Developer Design Document

Activity Management (Mobile CRM)

1. Overview

This document describes the technical design for the **Activity Management** feature in the mobile CRM application. The feature enables users to create, edit, delete, and sync activities associated with CRM records.

---

2. Architecture Overview

Components

- Mobile App (iOS and Android)
- Backend Activity Service
- Offline Storage and Sync Engine
- Authentication and Authorization Service

High-Level Flow

1. User creates or edits an activity on mobile.
2. Activity is saved locally (offline-first).
3. Sync engine pushes changes to backend when online.
4. Backend validates permissions and persists data.
5. Updates propagate to web and other devices.

---

3. Data Flow

Activity Creation

- Mobile UI validates mandatory fields.
- Activity stored in local DB with sync_status = PENDING.
- Sync worker posts activity to backend.
- Backend returns activity_id and timestamps.
- Local record updated to SYNCED.

Activity Editing

- Only allowed for user-owned activities.
- Offline edits create a new sync version.
- Conflict resolution uses last-write-wins.

---

4. Offline and Sync Strategy

- Local storage: SQLite or Room equivalent.
- Sync states:
    - PENDING
    - SYNCED
    - FAILED
- Retry with exponential backoff.
- Idempotency keys used for safe retries.

---

5. Permissions

- Activity visibility governed by record-level access.
- Edit and delete allowed only if:
    - User is owner OR
    - User has edit permission on activity entity.

---

6. Error Handling

- Validation errors shown immediately.
- Sync failures logged and retried silently.
- Hard failures surfaced in activity detail view.

---

7. Non-Functional Considerations

- Max activity payload size enforced.
- Sync must not block UI thread.

- Background sync throttled on low battery.