PROJECT ABSTRACT

Offline-First Field Survey Data Collection App

An offline-first Android app for field surveys: GPS-tagged forms, photo capture, on-device validation, and a sync queue with retry, resume and conflict flagging. Delivered built-to-order with Kotlin app source, sync engine, reference backend, report, PPT and viva kit.

Title

Offline-First Field Survey Data Collection App — a mobile data-collection system designed for zero connectivity, where the device is the database of record until the network returns.

Introduction / Background

Field surveys happen where the network does not: farms, construction sites, remote villages. Paper forms get lost and retyped; most mobile form tools assume connectivity and fail gracelessly without it. The hard problems are offline-shaped: records must queue durably on the device, photos must survive app restarts, uploads must resume rather than restart, and when two people edit the same record the system must surface the conflict instead of picking a winner silently. This project builds an Android app around those problems: a form library with per-form progress, structured entry with GPS tagging and photo capture, on-device validation, and a WorkManager-driven sync queue with retry, resume and conflict flagging.

Problem Statement

Connectivity-assuming survey tools lose data in the field: entries typed offline vanish on app kill, photos detach from records, interrupted uploads restart from zero, and server-side overwrites destroy a surveyor's work silently. The project addresses this with an offline-first architecture: Room/SQLite as the durable store, chunked acknowledged uploads that resume, and a conflict policy of flagging for human review — never silent overwrite. A supervisor view tracks per-form progress and sync states across the team.

Objectives

- Render assigned survey forms fully offline with text, numeric, select, date and GPS field types plus skip logic.
- Capture a GPS fix (with accuracy radius) and timestamp on every record.
- Support required/optional photo fields with on-device compression, surviving restarts.
- Validate required fields, ranges and formats on-device before queueing.
- Sync via WorkManager with ordered upload, exponential-backoff retry and resume.
- Flag server-newer records as conflicts for human review instead of overwriting.
- Provide supervisor views of per-form progress, sync states and conflicts.

Proposed Methodology

The app is built around a local-first data layer: form definitions and entries live in Room; a sync state machine (pending → uploading → acknowledged → synced) drives WorkManager jobs that upload in order when connectivity appears. Uploads are chunked with server acknowledgements so interruption resumes mid-record. The server returns version vectors per record; a newer server version marks the local record conflicted and surfaces it in the queue

for review. GPS comes from the fused location provider; photos compress on-device via CameraX capture callbacks.

System Architecture / Working

The Kotlin app has three layers: the form engine (renders field types, skip logic, validation), the data layer (Room database, photo store, sync queue), and the sync client (Retrofit/OkHttp against the reference Django REST backend). The supervisor view reads aggregated progress from the backend. The demo ships with sample surveys (crop damage, household census, water quality) so the full offline → queue → sync → conflict flow is demonstrable with the network toggled.

Software Requirements

- Kotlin (Android app)
- Room (on-device SQLite database)
- WorkManager (background sync scheduling)
- CameraX (photo capture)
- Fused Location Provider (GPS fixes)
- Retrofit with OkHttp (API sync client)
- Django REST backend (reference server)
- Material 3 UI components

Expected Outcomes

- A working Android app that collects complete survey records with zero connectivity.
- A sync engine demonstrating ordered upload, retry, resume and conflict flagging.
- GPS-tagged, photo-linked records with on-device validation.
- A complete report, PPT and viva Q&A; (Room, WorkManager, location APIs, conflict resolution).

Applications

Agricultural extension surveys, NGO field programs, construction site inspections and public-health data collection; the offline-first pattern adapts to any field-workforce app.

Future Scope

Automatic conflict merging for non-overlapping fields, map-based record review, supervisor assignment push, and on-device ML for photo quality checks.