Employee Tracking System

Complete Feature Specification — Mobile App (Flutter)

Feature Summary

This document covers all features of the Track Employee tracking mobile application. The system supports offline-first operation with automatic background sync, geo-tagged attendance, and a full organizational employee directory.

Module
Authentication & Onboarding
Attendance — Check In / Check Out
Dashboard & Working Hours Analytics
Organization & Employee Directory
Offline Mode & Auto Sync

Authentication & Onboarding

MODULE 01 **Authentication & Onboarding**

One-time organization setup and secure daily login

FEATURE	DESCRIPTION	PRIORITY
Organization URL Input *Onboarding / First Launch*	Employee pastes or scans a one-time invite URL to bind their account to the org. URL is validated, stored, and never asked again.	**Must Have**
Username & Password Login *Authentication*	Credential-based login after organization setup. Stored securely via Flutter Secure Storage (keychain/keystore).	**Must Have**
Persistent Session *Authentication*	Auth token saved locally so users stay logged in across app restarts. Token refreshed automatically when online.	**Must Have**
Biometric Login *Authentication*	Optional Face ID / fingerprint login after initial password setup for faster daily access.	**Nice to Have**
Logout & Session Clear *Authentication*	Secure logout that clears session tokens but retains local attendance data to ensure sync continuity.	**Must Have**

Attendance — Check In / Check Out

MODULE 02 **Attendance — Check In / Check Out**

Core attendance tracking with geo, time, and date capture on every action

FEATURE	DESCRIPTION	PRIORITY
Check In *Attendance*	Single-tap check in. Captures current time, date, GPS coordinates, and device info. Stored locally immediately, synced when online.	Must Have
Check Out *Attendance*	Single-tap check out with the same data capture. Automatically computes total worked hours from the check-in record.	Must Have
Geo Location Capture *Attendance*	GPS lat/lng captured at every check event. Falls back to last known location if GPS is unavailable. Address resolved via reverse geocoding (cached offline).	Must Have
Live Work Timer *Dashboard*	Real-time timer displayed from check-in moment, counting hours:minutes:seconds. Persists even if the app is closed or restarted.	Must Have
Break Logging *Attendance*	Optional break start/end events to track and deduct break duration from total working hours for accurate daily totals.	Should Have
Attendance History *History*	Scrollable log of all past check in/out records showing time, location, and worked hours for each day.	Must Have

Dashboard & Working Hours Analytics

MODULE 03 **Dashboard & Working Hours Analytics**

Real-time and historical summaries of employee working time and attendance

FEATURE	DESCRIPTION	PRIORITY
Daily Hours Summary *Dashboard*	Total worked hours for today. Breakdown of time-in, time-out, and break durations in a clear visual format.	Must Have
Weekly Hours Chart *Dashboard*	Bar chart showing hours worked each day of the current week. Highlights overtime days and missed check-outs.	Should Have
Monthly Attendance Score *Dashboard*	Attendance percentage, total days present, days absent, and days on leave for the current month.	Should Have
Today's Activity Timeline *Dashboard*	Chronological activity view showing check-in, breaks, and check-out with timestamps and geo-tagged locations.	Must Have

Organization & Employee Directory

MODULE 04 Organization & Employee Directory

Company details, employee listings, profiles, and reporting hierarchy — fully offline

FEATURE	DESCRIPTION	PRIORITY
Organization Details *Organization*	Company name, logo, industry, registered address, total headcount, and department listing.	Must Have
Employee Directory *Organization*	Full list of all employees with avatar, name, position, and online/offline status. Searchable and filterable by department.	Must Have
Employee Profile Card *Organization*	Detailed profile view: profile photo, full name, job title, department, email address, phone number, and join date.	Must Have
Reporting Manager Details *Organization*	Each employee profile shows their direct reporting manager with name, photo, title, and contact. Tap to navigate to the manager's own profile.	Must Have
Department Filter *Organization*	Filter the employee directory by department or team using tab-based or chip-based UI for fast navigation.	Should Have

Offline Mode & Auto Sync

MODULE 05 Offline Mode & Auto Sync

All critical features work without internet. Syncs automatically when connectivity is restored

FEATURE	DESCRIPTION	PRIORITY
Local SQLite / Hive Storage *Offline*	All attendance records, employee data, and org info stored in on-device database. App is fully functional without any network connection.	Must Have
Sync Queue *Sync*	Actions performed offline are queued in order. When internet is detected, the queue is processed sequentially to avoid conflicts.	Must Have
Auto Background Sync *Sync*	Connectivity listener triggers sync automatically when the device comes back online. No manual action required from the user.	Must Have
Network Status Indicator *Sync*	Persistent visual indicator showing online/offline state and the number of pending items in the sync queue.	Should Have

Recommended Flutter Packages

The following Flutter pub.dev packages are recommended for implementing all features in this specification.

PACKAGE	PURPOSE
sqflite	Local SQLite database for all on-device structured data storage
connectivity_plus	Network status stream — triggers auto-sync when back online
geolocator	GPS location capture with platform permission handling
flutter_secure_storage	Keychain / keystore for auth tokens and credentials
local_auth	Biometric authentication — Face ID and fingerprint support
geocoding	Reverse geocoding: converts GPS coordinates to human-readable address
hive / hive_flutter	Optional fast NoSQL local store for caching employee/org data
dio	HTTP client for API calls with interceptors for auth token injection

Offline & Sync Architecture

Write-Locally-First Strategy

Step 1 — Immediate Local Write

Every user action (check in, check out, break start/end) is immediately written to the local SQLite/Hive database. The UI responds instantly with no waiting for any network request.

Step 2 — Queue When Offline

If the device is offline at the time of action, the record is added to a SyncQueue table with status 'pending'. Each entry stores: entity type, entity ID, action (CREATE/UPDATE), JSON payload, timestamp, and retry count.

Step 3 — Auto Background Sync

A connectivity_plus listener monitors network state. When the device reconnects, the queue processor runs automatically in the background — no user action needed. Items are sent to the server in chronological order.

Step 4 — Conflict Resolution

Server-wins is applied to organization and employee data (master data controlled by admin). Client-wins is applied to attendance records — employees own their own check-in/out data. Failed sync attempts increment a retry counter; after 3 failures the item is flagged for manual review.