

CONCEPT STUDY · INTERACTIVE PROTOTYPE

RentFlow: Role-based tenant maintenance portal with multi-party audit trails and transparent repair scheduling.

Role-based lifecycle transparency with explicit audit trails. Status transitions (e.g. "Work Order Dispatched", "Parts Awaiting Approval", "Arrival Window Confirmed") are synchronized across tenant, manager, and contractor views with real-time audit milestones.

INDUSTRY VERTICAL

Property & Facility Management

PROTOTYPE STACK

Next.js / TypeScript

DOMAIN SECTOR

Tenant Operations & Facility Maintenance

TARGET PRODUCTION RUNTIME

React Native / Expo

Executive Summary

Sarah, an apartment tenant, discovers a sudden water leak beneath her kitchen sink. She needs urgent repair authorization without playing phone tag with property managers or wondering when a plumber will arrive.

Property maintenance is traditionally plagued by opaque communication: tenants receive generic "Ticket Received" confirmations with zero visibility into contractor dispatches, parts orders, or estimated arrival windows.

Operational Hurdles & The Failure of Conventional Approaches

In real-world mobile deployments within Property & Facility Management, applications encounter operational friction that desktop web portals never face.

Core Engineering Challenges Addressed:

- 1. Operational Bottleneck:** Property maintenance is traditionally plagued by opaque communication: tenants receive generic "Ticket Received" confirmations with zero visibility into contractor dispatches, parts orders, or estimated arrival windows.
- 2. Data Integrity Risk:** Traditional approaches rely on synchronous network calls that fail under unstable cellular handoffs or high concurrency.
- 3. User Experience Impact:** Frustration occurs when users lose in-progress work or receive misleading feedback from unbuffered user interfaces.

Role-based lifecycle transparency with explicit audit trails. Status transitions (e.g. "Work Order Dispatched", "Parts Awaiting Approval", "Arrival Window Confirmed") are synchronized across tenant, manager, and contractor views with real-time audit milestones.

Architectural Solution & Interface Design

Demonstrated UX & Architecture Decision:

Role-based lifecycle transparency with explicit audit trails. Status transitions (e.g. "Work Order Dispatched", "Parts Awaiting Approval", "Arrival Window Confirmed") are synchronized across tenant, manager, and contractor views with real-time audit milestones.

Principal Interface Screens (5 Views):

- 1 Tenant Property Home**
Active lease summary, quick issue reporter, and real-time maintenance timeline.
- 2 Structured Request Intake**
Categorized issue picker with required photo/video upload and urgency triage.
- 3 Interactive Ticket Timeline**
Auditable milestone tracker showing contractor assignment, quotes, and arrival times.
- 4 Contractor Window Selector**
Two-way calendar matching tenant availability with plumber dispatch slots.
- 5 Work Order Sign-Off**
Post-repair photographic verification and tenant satisfaction sign-off.

Interactive Journey: Plumbing Issue Intake to Dispatched Sign-Off

Submit issue with photo evidence → Manager assigns contractor → Tenant picks arrival window → Technician dispatched → Complete repair verification.

- 1 Open RentFlow tenant maintenance portal
- 2 Create new ticket: select "Plumbing" and capture sink leak photo
- 3 Select "Urgent" severity and submit with permission-to-enter notes
- 4 Switch to Manager view to approve contractor quote and work order
- 5 Switch back to Tenant view to confirm contractor arrival slot for 2:00 PM
- 6 Inspect synchronized status update showing "Technician En Route"

Reset Mechanism: One-click reset to initial unsubmitted tenant home screen.

Handled Edge Cases & Exception Recovery



Edge Condition #1:

Emergency severity selection routing urgent gas/water leaks directly to 24/7 emergency dispatch hotline



Edge Condition #2:

Contractor reschedule notification offering immediate 1-tap confirmation of alternative time windows



Edge Condition #3:

Automated manager escalation trigger after 48 hours without status transition update

Actual Prototype vs. Target Native Production Runtime



Actual Prototype Stack

Platform & Framework:

Next.js / TypeScript

Languages:

TypeScript, Tailwind CSS

Multi-role reactive portal simulating tenant, property manager, and maintenance contractor views.



Target Production Stack

Platform & Framework:

React Native / Expo

Languages & Storage:

TypeScript, Supabase Realtime (PostgreSQL)

Fast cross-platform UI iterations, real-time WebSocket state subscriptions, and Row Level Security for tenant privacy.

Explicit Prototype Boundaries & Validation Methodology

Documented Prototype Limitations:

- Simulated in browser using local state machine; does not connect to live property management PMS (e.g. Yardi / AppFolio) APIs.
- Push notification alerts simulated in-memory without APNs / FCM hardware tokens.

Target Production Telemetry:

Evaluate reduction in inbound tenant customer service calls across a 500-unit pilot residential building.

Commercial Offer Alignment & Handover Guarantees

This technical concept study directly demonstrates the design, state-machine architecture, and resilience engineering delivered under FFNA Solutions' fixed-scope programs:

Product Prototype Sprint

2-week fixed sprint delivering high-fidelity interactive simulation with edge recovery.

Mobile MVP Launch

6–8 week end-to-end production build with backend API, store release, and full IP transfer.

Handover Guarantee

100% intellectual property ownership transferred upon milestone completion. Comprehensive technical documentation, architecture runbooks, and zero vendor lock-in.