XCM / XFA Flight Ops App
Data Layer & API Wiring Technical Specification

Scope: This document specifies the data layer, SQL schema requirements, API interfaces, caching strategy, and refresh semantics for the XCM/XFA Flight Operations mobile application. UI layout and visual design are explicitly out of scope.

1. Architectural Overview

The system enforces a strict separation of concerns. The mobile client renders data only, the backend mediates all airline and airport API access, and SQL is the single source of truth for schedule and live flight status.

2. Legacy Parity Requirements

1 Schedule snapshot fetched when an airport is selected (Weekly Overview).

2 Schedule refresh only if older than 3 hours.

3 Live flight status refreshed every 15 minutes while a Day Details view is open.

4 Day Details always re-fetches full flight rows.

3. Time Model

1 UTC is authoritative: std_utc, sta_utc.

2 Local times stored for display: std_local, sta_local.

3 IANA timezones required: dep_tz, arr_tz.

4. Weekly Overview Fetch Window

1 Calendar-aligned to airport local time.

2 Start: today 00:00 local.

3 End: end of day N–1.

5. Classification & Bucketing Rules

1 Departures: dep_airport=X and std_utc in window.

2 Arrivals: arr_airport=X and sta_utc in window.

3 Bucketing by airport-local calendar day.

6. Flight Identity

1 flight_instance_id stored exactly as provided by airline API.

2 Used as primary merge key for status updates.

7. Status Refresh Model

1 Backend refreshes status and writes to SQL.

2 Client triggers refresh every 15 minutes while Day Details is open.

3 Client always re-fetches full rows for the day.

8. Caching Strategy

1 In-memory cache, frontend-only.

2 Keyed by airport + start date + day count.

3 Weekly schedule stale threshold: 3 hours.

9. SQL Schema Requirements

1 Identity: id, dep_airport, arr_airport, flight_number, flight_instance_id.

2 Times: std_local, sta_local, std_utc, sta_utc.

3 Timezones: dep_tz, arr_tz.

4 Status: status_code, status_last_updated_utc, etd_utc, eta_utc.

5 Freshness: schedule_last_updated_utc.

10. API Interfaces

1 GET /flights/window

2 GET /flights/day

3 POST /status/ensure-fresh

4 POST /schedule/ensure-fresh (optional)

11. Client Responsibilities

AppRoot defines navigation only. Screens call API and cache modules; no raw fetch logic inside UI components.

12. Out of Scope

All UI layout, spacing, colours, and visual design decisions are excluded from this specification.