WalkTracker - One-Page App Summary

Source basis: README, AndroidManifest, app/build.gradle, and Kotlin source files in this repo.

What It Is

WalkTracker is a Kotlin/Jetpack Compose Android app for tracking walking sessions with GPS, distance, and step counts on-device. It uses a foreground location service for live tracking and stores session history locally with Room and user settings with DataStore.

Who It Is For

Primary persona: people who want a simple, privacy-focused walking tracker that works offline. Formal persona definition document: Not found in repo.

What It Does

- Starts, pauses, resumes, and stops walk sessions from the home screen UI.
- Tracks distance via a foreground GPS service with accuracy, age, and movement filtering.
- Counts steps using device step sensor when available, with distance-based fallback.
- Saves completed sessions (distance, duration, steps) to local Room database history.
- Stores preferences in DataStore (units, step length, step-sensor toggle, maps toggle).
- Provides a step-length calibration screen based on known distance and step count.
- Supports optional Google Maps-based path display; current session/history path loading is currently stubbed to empty in MainActivity.

How It Works (Repo-Evidenced Architecture)

- UI layer: MainActivity + Compose screens (Home, Settings, Calibration, Map) and navigation.
- State layer: MainViewModel coordinates UI state, service binding, Room DAOs, and DataStore preferences.
- Tracking layer: LocationService uses FusedLocationProviderClient and exposes distance/path state flows.
- Data layer: AppDb (Room) with WalkSession/WalkPath entities and WalkDao/WalkPathDao for persistence.
- Data flow: User action -> ViewModel -> LocationService + StepCounter -> UI state updates -> session/path persisted on stop.
- External services found in code: Google Play Services Location and optional Google Maps SDK.
- Cloud backend/API service implementation: Not found in repo.

How To Run (Minimal)

- Prereqs: Android Studio, Android SDK 26+, Java 17, and Google Play Services.
- Open this project in Android Studio and sync Gradle.
- Optional maps: set google_maps_key in app/src/main/res/values/strings.xml.
- Run on device/emulator and grant location permissions; CLI build option: ./gradlew assembleDebug.