

CONCEPT STUDY • INTERACTIVE PROTOTYPE

FieldDesk: Offline-First Field Operations

Deterministic offline data synchronization, local SQLite persistence, and side-by-side conflict recovery for commercial field service technicians.

INDUSTRY VERTICAL Logistics & Field Operations	DEMONSTRATED FOCUS HVAC & Mechanical Service Automation
SIMULATED PROTOTYPE STACK Next.js / TypeScript / React	TARGET PRODUCTION RUNTIME Flutter / Dart (Drift SQLite)

EXECUTIVE OVERVIEW

In industrial field service, reliable mobile software cannot depend on persistent cellular connectivity. When HVAC technicians enter basement plant rooms, hospital boiler bunkers, or elevator shafts, cellular networks drop entirely.

Traditional client-server applications fail by freezing forms, timing out on submissions, or silently overwriting remote dispatch updates. FieldDesk solves this through an autonomous **local-first state machine** with local SQLite storage, immutable mutation logging, and transparent visual conflict recovery.

1. Operational Context & Why Naive Apps Fail

Commercial HVAC systems control cooling and ventilation across multi-story office towers, pharmaceutical plants, and hospitals. Technicians servicing these systems follow a rigorous operational routine:

- Arriving at remote customer job sites with high priority dispatches.
- Performing mandatory sequential safety protocols (Lockout / Tagout verification).
- Recording diagnostic pressure and voltage readings with strict tolerances.
- Logging parts consumed from truck inventory for warranty and billing.
- Capturing physical photographic evidence and customer approval signatures.

THE THREE CRITICAL FAILURE MODES OF STANDARD MOBILE ARCHITECTURES

Failure Mode 1: The Frozen Form Barrier

Standard web and hybrid applications communicate via synchronous HTTP requests. When a technician taps “Complete Job” in a basement, the request hangs until timeout, locking the interface and causing loss of uncommitted form fields.

Failure Mode 2: Destructive Last-Write-Wins Overwrites

When basic offline caches reconnect, they blast the server database with the entire job record. If a dispatch supervisor re-routed the job or logged customer notes while the technician was underground, the technician's blind overwrite deletes the supervisor's updates.

Failure Mode 3: Phantom Double-Submissions

Flapping connectivity causes retry requests where the server commits the first attempt but drops the response. The client retries, double-debiting inventory parts and billing the client twice.

2. System Architecture: Local-First Event Sourcing

FieldDesk abandons the fragile client-server model in favor of an autonomous local-first architecture. The client maintains complete operational sovereignty.

A. ENCRYPTED LOCAL SQLITE (DRIFT ORM)

All daily routes, equipment serials, maintenance histories, and checklist requirements are stored in a local SQLite database encrypted with SQLCipher. Queries execute in under 4ms locally with zero network dependency.

B. APPEND-ONLY MUTATION QUEUE

Rather than modifying record tables directly, user actions append atomic events to an immutable `offline\_mutations` queue. Each event records:

```
{ id: "MUT-UUID-01", type: "CHECKLIST_COMMIT", timestamp: 1727073722, idempotencyKey: "SHA256", payload: {...} }
```

C. DETERMINISTIC SYNCHRONIZATION ORCHESTRATOR

A background sync engine monitors network reachability. When connectivity is verified, the orchestrator transmits pending mutations in strict chronological order using HTTP idempotency headers (`X-Idempotency-Key`).

Idempotency Guarantee

If an intermittent cell connection drops during response delivery, the subsequent replay sends the identical idempotency key. The backend returns the cached success transaction without executing duplicate inventory debits.

3. Interface Walkthrough: Route & Job Detail

Screens 1 and 2 establish daily technician orientation, urgent priority tagging, and equipment specifications before entering basement plant rooms.

SCREEN 01 • ROUTE LIST

Daily Assigned Schedule

#JOB-2041

URGENT

Chiller Inspection & Valve Repair

Apex Commercial Tower • Basement B2

Cached route data displays site addresses, priority levels, and equipment model numbers without needing live GPS lookups.

SCREEN 02 • JOB DETAIL

Carrier 30XA Specifications

Serial Number:

CR-9921-X

Refrigerant:

R-134a (Dual Loop)

Safety Protocol:

Lockout / Tagout Required

Mandatory Lockout/Tagout protocol warning prevents the technician from accessing parts logs before safety checklist signoff.

Thumb-Reach Ergonomics

In field environments where technicians wear heavy nitrile or leather work gloves, buttons maintain a minimum 48px target height with high-contrast text and 12px margins, preventing accidental taps.

4. Interface Walkthrough: Checklist & Photo Evidence

Screens 3 and 4 demonstrate sequential verification, numeric pressure validation, and offline image hashing.

SCREEN 03 • CHECKLIST

Sequential Verification

1. Lockout / Tagout Applied

2. Suction Pressure: 71.4 PSI (OK)

3. Motor Bearings Inspected

Checklist completion is strictly sequential. Form validation prevents advancing until safety isolation is confirmed.

SCREEN 04 • PHOTO EVIDENCE

Offline Photo & Note

CARRIER_30XA_VALVE_INSPECTION.JPG

Captured: 10:42:15 AM • 37.7749°N, 122.4194°W

Note: "Suction line valve minor weeping; tightened & sealed."

Photos are compressed and hashed locally. Exif timestamps and cached GPS coordinates are burned directly into metadata.

Local Image Chunking

Camera images are split into 256 KB blocks in local storage. When uploading on weak connections, failed transfers resume from the last received chunk rather than restarting the entire 2 MB file.

5. Conflict Recovery & Sync Confirmation

Screens 5, 6, and 7 show the pending mutation queue, the visual conflict diff, and final server synchronization.

SIDE-BY-SIDE COLLISION DIFFING (SCREEN 06)

LOCAL TECHNICIAN DATA

Recorded: 10:43:30 AM

Status: **Completed**. Replaced Filter Drier #FD-408-S. Suction pressure verified 71.4 PSI. Signed by J. Miller.

REMOTE DISPATCH RECORD

Updated: 10:38:00 AM

Status: **Reassigned to Standby Crew**. Supervisor note: "No response from primary technician after 45 minutes."

The Three Conflict Resolution Paths

1. Option A: Intelligent Merge (Recommended Default)

Overrules the dispatch reassignment with physical technician proof; cancels standby crew dispatch, commits completed checklist, debits truck inventory, and updates customer invoice.

2. Option B: Force Overwrite (Client Wins)

Overwrites remote status completely, discarding the dispatch note.

3. Option C: Accept Remote Dispatch (Server Wins)

Accepts standby reassignment and files technician work as an internal draft note.

6. Handled Edge Cases & Measurable Validation

Connection Loss Mid-Signature Capture

Vector bezier curves are serialized directly into the local SQLite database on every stroke release. No network request is initiated until the technician taps “Confirm Signature.”

Device Clock Skew Across Time Zones

Technicians crossing regional borders may have out-of-sync system clocks. Every mutation includes a client Lamport logical timestamp to establish indisputable causal ordering regardless of wall-clock drift.

Corrupted Database Recovery

If the device OS suddenly terminates the application during a write transaction, SQLite WAL (Write-Ahead Logging) automatically replays uncommitted frames on restart, guaranteeing zero database corruption.

SIMULATION TELEMETRY VS. TARGET PRODUCTION BENCHMARKS

Metric Dimension	Simulated In Prototype	Target Field Benchmark
End-to-End Task Duration	42 Seconds	< 60 Seconds
Conflict Resolution Rate	100% Deterministic	> 99.8%
Reconnection Sync Turnaround	Instant (0.4s)	< 2.0s on LTE / 3G
Data Loss Rate	0.00%	0.00% Zero Tolerance

7. Explicit Limitations & Production Roadmap

TRUTHFUL PROTOTYPE DISCLOSURE

This prototype demonstration is implemented in the browser using React state and HTML5 LocalStorage event sourcing. While it faithfully demonstrates the interface state transitions, queue visualization, and conflict recovery UX, it does not interact with physical cellular baseband processors.

PRODUCTION ROADMAP: COMPILING TO NATIVE MOBILE (FLUTTER / DART)

- 1 **Native Background Daemons:** Migration to Flutter enables native iOS BackgroundTasks and Android WorkManager background execution, syncing queues even when the app is backgrounded.
- 2 **Hardware Barcode Scanning:** Native camera integration with ML Kit for instant barcode scanning of replacement part packaging directly into SQLite.
- 3 **Bluetooth Mesh Relay:** Optional peer-to-peer sync allowing a technician deep underground to relay sync packets via a colleague's device parked outside.

Engage FFNA for Your Mobile Engineering

We engineer mission-critical mobile applications with defined sprint scopes, visible bi-weekly progress, and clean source code handover.

Inquire: contact@ffnasolutions.com • **Web:** ffnasolutions.com