

Field Employee Visit Logger with Evidence, Role Hierarchy & Smart Background Sync

Android (Kotlin)

Objective

An **offline-first field workforce app** where:

- Manager manages **Sites, Supervisors, Employees**
 - Employees can log visits **only within 100m of site**
 - Evidence files attached using Scoped Storage
 - Data saved locally (Room) and synced to Firebase automatically
 - Position-wise verification & notifications
 - Full background sync using WorkManager + Foreground Service + BroadcastReceiver
 - App works **offline and online seamlessly**
-

Roles & Hierarchy (Firebase Auth + Firestore)

All users login using **Google Sign-In**.

Role	Capabilities
Manager	Create Sites, Supervisors, Employees, Assign mappings
Supervisor	View assigned employees, verify their visits

Employee Log site visits with evidence

Firestore holds:

```
users/  
  userId:  
    email  
    name  
    profilePic  
  role (manager/supervisor/employee)  
  assigned_employees: []    // for supervisor  
  my_supervisor: userId    // for employee
```



Site Management (Manager CRUD)

Manager can create/edit/delete sites.

Each site contains:

```
sites/  
  siteId:  
    siteName  
    latitude  
    longitude  
    workingDetails
```



Screens to Build (Role-wise)



Common Screens

1. Splash / Session check (DataStore)
2. Google Login

-
- 3. Sync Overlay (global component)
 - 4. Settings
-

Manager Screens

- 1. Manager Dashboard
 - 2. Site List
 - 3. Add/Edit Site (map location picker)
 - 4. Supervisor List
 - 5. Add/Edit Supervisor
 - 6. Employee List
 - 7. Add/Edit Employee
 - 8. Assign Employees to Supervisor (mapping screen)
 - 9. All Visits Monitoring Screen
 - 10. Visit Detail Viewer
-

Supervisor Screens

- 1. Supervisor Dashboard
- 2. Assigned Employee List
- 3. Employee Visit List
- 4. Visit Detail with Evidence
- 5. Approve / Reject Visit

Employee Screens

1. Employee Dashboard
 2. Start Visit (Geofence validation)
 3. Visit Form + File Attach
 4. Visit History
 5. Pending Sync Queue
-



100 Meter Geofence Validation (Employee)

When employee taps **Start Visit**:

1. Fetch site lat/long from Firestore
 2. Get current GPS location
 3. Allow only if within **100 meters**
 4. Else block visit
-



Local Storage Responsibilities

Data	Storage
Visits	Room
Evidence files	Scoped Storage
Session, role, last sync	DataStore

Sync Architecture

Component	Responsibility
Coroutines	Thread handling
WorkManager	Periodic + retry sync
Foreground Service	Show syncing notification
BroadcastReceiver	Network change, reboot trigger

Sync order:

1. Upload evidence → Firebase Storage
 2. Upload visit → Firestore
 3. Update Room status = Synced
-



Position-wise Notification (Realtime DB + FCM)

When employee logs visit:

1. Fetch supervisor device tokens from **Realtime DB**
 2. Send FCM notification
 3. Supervisor opens → Verification screen
-



Verification System (Supervisor)

Supervisor can:

- View files
- Approve / Reject
- Add remarks

Status synced back to Firestore → Room

Online / Offline Handling

Case	Behavior
No internet	Save in Room
Internet back	Auto sync via WorkManager
App killed	Sync continues
Phone reboot	BOOT receiver triggers sync

Firebase Usage

Feature	Usage
Firebase Auth	Google login
Firestore	Users, Sites, Visits
Realtime DB	Device tokens
Firebase Storage	Evidence files

Full Flow Summary

Manager Flow

Login → Add Sites → Add Supervisors → Add Employees → Assign Employees → Monitor Visits

Supervisor Flow

Login → See employees → Get notified → Verify visits

Employee Flow

Login → Reach site → Pass geofence → Log visit → Attach files → Save → Auto sync

Edge Cases Must Solve

- GPS off / permission denied
- File URI persistence after reboot
- Duplicate sync prevention
- Network fluctuation during upload
- Role-based screen restrictions
- Offline app usage for hours then sync

Evaluation Checklist

Must demonstrate:

- Site CRUD by manager

- Role hierarchy working
 - Employee restricted by 100m rule
 - Evidence attach & preview
 - Offline visit save
 - Background auto sync without opening app
 - Correct notification to supervisor
 - Verification updates reflected everywhere
 - DataStore reactive session handling
 - Overlay syncing UI
-