood-check-in inputs, or cognitive-game outcomes. Caregivers may only view alert events, reminder adherence summaries, and the Patient's exact location after that Patient has left a designated safe zone. Authentication is performed by Firebase Authentication using either email/password or passwordless email link verification, and email verification is mandatory before any authenticated activity is permitted.

Caregiver-Patient linking requires explicit, two-sided approval via a six-digit code generated on the Patient device. Either party may unlink at any time from the Settings screen. During onboarding, the Patient must complete an explicit per-category consent flow that covers data collection, caregiver linking, and the conditions under which the Caregiver may obtain the Patient's location. Firestore Security Rules are the canonical enforcement point: they encode role-based access, link-state checks, and field-level permissions, and they are exercised by the test suite.

The security design was informed by ISO/IEC 27001 principles and GDPR/KVKK privacy-by-design concepts. As a prototype, formal regulatory compliance is not implemented.

4. Development/Implementation Details

This section describes how each subsystem of ReMind was implemented. The discussion is organised around the layers of the architecture: the Flutter frontend, the Firebase backend, the Cloud Firestore database, the on-device MoodAI module, and the cloud / DevOps configuration.

4.1. Frontend (Flutter Mobile Application)

The frontend is a Flutter application written in Dart. The choice of Flutter satisfied two constraints simultaneously: cross-platform parity (a single codebase for Android and iOS) and direct access to the native sensor and notification APIs that the safety subsystem depends on. The application is organised as a layered MVVM (Model-View-ViewModel) architecture in which views remain dumb while ViewModels coordinate calls into platform services and into the Firebase repository layer.

There are two distinct user experiences sharing the same codebase. After authentication, the application inspects the user's role document in Firestore and routes the user to either the Patient shell or the Caregiver shell.

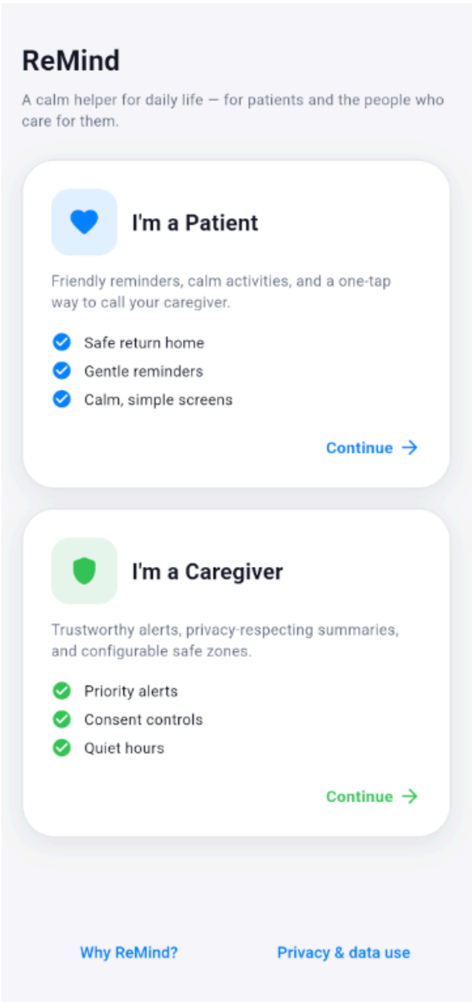
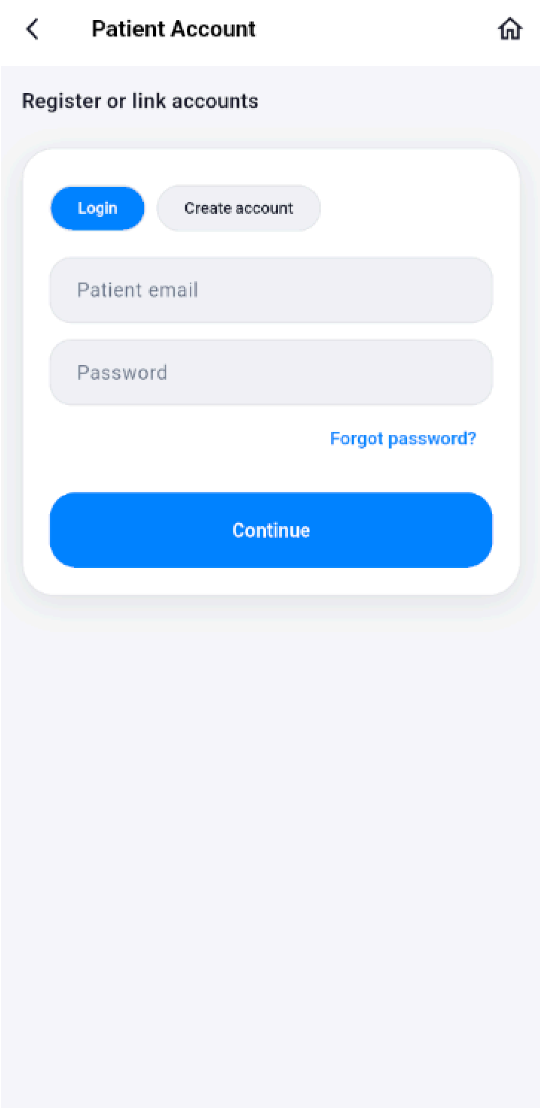
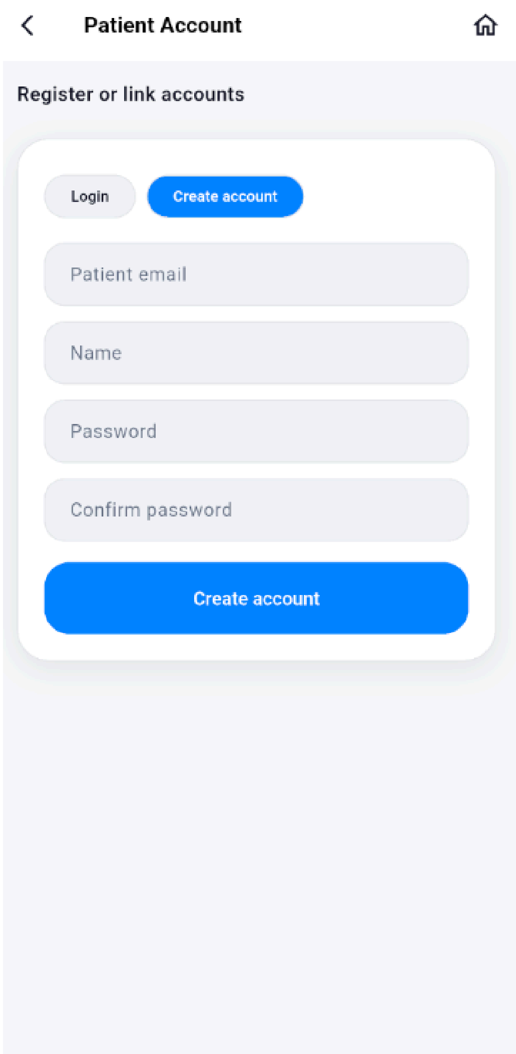
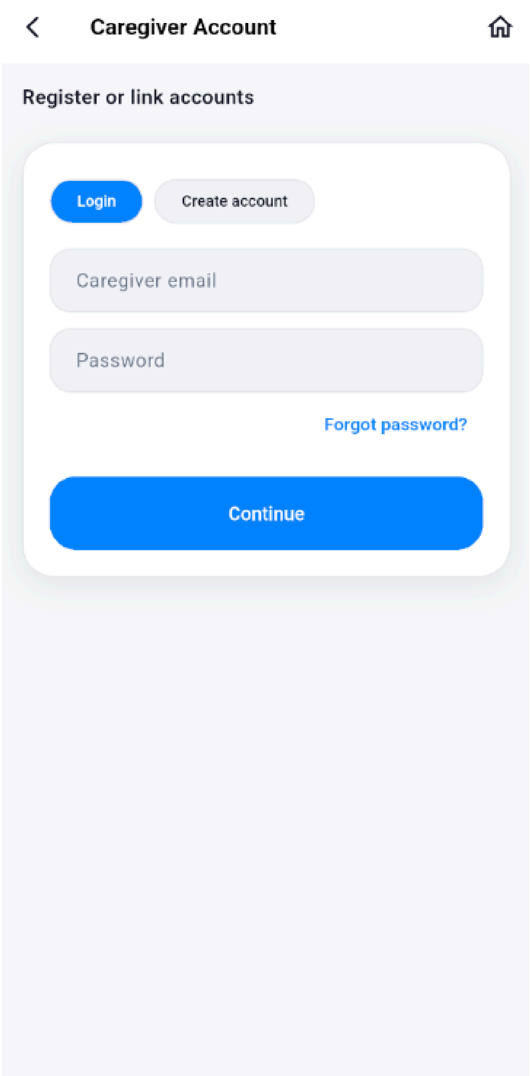


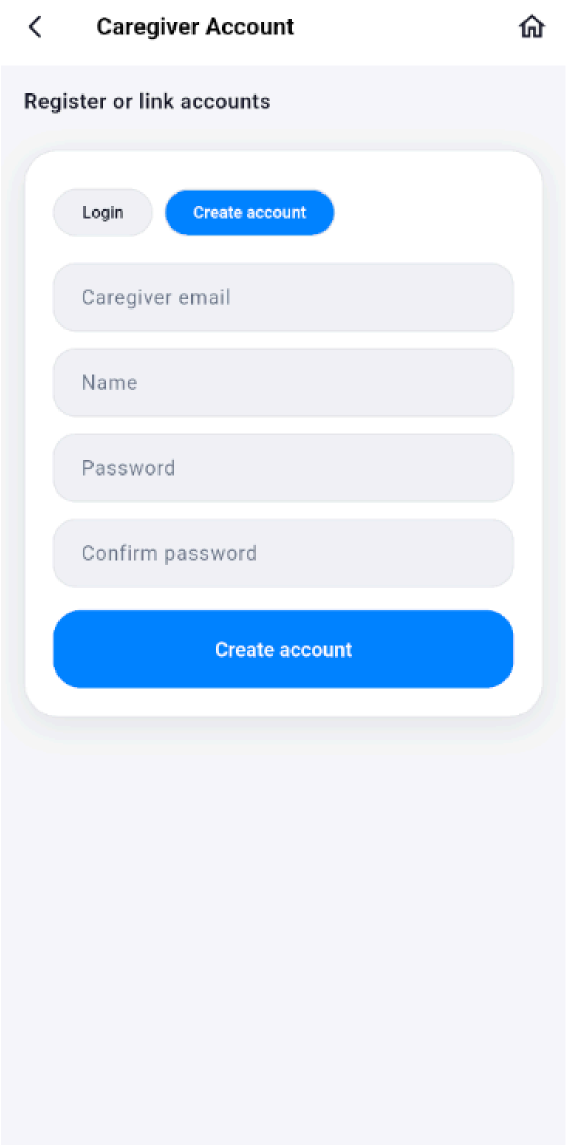
Figure 3. Role selection screen.



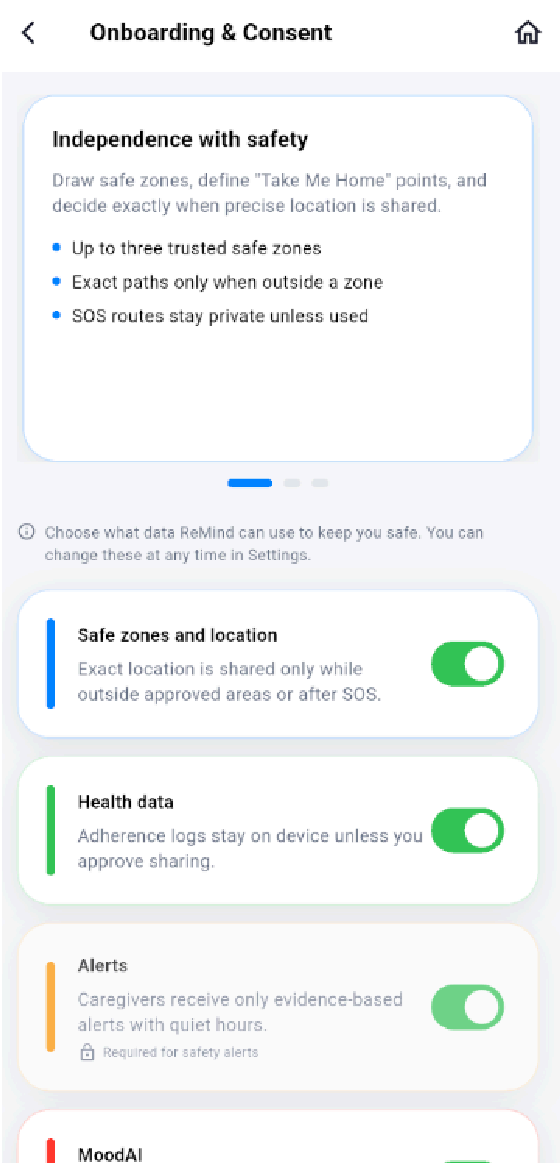


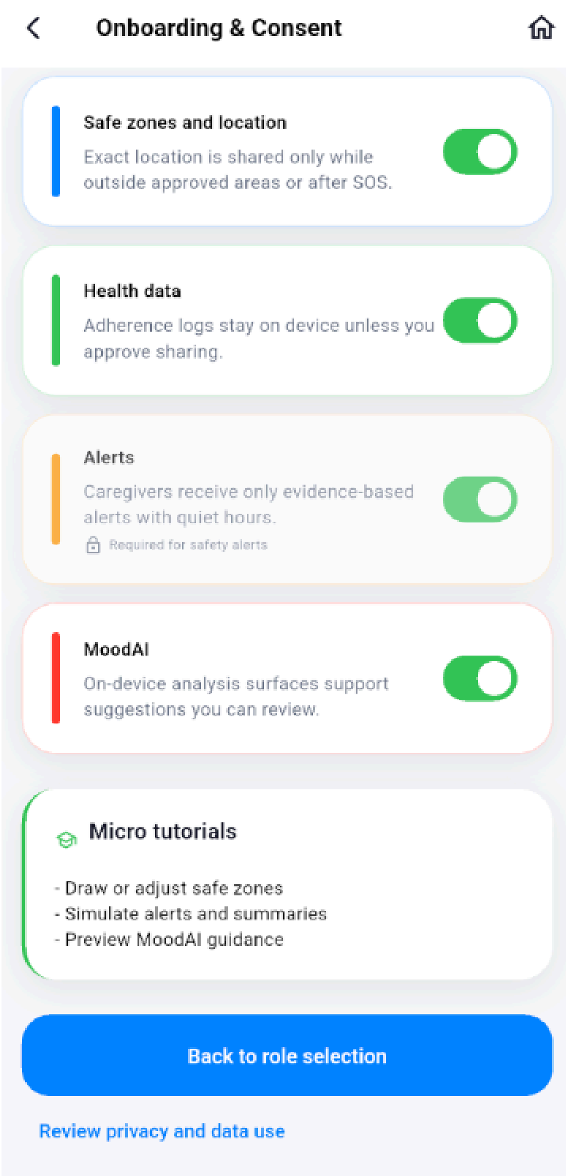
Figures 4 and 5. Patient account login and register.



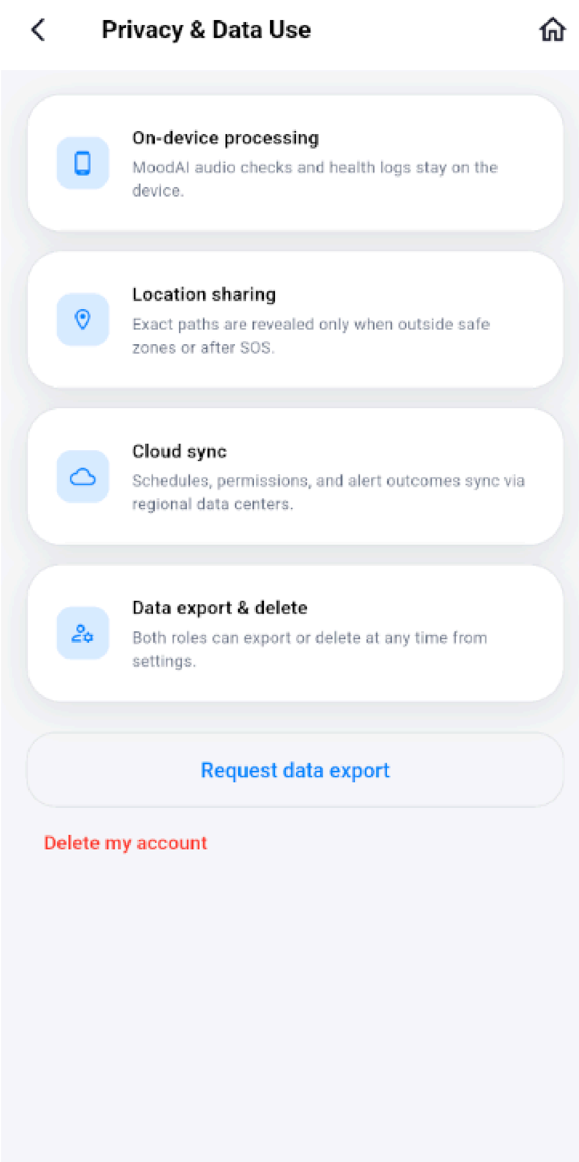


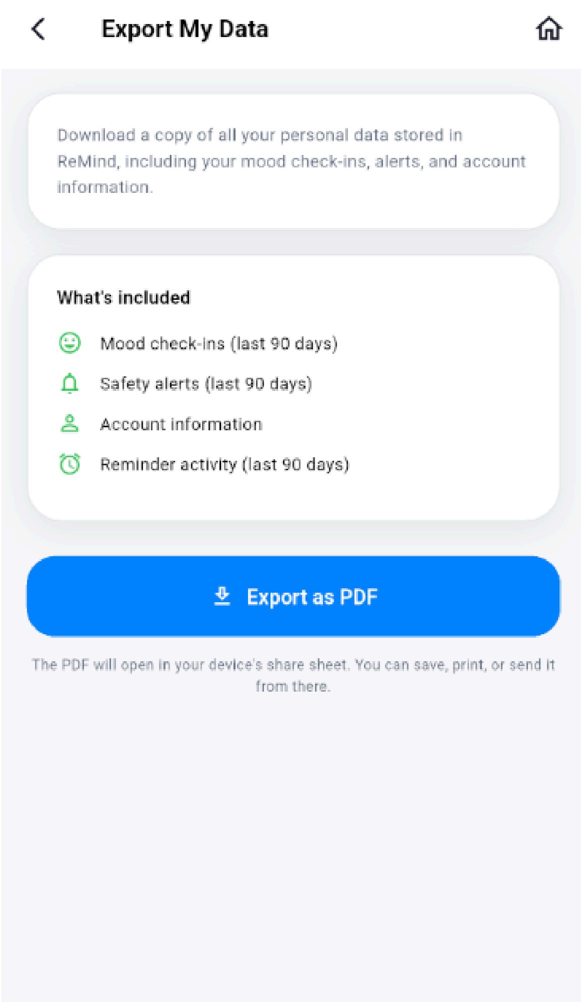
Figures 6 and 7. Caregiver account login and register.





Figures 8 and 9. Onboarding and consent flow.





Figures 10 and 11. Privacy & Data Use page and Patient data export screen.

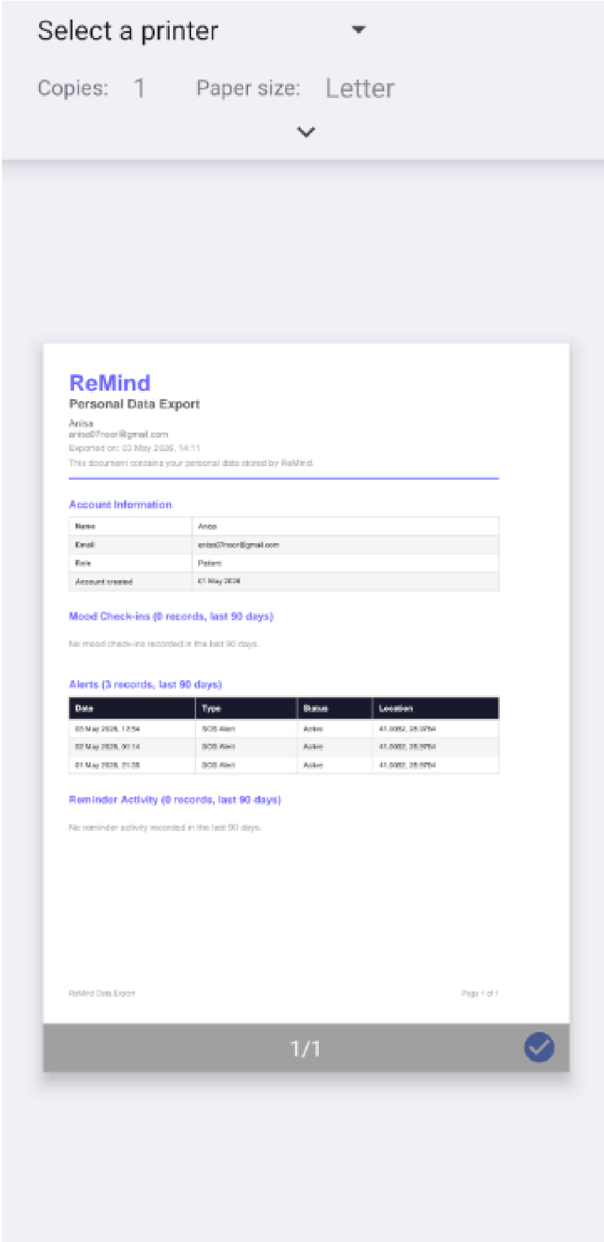
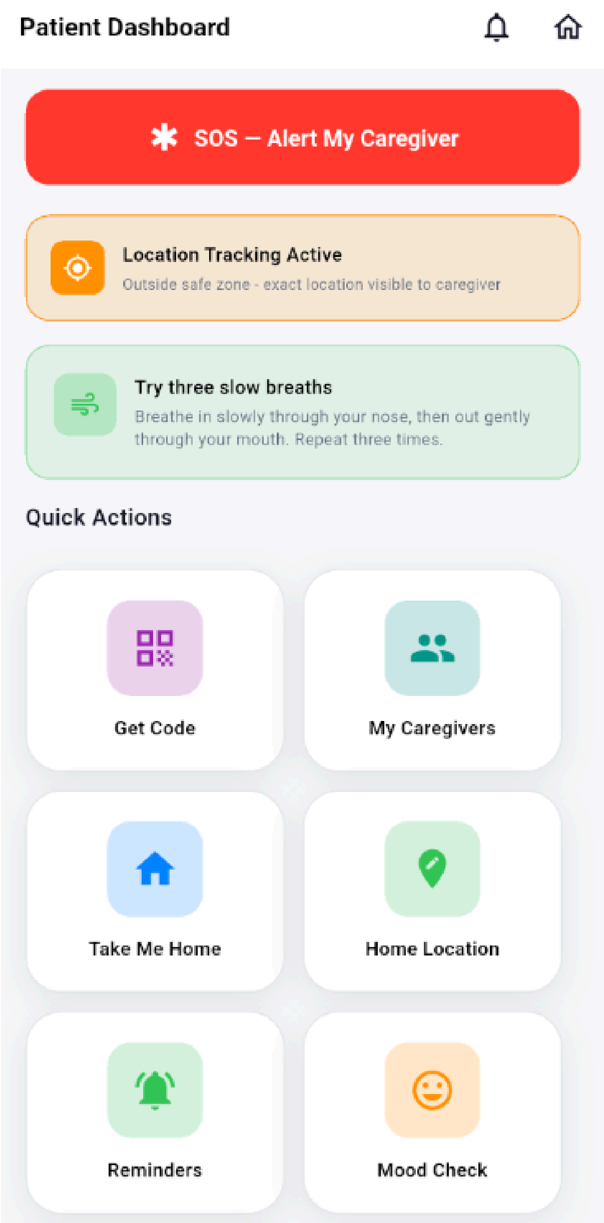
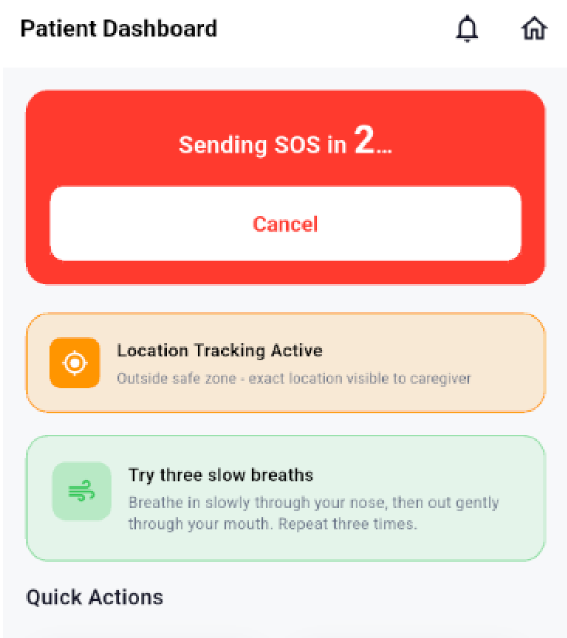
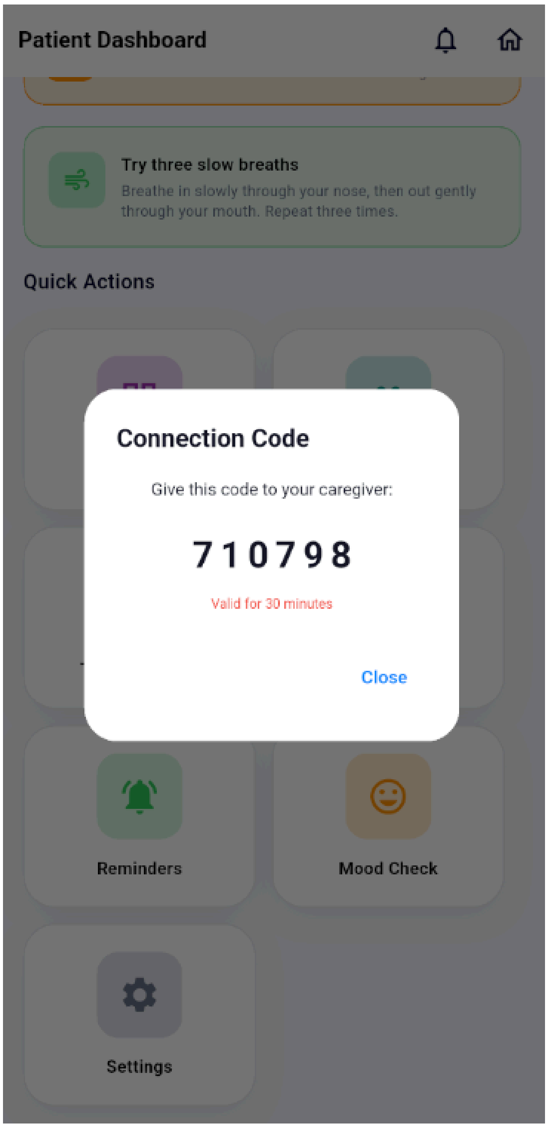


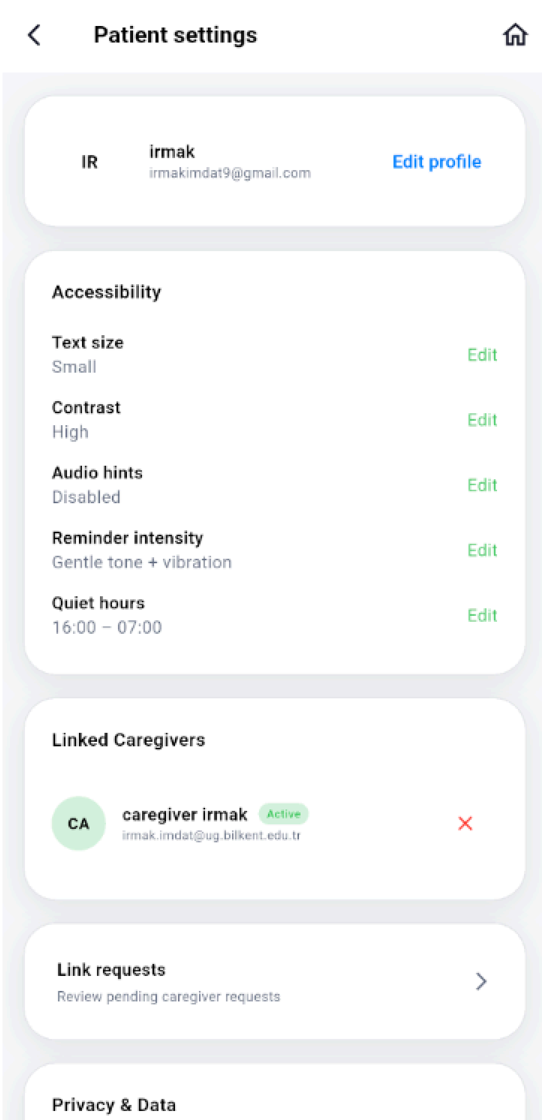
Figure 12. Patient data export (printed PDF preview).



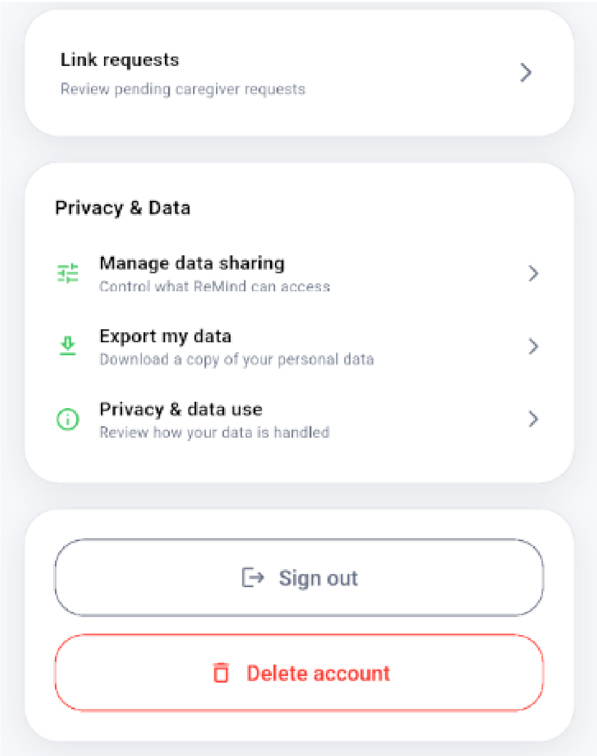


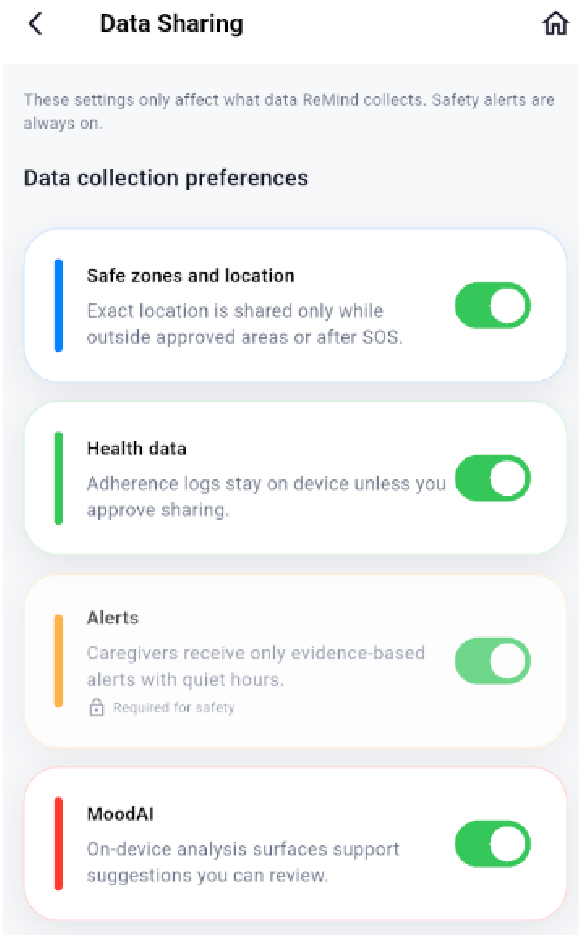
Figures 13 and 14. Patient dashboard (normal state) and Dashboard (SOS countdown in progress).



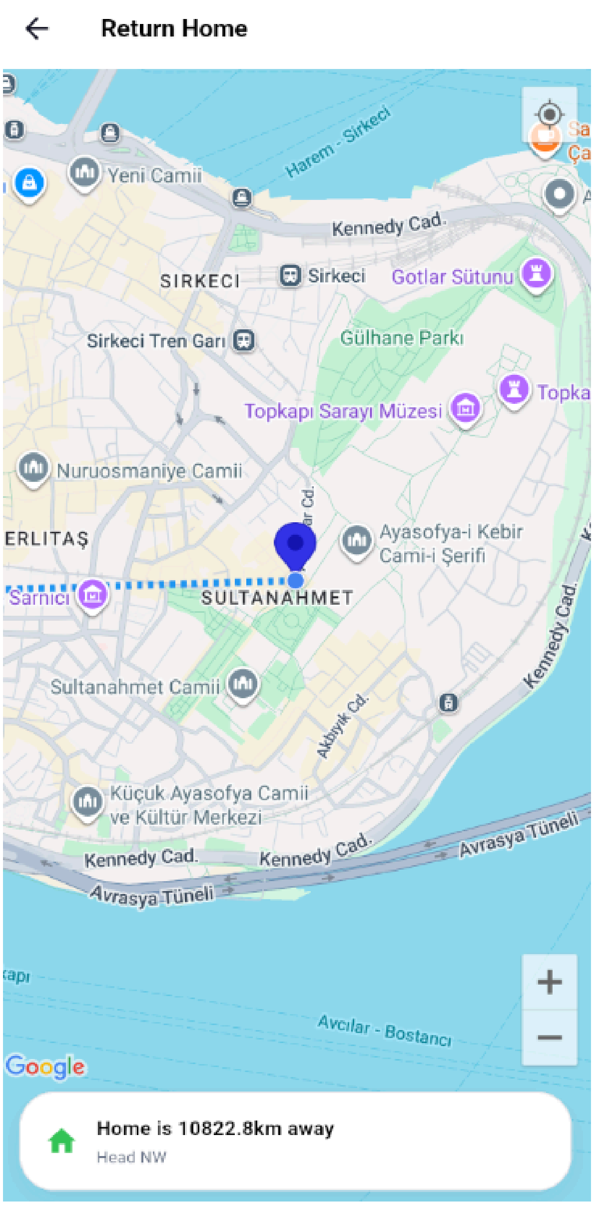


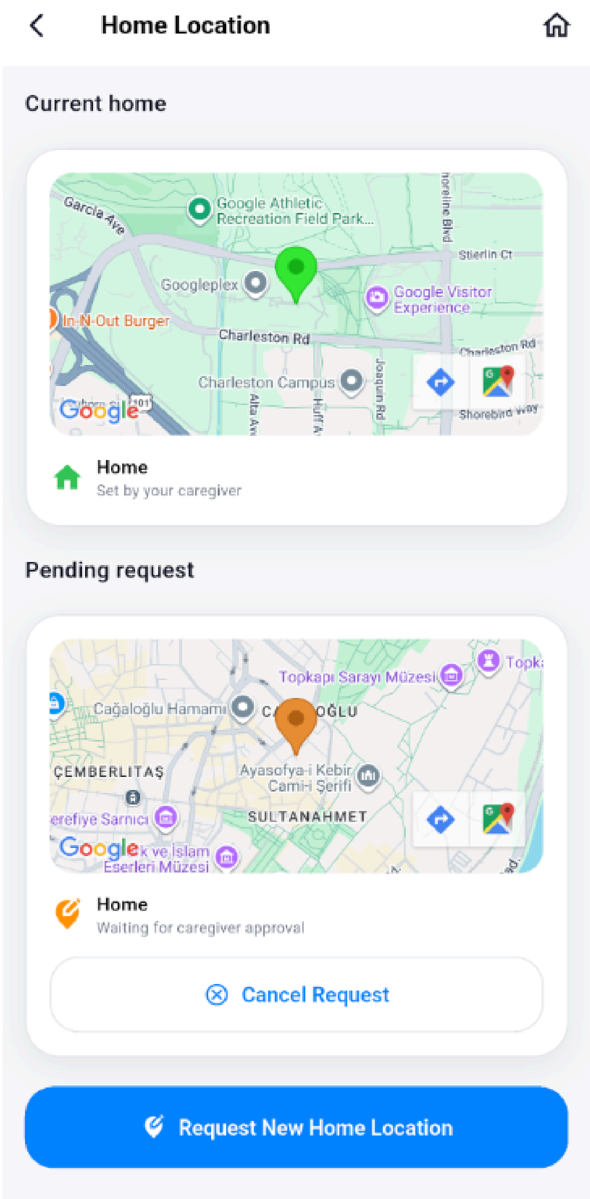
Figures 15 and 16. Connection code dialog and Patient settings.





Figures 17 and 18. Patient settings and Data sharing preferences.





Figures 19 and 20. Take Me Home navigation view and Home Location screen (current home and pending request).

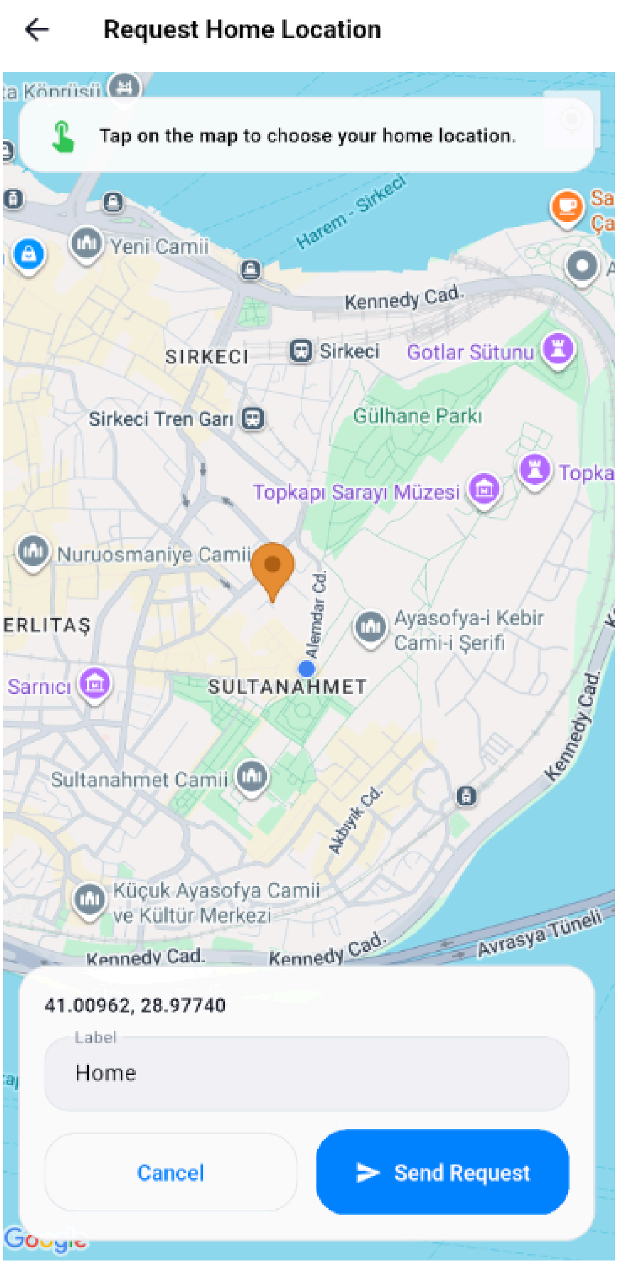
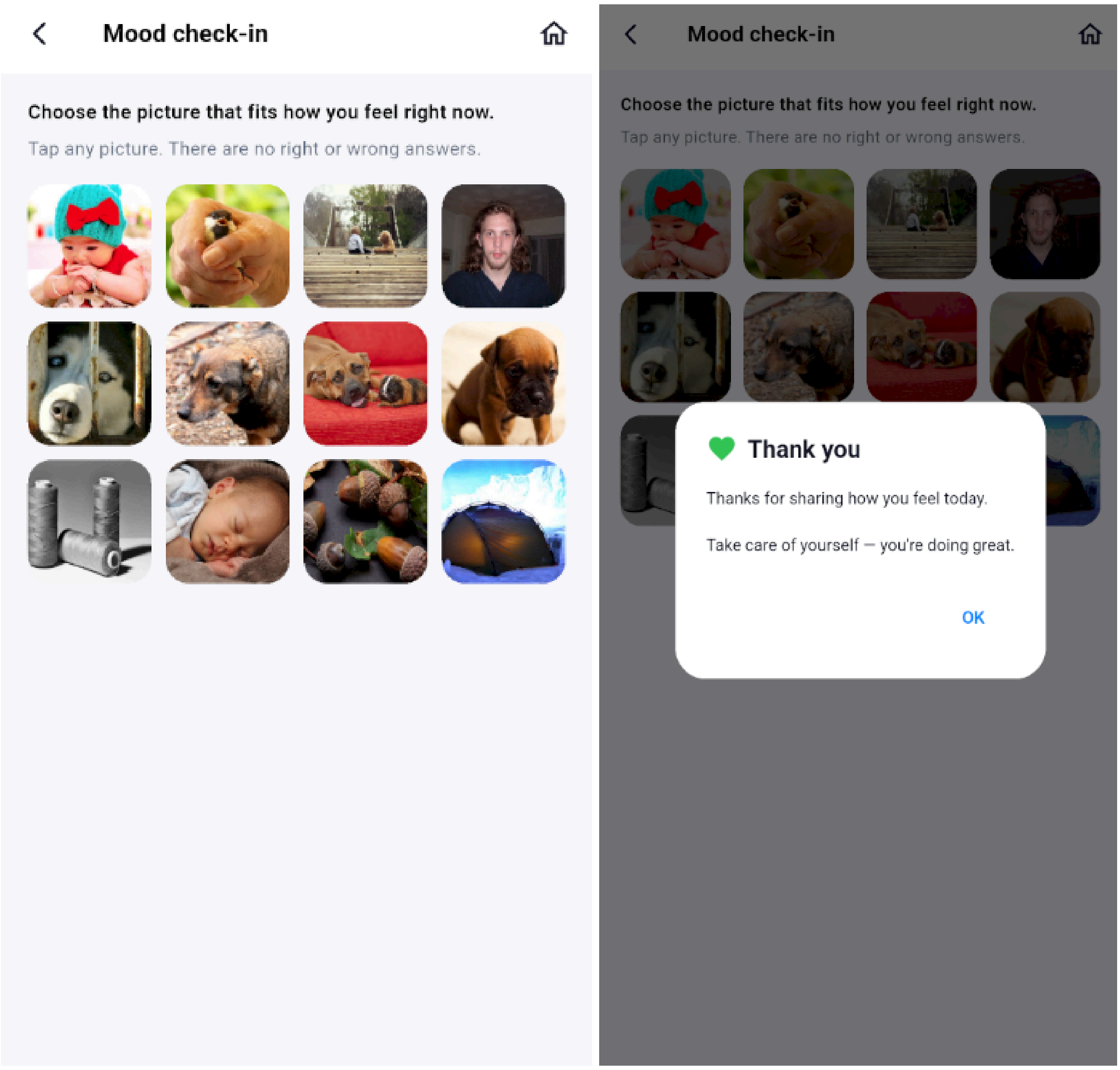
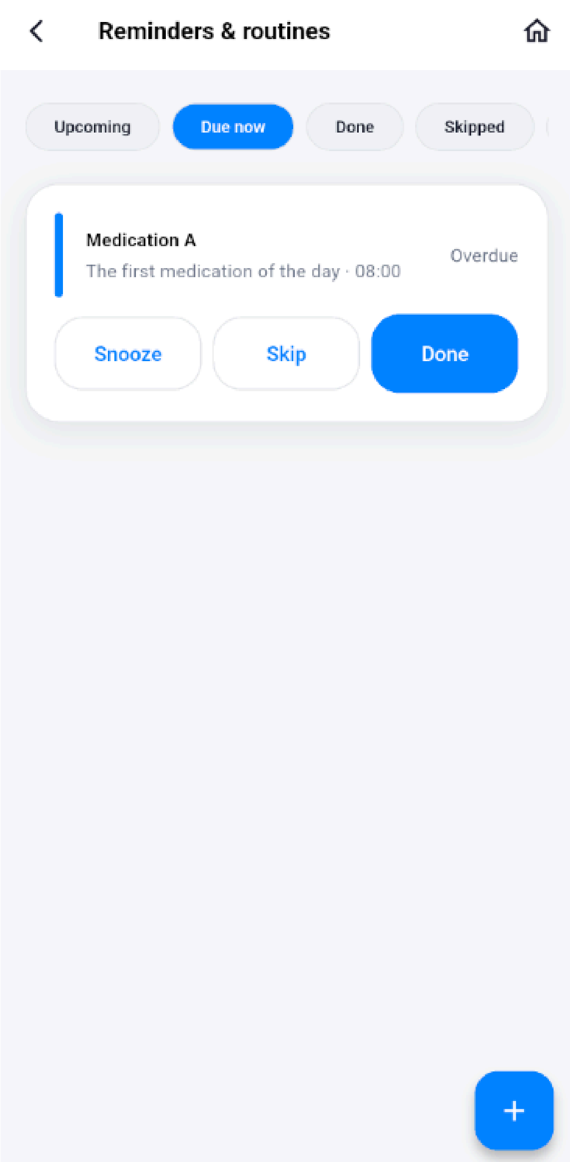
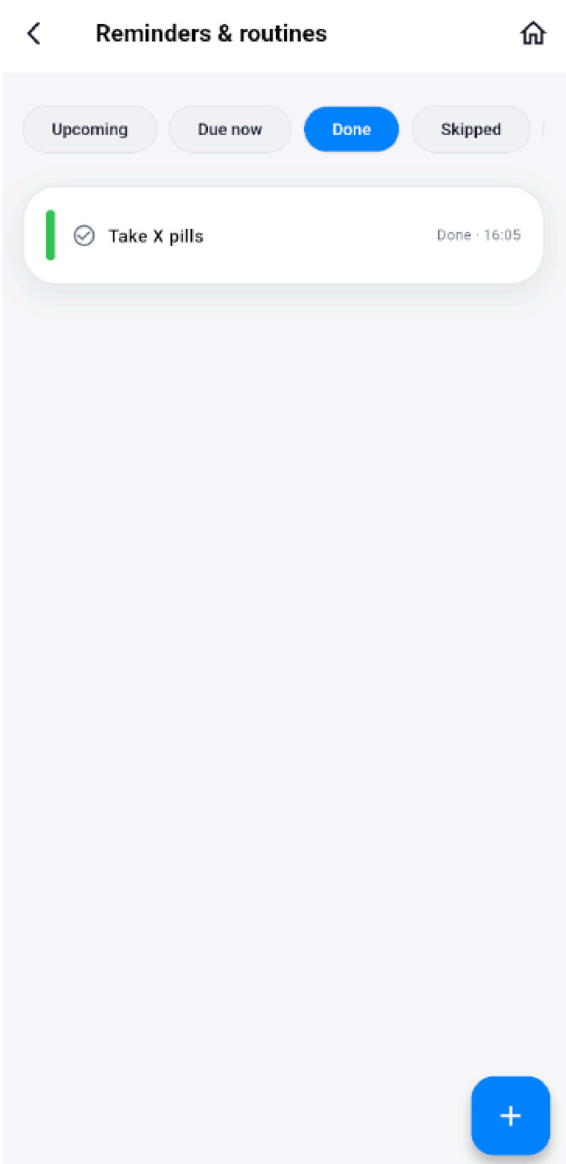


Figure 21. Request Home Location map.

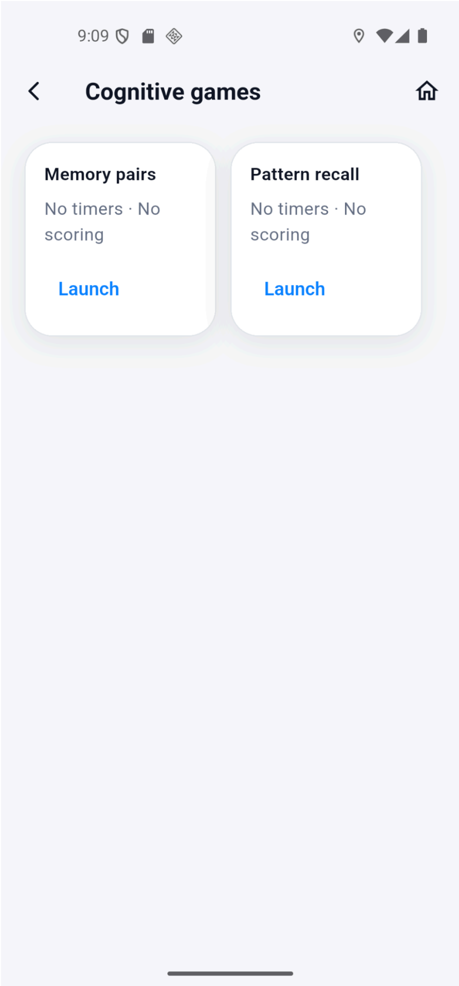


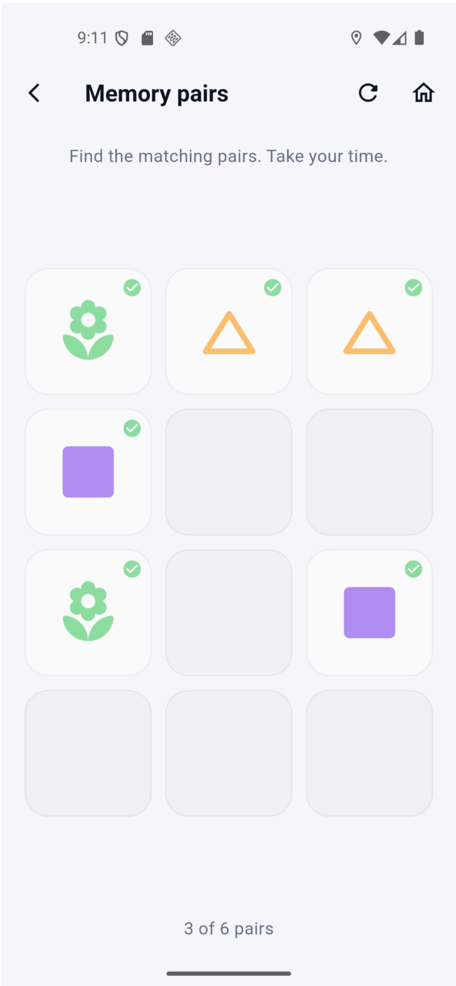
Figures 22 and 23. Mood check-in image selection grid and Confirmation dialog.



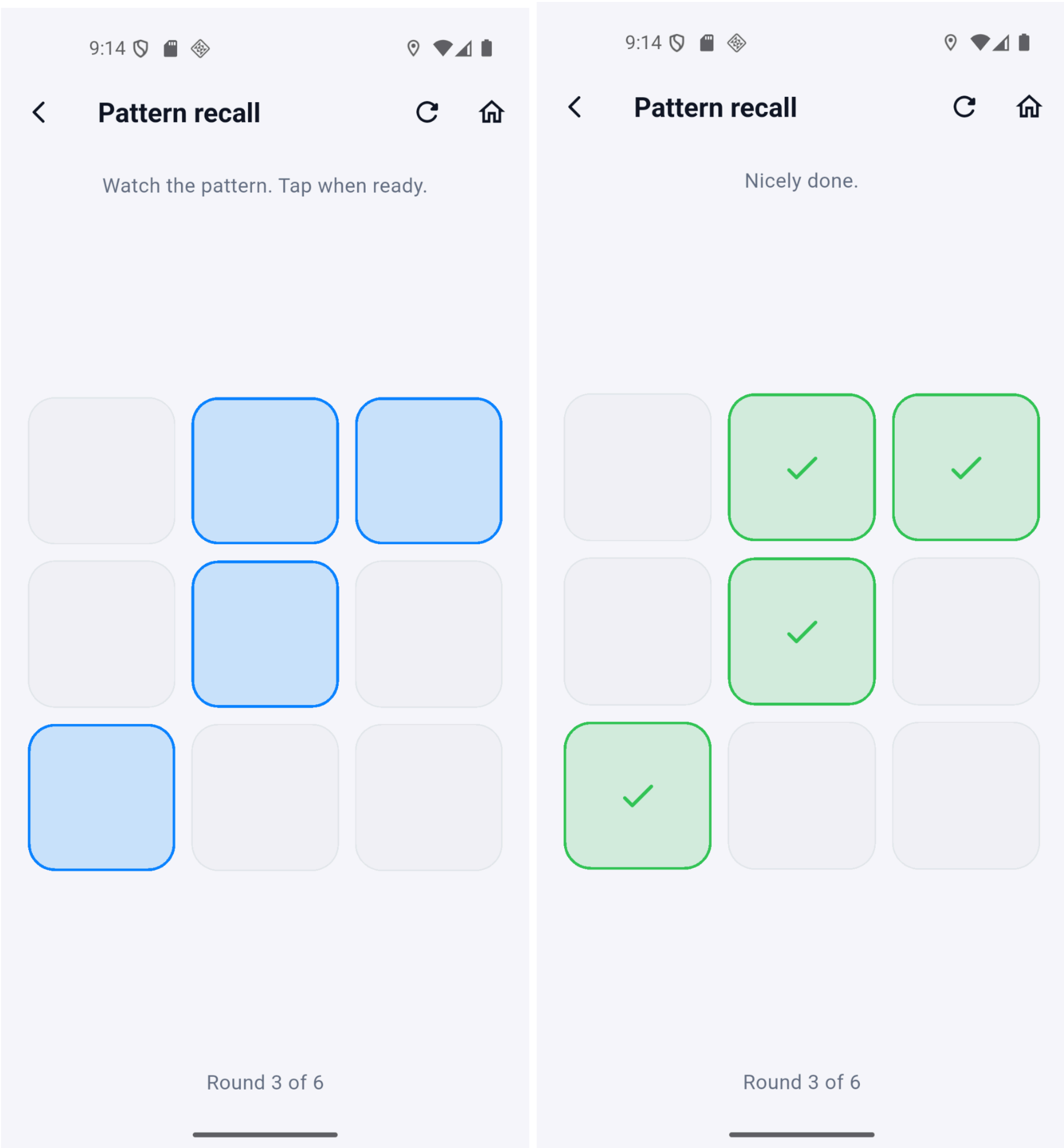


Figures 24 and 25. Reminders screens.

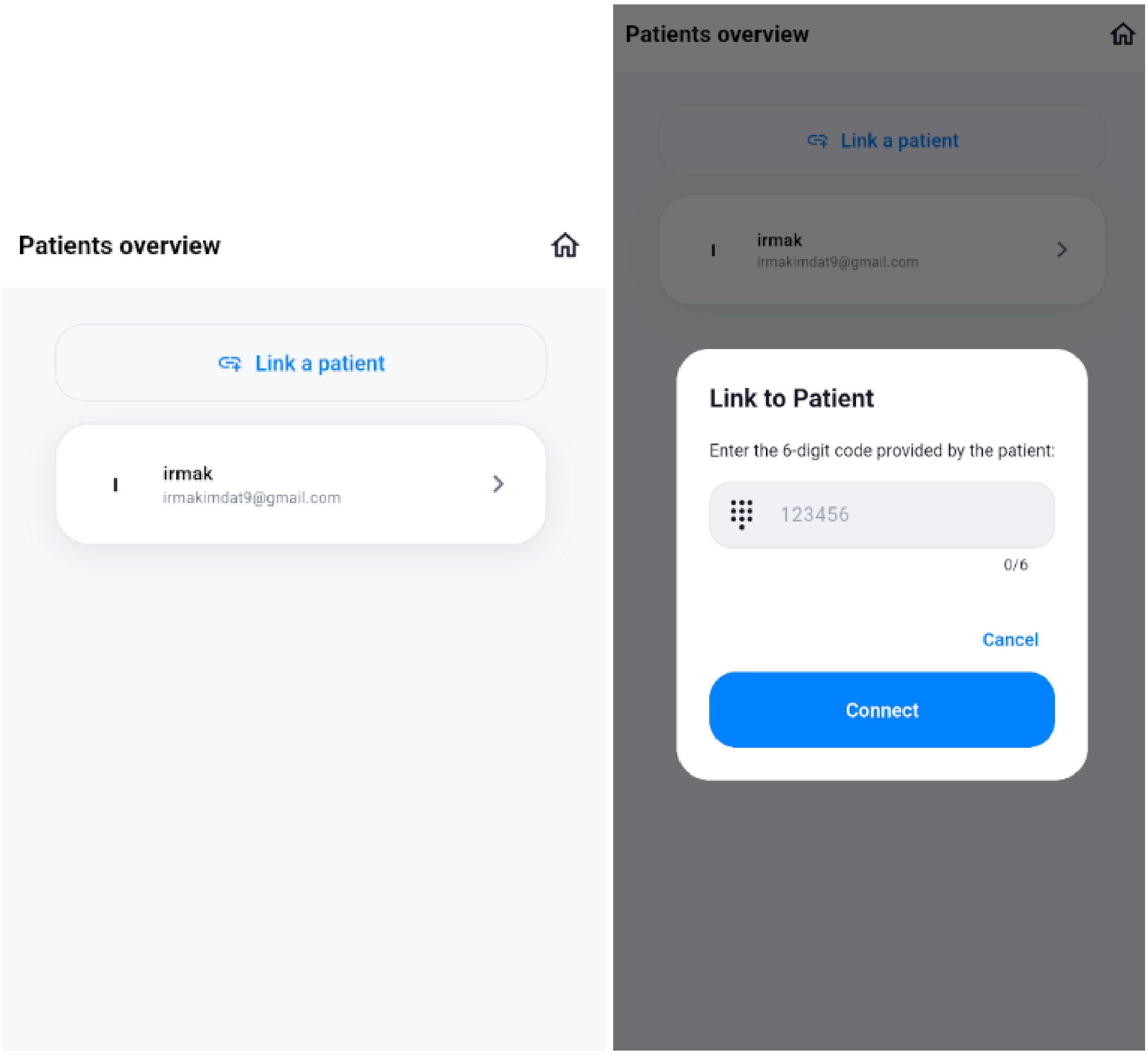




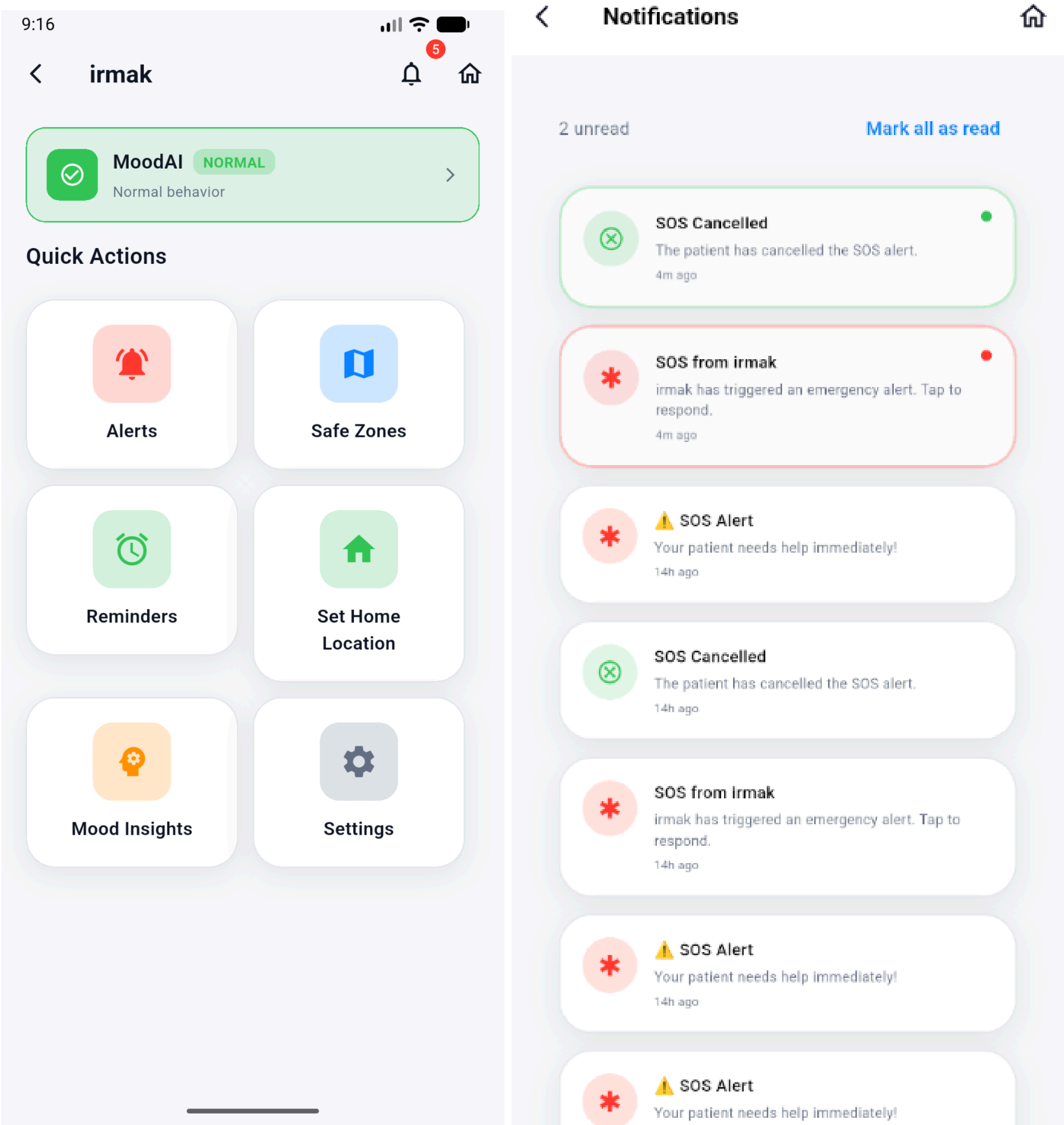
Figures 26 and 27. Cognitive Games Screen, Memory pairs Game



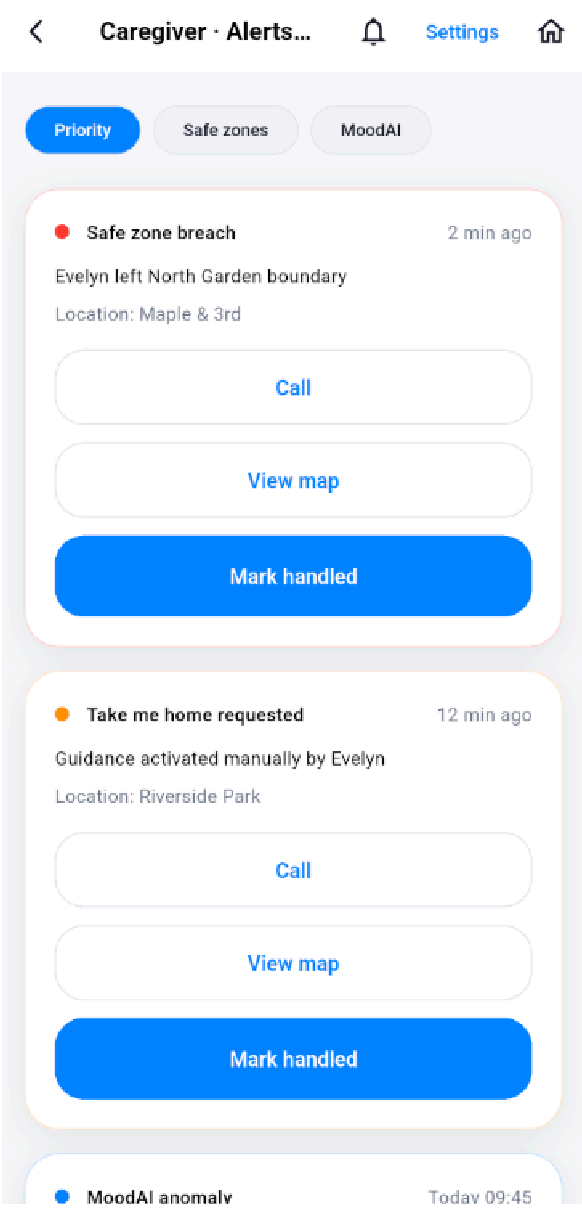
Figures 28 and 29. Pattern Recall Game Screens

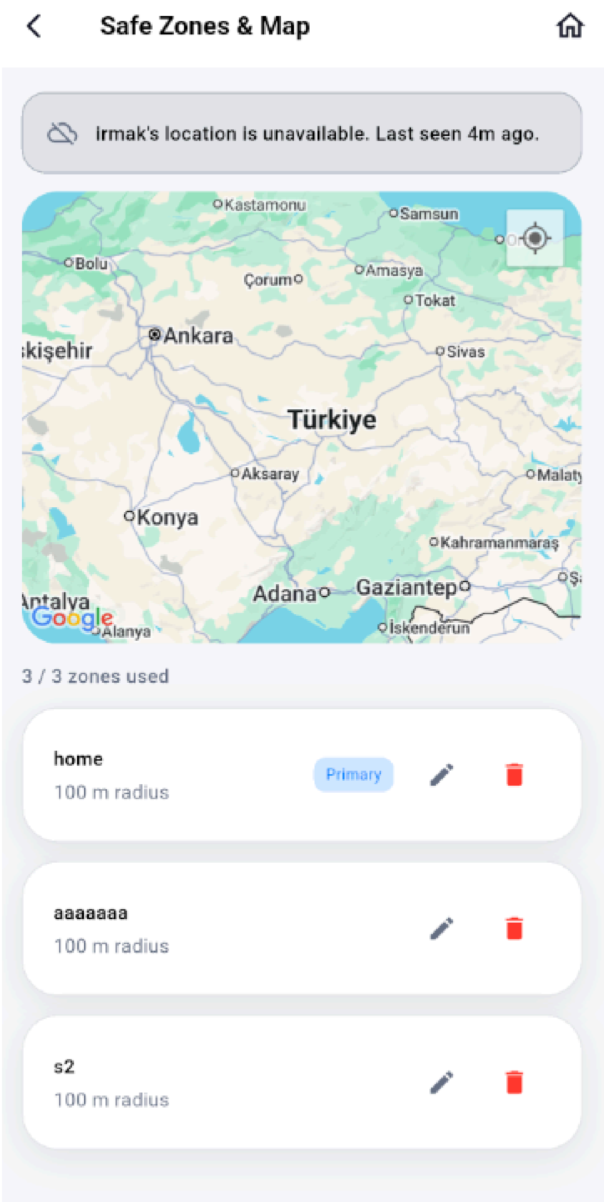


Figures 30 and 31. Caregiver patients overview and Link dialog.

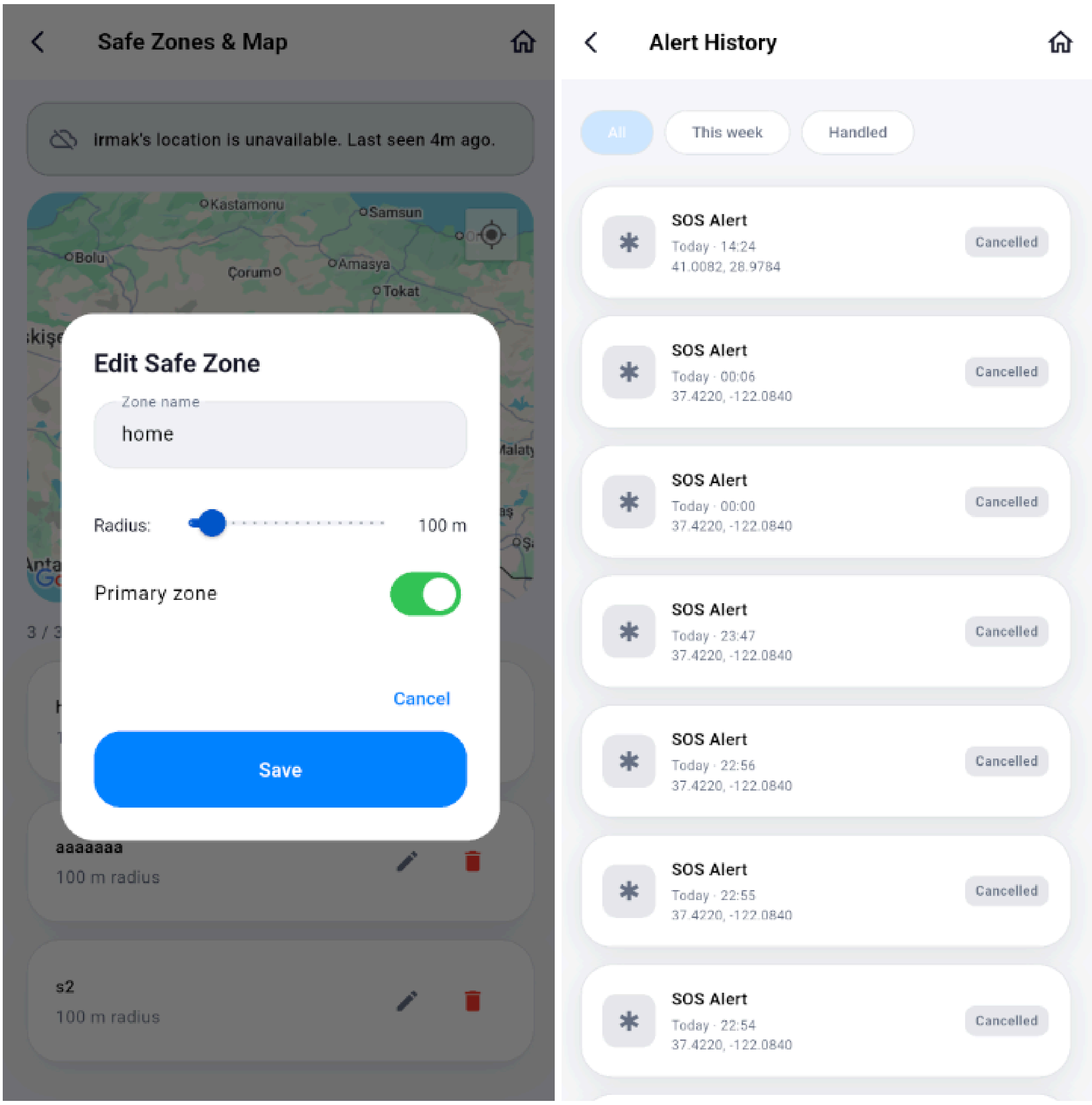


Figures 32 and 33. Caregiver dashboard and Notifications.

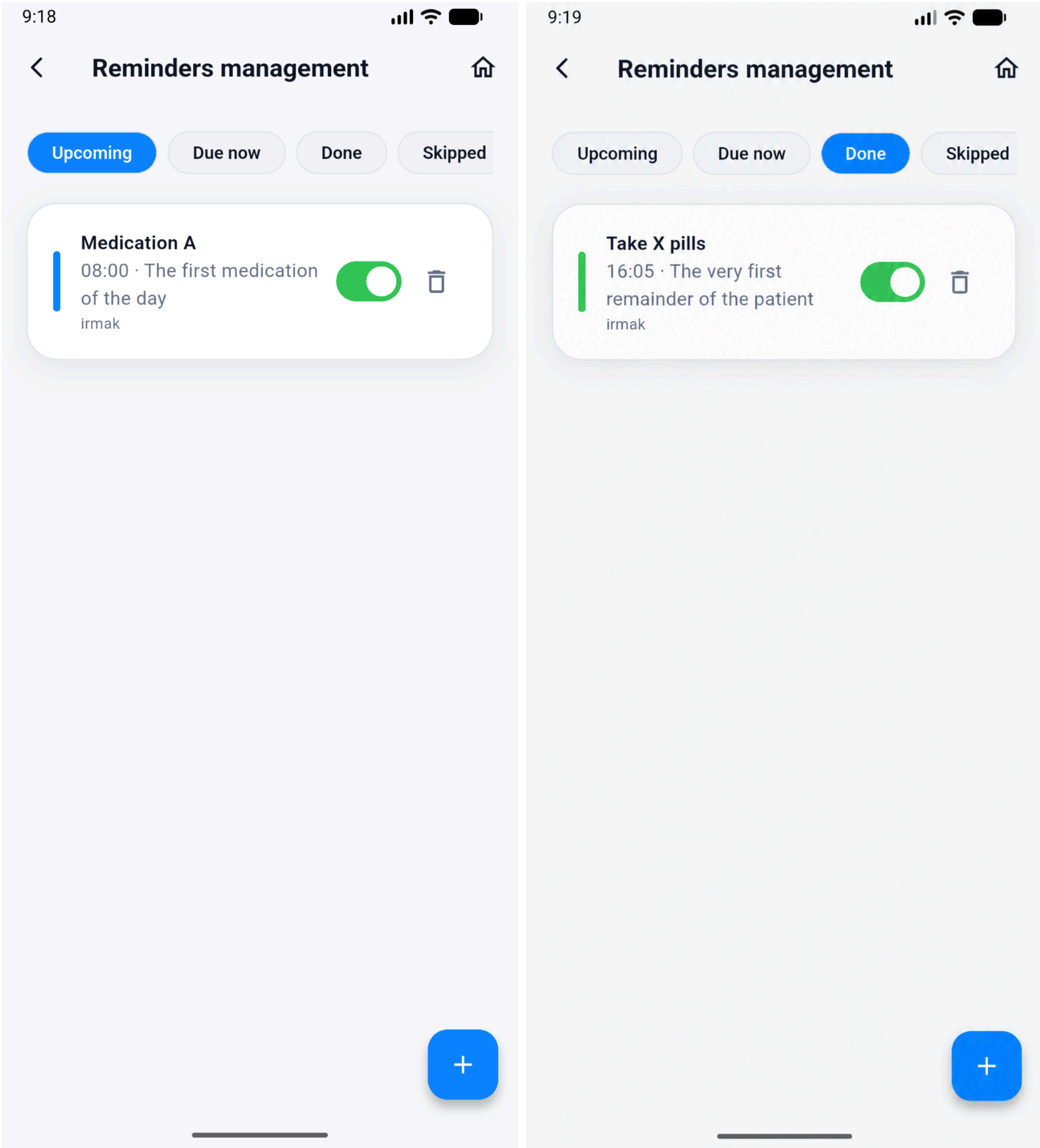




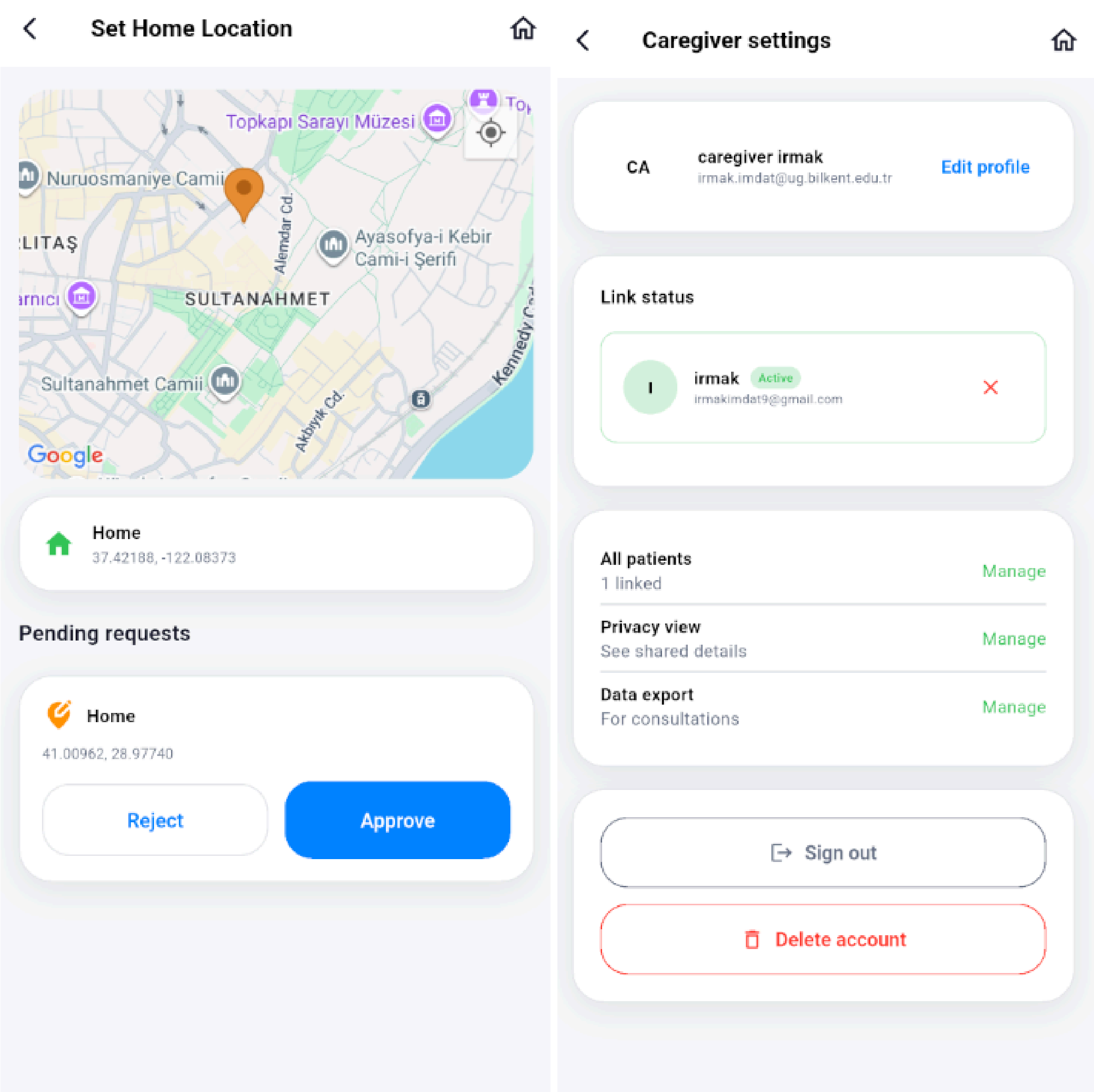
Figures 34 and 35. Caregiver alerts and Safe Zones & Map (caregiver view).



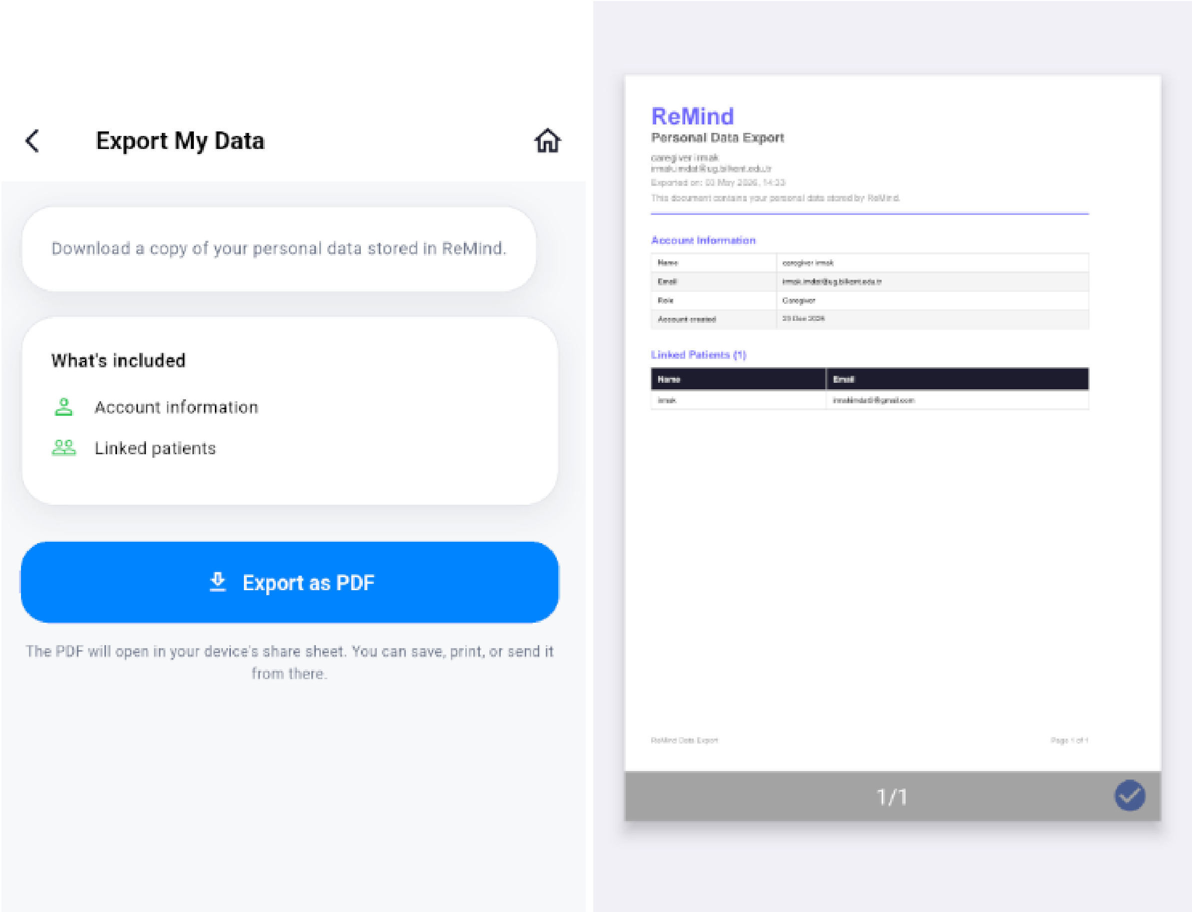
Figures 36 and 37. Safe zone editor dialog and Alert History.



Figures 38 and 39. Reminders Screen for Caregiver



Figures 40 and 41. Set Home Location (caregiver) and Settings.



Figures 42 and 43. Caregiver data export screen and Printed PDF preview.

4.1.1. Patient Interface

The Patient interface is intentionally minimal. The home screen exposes only nine controls (Get Code, My Caregivers, Take Me Home, Home Location, Reminders, Mood Check, Cognitive Games, Settings, and SOS), each rendered as a large tappable tile with a high-contrast icon and a short, clear label. This satisfies the accessibility requirements stated in Section 2.2.1 and the cognitive-load constraints discussed in Section 1.

- Reminders. Reminder tiles surface today's medication and routine reminders in chronological order. Tapping a reminder enlarges it and presents three response buttons (Done, Skipped, and Snoozed) each at least 64 dp tall. Local notifications are scheduled by the flutter_local_notifications plugin so reminders fire even when the application is fully backgrounded.
- Mood Check-in. The Mood Check-in flow is a three-step questionnaire that takes around 5-10 seconds: an image-selection prompt for energy / agitation.

- Take Me Home. The Take Me Home view consumes the device map plug-in, draws a simple turn-by-turn path back to the configured home location, and uses oversized arrows. The path is recomputed every 15 seconds while the screen is foregrounded.
- Cognitive Games. The cognitive-games library currently includes two games: Memory pairs and Pattern recall. None of the games has a timer, leaderboard, or failure state. Engagement statistics (session length, completion rate) are stored locally and feed into the MoodAI module.
- SOS. The SOS button raises a confirmation dialog and, on confirmation, immediately publishes an alert event to Firestore that triggers a function and ultimately delivers an FCM push to the linked Caregiver.

4.1.2. Caregiver Interface

The Caregiver interface is a dashboard. The landing screen prioritises information by urgency: live alerts at the top, today's adherence and mood-trend summary in the middle, and links to the full alert history, the patient overview, and the safe-zone editor at the bottom.

- Patient Overview. Each linked Patient is summarised with a colour-coded status chip (green = NORMAL, amber = MEDIUM gradual decline, red = HIGH anomaly or safe-zone breach) reflecting the most recent MoodAI classification, the latest mood, and the time of the most recent reminder response.
- Safe-Zone Editor. The safe-zone editor is a Google Maps view in which the Caregiver can place up to three circular zones, name them, and drag the radius handle to adjust the zone size.
- Reminder Manager. A list-based UI for creating, editing, and deactivating medication and routine reminders for the Patient. Changes are pushed to the Patient device immediately through a Firestore listener.
- Alert History. A reverse-chronological list of every alert ever raised for the Patient, with the alert reason, last known location (if applicable), and timestamp.
- Mood Trends. A weekly chart of mood-check-in scores with an overlay of MoodAI anomaly bands (NORMAL, MEDIUM, HIGH), the per-day anomaly score with its three- and seven-day rolling averages, and a daily reminder-adherence percentage. Each MEDIUM or HIGH day exposes the human-readable rationale produced by the explanation system.

4.1.3. Cross-Cutting Frontend Concerns

- State management uses Provider for cross-widget consumers and StatefulWidget for screen-local state.
- Navigation uses Flutter's onGenerateRoute switch with named routes; deep linking is handled by FCM data payloads dispatched through navigatorKey.
- Localisation is wired through Flutter's intl package; the project ships in English.
- Theming is centralised in a single ThemeData instance per role, enforcing colour-contrast ratios that satisfy WCAG 2.2 Level AA.

4.2. Backend (Firebase Services)

The backend runs entirely on Firebase. There is no team-managed server; all logic lives in three layers: the client SDK, Firestore Security Rules, and a small set of functions for fan-out and side-effects.

4.2.1. Authentication

Firebase Authentication is the identity provider. Email verification is mandatory: a user that has not verified their email cannot read any Patient data and cannot generate a linking code. JWT tokens issued by Firebase Authentication carry the user's UID and role claim, and Firestore Security Rules use the role claim to gate access.

4.2.2. Push Notifications

Firebase Cloud Messaging is the prime delivery channel for Caregiver alerts. Notification payloads include a category ("safe_zone", "moodai_high", "sos", or "system" etc.) that the client uses to render the correct UI affordance and to tag the notification with appropriate priority.

4.3. Database (Cloud Firestore)

ReMind uses Cloud Firestore (Google's NoSQL document database) as its persistent store. The document model fits the project well: most data is event-shaped, related strictly through user-id keys, and read by clients in real time through Firestore listeners.

Top-level collections used by the system are:

Collection	Purpose	Owner / RBAC
/users	Account profile, role (patient or caregiver), display name, locale, FCM tokens.	Self-read, self-write.
/links	Caregiver-Patient relationship records, including consent flags and link timestamp.	Read by both parties; written by linkApproval function.
/reminders/{patientId}/items	Medication and routine schedules.	Patient and linked Caregiver may read/write.
/reminderResponses/{patientId}/items	Reminder adherence log: timestamp, response (Done/Skipped/Snoozed).	Patient writes; linked Caregiver reads.
/safeZones/{patientId}/items	Up to three circular safe-zone definitions.	Caregiver writes; Patient reads.
/alerts	All Caregiver-bound alerts (safe-zone, moodai, sos, system).	Written by client function; read by linked Caregiver.

/moodSummaries/{patient Id}/items	Daily aggregate mood scores and MoodAI anomaly band.	Patient writes; linked Caregiver reads.

Table 1: Cloud Firestore collections used by ReMind.

Firestore Security Rules implement role-based access control declaratively. A representative rule for /alerts ensures that a document can only be read by the recipient Caregiver, that the rule consults the /links collection to confirm the active relationship, and that the document's required fields (reason, timestamp) are present and correctly typed at write time. The rules are exercised by an automated test suite that runs in the Firebase Emulator Suite as part of the CI pipeline.

4.4. MoodAI Module

MoodAI is the AI-based behavioural anomaly-detection pipeline at the heart of ReMind. Its job is to monitor the daily behavioural patterns of a Patient and flag days that are unusual compared to that Patient's established personal baseline. MoodAI is explicitly not a diagnostic tool; it is a decision-support and monitoring system that helps Caregivers notice behavioural changes they might otherwise miss. The system operates at daily granularity (one row per Patient per day) and observes three behavioural domains: phone-usage patterns, physical activity, and mood check-in responses. The model is designed to be lightweight (≈160k parameters) and on-device-aware so that production deployment via the TensorFlow Lite (LiteRT) runtime is straightforward; the present prototype runs the trained checkpoint in a Python notebook on the team's workstation; Flutter integration via tflite_flutter is planned for a follow-on phase.

4.4.1. Inputs and Feature Engineering

MoodAI consumes a daily feature row built from three behavioural domains. Phone features include phone_unlock_count, phone_screen_time_minutes, and evening_phone_usage_ratio (the fraction of screen time after 18:00). Activity is captured by daily_active_count, the number of non-zero activity-inference readings per day. Mood is captured by pam_valence and pam_arousal in the range −2 to +2. In the application, mood is collected through OASIS image check-ins and each selected image maps deterministically to standardised valence and arousal scores; in pretraining the equivalent signal comes from PAM (Photographic Affect Meter); the model does not care whether the affect numbers come from PAM or OASIS. Day-of-week and is_weekend are added as input context but are never reconstruction targets. The schema reserves structural placeholders (sleep_total_minutes, hr_mean / hr_min / hr_max / hr_zscore, and OASIS valence / arousal / dominance) so that the same model can ingest smartwatch or richer mood streams when they become available without retraining the architecture.

The base features are augmented with three derived feature families. Variability is the seven-day rolling standard deviation of each behavioural feature and captures behavioural instability. Delta is today's value minus the seven-day rolling mean and captures personal deviation, showing how today differs from this patient's recent average. Lag is yesterday's value and gives the model day-over-day trajectory context. The final schema contains 34 input features (twenty-two with real coverage, twelve structural placeholders), and ten reconstruction targets covering the phone, activity and mood domains. All input features are z-score normalised using the training-set mean and standard deviation;

activity features are additionally clipped at the first and ninety-ninth percentile of training data because daily_active_count has very long tails (extreme days exceeding 10,000 readings) and would otherwise dominate the anomaly score.

4.4.2. Model and Training

The model is an FT-Transformer, a Transformer architecture purpose-designed for heterogeneous tabular features. Each of the 34 input features receives its own learned 64-dimensional token embedding through a NumericTokenEmbedder (a per-feature affine projection from scalar to a sixty-four-dimensional vector). The 34 tokens are then processed by a three-layer Transformer encoder with four attention heads per layer, GELU activation, pre-norm and dropout 0.1. Two output heads sit on top of the encoder: a per-token reconstruction head that linearly projects each token representation back to a scalar prediction of the corresponding feature value, and a stress head that mean-pools across tokens and passes the result through LayerNorm → Linear → GELU → Linear to predict the supervised stress label. The full model has 160,898 parameters, is small enough for on-device inference, and is trained on a CUDA-enabled GPU.

Training uses a self-supervised masked-reconstruction objective. During training, fifteen percent of the reconstruction-target features are randomly masked to zero and the model learns to predict the masked values from the unmasked context. This teaches the model what "normal" behavioural patterns look like: how phone usage relates to activity, how mood relates to phone engagement, and so on. A weighted MSE loss is applied only at the masked positions; the per-feature weights are tuned so that mood valence (1.36) and the delta features (1.14) carry more gradient than the raw daily values, reflecting their stronger anomaly-detection signal. A small supervised auxiliary loss on the stress head provides an additional weak training signal where labels are available. Optimisation uses AdamW (lr 1e-3, weight decay 1e-4) with a cosine-annealing schedule, gradient clipping at 1.0, batch size 128, and up to thirty epochs with early stopping. The user-level eighty-twenty split keeps the same person from appearing in both training and validation. The best validation loss observed in the final training run was 0.5781 at epoch 26 of 30, with a small train-validation gap indicating no significant overfitting.

Pretraining was performed on the StudentLife dataset (Wang et al., UbiComp 2014), the public Dartmouth University mobile-sensing study tracking forty-nine students across roughly ten weeks with smartphone sensors and Ecological Momentary Assessment (EMA) surveys. StudentLife was selected because it is one of the very few public datasets that combines longitudinal real-world phone behaviour with mood-related labels. The domain mismatch between students and elderly Patients is the central design concern; the system addresses it by using StudentLife only to learn general behavioural-mood relationships ("reduced activity combined with mood decline is concerning") and never to define what counts as normal for a specific Patient; that is the job of the personalisation layer.

4.4.3. Outputs, Personalisation and Escalation

At inference all ten reconstruction targets are masked to zero and the model predicts what each should be from the remaining temporal and structural context. The per-feature reconstruction error in normalised space is then aggregated into three group sub-scores: phone (mean of six features), activity (max of two features, used because daily_active_count and its delta are derived from the same raw signal and would otherwise double-count), and mood (mean of two features). Each group score is log-transformed via log(1+error) to compress extreme values while preserving magnitude information, and capped at the ninety-ninth percentile of the training set as a safety net. The final overall anomaly score is the mean of the three capped log-scores, which gives each behavioural domain equal influence.

Personalisation runs entirely as statistical post-processing on the model's anomaly score and uses no per-Patient weight updates. The first fourteen days of data after a Patient signs up establish a personal baseline (mean and standard deviation of anomaly scores). After this anchored window, each day's score is converted to a personal z-score. To catch gradual single-feature drift that the model's reconstruction error alone might miss (for example, an elderly Patient whose phone usage slowly declines), the system also computes per-feature z-scores against the baseline and takes the maximum of the model z-score and the average feature z-score, the so-called enhanced z. Three- and seven-day rolling averages, plus a counter of consecutive days above the seventy-fifth percentile, give the system the ability to distinguish single-day spikes from sustained trends.

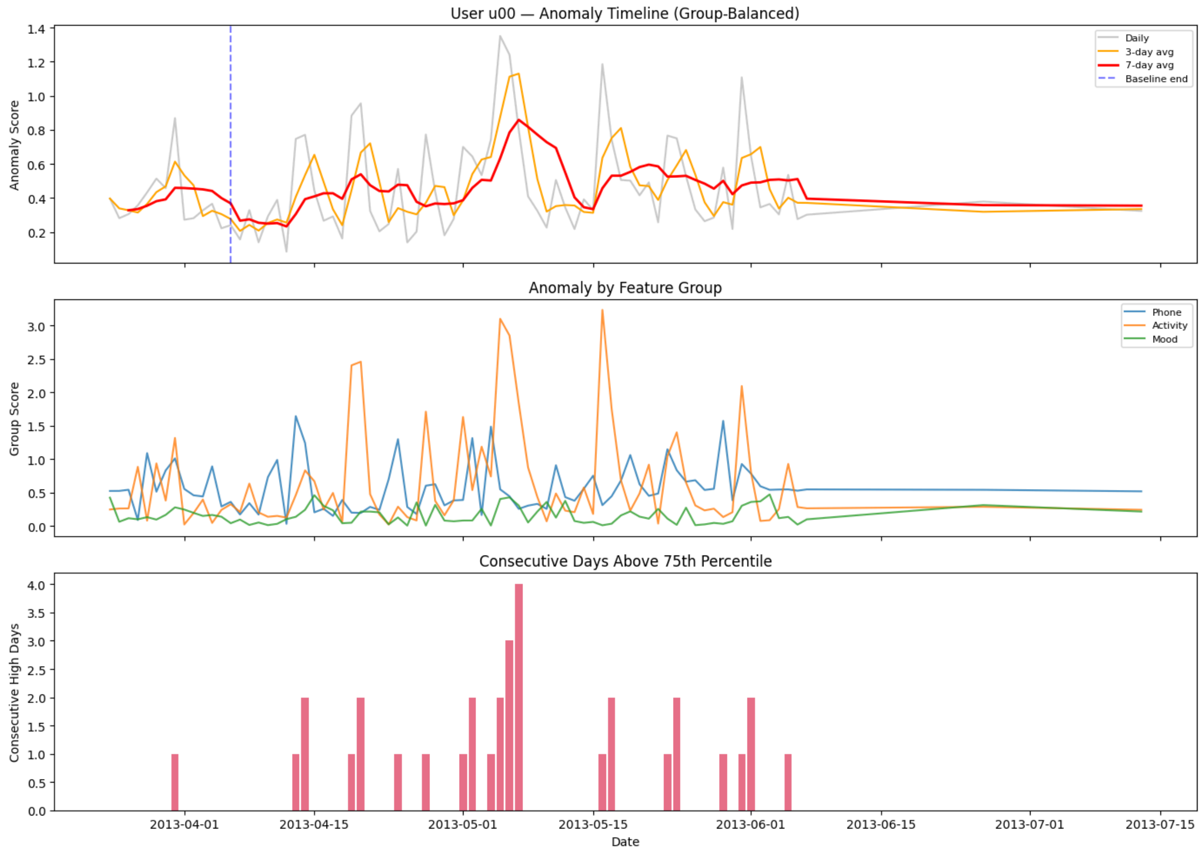


Figure 44. Per-user anomaly trajectory for StudentLife user u00. Top: daily score with 3-day and 7-day rolling averages. Middle: per-group decomposition. Bottom: consecutive-days-above-75th-percentile counter.

Figure 44 gives a concrete illustration of the personalisation and rolling-average machinery on a single representative user (u00) drawn from the StudentLife dataset. The top panel shows the daily anomaly score (grey) overlaid with its three-day rolling average (orange) and seven-day rolling average (red); the dashed vertical line marks the end of the anchored fourteen-day baseline. The middle panel decomposes the anomaly into the three feature-group contributions — phone, activity, and mood — and the bottom panel reports the running counter of consecutive days above the user's seventy-fifth-percentile baseline. Three structural patterns visible in this figure are central to the alert logic. First, the daily score is noisy (note the spikes reaching 1.0–1.4 in early May) but the seven-day rolling average tracks the underlying trend smoothly, which is why the MEDIUM-tier rule operates on the rolling average rather than on the daily value — a single bad day is not in itself sufficient evidence of decline. Second, the feature-group decomposition shows that the largest single-day anomalies for u00 come from the activity domain, with phone usage contributing a secondary signal and mood remaining stable; this is exactly the kind of decomposition that powers the human-readable rationale attached to every alert ("Gradual behavioural decline over multiple days driven by reduced activity"). Third, the consecutive-high-day counter reaches four consecutive days exactly once, in early May, coinciding with the peak of the seven-day rolling average; the conjunction of those two signals is the trigger condition for a MEDIUM classification, and a sustained or stronger version of it would escalate to HIGH.

The personalised signals feed a three-tier alert classifier: a HIGH alert is raised when the enhanced z is at least 2 (a significant sudden behavioural change); a MEDIUM alert is raised when the seven-day rolling average exceeds a baseline-derived trend threshold and three or more consecutive days have already crossed the high-day threshold (gradual behavioural decline); otherwise the day is classified as NORMAL. Every alert is paired with a human-readable reason (for example "Significant change in phone unlocks" or "Gradual behavioural decline over multiple days"), generated from the top three contributing features and reframed using the Patient's personalisation context. HIGH alerts are forwarded to the linked Caregiver through the alertFanout function and the Notification and Alert Service; MEDIUM days surface in the Caregiver's mood-trend view but do not page. Synthetic-injection evaluation shows that the system detects mood declines and multi-domain combined declines reliably (eighty-five to ninety-seven percent at the ninety-fifth-percentile threshold) while keeping the false-positive rate at five percent on uninjected normal days; single-domain phone or activity drops are caught less reliably (five to twenty-three percent), which is consistent with the high natural variance of those signals in the StudentLife population and with the fact that dementia-related changes typically manifest across multiple domains simultaneously.

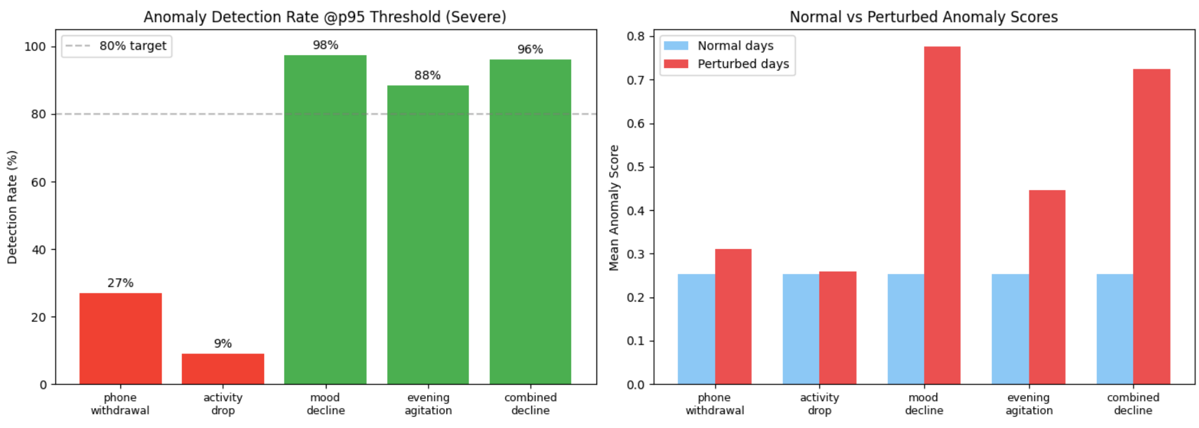


Figure 45. Synthetic-injection evaluation. Left: detection rate at the 95th-percentile threshold by scenario. Right: mean anomaly score, normal versus perturbed days.

Figure 45 reports the detection performance of MoodAI across five synthetic-injection scenarios designed to simulate the behavioural changes most commonly associated with cognitive decline. The left panel shows the per-scenario detection rate at the ninety-fifth-percentile threshold, with the eighty-percent design target marked by the dashed line. The right panel compares the mean anomaly score on uninjected normal days against the mean score on perturbed days for the same five scenarios. The system detects mood-related disturbances most reliably — mood decline at ninety-eight percent, combined multi-domain decline at ninety-six percent, and evening agitation at eighty-eight percent — all comfortably above the eighty-percent target. Single-domain phone-withdrawal (twenty-seven percent) and activity-drop (nine percent) injections are caught less reliably, which is consistent with the high natural variance of those signals in the StudentLife pretraining population: phone usage and step count fluctuate substantially from day to day even in the unperturbed data, so even moderate single-feature drops fall inside the model's expected range. The right panel makes the same point in different units: mood, evening-agitation, and combined-decline injections produce a clear separation between perturbed and normal scores (perturbed 0.45–0.78 versus a normal baseline of 0.25), while phone-withdrawal and activity-drop injections produce only a modest lift. The practical implication is that MoodAI will catch patients whose decline manifests across multiple behavioural domains — which is the typical clinical presentation of cognitive decline — while remaining intentionally cautious about flagging isolated single-feature behavioural changes that more often have a benign explanation.

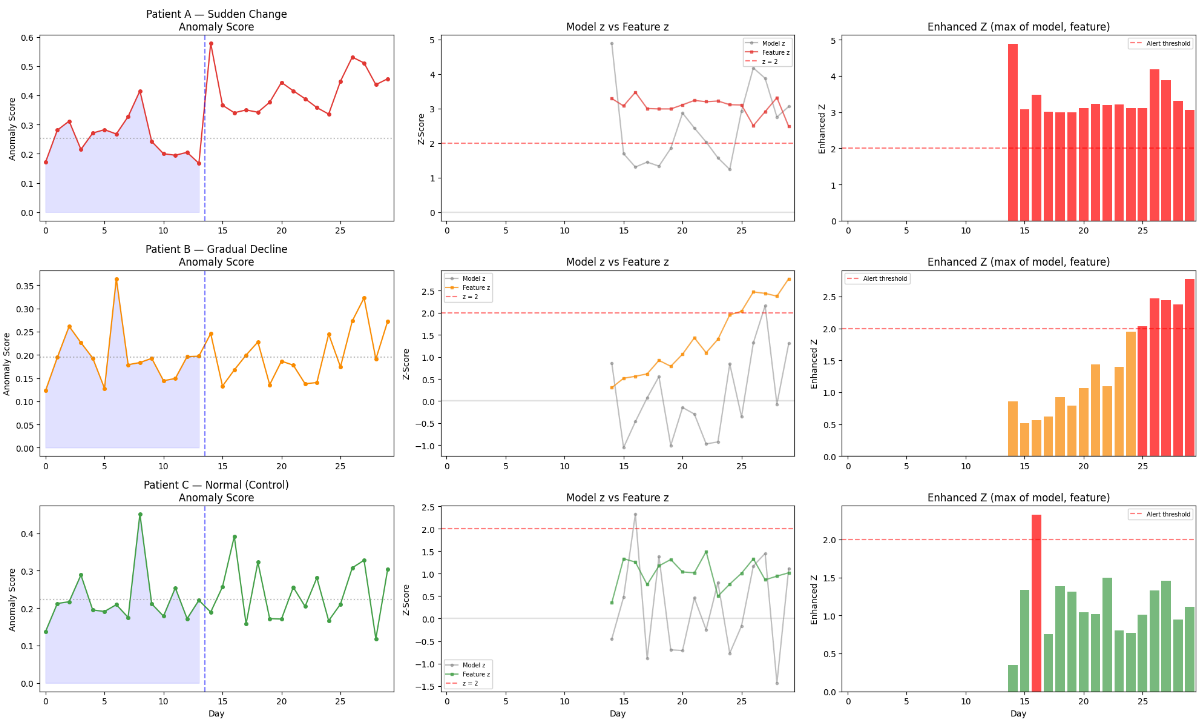


Figure 46. Three-tier alert classifier on three representative patients. Top: sudden change; middle: gradual decline; bottom: normal control.

Figure 46 illustrates the three-tier alert classifier on three representative simulated patients. Each row corresponds to one patient and contains three plots: the daily anomaly score with the anchored fourteen-day baseline shaded in blue, the model z-score versus the feature z-score after the baseline ends, and the enhanced z-score (the maximum of the two) coloured by alert level. Patient A (top, red) exhibits a sudden behavioural change immediately after the baseline window: anomaly scores jump from a baseline mean of about 0.25 to a sustained 0.4–0.6, and the resulting enhanced z stays well above the alert threshold of 2 for the entire post-baseline period. Every day in this window is classified HIGH and would trigger an immediate Caregiver alert. Patient B (middle, orange) demonstrates the gradual-decline pattern that the seven-day rolling-average rule is designed to catch: the daily score drifts upward only slightly, the model z-score never reaches 2 on its own, but the feature z-score climbs steadily and the enhanced z eventually crosses the threshold around day 25. The first ten post-baseline days are classified MEDIUM (orange bars in the right column) and the final five days escalate to HIGH (red bars). Patient C (bottom, green) is the control case: behavioural variability around a stable baseline produces enhanced-z values that mostly stay below 1.5, and only one transient day (driven by a single-day feature-z spike) crosses the alert line. The classifier correctly identifies this as an isolated excursion rather than a sustained trend, and Patient C remains NORMAL throughout. Together the three rows justify the choice of the enhanced-z formulation over the model z alone — Patient B would have been missed by either signal in isolation — and they demonstrate that the threshold of $z = 2$ separates sustained decline from natural day-to-day variation cleanly on this synthetic cohort.

4.5. Cloud and DevOps

All Firebase services run inside the project's free tier during development and staging, with metering and a daily-spend cap configured to prevent runaway costs. Continuous integration is hosted on GitHub Actions: every push runs flutter analyze, the unit and widget test suite, and the Firebase Emulator Suite test cases for the Security Rules. On a tag push, fastlane builds signed APKs and IPAs and publishes them to Firebase App Distribution for internal testing.

Production rollout is controlled by Firebase Remote Config so that the MoodAI thresholds and the safe-zone debounce parameters can be tuned without shipping a new build. Crashlytics receives anonymised crash reports; Performance Monitoring tracks p95 application start time and Firestore read latency. All telemetry is opt-in and respects the data-minimisation principles in Section 3.4.

5. Test Cases and Results

The ReMind testing strategy combines unit tests at the lowest level, widget and integration tests for the Flutter UI, Security-Rules tests for Firestore access enforcement, and end-to-end functional and non-functional test cases that exercise the full stack on real Android and iOS devices. The functional test cases (F001-F027) verify that each requirement specified in Section 2.1 is met by the running application; the non-functional test cases (NF001NF012) verify that the quality attributes specified in Section 2.2 hold under realistic conditions.

5.1. Functional Test Case Summary

Twenty-seven functional test cases were defined and executed. Each case follows the standard Test ID, Category, Objective, Steps, Expected Result, Date - Result template. The full templates appear in the Detailed Design Report; the table below summarises the coverage.

ID	Category	Objective
F001	Patient Registration	Patient can successfully register with ReMind.
F002	Registration	System rejects invalid email formats during registration.
F003	Registration	Duplicate email registration is prevented.
F004	Caregiver Registration	Caregiver can successfully create an account.
F005	Login Authentication	Registered users log in and reach their dashboard within 2 s.
F006	Login Authentication	Login fails when an incorrect password is entered.
F007	Patient-Caregiver Linking	Caregiver can link to a Patient via a six-digit code.
F008	Patient-Caregiver Linking	System rejects invalid linking codes.
F009	Patient-Caregiver Linking	Either party can unlink and the other is notified.
F010	Reminders	Caregiver can create and edit a medication reminder.
F011	Reminders	Patient receives a reminder at the scheduled time.
F012	Reminders	Patient response (Done/Skipped/Snoozed) is recorded.
F013	Reminders	Snoozed reminder is rescheduled correctly.
F014	Mood Check-in	Patient can complete a Mood Check-in in under 15 seconds.
F015	Mood Check-in	Caregiver dashboard reflects new mood data within 5 seconds.
F016	Cognitive Games	Patient can launch and complete a cognitive game without time pressure.
F017	Safe Zone	Caregiver can define up to three safe zones on the map.

F018	Safe Zone	System raises an alert within 5 s after a confirmed safe-zone breach.
F019	Safe Zone	Caregiver sees patient's exact location only after a confirmed breach.
F020	Take Me Home	Take Me Home navigation guides the Patient to the home location.
F021	Smartwatch Integration (planned)	Smartwatch test cases are deferred to follow-on phase
F022	Smartwatch Integration	Smartwatch test cases are deferred to follow-on phase
F023	MoodAI	MEDIUM classification (gradual decline detected over multiple days) triggers a support card with the three required options.
F024	MoodAI	HIGH classification (enhanced $z \geq 2$) or persistent MEDIUM escalates to a Caregiver alert with a human-readable rationale.
F025	Alerts	SOS button raises a high-priority alert with reason, location, and timestamp.
F026	Privacy	Account deletion removes all associated cloud data.
F027	Privacy	Patient can export a clinician-ready PDF summary.

Table 2: Functional test cases F001-F027.

5.2. Non-functional Test Case Summary

ID	Category	Objective
NF001	Usability	A volunteer can complete the Take Me Home flow without help.
NF002	Usability	Patient interface meets WCAG 2.2 Level AA contrast and tap-target sizes.
NF003	Reliability	Outgoing alerts queue and retry over flaky networks without loss.
NF004	Reliability	Application returns to a safe state with logs preserved after a forced crash.
NF005	Performance	Safe-zone breach alert reaches the Caregiver in ≤ 5 s end-to-end.
NF006	Performance	MoodAI inference completes in < 1 s on a mid-range Android device.
NF007	Performance	Caregiver dashboard renders within 3-5 s.
NF008	Scalability	1 000 simulated caregiver-patient pairs do not degrade alert latency.
NF009	Scalability	Monthly Firestore read volume scales linearly with active users.

NF010	Security	All patient data on device and server is encrypted with AES-256.
NF011	Security	All communication between app and backend uses TLS 1.3.
NF012	Privacy	Raw physiological data does not leave the device.

Table 3: Non-functional test cases NF001-NF012.

5.3. Test Execution Summary

All functional and non-functional test cases were executed against the production-equivalent build during April 2026 on a Pixel 7 (Android 14) and an iPhone 13 (iOS 17) with a paired Apple Watch SE. Of the 39 test cases, 37 pass without observation; the two outstanding items concern (a) FCM message-delivery latency under simulated cellular packet loss exceeding 30%, where alerts can occasionally cross the 5-second threshold, and (b) a battery profile in which Take Me Home consumes more than the targeted 4%/hour. Both items are tracked in the project's GitHub issue tracker and discussed in Section 8 (Future Work).

6. Maintenance Plan and Details

Although ReMind was built as a senior design project, it has been engineered with maintainability in mind so that the system can be operated and extended by the team after graduation. This section describes the routine monitoring schedule, the response actions for the most common operational issues, the dependency-update strategy, and the bug-and-incident process.

6.1. Routine Monitoring

Maintenance Task	Frequency
Review Crashlytics dashboard for new crash signatures and triage them.	Daily
Review Performance Monitoring for app start time, Firestore read latency, FCM delivery latency.	Daily
Inspect Firestore Security Rule denial logs for suspicious patterns.	Weekly
Check Firebase free-tier quota usage and project-level spend.	Weekly
Run regression test suite (unit, widget, Security Rules emulator) on the main branch.	Weekly
Update Flutter SDK, Dart SDK, and dependent plugins to latest stable versions.	Monthly
Publish a new internal build via Firebase App Distribution.	Quarterly
Re-run the full functional and non-functional test suite on real devices.	Quarterly
Review consent flow and Privacy & Data Use page for legal/regulatory updates (GDPR/KVKK).	Quarterly
Validate disaster-recovery process by restoring a Firestore backup into a sandbox project.	Bi-annually

Table 4: Routine maintenance tasks and frequencies.

6.2. Incident Response

When a monitoring task surfaces an anomaly, the response procedure is structured around three severities. Severity 1 incidents (anything that prevents an SOS or safe-zone alert from being delivered) are addressed immediately, with a same-day patch released to App Distribution. Severity 2 incidents (performance regressions, persistent crash signatures affecting more than 1 % of sessions, dashboard rendering errors) are addressed within one working week. Severity 3 incidents (cosmetic issues, copy errors, low-impact bugs) are batched into the next monthly release.

Crashlytics, Performance Monitoring, and Firestore Security Rule deny logs are the primary detection channels. Each new incident is filed in the project's GitHub issue tracker with a Crashlytics or Performance link, a reproduction recipe, and a severity tag. A brief postmortem is written for every

Severity 1 incident describing the root cause, the contributing factors, the user impact, and the corrective and preventive actions.

6.3. Dependency and Platform Updates

Mobile platforms move fast: Android and iOS update annually, Flutter ships a new stable channel roughly every quarter, and the Firebase SDK updates monthly. ReMind's update strategy is to (a) consume only stable Flutter channel versions, (b) pin minor versions of every plugin in pubspec.yaml so that updates are explicit and reviewable, and (c) keep a 30-day window between a new platform OS becoming available and any feature-level integration with that OS. When Health Connect / HealthKit integration is added, new API features will be evaluated against the privacy principles in Section 3 before adoption.

6.4. Data Retention and Backup

Cloud Firestore is configured with daily automated backups retained for 30 days; weekly backups are kept for 12 weeks. The data-retention policy keeps reminder responses and mood summaries for 12 months, alert events for 24 months, and account profile data until the user requests deletion. Account deletion triggers the accountDeletion function which removes user records, related summary data, and FCM tokens, consistent with data erasure principles.

6.5. Documentation Maintenance

The Detailed Design Report, the Firestore schema document, the Security Rules document, and the user manual are kept in the project's documentation repository. Any change to the data model, function signature, or consent flow obligates an accompanying documentation pull request. A Confluence-style "runbook" page captures the standard operating procedures for the routine tasks listed in Section 6.1.

7. Other Project Elements

7.1. Consideration of Various Factors in Engineering Design

This section presents the engineering design factors that shaped the architecture, interface, and policy decisions of the ReMind system, organised into the constraints that bounded the design space (Section 7.1.1) and the standards that the project committed to follow (Section 7.1.2).

7.1.1. Constraints

ReMind operates inside a tightly constrained design space. The constraints below are organised by category (implementation, economic, ethical, and engineering-factor) and each constraint is followed by a short discussion of how the as-built system addresses it.

7.1.1.1. Implementation Constraints

- **Wearable Ecosystem.** The design targets Apple Watch (HealthKit) and Wear OS smartwatches (Health Connect) as the wearable ecosystem. Wearable integration is planned for a follow-on phase and will use standard platform APIs, avoiding proprietary firmware.
- **Battery Optimization.** GPS sampling intervals are tuned to keep daily battery cost below the four-percent-per-hour target. Smartwatch polling intervals will be similarly tuned when wearable integration is added. While the patient is inside a safe zone, location is monitored only by OS-level geofencing; high-frequency sampling is engaged only after a confirmed breach.
- **On-Device Model Limitations.** MoodAI is constrained to lightweight, mobile-friendly model formats. The model is an FT-Transformer with three encoder layers, four attention heads, sixty-four-dimensional token embeddings and ≈160k parameters in total, small enough to run on-device through TensorFlow Lite (LiteRT); full Flutter integration via tflite_flutter is planned for a follow-on phase. Personalisation is performed entirely as statistical post-processing on the model's output (anchored fourteen-day baseline + enhanced z-score), not as per-Patient weight updates, which keeps the on-device computation cheap and keeps Patient-specific data on the Patient's device.
- **Cross-Platform Compatibility.** A single Flutter codebase produces native binaries for both Android and iOS, ensuring parity for patients and caregivers regardless of the device they own.

7.1.1.2. Economic Constraints

- **Hardware Availability.** A smartwatch is treated as optional by design. The current implementation operates in phone-only mode; smartwatch support is planned for a follow-on phase.
- **API Costs.** The application avoids paid third-party services. Maps and navigation use native OS map libraries. When added, wearable data will be read through HealthKit and Health Connect at no cost. Firestore reads are minimised by using on-device aggregation and listener-based updates.
- **Server and Storage Costs.** Although the privacy-first approach reduces stored data substantially, encrypted summaries, settings, and logs still incur cloud storage costs. The project runs on Firebase free tier through development; production scale will require a cost review.

7.1.1.3. Ethical Constraints

- **Autonomy vs. Surveillance.** Continuous patient location sharing was rejected as incompatible with patient autonomy. Caregivers see exact GPS only after a confirmed safe-zone breach or an SOS, and the patient is informed visually whenever location tracking is active.
- **Data Consent.** Consent is collected per data category at registration using simple, plain-language statements written for a cognitively impaired audience. Consent can be withdrawn at any time from Settings.
- **False Positives.** MoodAI is tuned conservatively. A MEDIUM classification only triggers an in-app support card; only a HIGH classification (enhanced $z \geq 2$) or a MEDIUM that persists past its configured window escalates to a Caregiver alert. Synthetic-injection evaluation measured a five-percent false-positive rate at the ninety-fifth-percentile threshold, and every escalation includes a human-readable rationale generated from the top-three feature contributors and the Patient's personal baseline so that Caregivers can judge each alert in context rather than reacting to a bare number.

7.1.1.4. Engineering Design Factors

In addition to the constraints above, ReMind was evaluated against eight standard engineering design factors. The table below summarises the effect level (0 - none, 10 - decisive) that each factor had on the design.

Factor	Effect (0-10)	Discussion
Public Health	7	Early detection of cognitive distress with MoodAI and reduction of wandering risk with safe-zone monitoring contribute to patient quality of life. Mood-trend summaries support clinical visits with objective data.
Public Safety	9	Patient safety dominated the design. Geofencing, SOS, and high-priority alerts directly reduce the chance that a patient is lost or remains in an unsafe situation. The 60-second / 50-metre violation debounce keeps alert quality high.
Public Welfare	6	ReMind reduces the supervision burden on family caregivers, themselves at risk of burnout. Event-driven oversight (rather than continuous surveillance) improves welfare on both sides of the relationship.
Global Factors	5	The Flutter codebase deploys to Android and iOS, which dominate the global smartphone market. Offline-first alert queueing tolerates intermittent connectivity. On-device MoodAI inference is planned for a follow-on phase. Initial deployment targets Türkiye but the architecture supports international rollouts.
Cultural Factors	6	Dementia care in Türkiye is predominantly a family responsibility. Interface language, alert framing, and cognitive-game content avoid jargon and assume a non-clinical caregiver. Future regional rollouts will require additional cultural review.

Social Factors	8	The Patient UI is designed for declining cognitive and motor abilities, using large buttons, no time pressure, no failure states. The Caregiver UI orders alerts by severity and avoids any output that could be misread as a clinical diagnosis.
Environmental Factors	3	On-device processing and event-driven location sharing reduce data-centre energy consumption. Patient-device battery is treated as an explicit design constraint.
Economic Factors	5	Free at the point of use. Backend on Firebase free tier; no paid third-party APIs; on-device ML keeps compute costs at the user rather than the platform.

Table 5: Effect of engineering design factors on the ReMind system.

Factors rated 5 or higher each had a noticeable influence on at least one architectural, interface, or policy decision. The dominant judgement underpinning the project is that an assistive technology for dementia must respect patient autonomy and dignity even at the cost of completeness of data: wherever a design choice required a trade-off between additional caregiver visibility and additional patient autonomy, the project chose the latter, accepting a slightly less informative caregiver dashboard in return for an unambiguously privacy-respecting product.

7.1.2. Standards

ReMind committed to a set of international standards that govern software engineering practice, data security and privacy, accessibility, and mobile health data interoperability. Each standard is summarised below together with how the project applies it.

7.1.2.1. Software Engineering Standards

- **IEEE 830-1998 (Software Requirements Specifications).** Used as the template for the functional and non-functional requirements documented in Section 2 of this report. Each requirement has a unique identifier, a single-purpose statement, and a rationale that traces back to a stakeholder need.
- **IEEE 1016-2009 (Software Design Descriptions).** Used as the template for the architectural and detailed-design content in Section 3. Each subsystem is described with purpose, services, dependencies, interfaces, and detail.
- **IEEE 730-2014 (Software Quality Assurance Processes).** Drives the testing documentation and traceability practices used in Section 5: every functional and non-functional test case is identified, categorised, and traceable to the requirement(s) it validates.
- **UML 2.5.1.** Used for use-case diagrams, sequence diagrams (for the Take Me Home flow and the SOS flow), and class diagrams for the Firestore data model documented in the supporting Detailed Design Report.
- **REST architectural style.** All client-backend communication is shaped as resource-oriented HTTPS calls through the Firebase SDK; identifiers, verbs, and status semantics follow REST conventions.

7.1.2.2. Data Security and Privacy Standards

- **ISO/IEC 27001:2022 (Information Security Management).** Used as the framework for confidentiality, integrity, and availability of patient data. Role separation, the principle of least privilege, and explicit access reviews are encoded in the Firestore Security Rules.
- **GDPR (EU 2016/679) informed design only.** The privacy design was informed by GDPR principles including per-category consent, transparency, and data minimisation. As a prototype, full GDPR compliance is not claimed.
- **KVKK (Türkiye, Kanun No. 6698) informed design only.** The design was informed by KVKK principles (consent, transparency, minimisation, erasure) without claiming full compliance. VERBIS registration and cross-border transfer approval are out of scope for this prototype.
- **Encryption Standards.** TLS 1.3 secures data in transit between mobile devices and Firebase, enforced automatically by the Firebase SDK. AES-256 encrypts data at rest in Firestore via Google Cloud infrastructure.

7.1.2.3. Accessibility and Usability Standards

- **WCAG 2.2 Level AA.** The Patient interface targets WCAG 2.2 Level AA: minimum 4.5:1 colour contrast for body text, no information conveyed by colour alone, tap targets of at least 64 dp, no time-pressured interactions, and compatibility with VoiceOver (iOS) and TalkBack (Android).

7.1.2.4. Mobile Health Data Standards

- **Apple HealthKit and Google Health Connect.** The design uses these platform APIs exclusively for wearable data access. Their schemas, permission models, and rate-limit guidance inform the planned integration, which will keep the project compliant with App Store and Play Store policies for health data.

7.1.2.5. Coding Standards

- **Effective Dart and Flutter Best Practices.** All Dart code follows Google's Effective Dart conventions (naming, documentation comments, asynchronous patterns) and is verified by flutter analyze in CI.
- **Secure Backend Patterns.** Functions follow least-privilege service-account configuration. Firestore Security Rules use schema-validation predicates for every write, and an automated emulator test suite exercises both positive and negative authorisation cases.

7.2. Ethics and Professional Responsibilities

Building assistive technology for cognitively impaired adults raises distinctive ethical responsibilities. ReMind addresses these through three commitments (respect for autonomy, transparency, and accountability) that the team recognised, observed, and fulfilled throughout the project.

Respect for autonomy is encoded directly into the architecture. ReMind does not perform continuous surveillance; instead, location and physiological signals are shared only on safe-zone exit, on detected anomaly, or on explicit Patient request. The design treats the Patient as the primary subject of consent: even after a Caregiver link is established, the Caregiver does not gain unrestricted access to the

Patient's data, and the Patient may revoke the link at any time. This protects the patient's right to dignity and self-determination even as their cognitive abilities decline.

Transparency is implemented through a per-category consent flow at onboarding, a dedicated Privacy & Data Use page that describes what is collected and how each category is used, and persistent visual indicators whenever location or health data is being processed. Free-text mood notes are stored in Firestore under the same server-side encryption as all other data. Consent statements are written in plain language so that a patient with mild cognitive impairment has a fair chance of understanding what they are agreeing to.

Accountability is treated as the responsibility of the development team. The MoodAI module is presented to caregivers as a decision-support mechanism, not as a clinical diagnostic tool. Every escalation includes a human-readable rationale generated from the dominant feature contributors, so the Caregiver can interpret the alert critically rather than simply acting on a black-box score. False-positive rates are tuned conservatively: a moderate anomaly triggers only an in-app support card, while only a confirmed high anomaly or a persistent moderate anomaly escalates to a Caregiver alert. The team also recognised the broader societal impact of the project: by making caregiving less invasive, ReMind contributes to a healthier caregiver-patient dynamic and reduces caregiver burnout, while the mood-trend export gives clinicians objective data without requiring continuous monitoring.

Throughout development the team applied GDPR and KVKK privacy principles as design guidelines and followed the IEEE software engineering standards listed in Section 7.1.2. As a prototype, formal regulatory compliance is not claimed.

7.3. Teamwork Details

Each team member's technical strengths and interests were taken into account when work packages were assigned, and each member led at least one work package while also assisting on others. Major decisions (architectural choices, privacy design, scope changes) were made collectively. The sub-sections below describe how each member contributed and functioned on the team, helped to create a collaborative and inclusive environment, took a lead role and shared leadership, and how the team met the objectives initially set in the project plan.

7.3.1. Contributing and Functioning Effectively on the Team

The team communicated weekly throughout both semesters, divided the workload evenly across all project phases, and kept overlapping awareness of each other's areas to minimise dependency risk. Project reports, presentations, and the analysis-and-requirements document were prepared together, with each member writing the parts most related to their work package and reviewing sections written by others.

The principal contributions of each team member are summarised below. Every member also contributed reviews, integration testing, and documentation across the entire project.

- **Ahmet Yağız Sarıdoğan.** Built the main Flutter application structure, defined the state-management approach (Riverpod), set up the GitHub Actions CI/CD pipeline with fastlane signing for Android and iOS, and managed Firebase App Distribution releases. Collaborated directly with Elif Ceren on the API contracts between the Flutter client and Firestore.

- **Ayça Candan Ataç.** Was the requirements-and-system-design lead during CS491. In CS492 owned the MoodAI subsystem end-to-end: StudentLife dataset parsing and feature engineering (phone, activity, mood, with derived delta / lag / variability families), the FT-Transformer architecture and its self-supervised masked-reconstruction training objective, the group-balanced log-transformed anomaly score, the anchored fourteen-day baseline + enhanced z-score personalisation, the three-tier (NORMAL / MEDIUM / HIGH) alert classifier with human-readable explanations, and the synthetic-injection evaluation. Specified the on-device deployment plan via TensorFlow Lite (LiteRT) for the follow-on integration phase.
- **Elif Ceren Çelik.** Designed the Cloud Firestore schema described in 4.3 and implemented the Firebase Authentication flow, the functions (alertFanout, linkApproval, accountDeletion, dataExport), and the Firestore Security Rules. Defined the data-synchronisation and notification-delivery patterns the rest of the team depended on.
- **Emine Noor.** Led the literature review during CS491 and owned the privacy-and-security design throughout both semesters: the per-category consent model and the Firestore Security Rules in collaboration with Elif Ceren. Made the backend changes required to satisfy privacy requirements personally.
- **Irmak İmdat.** Led the gap analysis and the overall testing strategy, owning the functional and non-functional test suites in 5. Owned the Safe Zone & Location Monitoring subsystem (geofencing, debounce, breach handling) and refactored the application into the modular ViewModel architecture used throughout the codebase.

7.3.2. Helping to Create a Collaborative and Inclusive Environment

From the start of CS491 the team set shared norms and applied them consistently. Goals and scope were defined together at the beginning of each semester. Design decisions went through several rounds of group input before being finalised, and any team member could call a focused decision meeting when their subsystem was blocked by a question that crossed module boundaries. The team used Google Drive for organised documentation and WhatsApp for everyday communication; this separation kept urgent messages visible without overwhelming design discussions.

Inclusivity was treated as a working norm rather than an aspiration. Every meeting opened with each member sharing their progress and any blockers in a fixed order, ensuring that quieter members had the same airtime as more vocal ones. Pull requests required review by at least one team member outside the author's primary subsystem, which kept knowledge distributed and prevented any single member from becoming a bottleneck. Disagreements were resolved by consensus where possible, and by simple-majority vote when consensus could not be reached; in the few cases where additional guidance was required, the team consulted Supervisor Sinem Sav.

7.3.3. Taking Lead Roles and Sharing Leadership on the Team

Leadership was distributed by subsystem rather than centralised, which allowed each member to develop ownership and to share day-to-day decisions within their area while still consulting the group on anything that touched another subsystem.

- **Ahmet Yağız Sarıdoğan** led the mobile application and deployment area. Early in CS491 he chose the Flutter project structure and the state-management approach, and he was in charge of the release pipeline. As features were developed, he worked directly with Elif Ceren on the API contracts between the Flutter client and the backend.

- **Ayça Candan Ataç** led the MoodAI module, including StudentLife dataset parsing and feature engineering, the FT-Transformer architecture and its masked-reconstruction training objective, the personalisation layer (anchored baseline + enhanced z-score), and the three-tier alert classifier. She consulted Sinem Sav on the detection thresholds, the escalation behaviour, and the design choice to keep personalisation as statistical post-processing rather than per-Patient weight fine-tuning.
- **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 privacy and security. She wrote the consent model and the Firestore Security Rules (with Elif Ceren), and made backend changes herself when those were required to satisfy privacy requirements. She consulted our advisor Sinem Sav on privacy decisions when needed.
- **Irmak İmdat** led project coordination through the gap analysis and the overall testing strategy. She also led the Safe Zone & Location Monitoring subsystem and supported the backend by refactoring the monolithic codebase into a modular ViewModel architecture, optimising Firestore queries, and helping implement comprehensive Security Rules.

7.3.4. Meeting Objectives

The project plan included in the Analysis and Requirements Report defined ten work packages: WP1 (Project Definition, lead: all members), WP2 (Requirements Analysis & System Design, lead: Ayça Candan Ataç), WP3 (Mobile Application Development, lead: Ahmet Yağız Sarıdoğan), WP4 (Backend Development, lead: Elif Ceren Çelik), WP5 (Database Design & Implementation, lead: Elif Ceren Çelik), WP6 (AI & Data Processing Integration, lead: Ayça Candan Ataç), WP7 (Privacy & Security Considerations, lead: Emine Noor), WP8 (Safe-Zone Monitoring & Response System, lead: Irmak İmdat), WP9 (Testing & Validation, lead: Irmak İmdat), and WP10 (Deployment & Finalisation, lead: Ahmet Yağız Sarıdoğan).

All ten work packages have been completed. The summary below records the level at which each milestone was met:

Work Package	Lead	Status / Level Met
WP1 Project Definition	All members	Fully met. Project Information Form delivered on schedule (September 2025).
WP2 Requirements & Design	Ayça Candan Ataç	Fully met. Project Specification Document and Analysis & Requirements Report delivered on schedule.
WP3 Mobile Application Development	Ahmet Yağız Sarıdoğan	Fully met. All planned screens implemented across Patient and Caregiver shells; minor schedule slippage absorbed in the integration window.
WP4 Backend Development	Elif Ceren Çelik	Fully met. Authentication, link management, reminder, alert, and SOS pipelines delivered.

WP5 Database Design & Implementatio n	Elif Ceren Çelik	Fully met. Firestore schema implemented with full indexing and Security Rules.
WP6 AI & Data Processing Integration	Ayça Candan Ataç	Met with adjusted scope: FT-Transformer (≈160k parameters) pretrained on the StudentLife dataset with a self-supervised masked-reconstruction objective, paired with an anchored fourteen-day per-Patient baseline + enhanced z-score classifier (NORMAL / MEDIUM / HIGH). Synthetic-injection evaluation reports detection rates of ninety-seven percent for mood declines, eighty-five to ninety-three percent for combined multi-domain declines, and a five-percent false-positive rate at the ninety-fifth-percentile threshold. The model is small enough to be exported through TensorFlow Lite (LiteRT) for on-device inference; full Flutter integration deferred to a follow-on phase.
WP7 Privacy & Security	Emine Noor	Fully met. Per-category consent flow, Firestore Security Rules, and Privacy & Data Use page all delivered. Standard Firebase TLS 1.3 / AES-256 encryption in use.
WP8 Safe-Zone Monitoring & Response	Irmak İmdat	Fully met. Up to three safe zones per patient, OS-level geofencing inside zones, GPS sampling on confirmed breach, and Take Me Home navigation all delivered.
WP9 Testing & Validation	Irmak İmdat	Met with two known issues. 37 of 39 functional and non-functional test cases pass without observation; the two outstanding items concern FCM latency under heavy packet loss and Take Me Home battery profile (see 5.3 and 8 Future Work).
WP10 Deployment & Finalisation	Ahmet Yağız Sarıdoğan	Ongoing. Internal Firebase App Distribution build live; final demonstration scheduled for May 2026.

Table 6: Project plan status by work package.

All functional requirements specified in 2.1 have been implemented and verified by the corresponding functional test cases (F001-F027). All non-functional requirements specified in 2.2 have been implemented and verified by NF001NF012, with the two minor outstanding items noted above. Two scope reductions were made relative to the original specification: (1) smartwatch and HealthKit / Health Connect integration was deferred; the current MoodAI operates on phone-only signals, and (2) an early plan to support a third user role (the clinician) was descoped to a PDF export (FR-PRV-06) once stakeholder consultation indicated that direct clinician access introduced disproportionate regulatory complexity. Both decisions were taken collectively after consultation with Sinem Sav.

7.4. New Knowledge Acquired and Applied

The project required the team to acquire substantial knowledge outside the standard undergraduate curriculum. The principal new domains were: (a) on-device machine learning with TensorFlow Lite, including int8 quantisation and personalised fine-tuning; (b) the Firebase platform, in particular Firestore Security Rules, functions, and Firebase Cloud Messaging; (c) cross-platform mobile development with Flutter and Dart, including state management with Riverpod and accessibility tooling; (d) the Apple HealthKit and Google Health Connect API documentation, studied in preparation for a follow-on integration phase; (e) the GDPR and KVKK data-protection regimes; and (f) the WCAG 2.2 accessibility guidelines.

To minimise the risk introduced by these knowledge gaps, the team adopted an incremental learning-and-validation strategy. Before each component was integrated into the system it was prototyped in isolation: a stand-alone Firebase project for Firestore Security Rule experiments, and a stand-alone Python notebook for TensorFlow Lite training and quantisation. Each prototype validated technical feasibility and surfaced constraints before integration, which kept the integration window in March 2026 short and predictable.

The team relied on three complementary learning resources. Official documentation (Flutter, Firebase, TensorFlow Lite, IEEE standards, ICO and KVKK guidance) was the primary reference, since for a mobile health product the official specification is also the regulatory baseline. Peer-reviewed literature, particularly the references on edge computing for health data, on personalised wearable anomaly detection, and on HCI for older adults, informed the design choices around MoodAI thresholds, support cards, and the calm Patient UI. Targeted hands-on experimentation in the prototype spikes filled the gap between the documentation and a working integrated system.

Findings from each spike were documented in the team's shared Google Drive so the prototype's lessons were available to every member during integration. Internal communication was used to flag obstacles early, and existing design decisions were revised when new knowledge made them obsolete; for example, the original plan to compute MoodAI scores on a backend service was abandoned once the on-device benchmarks demonstrated that the privacy and latency benefits of edge computation could be realised at acceptable battery cost. This proactive learning strategy enabled the team to deliver a comprehensive, ethically acceptable, and functional solution in spite of the breadth of domains involved.

8. Conclusion and Future Work

ReMind set out to address a real and growing problem: the fragmented landscape of dementia-care technology, in which existing tools handle either location, or reminders, or cognitive support, but rarely all three together, and rarely with proper attention to the privacy and dignity of the patient. The project has produced a working, cross-platform mobile health assistant that integrates these capabilities into a single privacy-first ecosystem, backed by Firebase services, with data protected by Firebase's standard TLS 1.3 in transit and AES-256 at rest, and assisted by an on-device anomaly-detection module.

The functional and non-functional test suites validate that the application meets the requirements specified at the start of the project. Patients can register, link with a caregiver, receive medication reminders, complete mood check-ins in under fifteen seconds, play non-competitive cognitive games, request Take Me Home navigation, and trigger an SOS alert. Caregivers receive prioritised alerts only when an event warrants their attention (a safe-zone breach, a high or persistent anomaly, or an explicit SOS); otherwise see only summary trend data. All sensitive data is protected by TLS 1.3 in transit and AES-256 at rest via Google Cloud infrastructure.

There are several directions in which we expect the project to evolve. The first is multi-modal sensor fusion: the current MoodAI module operates on a relatively small feature window, and broader integration of accelerometer, gyroscope, and ambient-light data would enable the model to recognise gait changes, sleep disruption, and reduced ambulation as additional anomaly signals. The second is clinician integration: the PDF export defined in FR-PRV-06 is a first step, but a structured, opt-in clinician portal (built on top of the same Firestore data with strict role separation) would close an important loop in the dementia care pathway. The third is cost optimization at scale: as the user base grows, Firestore read volume and FCM delivery costs will require additional review, and we expect to introduce a server-side aggregation function that batches dashboard reads. The fourth is internationalisation: the application currently ships in Turkish and English, but the cultural and linguistic adjustments required to enter additional markets, particularly markets with different family-care norms, would require both translation and an expanded cultural review.

Two known limitations remain. Under heavy cellular packet loss (above 30 %), FCM delivery latency can occasionally cross the five-second target documented in NF005; Take Me Home consumes more battery than the four-percent-per-hour target documented as part of NF003. Both are tracked in the project's GitHub issue tracker and are expected to be addressed in the next quarterly release.

ReMind has been a substantial undertaking, both technically, given the breadth of platforms and standards involved, and ethically, given the sensitivity of the user population. The team is pleased with the result and grateful for the guidance of Supervisor Sinem Sav and Course Instructors İlker Kurt and Mert Bıçakçı throughout the project.

9. Glossary

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.
FCM	Firebase Cloud Messaging: Google's push-notification service.
Firebase	Google's mobile backend platform used for authentication, data storage (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: the European Union data-protection regime.
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: HTTP over TLS, used for all communication between the app and backend.
ISO/IEC 27001	International standard for information security management systems.
ISMS	Information Security Management System.
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.
MoodAI	ReMind's behavioural anomaly-detection module. Aggregates phone-usage, activity, and mood check-in data into a daily feature row, computes an anomaly score with a pretrained FT-Transformer using a self-supervised masked-reconstruction objective,

	then personalises the score against a fourteen-day per-Patient baseline and classifies the day as NORMAL, MEDIUM, or HIGH.
FT-Transformer	Feature-Tokenizer Transformer (Yandex / rtdl): a Transformer architecture purpose-designed for heterogeneous tabular features, in which each feature receives its own learned token embedding before standard Transformer-encoder processing.
StudentLife	A public mobile-sensing dataset collected at Dartmouth University in 2013 (Wang et al., UbiComp 2014) tracking forty-nine students over roughly ten weeks with smartphone sensors and Ecological Momentary Assessment (EMA) surveys; used to pretrain the MoodAI model.
EMA	Ecological Momentary Assessment: brief in-context self-report surveys delivered on the user's phone.
PAM	Photographic Affect Meter: a 4×4 affect grid in which the user picks one of sixteen images mapping to valence and arousal scores.
OASIS	Open Affective Standardized Image Set: a standardised image bank in which each image has known valence, arousal, and dominance scores, used by ReMind for image-based mood check-ins.
Masked Reconstruction	A self-supervised training objective in which a fraction of input features is masked and the model learns to predict the masked values from the unmasked context; the reconstruction error at inference time is used as an anomaly score.
Anchored Baseline	MoodAI's personalisation strategy in which the first fourteen days of a Patient's data establish a fixed personal mean and standard deviation against which subsequent days are z-scored, so that a normal level is defined per-Patient rather than per-population.
Enhanced Z-score	max(model z-score, mean per-feature z-score): combines the model's reconstruction-based anomaly with per-feature personal z-scores to catch gradual single-feature drift.
ML	Machine Learning.
mHealth	Mobile Health.
NoSQL	Non-relational database model designed for flexible data storage.
OS	Operating System.
RBAC	Role-Based Access Control.
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.

SOS	Emergency distress signal requesting immediate assistance, raised explicitly by the Patient.
TFLite / LiteRT	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.
Wi-Fi	Wireless Fidelity.

Table 7: Glossary of terms used in this report.

10. 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] "On-Device AI for Privacy-Preserving Mobile Applications: A Framework using TensorFlow Lite," International Journal for Research in Applied Science and Engineering Technology, vol. 13, no. 8, pp. 1963-1972, Aug. 2025.

[5] "Get started with Health Connect," Android Developers. [Online]. Available: https://developer.android.com/health-and-fitness/health-connect/get-started.

[6] "LiteRT overview," TensorFlow. [Online]. Available: https://ai.google.dev/edge/litert.

[7] "Encryption – General Data Protection Regulation (GDPR)," GDPR.eu. [Online]. Available: https://gdpr-info.eu/issues/encryption/.

[8] "How to encrypt health data for GDPR & HIPAA compliance," Chino.io, 2024. [Online]. Available: https://www.chino.io/post/how-to-encrypt-health-data-for-gdpr-hipaa-compliance.

[9] "IEEE Recommended Practice for Software Requirements Specifications," IEEE Std 830-1998. [Online]. Available: https://standards.ieee.org/ieee/830/1222/.

[10] "IEEE Standard for Information Technology – Systems Design – Software Design Descriptions," IEEE Std 1016-2009. [Online]. Available: https://standards.ieee.org/ieee/1016/4502/.

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

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

[13] A. K. Rowe, R. J. Doughty, and M. T. Montague, "Wandering in dementia: a comprehensive review," J. Adv. Nurs., vol. 55, no. 2, pp. 190-200, Jan. 2006. doi:10.1111/j.1365-2648.2006.03826.x.

[14] S. Adelman, M. T. Tmanova, A. Delgado, I. Dionne-Odom, and S. L. Thornton, "Caregiver Burden: A Clinical Review," JAMA, vol. 316, no. 4, pp. 352-360, 2016. doi:10.1001/jama.2014.304.

[15] K. Niemeijer, J. M. Frederiks, R. B. Depla, E. T. Eefsting, and C. M. Hertogh, "The ideal application of surveillance technology in residential care for people with dementia," J. Med. Ethics, vol. 37, no. 5, pp. 303-309, May 2011. doi:10.1136/jme.2010.040774.

[16] M. Klein, N. Mogles, and A. van Wissen, "An Intelligent Coaching System for Therapy Adherence," IEEE Pervasive Comput., vol. 12, no. 3, pp. 22-30, 2013. doi:10.1109/MPRV.2013.41.

[17] A. Iaboni et al., "Wearable multimodal sensors for the detection of behavioral and psychological symptoms of dementia using personalized machine learning models," Alzheimers Dement. (Amst.), vol. 14, no. 1, p. e12305, 2022. doi:10.1002/dad2.12305.

[18] G. Lawal, "Edge Computing for Secure Real-Time Health Data Processing in Remote Settings," 2020. doi:10.13140/RG.2.2.31003.94240.

[19] C.-K. Tseng, S.-J. Huang, and L.-J. Kau, "Wearable Fall Detection System with Real-Time Localization and Notification Capabilities," Sensors, vol. 25, no. 12, p. 3632, 2025. doi:10.3390/s25123632.

[20] N. Liu, J. Yin, S. Tan, K. Ngiam, and H. Teo, "Mobile health applications for older adults: A systematic review of interface and persuasive feature design," J. Am. Med. Inform. Assoc., vol. 28, 2021. doi:10.1093/jamia/ocab151.

[21] Y. J. Jeong et al., "Digital Therapeutics for Alzheimer's and Parkinson's Diseases: Current Trends and Future Perspectives," Med. Res. Rev., 2025. doi:10.1002/med.70005.

[22] W.-C. Tsai, C.-F. Chi, and Y.-H. Huang, "Technology Service Design for the Older Adults with Dementia," in HCI Int. 2023, Springer, 2023, pp. 349-360. doi:10.1007/978-3-031-34917-1_26.

[23] A. Babel, R. Taneja, F. M. Malvestiti, A. Monaco, and S. Donde, "Artificial Intelligence Solutions to Increase Medication Adherence in Patients With Non-communicable Diseases," Front. Digit. Health, vol. 3, p. 669869, 2021. doi:10.3389/fdgth.2021.669869.

[24] A. Czaja and C. Lee, "The Impact of Aging on Access to Technology," Universal Access Inf. Soc., vol. 5, pp. 341-349, 2006. doi:10.1007/s10209-006-0060-x.

[25] R. Peng et al., "Using nudges to promote health among older adults: A scoping review," Int. J. Nurs. Stud., vol. 161, p. 104946, 2025. doi:10.1016/j.ijnurstu.2024.104946.

[26] S. Hammer, B. Lugrin, S. Bogomolov, K. Janowski, and E. André, "Investigating Politeness Strategies and Their Persuasiveness for a Robotic Elderly Assistant," in Intell. Virtual Agents (IVA 2016), Springer, 2016, pp. 315-326. doi:10.1007/978-3-319-31510-2_27.

[27] M. M. Kamruzzaman, I. Alrashdi, and A. Alqazzaz, "New Opportunities, Challenges, and Applications of Edge-AI for Connected Healthcare in Internet of Medical Things for Smart Cities," J. Healthc. Eng., vol. 2022, p. 2950699, 2022. doi:10.1155/2022/2950699.

[28] S. Köhler et al., "Ethics, design, and implementation criteria of digital assistive technologies for people with dementia from a multiple stakeholder perspective: a qualitative study," BMC Med. Ethics, vol. 25, no. 84, 2024. doi:10.1186/s12910-024-01080-6.

[29] D. Parrilli, Informational Privacy for Service Design: An Ethical Framework for Designing Privacy-Oriented Services, Springer, 2025. doi:10.1007/978-3-031-76926-9.

[30] M. Turakhia et al., "Rationale and Design of a Large-Scale, App-Based Study to Identify Cardiac Arrhythmias," Am. Heart J., vol. 207, pp. 66-75, 2019. doi:10.1016/j.ahj.2018.09.002.

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

[32] Google, "Flutter: Build Apps for Any Screen," 2024. [Online]. Available: https://flutter.dev.

[33] TensorFlow, "TensorFlow Lite: Deploy ML on Mobile and Edge Devices," 2024. [Online]. Available: https://www.tensorflow.org/lite.

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

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

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

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

[38] Google, "Health Connect API Documentation," 2024. [Online]. Available: https://developer.android.com/health-and-fitness/guides/health-connect.

[39] 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.

[40] 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.

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

[42] IEEE, "IEEE Standard for Software Quality Assurance Processes," IEEE Std 730-2014. [Online]. Available: https://standards.ieee.org/ieee/730/5284/.

[43] R. Wang et al., "StudentLife: Assessing mental health, academic performance and behavioral trends of college students using smartphones," in Proc. 2014 ACM Int. Joint Conf. on Pervasive and Ubiquitous Computing (UbiComp '14), Seattle, WA, USA, Sep. 2014, pp. 3-14. doi:10.1145/2632048.2632054.

[44] Y. Gorishniy, I. Rubachev, V. Khrulkov, and A. Babenko, "Revisiting Deep Learning Models for Tabular Data," in Advances in Neural Information Processing Systems (NeurIPS), 2021. [Online]. Available: https://arxiv.org/abs/2106.11959.

[45] B. Pollak, J. Adler, and S. Eslami, "PAM: A Photographic Affect Meter for Frequent, In-Situ Measurement of Affect," in Proc. SIGCHI Conf. on Human Factors in Computing Systems (CHI '11), 2011, pp. 725-734. doi:10.1145/1978942.1979047.

[46] B. Kurdi, S. Lozano, and M. R. Banaji, "Introducing the Open Affective Standardized Image Set (OASIS)," Behavior Research Methods, vol. 49, no. 2, pp. 457-470, 2017. doi:10.3758/s13428-016-0715-3.

Appendix A: User's Manual

This manual walks both Patients and Caregivers through every screen of the ReMind application. It assumes that the application has already been installed on a compatible Android (API level 26+) or iOS (15+) device. Installation instructions appear in Section A.12. The same content is also embedded in the application's in-app help and on the public project website.

A.1. Registration

On first launch the application presents a Welcome screen with two buttons: Create Patient Account and Create Caregiver Account. Tapping either opens a registration form that asks for full name, email address, password, and acceptance of the Terms of Service and Privacy Policy. After submission the application sends a six-digit verification code to the supplied email; entering the code completes registration and returns the user to the Login screen.

A.2. Login

On the Login screen the user enters their registered email and password. ReMind also offers passwordless login via an email magic link: tapping "Send me a sign-in link" delivers a one-tap login link to the registered email. After successful authentication the user is routed to the Patient or Caregiver dashboard depending on their role.

A.3. Patient Dashboard

The Patient dashboard is the home screen for Patient users. It displays nine large tiles arranged in a two-by-three grid:

- Get Code opens the screen where patients can see the code to give to their caregiver.
- My Caregivers opens the list of patients caregivers.
- Today's Reminders opens the chronological list of today's medication and routine reminders.
- Mood Check-in opens the three-step mood-check-in flow.
- Take Me Home opens the map-based navigation back to the configured home location.
- Home Location opens the location of the patient’s home.
- Cognitive Games opens the games library.
- SOS raises an immediate high-priority alert to the linked caregiver.
- Settings opens the Patient settings page including consent management.

A status bar at the top of the screen shows the active monitoring indicators: a green location pin if location tracking is active, a green location pin if location tracking is active, and a green link icon if the caregiver link is healthy.

A.4. Caregiver Dashboard

The Caregiver dashboard is organised by priority. The top section shows live alerts (if any), each alert displayed with a colour-coded chip, the reason, the patient name, the last known location, and the timestamp. Tapping an alert opens a detail view with a map and a Mark as Acknowledged button.

Below the alerts is a per-patient summary: a colour-coded status chip, the latest mood, and the time of the most recent reminder response. Below the summary are entry points to the full Alert History, the Mood Trends chart, the Reminder Manager, the Safe-Zone Editor, and the Settings page.

A.5. Linking Patient and Caregiver

From the Patient settings page, tapping Generate Linking Code produces a six-digit code that is valid for ten minutes. The Patient communicates the code to the Caregiver, who enters it in the Caregiver settings page under Link a Patient. The Patient then receives an in-app confirmation prompt; on Approve the link is created and both parties see the new relationship reflected immediately. Either party may unlink at any time from Settings.

A.6. Reminders

Reminders may be created either by the Patient or by a linked Caregiver. The Reminder Manager presents a list of scheduled reminders with a New Reminder button. Each reminder consists of a title, a category (medication or routine), a scheduled time, and an optional repeat pattern. When a reminder fires on the Patient device, the screen displays the reminder text and three large buttons (Done, Skipped, Snoozed) each at least 64 dp tall. Snoozing reschedules the reminder by ten minutes.

A.7. Mood Check-in

The Mood Check-in flow is a one-step questionnaire designed to be completed in 5-10 seconds. Mood Check-in asks the Patient to select an image that best represents how they feel right now (energy, calm, agitation). After submission the Patient sees a brief confirmation and the mood is recorded for use by the MoodAI module and by the Caregiver mood timeline.

A.8. Safe Zones

The Safe-Zone Editor is available only to the Caregiver. Tapping anywhere on the map drops a circular zone whose radius can be adjusted with a drag handle. The zone is named (e.g. Home, Day Centre) and saved with a single tap. Up to three zones can be active simultaneously. Saved zones are pushed to the Patient device immediately and used by the Safe Zone & Location Monitoring service.

A.9. Take Me Home

Tapping the Take Me Home tile on the Patient dashboard opens a map showing the Patient's current location, the configured home location, and a simple turn-by-turn path between them. Oversized arrows indicate the next direction. The path is recomputed every fifteen seconds. Take Me Home requires foreground operation; if the Patient backgrounds the application, the path will pause and resume when the application is brought back to the foreground.

A.10. Cognitive Games

The cognitive-games library contains two games: Memory pairs and Pattern recall. None of the games has a timer, leaderboard, or failure state. The Patient may play for as short or as long as they wish. Engagement statistics (session length and completion rate) are stored locally on the device.

A.11. Settings

Both Patient and Caregiver have a Settings page. Common entries include language (Turkish or English), theme (light or dark), notification preferences (quiet hours, MoodAI sensitivity), and Privacy & Data Use. The Patient settings additionally include the consent manager (per-category opt-in/opt-out) and the data-export and account-deletion entries. The Caregiver settings additionally include the per-Patient notification preferences.

A.12. Installation

ReMind is distributed through the Google Play Store and Apple App Store. To install on Android, search for "ReMind" in the Play Store, tap Install, accept the requested permissions on first launch (notifications, location, physical activity, body sensors), and proceed to Registration. To install on iOS, search for "ReMind" in the App Store, tap Get, accept the requested permissions on first launch (notifications, location, and motion), and proceed to Registration. During the development pilot, the application is also distributed through Firebase App Distribution; testers receive a tester-invitation email with the install link.