ChatGPT

Weekly Journal – AI-Enhanced Alzheimer’s Caregiver Support Project

Week 1 – Mobile Lead

Date: 04-08-2025 to 09-08-2025

What I Worked On: I set up the Android project structure for the **patient-facing app**, creating a dual-module workspace and implementing the core navigation framework. I established the MVVM architecture in Android Studio and defined Room database entities (`PatientEntity`, `TaskEntity`, `ReminderEntity`) for local storage. I collaborated with the team to configure Firebase Authentication and Firestore dependencies in Gradle, enabling future cloud sync. I also implemented the initial UI layouts (main dashboard, patient profile entry) with a focus on accessibility.

Work Summary: This week I laid the foundation of the patient app. I built the skeleton of the application with separate packages and modules, implementing `MainActivity` and navigation components to link patient screens. I created the Room database layer, adding the core entities and DAOs for patient data. In Android Studio, I configured Firebase for user sign-in and prepared Firestore so we can sync patient data to the cloud. Finally, I sketched out the first UI wireframes and developed basic layouts (e.g., patient dashboard and profile form) following Material Design guidelines with senior-friendly buttons and fonts.

Hours Worked: 32 hours

Key Learnings:
- Gained expertise in Android multi-module project setup and MVVM architecture.
- Deepened understanding of Room database relationships and entity design for health data.
- Learned Jetpack Navigation Component to manage complex app flows.
- Improved Git branching strategies for collaborative Android development.

Blockers:
- Designing a dual-app (patient vs caregiver) codebase structure initially felt complex.
- Configuring Firebase with proper security rules (HIPAA compliance) requires further attention.
- Balancing local database schema with future cloud sync logic needed careful planning.

Week 1 – Cloud & Data Lead

Date: 04-08-2025 to 09-08-2025

What I Worked On: I designed the backend infrastructure for the project. Specifically, I defined the Firestore data schema for patients, tasks, and reminders, and drafted initial security rules to isolate patient-caregiver data. I implemented Firebase Authentication, enabling email/password sign-up flows for patients and caregivers. In our Room database setup, I mapped the entities to mirror the cloud schema, preparing for bi-directional sync. I also configured offline persistence for Firestore to support intermittent connectivity.

Work Summary: This week I focused on cloud setup and data modeling. I created the initial Firestore collections for patient profiles and linked data, ensuring they correspond to our local Room entities. I set up Firebase Auth with a basic login/register flow. Then I enabled Firestore offline mode so local changes queue when the network is down. I also started writing the first version of security rules that restrict caregivers' access to only their assigned patients. In addition, I established database migrations in Room to smoothly evolve our schema.

Hours Worked: 32 hours

Key Learnings:
- Learned best practices for structuring Firestore collections and documents for healthcare data.
- Gained proficiency with Firebase Auth and Firestore offline-first capabilities.
- Understood basic considerations for HIPAA-compliant storage and access control (role-based rules).
- Learned to manage versioned database migrations in Room.

Blockers:
- Determining the correct access levels for patients versus caregivers in Firestore rules required discussion.
- Ensuring seamless sync logic for local and cloud databases will need further testing.
- Researching encryption requirements for patient data storage on Firebase is ongoing.

Week 1 – AI/ML Lead

Date: 04-08-2025 to 09-08-2025

What I Worked On: I initiated the AI and ML foundations of the project. I researched conversational AI options (Google Gemini API, etc.) and drafted the chatbot integration plan. I set up a Python/Android environment for developing NLP components. I also explored available cognitive assessment models, studying the Mini-Mental State Examination (MMSE) protocol and how to implement its scoring algorithm digitally. Early on, I experimented with TensorFlow Lite to ensure our app could eventually run ML models offline.

Work Summary: This week was dedicated to planning the AI features. I evaluated AI chatbot APIs for multi-language support, outlining how we'll integrate them into `ChatbotActivity`. I sketched the architecture for processing and analyzing patient conversations (for memory extraction). I also reviewed clinical guidelines for cognitive tests and identified the data needed for our first assessment (MMSE). On the technical side, I installed and tested TensorFlow Lite samples in the Android project to ensure on-device inference is feasible. These steps set the stage for next week's development.

Hours Worked: 32 hours

Key Learnings:
- Surveyed large language model APIs (e.g., Gemini) and integration via REST in Android.

- Deepened understanding of cognitive assessment protocols (MMSE) and how to quantify results.
- Learned about deploying TensorFlow Lite on Android for offline AI inference.
- Recognized challenges in multilingual NLP and domain-specific conversation analysis.

Blockers:
- There is limited pre-trained NLP support for certain Indian dialects; may need custom data collection.
- Regulatory compliance for AI in healthcare (avoiding medical advice) needs careful monitoring.
- Coordination between the AI services and Firebase backend (e.g., logging conversations) must be defined.

Week 1 – Caretaker App + UX/QA Lead

Date: 04-08-2025 to 09-08-2025

What I Worked On: I began designing the caregiver-facing app and UX guidelines. I sketched wireframes for the caregiver dashboard, including patient lookup and data screens. I researched accessibility standards for elderly users, planning high-contrast, large-touch interfaces. I also set up the skeleton of the CaretakerApp module in Android Studio, aligning package structure with the patient app for shared code. Finally, I prepared a UX checklist to ensure color contrast, font sizes, and iconography are appropriate for our demographic.

Work Summary: This week I focused on the caregiver user experience. I defined the initial navigation flow for caregivers: linking to a patient's profile, viewing alerts, and managing tasks. I created prototype screens that emphasize simplicity – big buttons, clear labels, and minimal steps. I coordinated with the Mobile Lead to ensure both apps use consistent styles and shared utilities. I also outlined a QA plan, drafting test scenarios that include accessibility checks (e.g., screen reader behavior). This foundation will guide the caregiver app development.

Hours Worked: 32 hours

Key Learnings:
- Learned about dual-app architecture and how to maintain design consistency across modules.
- Became familiar with Android accessibility best practices (WCAG compliance) for seniors.
- Gained insight into caregiver workflows and how they differ from patient needs.
- Improved proficiency with UI design tools (e.g., Figma) for creating interactive wireframes.

Blockers:
- Defining user roles and permissions (patient vs caregiver) in the app logic needs careful planning.
- Choosing the right level of detail on the dashboard (enough info without cluttering) is challenging.
- Limited initial feedback from test users means assumptions on UX must be validated later.

Week 2 – Mobile Lead

Date: 11-08-2025 to 16-08-2025

What I Worked On: I developed patient-side features this week. I implemented `PatientProfileActivity` and `DetailedPatientProfileActivity` to allow patients to enter and view personal info (name, age, medical history). I created `EmergencyActivity` that places a one-touch call to a preset contact and triggers an alert to the caregiver. I also built the foundation of `MedicationActivity` for patients to add medication reminders. In all these screens, I improved the UI for accessibility: using large buttons, high-contrast colors, and a simple layout. I integrated Android's SpeechRecognizer to add voice input hints (e.g., for form entries).

Work Summary: The focus this week was on core patient functionalities with an emphasis on ease of use. The patient profile system was fully implemented, including form validation and data persistence. I prioritized an easy-to-use emergency contact feature: with one tap the patient's caregiver is alerted and the emergency number is dialed. The medication tracking screen was created so patients (or caregivers on the patient's device) can schedule reminders. Throughout, I enhanced the UI (in `activity_medication.xml`, etc.) to be senior-friendly by increasing touch target sizes and ensuring clear text. Voice input helpers were added to reduce typing for users with difficulty.

Hours Worked: 32 hours

Key Learnings:
- Mastered form handling and validation in Android activities.
- Learned to use Android Intents for emergency calling and integrating with Contacts.
- Gained experience applying accessibility guidelines (content descriptions, tactile feedback).
- Implemented basic voice input (SpeechRecognizer) to assist user entry.

Blockers:
- Managing runtime permissions for phone calls and speech input required careful handling.
- Ensuring data privacy of personal info requires implementing encryption later.
- Offline functionality (reminders when offline) will need more work on the sync side.

Week 2 – Cloud & Data Lead

Date: 11-08-2025 to 16-08-2025

What I Worked On: I expanded the cloud backend to support new patient features. I updated the Firestore schema to store patient profile details, medication schedules, and emergency contacts. I configured Firestore listeners so that when the patient updates their profile or medication, the caregiver's device will sync automatically. I also set up FCM (Firebase Cloud Messaging) to handle alerts: for example, an emergency trigger will send a push to the caregiver app. I fine-tuned our Room database design to match the new Firestore structure for seamless syncing, and I began drafting enhanced security rules for these new collections.

Work Summary: This week I ensured the cloud side is aligned with the patient app enhancements. I created the Firestore collections and rules for profiles, medications, and reminders. I linked the patient's local changes to propagate to the cloud and vice versa. For emergency alerts, I integrated FCM so that notifications can be pushed instantly. I also worked on offline persistence so that if the patient enters data without internet, it queues up for later sync. By the end of the week, caregiver and patient data flows are in place and secured by preliminary rules.

Hours Worked: 32 hours

Key Learnings:
- Enhanced my understanding of role-based access in Firestore security rules.
- Learned how to configure FCM topics and alerts for critical patient events.
- Improved data modeling skills to keep local Room and remote Firestore in sync.
- Explored offline data handling (caching, write queues) in Firebase.

Blockers:
- Defining precise permission rules for caregiver vs patient was non-trivial.
- Handling FCM token management (registration, topics) will need careful error handling.
- Ensuring the app correctly retries failed writes in offline mode is still to be tested thoroughly.

Week 2 – AI/ML Lead

Date: 11-08-2025 to 16-08-2025

What I Worked On: I enhanced the app's voice interaction capabilities. I implemented a `SpeechRecognitionHelper` that allows patients to use voice input for common actions (e.g., adding a reminder by speaking). I also integrated Android's `TextToSpeech` engine to provide spoken feedback for some key UI events (e.g., confirmation of emergency call). I started developing basic NLU routines to parse simple commands from patient speech in multiple languages. In parallel, I began prototyping NLP models to analyze cognitive game responses.

Work Summary: The focus this week was on improving the natural interaction methods. I added voice input to reduce patient effort for data entry and commands. The system now reads back critical information to the patient through TTS. I also laid groundwork for future cognitive assessments by writing initial NLP code that can interpret a patient's spoken or typed answers in games. These features aim to make the app more accessible to patients who have difficulty typing or seeing the screen.

Hours Worked: 32 hours

Key Learnings:
- Gained practical experience with Android SpeechRecognizer and TextToSpeech APIs.
- Learned about designing simple grammars for speech recognition to improve accuracy.
- Began exploring NLU libraries for tokenizing and intent recognition in different languages.
- Recognized the need for domain-specific language models for better accuracy.

Blockers:
- Achieving reliable recognition for elderly speech with background noise is challenging.
- Handling multilingual speech input (Hindi, Tamil, etc.) may require separate models.
- Ensuring privacy (processing of voice data) remains compliant in patient context.

Week 2 – Caretaker App + UX/QA Lead

Date: 11-08-2025 to 16-08-2025

What I Worked On: I focused on caregiver UI and usability. I improved the layouts with large buttons and clear labels (updated `activity_patient_link.xml`, etc.) to make navigation simpler. I designed and implemented the caregiver's view of the emergency system: when the patient's app triggers an emergency, the caregiver app now displays a prominent alert with the patient's location. I also refined the color scheme and element sizing on all caregiver screens for consistency and readability. Throughout the week, I began writing test cases to validate these features, particularly around accessibility (e.g., VoiceOver/TalkBack support).

Work Summary: This week I enhanced the caregiver experience of the new features. I polished the interface so caregivers can easily see patient status and receive notifications. The emergency flow was integrated on the caregiver side: a "Press to Call" prompt now appears when an alert is received. I conducted preliminary usability tests on font size and button reachability, adjusting styles in `values/colors.xml` and `dimens.xml` accordingly. I also started formulating QA plans to cover voice interaction and accessibility, ensuring our design choices are verified.

Hours Worked: 32 hours

Key Learnings:

- Strengthened understanding of accessibility guidelines for cognitive impairment (e.g., simple language).
- Gained experience in integrating real-time notifications into the caregiver UI.
- Learned importance of consistent UI across both patient and caregiver apps for familiarity.
- Improved QA planning skills for cross-module features (voice input, emergency alerts).

Blockers:

- Coordinating with the patient app for emergency alerts required careful cross-app testing.
- Verifying emergency location sharing needed ensuring location permissions and privacy concerns.
- Balancing detailed information versus simplicity in the caregiver dashboard was challenging.

Week 3 – Mobile Lead

Date: 18-08-2025 to 23-08-2025

What I Worked On: I implemented patient-side task and voice features. I completed `TasksActivity` to allow patients to create and schedule tasks, including setting priority levels and due dates. I integrated Android's TextToSpeech so that chatbot replies can be spoken aloud to the patient. I worked on linking the new AI `ChatbotActivity` into the navigation flow. I also made sure our layouts and resources support all required languages, using localized string files for English, Hindi, Tamil, etc.

Work Summary: Week 3 was a big milestone for patient app features. The task management screen now supports complex workflows: recurring tasks, priority tagging, and completion tracking all work smoothly. I added voice output to the chatbot interface, so that the patient hears responses. I integrated the AI chatbot module into our patient app, enabling conversational interaction. Throughout, I ensured that the UI is fully localized: the app can switch between English and regional languages seamlessly. These improvements significantly enhance the usability and accessibility of the patient app.

Hours Worked: 32 hours

Key Learnings:
- Mastered Android TextToSpeech initialization and controlling speech queues.
- Learned how to structure multi-lingual resources (`res/values-hi/`, etc.) for dynamic language support.
- Improved skills in building dynamic RecyclerView lists (for tasks) with diverse item types.
- Understood importance of performance considerations when updating UI in multiple languages.

Blockers:
- Managing concurrent task updates (e.g., recurring tasks) requires careful state handling.
- Ensuring text-to-speech performance on older devices (e.g., latency) can be tricky.
- Testing all language variants to catch UI overflow issues is time-consuming.

Week 3 – Cloud & Data Lead

Date: 18-08-2025 to 23-08-2025

What I Worked On: I supported the caretaker integration in the backend. I created a new Firestore collection and rules to implement the patient-caregiver linking (used by `PatientLinkActivity`). I also set up real-time Firestore listeners so that any tasks or reminders created on one device instantly appear on the other. I enhanced our security rules to ensure that a caregiver can only read/write data for linked patients. Additionally, I made sure that multi-language fields (e.g., patient responses) are stored properly for analysis.

Work Summary: My work this week centered on data flow between patient and caregiver apps. I implemented the linking mechanism in Firestore: when a caregiver enters a patient code, the backend associates them. I verified that updates to tasks or medication on one side are immediately synced thanks to real-time listeners. I also tested that our rules prevent unauthorized access. Behind the scenes, I ensured the database can handle content in different languages (UTF-8/Unicode handling) without issues. With these changes, cross-device synchronization is robust.

Hours Worked: 32 hours

Key Learnings:
- Learned to manage many-to-many relationships in Firestore (caregivers to patients).
- Became adept at writing Firestore rules that enforce link-based access control.
- Improved knowledge of real-time listeners and how to prevent sync loops.
- Understood encoding challenges with storing multilingual text data.

Blockers:
- Simulating the linking process end-to-end for testing required setting up dummy accounts.
- Handling cases where a caregiver attempts to connect to a non-existent patient ID needed graceful error handling.
- Ensuring low latency in updates for task sync involved optimizing query usage.

Week 3 – AI/ML Lead

Date: 18-08-2025 to 23-08-2025

What I Worked On: I built and integrated advanced AI features. I implemented `ChatbotActivity` with a backend `GeminiChatService` that connects to the Gemini API for conversation. I also added a multi-language detector (utilizing `LanguageDetectionHelper`) so the chatbot can switch its model based on the patient's language (Hindi, Tamil, etc.). I developed `MemoryExtractionService` to analyze the conversation text in real time, extracting memory-related keywords for cognitive assessment. Additionally, I integrated Android TTS to read the chatbot's replies to the patient.

Work Summary: This week marked a significant milestone: the AI chatbot is now fully functional. I connected our app to the Gemini conversational AI API and built the chat UI (`activity_chatbot.xml`). The system automatically detects the patient's language and uses the appropriate responses. I also implemented conversation analysis that flags indicators of memory loss (e.g., forgotten names) and logs them for later review. The chatbot now communicates in voice using TextToSpeech, making it more accessible for patients who prefer listening. These features bring the application to life with intelligent, language-aware interaction.

Hours Worked: 40 hours

Key Learnings:
- Gained experience integrating a conversational AI API and handling its asynchronous responses.
- Learned advanced multilingual support: dynamic locale switching and resource management.
- Improved NLP skills in building pattern recognition (extracting memory cues from text).
- Understood the ethics of AI in healthcare by designing the chatbot to avoid giving medical advice.

Blockers:
- The Gemini API sometimes generates medical-sounding recommendations, so we must filter its output carefully.
- Certain regional dialect phrases still confuse the language detector, leading to misrouted queries.
- API rate limiting required us to implement caching for repeated questions to reduce calls.

Week 3 – Caretaker App + UX/QA Lead

Date: 18-08-2025 to 23-08-2025

What I Worked On: I began development of the caregiver app module. I implemented `PatientLinkActivity` so a caregiver can enter a patient code to connect with that patient's account. I also stubbed out `TaskListActivity` to display assigned tasks, and an `AddMedicationActivity` to schedule medications remotely. I worked on `EmergencyActivity` integration so that when the patient's app triggers an emergency, the caregiver receives a notification with location. I ensured all new screens follow our UX guidelines (consistent colors, large touch targets). Finally, I ran initial end-to-end tests of the patient-to-caregiver flows.

Work Summary: This week I laid the foundation for the caretaker app. Caregivers can now join with a patient via a secure code (handled by the new `PatientLinkActivity`). I created prototype screens for caregivers to view tasks and manage medications (`activity_task_list.xml`, etc.), linking them to the patient's data through Firebase. I also verified that the emergency protocol works in both apps.

Throughout, I prioritized clarity and accessibility in the caregiver UI. By the end, the caregiver app had its basic navigation and could talk to the patient's side through Firebase, with real data syncing in tests.

Hours Worked: 40 hours

Key Learnings:
- Learned to implement cross-app linking via shared backend (patient-to-caregiver).
- Gained experience in building CRUD UIs for remote data entry (tasks, meds).
- Improved understanding of real-time update workflows (caregiver action → patient app update).
- Enhanced ability to conduct integration testing for multi-device scenarios.

Blockers:
- Generating unique, secure patient codes and managing them proved tricky.
- Ensuring that caregiver notifications arrive even when the caregiver app is backgrounded required careful handling of Android notification settings.
- We still need to refine the test workflow (e.g., simulating a caregiver on one device, patient on another) for end-to-end QA.

Week 4 – Mobile Lead

Date: 25-08-2025 to 30-08-2025

What I Worked On: I advanced the patient-side functionality in Week 4. I enhanced `TasksActivity` to support task priorities, recurring schedules, and completion tracking. I implemented `GeofenceDefinition` using the Google Location API to create safe-zone alerts when a patient enters or exits a geofence. I upgraded `MedicationActivity` and built a new `RemindersActivity` to handle intelligent reminders (missed dose tracking and caregiver alerts). I also added a comprehensive notification framework (`NotificationService`) so that all reminders and alerts trigger local push notifications on the patient device.

Work Summary: This week I focused on critical safety and management features. The task management system was completed: patients can now set recurring tasks and mark them done, and caregivers see updates in real time. I successfully added geofencing: the app now monitors the patient's location and can trigger alerts for out-of-bounds events. The medication reminder system was improved with automatic rescheduling and real-time caregiver notifications. I implemented the notification channels and services needed for reminders, alerts, and emergencies. Overall, these features make the patient app a reliable hub for daily care routines.

Hours Worked: 38 hours

Key Learnings:
- Mastered Android geofencing APIs and optimized battery usage with balanced location requests.
- Learned to use `AlarmManager` and `WorkManager` for precise scheduling of background tasks.
- Implemented Android notification channels (high priority for emergencies, etc.) for differentiated alerts.
- Gained experience in designing continuous foreground services for ongoing monitoring.

Blockers:
- Location accuracy indoors remains an issue; we may need supplemental sensors or Wi-Fi-based positioning.
- Android background execution limits required careful use of foreground services and user notifications.
- Too many notifications risk user fatigue; we need to refine which alerts are essential.

Week 4 – Cloud & Data Lead

Date: 25-08-2025 to 30-08-2025

What I Worked On: I scaled up our cloud synchronization and notification system this week. I wrote Firestore real-time update logic so that whenever the patient's app updates a task or medication, the caregiver's app receives the change instantly. I also set up and configured FCM (Firebase Cloud Messaging) channels for our various alerts (location alerts, reminders, emergencies). I wrote and tested the backend service that detects geofence breaches and sends the proper notifications. Finally, I reviewed and tightened the Firestore security rules related to these new features.

Work Summary: My efforts in week 4 ensured both apps stay in sync. I enabled real-time listeners on the Firestore collections used by tasks and medications. Now, patient and caregiver can see updates immediately. I also built the notification infrastructure on the cloud side: for example, if a patient misses a dose or crosses a geofence boundary, FCM delivers an alert. I ensured that our security rules allow these updates while preventing unauthorized access. The caregiver app now has remote control features that feel instantaneous thanks to these enhancements.

Hours Worked: 38 hours

Key Learnings:
- Gained proficiency with Firebase Cloud Messaging for multiple notification types (data vs notification messages).
- Improved understanding of conflict resolution strategies in real-time sync.
- Learned to optimize Firestore queries and rules for a rapidly growing data model.
- Expanded knowledge of handling geofence triggers server-side (processing location updates).

Blockers:
- Some edge cases (e.g., simultaneous edits on both apps) still cause sync delays.
- Ensuring timely delivery of FCM in low-connectivity areas is inherently unreliable.
- Monitoring costs and limits for real-time features (reads/writes, FCM usage) requires attention.

Week 4 – AI/ML Lead

Date: 25-08-2025 to 30-08-2025

What I Worked On: I continued integrating AI into the core system. I provided voice prompts for the geofencing alerts (e.g., the chatbot now warns the patient verbally if they stray from home). I also optimized the chatbot to contextually reference location (for example, saying “I see you're out for a walk” when outside). I contributed to the UI by advising on how to display cognitive data. I examined conversation transcripts after these features were added to ensure our memory extraction still works correctly, making minor adjustments to the algorithms.

Work Summary: This week I focused on context-awareness. I enhanced the chatbot so it reacts to patient location (telling them to stay safe) and so voice instructions are clear in these scenarios. I refined the language of the responses to account for new alerts. Meanwhile, I ran our analysis tools on the latest data to ensure no logic breaks; the memory extraction and sentiment analysis continue to perform as expected. These improvements help make the AI assistant more proactive and integrated with the patient's environment.

Hours Worked: 32 hours

Key Learnings:
- Learned how to integrate contextual data (location, tasks) into conversational AI responses.
- Improved my skills in crafting clear, calm voice messages for patients.
- Gained insight into testing AI algorithms within an evolving feature set.
- Understood the trade-offs of adding AI commentary (keeping it helpful, not intrusive).

Blockers:
- Generating voice prompts that sound natural and not alarmist is difficult.
- Ensuring the AI does not inadvertently reveal private location info requires caution.
- Balancing API calls for context (e.g., geofence status checks) with performance concerns.

Week 4 – Caretaker App + UX/QA Lead

Date: 25-08-2025 to 30-08-2025

What I Worked On: This week I expanded the caregiver app significantly. I implemented `TaskListActivity` to display and assign tasks remotely, and `AddMedicationActivity` so caregivers can add patient medications from their device. I created `PatientLocationActivity`, a map view where caregivers see real-time patient position and geofence overlays. I also built the notification interfaces so caregivers are immediately alerted to patient events. I continued to refine the UI, ensuring new screens follow our high-contrast, large-font design. Finally, I began writing automated tests for these new features to verify data sync.

Work Summary: Week 4 was all about enabling remote patient management. Caregivers can now assign and track tasks through the app, and they can schedule medications that sync to the patient's reminders. The geofence map allows defining safe zones and shows live location, giving caregivers peace of mind. I ensured the caregiver app's UI handles these functions intuitively and tested them across devices. The remote management flow between the two apps is now end-to-end functional, and I have a suite of QA tests drafted to catch any regressions.

Hours Worked: 32 hours

Key Learnings:
- Learned advanced Google Maps integration on Android (drawing, markers, geofences).

- Developed a stronger understanding of synchronous workflows for remote task assignment.
- Improved ability to write UI automation tests for Android activities (Espresso).
- Honed cross-app testing skills to ensure consistency between patient and caregiver views.

Blockers:
- Some geographic features (map gestures) were hard to make intuitive for caregivers.
- Ensuring consistent real-time updates on both sides required tuning sync intervals.
- Setting appropriate default geofence sizes (too small caused false alerts, too large reduced safety).

Week 5 – Mobile Lead

Date: 01-09-2025 to 06-09-2025

What I Worked On: I developed the advanced AI-driven patient features and security enhancements. I implemented `FaceRecognitionActivity` using Google ML Kit to verify identities via the camera. I also created a new `SequenceGameActivity` with multiple memory game levels and progress tracking for cognitive stimulation. I built `StoryGenerationActivity` integrating the Gemini API to generate personalized therapeutic stories. To support offline functionality, I integrated TensorFlow Lite models for on-device face recognition and cognitive scoring. Finally, I added biometric authentication (fingerprint/PIN) using Android's BiometricPrompt to secure the app.

Work Summary: Week 5 introduced powerful AI and security features. The face recognition system now allows the app to recognize familiar faces, which aids patients who may forget names. The cognitive game was enriched with new stages that adapt to the patient's performance. The therapeutic storytelling feature generates custom stories based on patient data. Critically, all this AI work is backed by on-device models (via TFLite) so it works even without internet. I also strengthened security by enabling fingerprint or PIN unlock. These enhancements make the patient app more engaging and secure.

Hours Worked: 36 hours

Key Learnings:
- Mastered Google ML Kit for face detection and how to integrate on-device models.
- Learned to design engaging cognitive game mechanics with adaptive difficulty.
- Improved prompt engineering for AI story generation with Gemini API.
- Gained proficiency with Android BiometricPrompt for secure logins.

Blockers:
- On-device ML models are large, requiring us to optimize for memory and storage.
- Tuning face recognition thresholds for elderly faces (which can change with age) was tricky.
- Ensuring the AI-generated stories remain appropriate and therapeutically useful needs oversight.

Week 5 – Cloud & Data Lead

Date: 01-09-2025 to 06-09-2025

What I Worked On: This week I bolstered the cloud side for AI and security. I set up encrypted Firestore storage for facial recognition data (face embeddings) and biometric hashes, ensuring sensitive data is never stored in plaintext. I extended our cloud functions to process the new cognitive game results and store them. I also created a secure content management system: caregivers can now upload or edit “therapeutic content” (e.g., preferred topics for story generation) which is stored and retrieved from the database. Lastly, I audited our Firebase security rules to cover these new data types and actions.

Work Summary: Week 5 was about integrating advanced features with our backend. I configured Firestore to hold the data required by the new AI features, encrypting face recognition templates and game scores. I implemented logic so that cognitive assessment results can be compiled into provider-ready reports (if needed). I also prepared the infrastructure for managing custom therapeutic content (like uploading memory cues). Security rules were reviewed to ensure that only authenticated users can access this sensitive data. The cloud now fully supports the AI and security modules with proper safeguards.

Hours Worked: 36 hours

Key Learnings:
- Learned about secure storage practices for biometrics and sensitive patient data (encryption at rest).
- Gained experience with Cloud Functions for processing and storing ML outputs.
- Became familiar with designing a simple CMS using Firebase for dynamic therapeutic content.
- Understood the additional compliance requirements when handling biometric and cognitive data.

Blockers:
- Managing encryption keys securely on the backend introduced operational overhead.
- Ensuring zero downtime during database schema updates for new content types was critical.
- We must plan for data retention policies (how long to keep medical/AI data) which is still under review.

Week 5 – AI/ML Lead

Date: 01-09-2025 to 06-09-2025

What I Worked On: I implemented several AI-powered patient therapies. I built `StoryGenerationActivity` that feeds patient preferences into Gemini to craft personalized healing stories. I also developed a therapeutic content framework: a database of prompts and variables that can be mixed and matched by the AI for realism. I integrated TensorFlow Lite for offline processing in cognitive assessments, enabling the app to adjust game difficulty based on an on-device ML model. Finally, I coded the algorithms to handle cognitive game scoring automatically on the device.

Work Summary: Week 5 delivered the heart of our AI-driven care features. The story generator now produces custom narratives using patient details (like names, past experiences) for emotional engagement. The cognitive games update dynamically: patient performance is analyzed (using TFLite) to increase or decrease challenge. This ensures the exercises remain therapeutic. The content management system allows us to maintain a library of story elements and game patterns that the AI can use. All these services are optimized to run offline, which is essential for our patient demographic.

Hours Worked: 36 hours

Key Learnings:
- Gained expertise in prompt engineering to control AI story output quality.
- Learned techniques for on-device machine learning model optimization and integration.
- Improved understanding of content management for AI (organizing prompts and templates).
- Enhanced skills in linking app state (game score) to AI-driven adjustments.

Blockers:
- Keeping AI-generated stories free of any unwanted content (e.g., triggering memories that upset) requires manual curation of templates.
- Balancing model complexity with mobile performance constraints was challenging.
- Verifying the correctness of automated cognitive scoring against clinical standards will need expert review.

Week 5 – Caretaker App + UX/QA Lead

Date: 01-09-2025 to 06-09-2025

What I Worked On: I focused on bringing the new AI features to the caregiver's perspective. I designed a dashboard where caregivers can view cognitive game results and story sessions (e.g., transcripts or recordings). I enabled biometric login for caregivers as well, so they can easily access the app. I also implemented screens for caregivers to input patient information (name, interests) into the system for personalized stories. Throughout, I ran end-to-end tests of the new features, including face-login and game interactions, to ensure the caregiver app displays all data correctly.

Work Summary: Week 5 enhanced the caregiver-side utility of our AI features. Caregivers can now see logs of the patient's cognitive games and stories, which helps in tracking patient engagement. I also made the caregiver app more secure by adding fingerprint/PIN login, mirroring the patient app's security. I updated the profile input screens so that caregivers can directly specify patient preferences that feed into the story generator. I tested these new flows extensively to ensure data syncs between the apps and that the UI remains consistent and accessible.

Hours Worked: 36 hours

Key Learnings:
- Learned to design caregiver-specific interfaces for interpreting AI-generated data.
- Improved proficiency with Android BiometricPrompt for secondary module.
- Gained insight into how caregiver input can directly influence AI outputs (UX for personalization).
- Enhanced QA skillset by testing complex, multi-component features across apps.

Blockers:
- Ensuring caregivers correctly enter data for AI personalization (e.g., framing questions) required iteration.
- Communicating the AI outcomes (e.g., story content) to caregivers in a non-technical way is still a work in progress.
- Testing biometric login on multiple devices revealed OS version differences to accommodate.

Week 6 – Mobile Lead

Date: 08-09-2025 to 13-09-2025

What I Worked On: I implemented the clinical assessment feature on the patient app. I created `MMSEActivity` which presents the digital Mini-Mental State Exam to the patient. This includes various question types (orientation, recall, etc.) with auto-scoring. I built UI layouts for the exam questions and result display. I also added a `ProgressChartActivity` using a chart library to visualize the patient's MMSE score over time. Additionally, I fine-tuned several UI elements in other activities to accommodate the new data fields (for example, adding a "Cognitive Score" display on the dashboard).

Work Summary: Week 6 was all about integrating a standardized cognitive test. Patients can now take a digitally administered MMSE within the app. The activity captures their answers, computes a score (0–30 scale), and saves it in the database. I then implemented a chart that shows how their score has changed over multiple sessions. This provides visual feedback to both patients and caregivers. The interface is designed to be as clear as possible, with each question on one screen and large input controls. These additions make the app a credible tool for monitoring cognitive function.

Hours Worked: 45 hours

Key Learnings:

- Gained experience in designing clinical survey forms on mobile (handling user inputs and validations).
- Learned to integrate a chart library (MPAndroidChart) for data visualization on Android.
- Improved my understanding of on-device ML model deployment by using TFLite for MMSE scoring to reduce computation time.
- Learned about the nuances of translating clinical procedures (MMSE) into app workflows.

Blockers:

- Implementing all branches of the MMSE logic (e.g., conditional questions) required careful flow control.
- Ensuring the scoring algorithm matches clinical calculations exactly was critical for validity.
- Making the chart readable for patients (choosing colors, scales) took iteration.

Week 6 – Cloud & Data Lead

Date: 08-09-2025 to 13-09-2025

What I Worked On: I enabled clinical report capabilities. I implemented `HealthcareReportGenerator` that formats the patient's MMSE results and chat analysis into standardized report documents (using HL7 FHIR guidelines). These reports can be exported or emailed to healthcare providers. I stored them in Firestore so caregivers can access them. I also updated the database to record longitudinal data (MMSE scores, game performance) for trend analysis. For data compliance, I ensured all clinical data is encrypted in transit and at rest, and added audit logging for report generation events.

Work Summary: This week I built the backend support for clinical integration. When the patient completes an MMSE, the app automatically generates a professional report summarizing the score and any flagged cognitive changes. I created templates (`medical_report_template.xml`) and conversion tools to produce PDF or XML outputs that adhere to medical standards. These reports are stored and can be shared via the caregiver app. I also refined our Firestore structure to hold this time-series health data securely and in an organized manner. The system now has a full pipeline from patient input to professional-quality documentation.

Hours Worked: 45 hours

Key Learnings:
- Learned the basics of HL7 FHIR and how to map app data to clinical report formats.
- Gained experience in generating templated documents (XML/PDF) on Android for interoperability.
- Understood regulatory requirements for logging sensitive health data (audit trails).
- Improved skills in designing databases for time-series health tracking.

Blockers:
- Ensuring our export formats were compatible with various hospital systems required multiple iterations.
- Handling PHI in logs and reports necessitated strict access controls in the code.
- We will need to get professional validation of the reports' medical accuracy.

Week 6 – AI/ML Lead

Date: 08-09-2025 to 13-09-2025

What I Worked On: I expanded the AI algorithms for clinical assessment. I implemented `MMSEScoring` which automatically computes and verifies the patient's score from the input answers. I enhanced `ConversationAnalyzer` to identify subtle decline cues (e.g., repetition, hesitation) by analyzing the chat logs. I also improved the multi-language content pipeline: `CulturalContentManager` now ensures that cognitive questions and story elements are appropriately localized for each language. Lastly, I developed the `CognitiveDeclineTracker` to chart and detect downward trends in the patient's scores over time.

Work Summary: This week focused on making the assessment features smart and culturally aware. I coded the logic that evaluates MMSE answers and flags any anomalies. In the chat analysis, I upgraded the NLP to spot patterns that correlate with cognitive decline. For example, if the patient repeatedly forgets certain details in conversation, the system notes it. I ensured all these tools respect cultural context by handling language-specific scoring (via `LanguageSpecificAssessment`). I also enabled visual charting of cognitive decline, so the app can highlight if scores are dropping consistently.

Hours Worked: 45 hours

Key Learnings:
- Mastered clinical algorithm implementation (digital MMSE and trend analysis).
- Learned advanced NLP techniques to quantify language changes over time.
- Improved understanding of cross-cultural differences in assessments (adjusting content for context).
- Developed ability to design algorithms for longitudinal health tracking and alerting.

Blockers:
- Distinguishing between normal variance and significant decline in scores can be subtle and might require

more data.
- Risk of cultural bias: ensuring questions mean the same across languages and cultures.
- Future work needed on formally validating our AI assessments in a clinical study setting.

Week 6 – Caretaker App + UX/QA Lead

Date: 08-09-2025 to 13-09-2025

What I Worked On: I implemented caregiver views for the new assessments. I created an activity to display the patient's MMSE score and cognitive trend chart (showing scores over multiple sessions). I added a section where caregivers can download or share the clinical report PDF. I also refined the notifications: now if the patient's score drops significantly, the caregiver receives an alert. Throughout, I ensured the caregiver screens use clear terminology (avoiding jargon) and large, readable charts. I performed cross-language checks on these new UIs as well.

Work Summary: This week added important caregiver-facing analytics. Caregivers can now easily see the patient's cognitive score history in a chart and review detailed assessment reports. I also enhanced the alert system so caregivers are notified when there's a marked decline. I tested the interface in each supported language to ensure clarity. These features allow caregivers to monitor the patient's mental state and share insights with healthcare providers if necessary, all through a simple, consistent UI.

Hours Worked: 35 hours

Key Learnings:
- Learned how to present clinical data in an empathetic, user-friendly manner for non-professionals.
- Gained experience incorporating dynamic charts and sharing/export workflows in Android.
- Improved UX writing for medical contexts (simplifying terms like "MMSE" to "Cognitive Score").
- Enhanced cross-language QA to ensure medical terms translate correctly.

Blockers:
- Explaining cognitive scores in plain language to caregivers required iterative phrasing.
- Ensuring alerts are informative but not alarmist (e.g., "Significant change detected" messages).
- Coordinating chart data updates with new backend scores needed careful timing controls.

Week 7 – Mobile Lead

Date: 15-09-2025 to 20-09-2025

What I Worked On: I focused on fine-tuning the patient app to support caregiver features. I updated `EmergencyActivity` to handle edge cases (e.g., queueing the alert if connectivity is lost). I also optimized the patient UI in `MedicationActivity` and `TasksActivity` to reflect real-time updates sent by the caregiver. I improved navigation flows so if a caregiver assigns a new task, the patient app auto-refreshes that screen. I performed some UI polish (standardizing button styles and improving responsiveness) across existing activities.

Work Summary: Week 7 was about synchronizing and refining. I made sure the patient app robustly sends emergency alerts even under poor network conditions. I enabled real-time updates so that any caregiver-assigned medication or task appears immediately on the patient's device without needing a restart. I also cleaned up the UI: for example, I made notification messages consistent and adjusted layouts slightly for better touch response. Overall, these changes create a seamless experience as both apps share more data in real time.

Hours Worked: 35 hours

Key Learnings:
- Became proficient in handling edge-case logic (e.g., storing pending alerts during downtime).
- Enhanced knowledge of data binding to update the UI upon live Firestore changes.
- Learned best practices for incremental UI updates to avoid jarring the user.
Blockers:
- Ensuring perfect sync under all conditions proved difficult; network tests still uncovered rare lags.
- Some UI elements required rewrites to handle dynamic content without flickering.

Week 7 – Cloud & Data Lead

Date: 15-09-2025 to 20-09-2025

What I Worked On: I refined the synchronization rules and performance this week. I improved the Firestore security rules and real-time listeners to better handle rapid updates from both the patient and caregiver apps. I optimized our conflict resolution by introducing timestamp versioning for tasks and medications. I also audited our Firestore indexes to speed up queries for patient data retrieval. In addition, I implemented small enhancements for offline behavior, such as caching recently fetched task lists.

Work Summary: My focus in Week 7 was stability. I ensured that when multiple devices update the same record (e.g., two caregivers editing tasks), our Firestore rules resolve it predictably. I also indexed key fields so that fetching a patient's tasks, reminders, and reports is quicker. I verified that even offline edits queue and sync properly. All these improvements mean that data consistency is stronger and the app remains responsive even under heavy use.

Hours Worked: 35 hours

Key Learnings:
- Learned advanced Firestore techniques (timestamp-based conflict resolution).
- Improved my query optimization skills (creating compound indexes based on usage patterns).
- Enhanced understanding of caching strategies in Firebase.
Blockers:
- Some race conditions in updates were non-trivial to fix without redesigning parts of the model.
- Offline synchronization edge cases still exist when devices reconnect out of order.

Week 7 – AI/ML Lead

Date: 15-09-2025 to 20-09-2025

What I Worked On: This week I mostly provided maintenance for the AI modules. I tested the memory extraction and chatbot under new multi-device conditions to ensure no regressions. I also performed minor refinements: for example, I updated prompts and chat flows to account for caregiver-initiated events (like acknowledging a task). I reviewed performance logs to double-check that our neural network models are running as efficiently as before. Finally, I documented some internal API changes so the team understands the updated AI interfaces.

Work Summary: Week 7 was largely about verification and small tweaks. I made sure that our conversation analysis still works correctly after all the synchronization updates. I also tweaked the chatbot's dialogue to include new context awareness (e.g., it knows if the patient just assigned themselves a task). Performance remained stable, and I updated internal docs to reflect the final AI architecture as it stands. These minor adjustments ensure our AI features continue to operate smoothly in the integrated system.

Hours Worked: 35 hours

Key Learnings:
- Reinforced the value of continuous integration testing for AI services.
- Learned to adapt NLP outputs to new app contexts (like tying chatbot responses to tasks).
- Refined understanding of version control for models and prompts.

Blockers:
- Adjusting language models on the fly was limited without retraining.
- Confirming AI accuracy required careful manual verification.

Week 7 – Caretaker App + UX/QA Lead

Date: 15-09-2025 to 20-09-2025

What I Worked On: I enhanced the caregiver app's workflow and UI. I improved `PatientLinkActivity` by adding email/SMS notifications so patients see that a caregiver is linking. I refined `AddMedicationActivity` so that any medication scheduled by a caregiver immediately shows up in the patient's reminders. I enhanced `TaskListActivity` with better assignment controls and progress indicators (checkboxes, timestamps). I also strengthened the emergency notification integration: caregivers now receive location and status even if they weren't online at the time of the alert. Finally, I polished the UI of existing screens for clarity.

Work Summary: Week 7 was about refining the caregiver user experience. The patient linking flow now has a confirmation step and retry logic. Medication and task management are faster and more visual: caregivers can instantly see which tasks are pending and when the patient completes them. I also made sure emergency alerts are sent reliably (using a confirmation receipt). I updated the design of all caregiver screens to reflect the finalized color palette and button styles. Overall, these improvements made the app more robust and user-friendly.

Hours Worked: 35 hours

Key Learnings:
- Learned to design interactive lists (tasks) with smooth assignment and progress workflows.

- Gained experience in handling asynchronous events (e.g., linking confirmation dialogues).
- Improved understanding of user feedback mechanisms (e.g., showing timestamps of actions).
Blockers:
- Handling concurrency (two caregivers assigning the same task) required locking logic.
- Some UI enhancements (like drag-to-reorder tasks) were beyond scope, so we chose simpler alternatives.

Week 8 – Mobile Lead

Date: 22-09-2025 to 27-09-2025

What I Worked On: I continued building advanced features. I expanded our geofencing logic with an `AdvancedGeofenceManager` class that supports multiple safe zones and adaptable alert thresholds (e.g., ignore brief exits). I improved voice interaction in the patient app: users can now use voice commands (via `VoiceCommandProcessor`) to navigate common actions hands-free (e.g., saying “Play game” launches the cognitive exercises). I also assisted in integrating the data export feature by adding a preview dialog in the app for reports. Throughout, I made sure these features respect the existing UI conventions.

Work Summary: Week 8 involved adding complex enhancements to core features. The geofencing was made smarter (for example, distinguishing indoor vs outdoor movement patterns). I implemented an overlay for voice commands that listens for key phrases to trigger actions, improving accessibility. I also played a role in the backend data export by implementing a local interface to initiate and display exported reports. These additions make the patient app more intuitive: it can now be controlled by voice, and emergency alerts are more reliable due to smarter location management.

Hours Worked: 37 hours

Key Learnings:
- Gained experience in advanced Android geofencing scenarios (handling multiple zone shapes).
- Mastered the Android voice interaction APIs for enabling hands-free navigation.
- Learned about integrating file export workflows into the Android UI.
Blockers:
- Balancing the sensitivity of voice commands to avoid false positives was tricky.
- Testing geofence logic in different environments (indoors vs outdoors) revealed gaps that required iterative fixes.

Week 8 – Cloud & Data Lead

Date: 22-09-2025 to 27-09-2025

What I Worked On: I implemented the comprehensive data export system. I developed `DataExportHelper` to compile patient records, assessments, and history into formats (JSON, CSV, and FHIR-compliant XML). I integrated hospital-friendly report templates (e.g., `medical_report_template.xml`) and coding to convert our data into HL7 FHIR-compatible output. I also updated the backend to store export logs so caregivers can access past exports. Additionally, I ensured that exported data is anonymized where required and that PHI fields are flagged for compliance.

Work Summary: Week 8 added an important bridge between our app and the healthcare ecosystem. I built functionality to bundle all patient data (chat logs, game results, medical reports) into standardized files. Caregivers can now generate and download these reports directly from the app. I utilized templating to create professional-looking medical documents. Importantly, I enforced privacy measures in the export process: sensitive identifiers are either encrypted or redacted according to policy. With this, our application can integrate with external health records and research workflows.

Hours Worked: 37 hours

Key Learnings:
- Became proficient in data serialization and templating for healthcare formats (HL7 FHIR, CSV, JSON).
- Learned how to structure export pipelines to handle complex data sets efficiently.
- Improved skills in data privacy (anonymization, encryption) during export.

Blockers:
- Ensuring compatibility with various hospital software (each may expect slightly different FHIR fields) was challenging.
- Large export sizes on-device required implementing streaming and progress indicators.

Week 8 – AI/ML Lead

Date: 22-09-2025 to 27-09-2025

What I Worked On: This week I focused on advanced voice and data features. I enhanced the chatbot's speech recognition by integrating noise suppression and fine-tuning the microphone input parameters, improving accuracy for elderly voices. I developed the `VoiceCommandProcessor` module to parse spoken navigation commands on the fly. I also assisted in generating the medical report narrative, refining how the AI describes patient data (to be concise and clear). Lastly, I added neural network-based smoothing in the geofencing alerts to reduce false positives.

Work Summary: Week 8 was about making voice and alerts intelligent. I improved the speech-to-text pipeline so the chatbot better understands hesitant or slurred speech. I implemented a system that listens for key phrases ("remind me", "set timer") and triggers app functions. In report generation, I ensured the language model's output was formatted sensibly for medical contexts. On the location front, I applied a simple ML filter to geofence triggers to avoid false alarms due to GPS noise. These optimizations enhance reliability of voice and AI features.

Hours Worked: 37 hours

Key Learnings:
- Gained experience in noise reduction techniques and customizing speech models for specific demographics.
- Learned to implement a voice-triggered command framework on Android.
- Improved my skills in natural language generation to create professional medical text.

Blockers:
- Voice recognition still struggles with very soft or incoherent speech (limit of current tech).

- Ensuring the voice command system doesn’t interfere with normal speech in chat required careful toggling.

Week 8 – Caretaker App + UX/QA Lead

Date: 22-09-2025 to 27-09-2025

What I Worked On: I built the caregiver interface for the new features. I implemented `GeofenceManagementActivity` with an interactive map so caregivers can define, modify, and monitor safe zones for the patient. I added voice navigation aids (text hints) to caregiver activities so they can use voice commands too. I also incorporated the medical report generation: caregivers can now view and download HL7-formatted patient summaries. Finally, I performed QA to ensure all caregiver UIs (especially the map and export screens) worked smoothly and were accessible.

Work Summary: Week 8 completed the caregiver’s toolkit for advanced monitoring. Caregivers can draw or edit geofences on a map and see status updates in real time. I made sure that voice assistance (e.g., button hints) is available on these new screens. I implemented the `ReportActivity` so caregivers can initiate and review patient exports directly. I conducted thorough testing, confirming that each feature is intuitive and error-free. The caregiver app is now fully equipped to manage the patient’s environment and data, just like we envisioned.

Hours Worked: 37 hours

Key Learnings:
- Learned to implement advanced map interactions (drawing polygons, list of zones) in Android.
- Developed proficiency in voice-guided UI prompts to assist users.
- Gained experience ensuring complex features remain user-friendly for caregivers.
Blockers:
- Ensuring map editing works well with all screen sizes (especially for caregivers with vision issues) was challenging.
- The export function sometimes required repeated taps to start (a minor UI fix was needed).

Week 9 – Mobile Lead

Date: 29-09-2025 to 04-10-2025

What I Worked On: I spent week 9 optimizing performance and polishing features. I refactored several activities (e.g., `ChatbotActivity`, `FaceRecognitionActivity`) to improve memory management, which reduced our app’s footprint by about 25%. I improved the face recognition processing speed by using a smaller ML model and optimized camera capture. I enhanced the cognitive games (`SequenceGameActivity`) by adding more difficulty levels and smoother animations. I also optimized Room database queries across all entities, adding necessary indexes for faster data retrieval. Lastly, I refined the UI layouts and navigation flows for better responsiveness on older devices.

Work Summary: The focus this week was on refinement. I optimized the codebase so the app runs smoothly even on low-end smartphones. Face recognition now processes faster, and games respond instantly to touch. Database operations (loading tasks, reminders) execute much quicker. The result is a noticeably snappier app: screens load faster, transitions are fluid, and animations are smoother. I also made minor UI tweaks (alignments, button feedback) to polish the user experience. The app is now more robust and ready for final release.

Hours Worked: 32 hours

Key Learnings:
- Mastered Android profiling tools (Memory Profiler, CPU Profiler) for identifying bottlenecks.
- Learned techniques for optimizing ML Kit and TFLite performance on mobile.
- Improved SQL query optimization with proper indexing and avoiding expensive calls.
- Refined UI responsiveness through layout hierarchy simplification and view recycling.

Blockers:
- Performance varied greatly across devices; what's fast on a new phone still lagged on very old hardware.
- Reducing resource usage (especially ML model sizes) sometimes required trade-offs in accuracy.

Week 9 – Cloud & Data Lead

Date: 29-09-2025 to 04-10-2025

What I Worked On: I optimized the database and backend this week. I enabled offline persistence caching more aggressively to improve perceived performance. I also reviewed our Firestore indexes and created new ones to speed up common queries (e.g., fetching a patient's latest assessments). I implemented query optimizations in the app code (limiting fields returned) to reduce data transfer. Additionally, I enabled Firebase Realtime Database for small, instant updates (like geofence triggers) where Firestore's latency was an issue. These changes reduced sync delays noticeably.

Work Summary: Week 9's work was about backend tuning. By refining our Firestore usage, the app's data operations became faster. We now cache frequently used data so minor network hiccups don't interrupt the experience. I also introduced a hybrid approach using Firebase Realtime Database for super-fast, lightweight data (like emergency pings). Overall, data retrieval and sync are much snappier. This complements the mobile optimizations to ensure the user experience is uniformly fast.

Hours Worked: 32 hours

Key Learnings:
- Understood the trade-offs between Firestore and Realtime DB for different use cases.
- Gained experience in optimizing network payloads (using `select()` to fetch only needed fields).
- Learned strategies for offline caching to make the app resilient.

Blockers:
- Some complex data joins are still slow on Firestore (we avoided joins by denormalizing data).
- Ensuring real-time updates (e.g., geofence alerts) didn't overload FCM was a balancing act.

Week 9 – AI/ML Lead

Date: 29-09-2025 to 04-10-2025

What I Worked On: I continued refining AI features for performance and sensitivity. I added more difficulty levels to the cognitive games based on patient age/preferences (making them adaptive). I improved `StoryGenerationActivity` by tweaking the AI prompts to produce more culturally appropriate narratives (added local references). I also further tuned the text-to-speech voices to sound more natural when reading story and chat content. On the NLP side, I updated the language models to include new slang or medical terms gleaned from user data.

Work Summary: Week 9 was dedicated to refining the patient experience. The cognitive games now have a smooth progression from easy to hard, keeping patients engaged. The therapeutic stories have improved relevance (mentioning local festivals, for example). Voice narration was fine-tuned for clarity. The backend AI models were trained with additional vocabulary so the chatbot understands more varied inputs. These iterative improvements make the AI feel more personalized and relatable to users.

Hours Worked: 32 hours

Key Learnings:
- Learned methods to scale game difficulty based on simple user metrics.
- Gained experience in culturally localizing AI content generation.
- Improved voice synthesis tuning for clarity (adjusting pitch/speed parameters).

Blockers:
- Balancing story complexity vs familiarity (not all local references translate well across regions).
- Ensuring games remain engaging for different cognitive levels requires constant tweaking.

Week 9 – Caretaker App + UX/QA Lead

Date: 29-09-2025 to 04-10-2025

What I Worked On: I focused on polishing the caregiver app interface. I standardized all UI elements (colors, fonts, icons) across the entire app. I made sure every screen follows our design guidelines. I also improved layout performance by simplifying view hierarchies in key screens. I fixed a few minor issues discovered during usability testing (for example, improving button touch areas and adjusting label texts for clarity). Overall, I did one final pass on consistency and responsiveness.

Work Summary: Week 9 was about giving the caregiver app a professional polish. Every screen now has uniform styling and accessibility labels. I ran our design system audit to ensure no discrepancies remained. I also reviewed the app on different screen sizes to fix any layout issues. These final refinements mean the app looks cohesive and performs smoothly, providing the user with a high-quality experience.

Hours Worked: 32 hours

Key Learnings:
- Learned advanced theming and styling in Android to maintain consistency easily.
- Mastered Android UI optimization techniques (e.g., using ConstraintLayout for flatter hierarchies).
- Enhanced QA attention to minor details that affect polish (animation smoothness, typography).

Blockers:
- Achieving pixel-perfect consistency required careful cross-checking of multiple layouts.
- Testing UI on older devices to ensure all elements were visible and legible was time-consuming.

Week 10 – Mobile Lead

Date: 06-10-2025 to 11-10-2025

What I Worked On: I addressed critical bug fixes on the patient app. I resolved crashes in `ChatbotActivity`, `TasksActivity`, and `MedicationActivity` by fixing null-pointer issues and data binding errors. I improved Firebase sync robustness in these activities (handling edge cases like intermittent connectivity). I also tweaked the conversation analysis code (`GeminiChatService`) to better handle partial responses. Additionally, I updated the German chatbot resource file (and others) based on user feedback for cultural accuracy.

Work Summary: Week 10 was dedicated to stabilization. I fixed several critical bugs reported during testing, ensuring all core patient features are stable. The chatbot and task/reminder screens are now crash-free, and data sync logic has been hardened. I also made minor content updates for language packs. After these fixes, the app is much more reliable and ready for the final phase.

Hours Worked: 30 hours

Key Learnings:
- Enhanced my debugging skills, especially for multi-module Android apps.
- Learned about systematic internationalization testing to catch translation issues.
- Gained insight into improving error handling for offline scenarios.

Blockers:
- Some bugs were device-specific and hard to reproduce (e.g., on older Android versions).
- Balancing quick fixes versus longer refactors was challenging under time pressure.

Week 10 – Cloud & Data Lead

Date: 06-10-2025 to 11-10-2025

What I Worked On: I improved the reliability of the cloud services. I addressed data consistency issues in Firebase: for instance, I implemented retry logic for failed writes and optimized our offline cache settings. I also monitored the real-time sync stream closely, resolving any dropped updates. Additionally, I refined our Firestore security rules to close any loopholes discovered during testing. I reviewed API usage for the Gemini service, implementing better throttling to avoid rate-limit errors.

Work Summary: During week 10 I ensured our backend could withstand real-world use. I added logic to gracefully handle network interruptions in our database sync. I also monitored our API quotas to ensure the chatbot would not be suddenly blocked, and optimized our use of the Gemini API (e.g., reducing redundant calls). After these improvements, both data sync and AI integration are much more stable under stress.

Hours Worked: 30 hours

Key Learnings:
- Learned advanced patterns for handling intermittent connectivity in Firebase (automatic retries, offline persistence tuning).
- Improved understanding of API rate limiting strategies and how to manage them.
- Gained experience in live monitoring of cloud services for reliability.

Blockers:
- Some network errors (like mid-sync disconnects) were hard to simulate and fully test.
- Ensuring zero downtime on backend changes meant coordinating updates during off-hours.

Week 10 – AI/ML Lead

Date: 06-10-2025 to 11-10-2025

What I Worked On: I refined the language processing and content accuracy. I updated the conversation analysis algorithms to improve their accuracy (tuning parameters to reduce false triggers). I also reviewed and updated the multilingual content: e.g., I rewrote several Tamil prompts for clarity. On the geofencing side, I fixed issues that were causing notifications too frequently and fine-tuned the thresholds. I also expanded our test suite for the chatbot service to automate detection of obvious logic errors.

Work Summary: Week 10 was about polishing our AI pipelines. The memory extraction and NLU components now have higher precision. We updated the multilingual resource files for better cultural sensitivity in each language. The geofence system became more reliable with reduced false alerts. I ran automated tests on these modules, catching a few edge-case bugs. These refinements ensure that our AI features function smoothly and accurately.

Hours Worked: 30 hours

Key Learnings:
- Learned advanced parameter tuning for NLP to balance precision and recall.
- Gained experience in localizing content (revising translations for nuance).
- Improved skills in writing automated tests for AI algorithms.

Blockers:
- Subtle language issues in regional dialects still require manual oversight.
- Overfitting the AI to our limited test data can potentially reduce its generalizability.

Week 10 – Caretaker App + UX/QA Lead

Date: 06-10-2025 to 11-10-2025

What I Worked On: I led the final testing phase. I performed comprehensive regression tests on the caregiver app across multiple Android versions. I documented and helped fix remaining bugs (e.g., ensuring chat and emergency alerts sync correctly). I also tested cross-module interactions, like verifying that a new task added by a caregiver appears properly on the patient side. I updated our QA checklist and confirmed that each item passes. The result is an app ready for production with no known critical issues.

Work Summary: In week 10, all efforts were focused on quality assurance. I executed a full round of manual and automated tests across both apps. All identified issues were tracked and fixed in collaboration with the team. We achieved stability: tasks, meds, geofence, and communication features now work reliably in all test scenarios. With these final validations complete, the project is ready to move on to launch preparations.

Hours Worked: 30 hours

Key Learnings:
- Mastered end-to-end testing strategies for multi-app workflows.
- Learned to prioritize bug fixes based on user impact.
- Strengthened skills in writing clear QA documentation and reports.

Blockers:
- Testing on a wide range of devices was time-consuming but necessary.
- Some rare race-condition bugs may still emerge under extreme scenarios (to be monitored post-launch).

Week 11 – Mobile Lead

Date: 13-10-2025 to 18-10-2025

What I Worked On: I focused on code quality and maintainability. I added detailed logging across all patient-side activities to track user behavior and app state (using Android’s `Log` and a `Logger` utility). I wrote comprehensive JavaDoc for all major classes and methods in the patient app. I standardized UI elements by applying a common style library (colors, typography, button styles) so every screen looks cohesive. I also refactored parts of the codebase: removed deprecated code, organized imports, and cleaned up the package structure for clarity.

Work Summary: Week 11 was dedicated to documentation and cleanup. The patient app now has a robust logging system, which will be invaluable for diagnosing issues later. Every class and function has descriptive comments, which will help future developers. The UI theme was applied uniformly, making the interface more polished. These changes do not affect functionality but significantly improve the readability and maintainability of the code. The app is now well-documented and organized.

Hours Worked: 28 hours

Key Learnings:
- Learned best practices in logging without impacting performance or leaking sensitive data.
- Improved my technical writing skills by creating clear, useful JavaDoc.
- Understood the importance of a unified design system for brand consistency.

Blockers:
- Writing thorough documentation for all classes was time-intensive.
- Ensuring logging captures useful context without being too verbose required careful planning.

Week 11 – Cloud & Data Lead

Date: 13-10-2025 to 18-10-2025

What I Worked On: I improved cloud code quality and documentation. I instrumented detailed logging for all cloud functions and database operations, using Stackdriver, to monitor runtime behavior. I created documentation for the Firebase setup, including an `API_Documentation.md` and `Architecture_Overview.md`, so that new team members can easily understand our backend. I also cleaned up the Firebase project: removing unused references, organizing rules, and archiving obsolete test data. Finally, I documented all build and environment configurations (API keys, service accounts).

Work Summary: This week was about making our backend maintainable. All cloud services now log extensively, aiding in future debugging and auditing. The project's Firebase settings and data model are fully documented, which will simplify project handovers. I also standardized the naming of collections and rules. With these updates, our cloud infrastructure is not only robust but also transparent and well-documented.

Hours Worked: 28 hours

Key Learnings:
- Learned how to set up and use Stackdriver logging and monitoring effectively.
- Improved skills in writing clear API and architecture documentation.
- Gained experience in organization and naming conventions for cloud projects.

Blockers:
- Ensuring none of the documentation includes sensitive information was a careful process.
- Keeping documentation up to date with last-minute changes required coordination with the team.

Week 11 – AI/ML Lead

Date: 13-10-2025 to 18-10-2025

What I Worked On: I devoted this week to documenting the AI components. I added JavaDoc and inline comments to all AI/NLP classes (chat service, analysis modules, ML utilities). I created architecture diagrams and a markdown overview (`Architecture_Overview.md`) that map out data flow between these components. I standardized code style and cleaned up the AI module structure (renaming classes for consistency). Additionally, I organized our Python/ML script repository and updated the requirements. This ensures every aspect of our AI system is well-documented.

Work Summary: Week 11 improved the transparency of our AI code. Every class in the AI package now has detailed comments and usage examples. The architecture doc illustrates how the chatbot, analysis engine, and ML models work together. I also wrote an `API_Documentation.md` for any server-side AI APIs. These deliverables will help anyone (developers or auditors) understand the AI/ML pipelines. The codebase is now more maintainable and easier for team members to navigate.

Hours Worked: 28 hours

Key Learnings:
- Learned to express complex AI logic in clear documentation.
- Enhanced ability to create system diagrams that clarify software structure.
- Recognized the importance of aligning code and documentation for collaboration.

Blockers:
- Documenting every edge-case in the AI logic was lengthy and required in-depth knowledge.
- Some algorithms were proprietary or complex, making them hard to explain succinctly.

Week 11 – Caretaker App + UX/QA Lead

Date: 13-10-2025 to 18-10-2025

What I Worked On: I standardized the caregiver app's design and documentation. I applied consistent UI themes (colors, styles) across all screens. I implemented logging for user actions and errors within the caregiver module. I also wrote a setup guide and component documentation (`Component_Relationships.md`) explaining how each screen connects. I cleaned up the code by removing unused resources and organizing the layout files. These tasks finalize the app's readiness for maintenance and handover.

Work Summary: In week 11, I polished the caregiver app's presentation and internal documentation. The interface now fully adheres to our design guidelines, making navigation intuitive. I added logs in critical screens to track app usage. I documented the overall architecture, including API endpoints and data flow, so that our knowledge is not just in our heads. Finally, I refactored the project structure for clarity. This comprehensive cleanup ensures the app is professional and maintainable.

Hours Worked: 28 hours

Key Learnings:
- Learned the value of a unified design language across multiple activities.
- Improved skills in organizing project resources for long-term maintainability.
- Understood best practices for writing cohesive module documentation.

Blockers:
- Consolidating documentation from multiple developers into one cohesive guide took significant effort.

- Ensuring no functionality was inadvertently removed while cleaning up resources required thorough testing.

Week 12 – Mobile Lead

Date: 20-10-2025 to 25-10-2025

What I Worked On: I performed a final audit of the patient app features. I systematically tested each implemented feature (chatbot, face recognition, games, tasks, etc.) to verify completeness and find any hidden bugs. I identified and documented over 15 bugs and issues (such as layout glitches and edge-case crashes). I also reviewed the code for potential optimizations and marked several areas for improvement. Finally, I established performance baselines by measuring current app launch time and memory usage on benchmark devices.

Work Summary: In this final refinement week, I meticulously reviewed all patient app functionality. Every feature was tested end-to-end, and any missing behaviors were noted. I cataloged all discovered bugs and prioritized them. I also assessed the code structure for any remaining inefficiencies. I recorded performance metrics (like average task list load time) so that our final tweaks can be measured against them. This sets the stage for the last round of fixes in Week 13.

Hours Worked: 32 hours

Key Learnings:
- Learned systematic testing and feature verification methods for large codebases.
- Gained insight into categorizing bugs by user impact (critical vs minor).
- Developed techniques for performance profiling and baseline establishment.

Blockers:
- Some bugs were elusive because they required very specific user interactions to reproduce.
- It was difficult to test certain AI features without real patient interactions; we did our best with simulated data.

Week 12 – Cloud & Data Lead

Date: 20-10-2025 to 25-10-2025

What I Worked On: I conducted a thorough review of the cloud backend. I audited all implemented features (authentication, database, AI integrations, etc.) and verified each against our project requirements checklist. I systematically tested for bugs and performance issues across modules. I documented interactions between patient and caregiver apps, noting any potential cross-module dependency issues. I also captured performance baselines for data-intensive operations (like syncing large data sets) to use in final optimizations.

Work Summary: Week 12 was a full system audit week. I mapped out the entire system’s features and confirmed each is implemented. I ran through data sync scenarios (creating and editing records simultaneously) and noted any failures. I created a prioritized list of bugs (over 15 found) and outlined fixes.

I also documented the cross-app data flows to highlight potential integration risks. By establishing current performance metrics, I prepared our team to target specific areas for improvement in the last weeks.

Hours Worked: 32 hours

Key Learnings:
- Improved my ability to analyze system-wide dependencies and identify hidden coupling.
- Learned advanced testing approaches for distributed apps (patient ↔ caregiver).
- Enhanced understanding of performance metrics and bottleneck identification in Firebase.

Blockers:
- Some integration bugs were only apparent when multiple features interacted (e.g., chat updates during a task sync).
- Establishing meaningful performance baselines required simulating real-world usage patterns.

Week 12 – AI/ML Lead

Date: 20-10-2025 to 25-10-2025

What I Worked On: I performed a comprehensive review of all AI/ML components. I verified feature completeness for the chatbot, memory extraction, translation services, story generation, and cognitive games. I systematically tested edge cases, identifying over a dozen issues (e.g., memory leaks in analysis service, incorrect localization strings). I assessed the code quality and flagged sections for optimization. I also measured current performance (API call latency, on-device inference time) to establish baselines.

Work Summary: Week 12 was the final QA pass for the AI engine. I audited every ML feature, confirming that it is present and works as intended. I compiled a detailed bug report (15+ issues) covering functional and performance bugs. By documenting the findings, I created a clear action plan for final bug fixes. I also collected metrics on AI response times and resource usage to guide upcoming optimizations. This thorough review positions us to confidently finalize the system.

Hours Worked: 32 hours

Key Learnings:
- Enhanced techniques for validating AI outputs against expected behavior.
- Learned to categorize AI/ML issues (model vs. code vs. data problems).
- Gained experience in tracking performance of cloud AI services and on-device inference.

Blockers:
- Fully simulating human conversation and cognition for AI testing is inherently limited without real subjects.
- Some issues (like dialect misrecognition) may persist beyond initial fixes without new data.

Week 12 – Caretaker App + UX/QA Lead

Date: 20-10-2025 to 25-10-2025

What I Worked On: I led the final feature audit on the caregiver side. I checked each feature's completeness against our original list (patient linking, task management, emergency alerts, reports, etc.), and identified gaps. I cataloged all bugs found during cross-app testing (over 15 issues ranging from UI glitches to data mismatches). I established a performance baseline by measuring load times and memory usage in the caregiver app. This thorough review ensures we know exactly what needs to be addressed in the final iteration.

Work Summary: Week 12 wrapped up the feature verification for the caregiver app. I walked through every use case and documented the results. This produced a prioritized list of remaining bugs and improvements. I also noted the app's current performance metrics for the final optimization push. By creating this improvement roadmap, I have given the team clear targets for ensuring the caregiver app is fully functional and performant by the end of the project.

Hours Worked: 32 hours

Key Learnings:
- Further refined QA methodologies for ensuring feature parity between interconnected apps.
- Learned to create comprehensive bug reports with clear reproduction steps.
- Improved ability to set objective performance goals for mobile apps.

Blockers:
- Some issues span multiple modules (patient and caregiver), requiring careful coordination to fix.
- Limited time remains to verify fixes, so we must prioritize effectively.

Week 13 – Mobile Lead

Date: 27-10-2025 to 01-11-2025

What I Worked On: I addressed the critical bugs identified in Week 12 for the patient app. I fixed several memory leaks by properly disposing of TensorFlow models in `ChatbotActivity` and releasing camera resources in `FaceRecognitionActivity`. I resolved database update conflicts in `MedicationActivity` and `TasksActivity` by implementing optimistic locking and retry logic. I optimized `GeminiChatService` by adding response caching to reduce repeat API calls. I also enhanced `SequenceGameActivity` touch handling to eliminate lag.

Work Summary: Week 13 was about fixing high-priority issues. I eliminated the major memory leaks and ensured GPU/CPU resources are freed when not needed. I fixed sync conflicts so that caregiver edits no longer overwrite patient data erroneously. I improved the AI service's efficiency by caching repeated requests (cutting response time by ~40%). The cognitive games now respond instantly without stutter. These changes significantly improve the app's stability and performance just in time for launch.

Hours Worked: 38 hours

Key Learnings:
- Mastered memory profiling and leak detection tools on Android (LeakCanary).

- Learned advanced synchronization techniques (optimistic locking in Firestore transactions).
- Enhanced knowledge of caching strategies for network-based AI calls.

Blockers:
- Some fixes required deep changes to data flow, risking subtle side-effects.
- Testing memory leaks in all scenarios (e.g., switching apps mid-chat) took considerable effort.

Week 13 – Cloud & Data Lead

Date: 27-10-2025 to 01-11-2025

What I Worked On: I resolved the remaining synchronization issues this week. I implemented optimistic locking in Firestore transactions for medication and task updates to prevent conflicts. I also addressed edge cases where updates were lost due to simultaneous writes from patient and caregiver. I optimized database queries further, adding indices to `PatientDao` and related tables to speed up common operations. Additionally, I continued monitoring API usage, ensuring our caching solution is effective and does not exceed quotas.

Work Summary: In Week 13 I finished the last technical fixes for data consistency. The app now reliably syncs even when both users update the same record concurrently. I also improved the underlying data queries on the cloud side for efficiency. My final task was performance tuning: every database call and cloud write was profiled and optimized. These efforts mean that data sharing between patient and caregiver is now bulletproof and snappy, setting the stage for deployment.

Hours Worked: 38 hours

Key Learnings:
- Gained deeper understanding of Firestore transactions and consistency models.
- Learned to optimize query performance in both Firestore and Room ORM.
- Fine-tuned strategies for staying within API quotas (cache invalidation policies).

Blockers:
- Handling edge-case conflicts (e.g., simultaneous deletions) required manual conflict resolution logic.
- Monitoring production-level usage and performance is limited until post-launch.

Week 13 – AI/ML Lead

Date: 27-10-2025 to 01-11-2025

What I Worked On: I optimized the AI services. I resolved 8 critical memory leaks in AI-heavy activities by ensuring proper cleanup of ML and audio resources. I fixed 5 major synchronization conflicts in the data flow between patient and caregiver (e.g., finalizing data from `AddMedicationActivity`). I improved the chatbot's performance by adding a caching layer in `GeminiChatService`, which reduced average response time by ~40%. I also enhanced `MemoryExtractionService` efficiency by streamlining text processing. These changes significantly reduced resource use and made the AI features faster.

Work Summary: Week 13 focused on the most urgent bug fixes. Memory management in the chatbot, face recognition, and story generation activities is now robust – no more leaks under testing. Data synchronization between modules is now consistent after implementing conflict resolution. The AI response pipeline is much faster thanks to caching and code optimizations. Overall, the app's performance and stability have improved dramatically, ensuring the AI features will work reliably for end-users.

Hours Worked: 38 hours

Key Learnings:
- Advanced memory profiling on Android for AI use-cases.
- Learned to implement optimistic locking patterns in data synchronization.
- Understood the trade-offs in caching AI API responses (stale vs. speed).

Blockers:
- Caching logic can lead to slightly outdated responses if patient info changed, which we accepted as a trade-off.
- The largest models still push the limits of device memory; future work could involve model compression.

Week 13 – Caretaker App + UX/QA Lead

Date: 27-10-2025 to 01-11-2025

What I Worked On: I finalized fixes on the caregiver side. I eliminated synchronization bugs that caused data to disappear (e.g., missed medication entries) by improving the update logic. I optimized UI responsiveness by reducing view overdraw and preloading common layout elements. I fixed several animation stutters and ensured all touch targets respond promptly. I also compressed and removed unused assets to shrink the app size.

Work Summary: In Week 13, I polished performance and fixed last bugs. The caregiver UI now runs smoothly, even with large data sets. All screens load and scroll without lag. I refined the rendering of lists and maps to be more efficient. With these optimizations, the caregiver app is fast and light, giving a smooth experience even on older devices. We are now ready for final quality assurance checks.

Hours Worked: 38 hours

Key Learnings:
- Learned advanced techniques for reducing Android view overdraw (flattening layouts).
- Gained experience in optimizing asset pipelines (vector drawables, 9-patch).
- Improved understanding of user-centric performance (perceived vs actual speed).

Blockers:
- Some third-party map components were hard to optimize further.
- Ensuring consistent 60fps animations required trimming some visual effects.

Week 14 – Mobile Lead

Date: 03-11-2025 to 08-11-2025

What I Worked On: In the final week, I tackled the remaining edge cases and optimizations. I improved geofencing to handle indoor/outdoor transitions more gracefully by combining GPS with Wi-Fi/cell data. I enhanced `EmergencyActivity` with better error handling: if network fails, the app now attempts SMS as a backup. I focused heavily on battery optimization: I reduced location polling frequency, paused background tasks when idle, and batched network sync to avoid excessive wake-ups. I implemented comprehensive error handling across all activities so that any failure shows a user-friendly message rather than crashing.

Work Summary: Week 14 was all about reliability. I implemented hybrid location fixes which eliminated most false geofence alerts indoors. Emergency alerts now have offline pathways, ensuring caregivers always get notified. I dramatically cut battery usage by optimizing background services and sync scheduling, extending device uptime. I also added global exception handlers so any unexpected error yields a graceful fallback. These final changes solidify the app's robustness, making it ready for real-world usage.

Hours Worked: 36 hours

Key Learnings:
- Mastered hybrid positioning strategies to improve indoor location accuracy.
- Learned sophisticated Android power management (Doze mode, JobScheduler best practices).
- Developed patterns for graceful degradation and retry logic in health-critical features.

Blockers:
- Some hardware limitations (inconsistent Wi-Fi/GPS indoors) are beyond software fixes.
- Android version fragmentation means certain optimizations behave differently across devices.

Week 14 – Cloud & Data Lead

Date: 03-11-2025 to 08-11-2025

What I Worked On: I finalized the data reliability and security improvements. I double-checked the hybrid sync logic (combining push and periodic pull) to ensure data consistency under all conditions. I optimized the backend triggers for low-power mode, making sure location updates and emergency signals are batched. I also performed a final audit of our Firebase rules and encryption to ensure all sensitive operations are covered. Finally, I prepared the final backup scripts for patient data before launch.

Work Summary: In the final week, I ensured the backend is rock-solid. I confirmed that our last-minute optimizations (like reduced sync frequency) do not break data flow. I also verified that backup and restore mechanisms are in place. Security-wise, the Firestore rules have been tightened and all data at rest is encrypted. These last steps complete our backend readiness, ensuring that patient data is safe and always available.

Hours Worked: 36 hours

Key Learnings:
- Reinforced importance of end-to-end validation for hybrid cloud-mobile systems.
- Learned about final post-deployment preparations (data backup, rollback plans).
- Solidified best practices for data security in production environments.

Blockers:
- Live testing of backups and restores is limited without a production environment.
- Ensuring all user data (especially AI logs) is cleaned up before archiving needed attention.

Week 14 – AI/ML Lead

Date: 03-11-2025 to 08-11-2025

What I Worked On: I completed the final AI enhancements. I retrained the speech recognition thresholds to better handle elderly speech patterns and medical terminology, improving recognition accuracy by ~25%. I tuned the chatbot's vocabulary filtering to avoid misinterpreting specific patient terms. I conducted final end-to-end tests of the AI features under varied conditions (poor network, background noise, etc.) to verify robustness. I also performed a last review of the AI content to ensure it is culturally appropriate and non-offensive.

Work Summary: In the last week, I gave the AI one final polish. Voice recognition is now more accurate for the patient demographic, and the chatbot is more reliable. I verified that the AI gracefully handles all final edge cases we identified. All language packs have been updated and vetted. These efforts ensure the AI functions are not only technically complete but also tuned for our users' real-world environment.

Hours Worked: 36 hours

Key Learnings:
- Achieved deeper insight into fine-tuning on-device AI for specific user speech patterns.
- Improved ability to validate AI outputs against expected norms in a clinical setting.
- Learned how small parameter adjustments can have significant impact on recognition rates.

Blockers:
- Perfect voice recognition for all elderly accents is likely unattainable; some error rate will remain.
- Last-minute changes to AI content require careful regression testing to avoid unintended consequences.

Week 14 – Caretaker App + UX/QA Lead

Date: 03-11-2025 to 08-11-2025

What I Worked On: I wrapped up any remaining caregiver-side refinements. I conducted final acceptance testing of emergency, reminder, and geofence flows to ensure end-to-end reliability. I optimized the caregiver app's battery usage (e.g., background sync interval adjustments) to achieve a 35% reduction in power draw. I implemented robust error recovery mechanisms across caregiver activities: for example, if the network is lost, the app now caches actions and syncs later. I also tuned the speech recognition prompts so caregivers can use hands-free navigation as well.

Work Summary: The final week was about bulletproofing the caregiver app. All emergency and notification scenarios have been tested and validated offline. The app now uses significantly less power when idle, which is critical for on-the-go caregivers. Error handling was implemented to cover all network failures without data loss. Voice command reliability was improved for caregiver workflows. These final touches complete our project: the caregiver app is now polished, reliable, and optimized for real-world use.

Hours Worked: 36 hours

Key Learnings:
- Deepened understanding of mobile battery optimization techniques in Android.
- Learned to implement fault-tolerant design patterns for critical features.
- Gained experience creating user support materials to complement the app.

Blockers:
- In-field testing (e.g., actual car crashes or emergencies) cannot be fully replicated, so ongoing monitoring will be necessary.
- Differences in mobile carrier networks may still cause occasional sync delays, a factor beyond our control.
