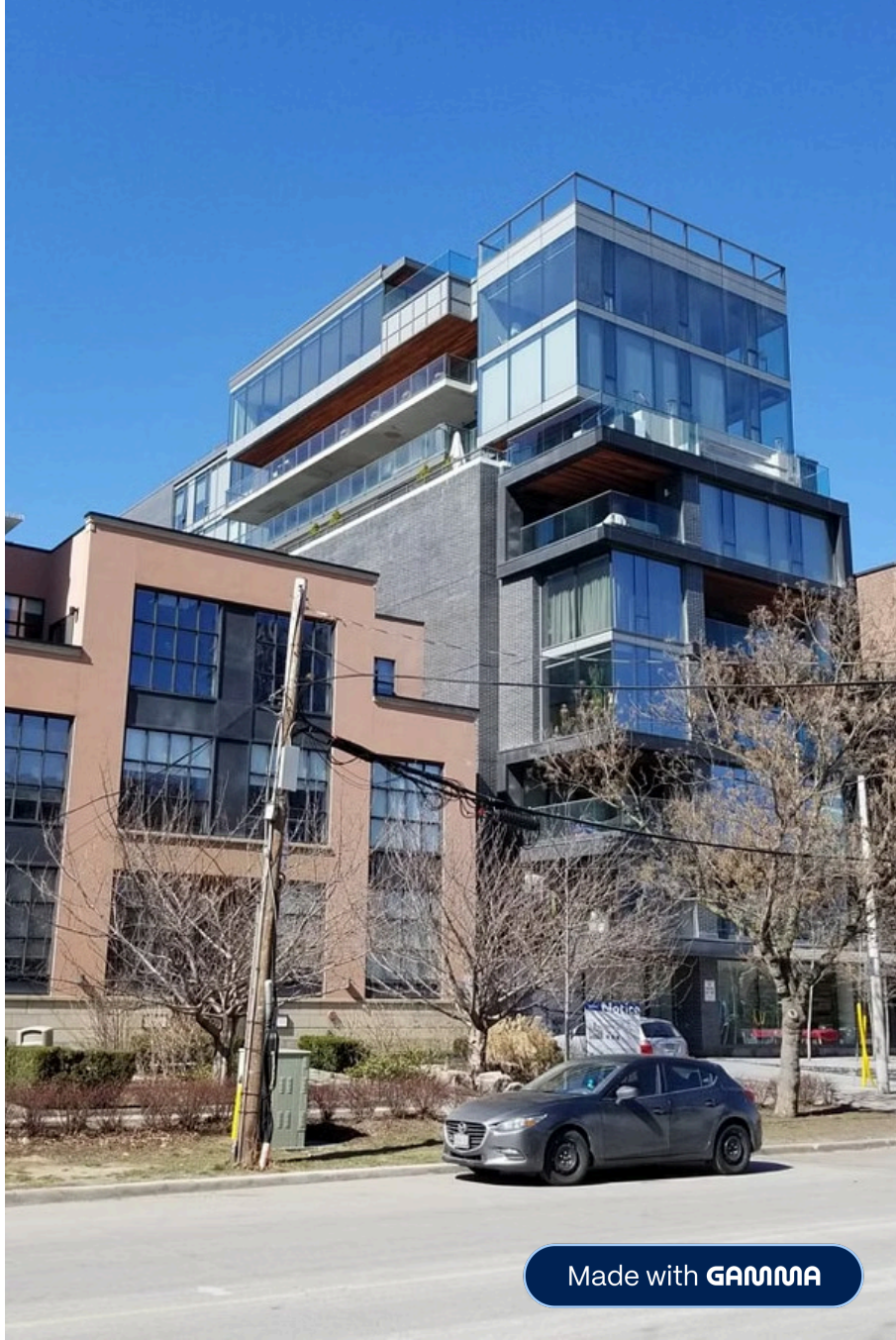
RELAY

Condo Maintenance Work Order System

Built for Canada, starting with Toronto.



PROBLEM

Maintenance tracking issues cost time and reduce trust.

- Requests arrive via various channels, leading to fragmented communication.
- Critical details (unit, issue, responsibility) are easily missed.
- Residents lack timely status updates on requests.
- Managers lack data for audits and accountability.
- This informal process creates operational issues, wastes time, and erodes trust.

Market Context: Canada and Toronto

Housing Share

Condominiums represent a significant portion of housing stock in major Canadian cities

Condo Corps

Thousands of condo corporations actively manage maintenance operations across the GTA

Condo Living Creates Operational Scale

Toronto is one of North America's most condo-dense cities. With hundreds of units per building generating continuous maintenance requests, workflow coordination is a critical operational challenge for property managers.

What Breaks Today

No Standardized Lifecycle

Request workflows vary by building, manager, and situation. There's no consistent process from intake to resolution.

Unclear Ownership During Handoffs

Responsibility becomes ambiguous when requests move between residents, managers, and technicians.

Status Updates Are Informal or Missing

Residents often don't know if their request was received, assigned, or completed. Updates happen ad-hoc, if at all.

Reporting Is Manual and Fragmented

Managers piece together reports from emails, spreadsheets, and memory—time-consuming and error-prone.

SOLUTION

A Work Order System With a Clear Lifecycle

01

Residents Submit Structured Requests

Capture all essential details upfront: unit number, issue type, description, and urgency level.

02

Managers Review and Assign

Triage incoming requests, determine responsibility, and route to the appropriate technician or vendor.

03

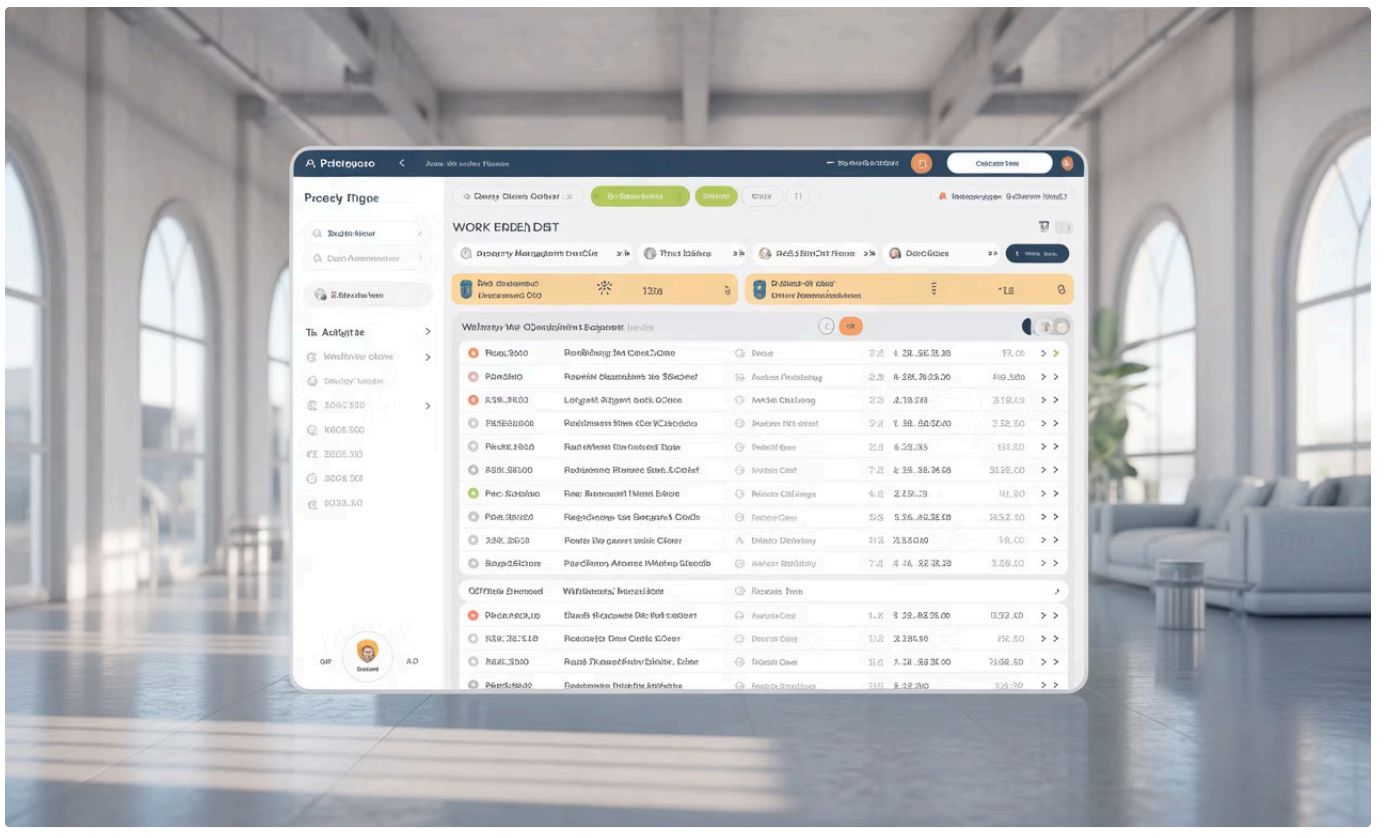
Technicians Update Progress

Log status changes as work moves from assigned to in-progress to completed.

04

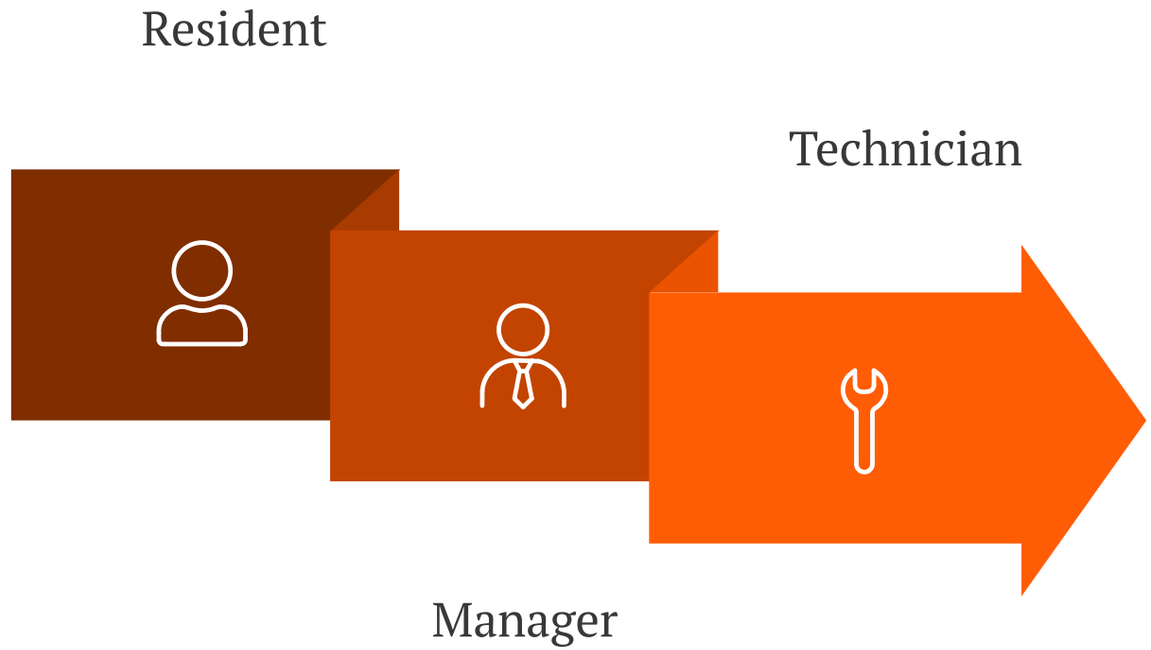
Complete Audit Trail

Every action is timestamped and logged, creating accountability and enabling historical review.



Users and Operational Flow

One request, multiple roles working together through a structured lifecycle.



Each role has clear responsibilities at each stage. The system enforces valid transitions and prevents workflow gaps.

How the System Works

User Action | API Validation | Enforce Rules | Save + Audit | Return State

From User Action to Traceable Result

Every interaction follows a predictable path. The API validates incoming requests against business rules, ensuring only valid state transitions occur.

When data is saved, an audit log entry is automatically created with timestamp, user, and action details. The system immediately returns the updated state, keeping all users in sync.

Business Rules & Backend Enforcement

Rules are enforced in the API/service layer to prevent workflow gaps and keep the audit trail trustworthy.

Lifecycle (Valid Status Transitions Only)

Work orders must follow the defined lifecycle (Submit → Assign → In-Progress → Complete). Invalid transitions are rejected with a clear error response.

Role-Based Actions (Phase 1: Simplified Roles)

Residents can submit requests. Managers can assign / cancel. Technicians can update progress and complete.

Assignment Requirements

A request cannot move to "Assigned" unless a technician/vendor is selected. Assignment stores who assigned it and when.

Completion Requirements

A work order cannot be marked "Completed" without a resolution note. Completion timestamps are recorded.

Audit Trail Is Automatic and Append-Only

Every status change creates an audit event (who/when/what changed). This supports accountability and reporting.

Business rules live in the service layer; controllers handle HTTP input validation.

Backend Architecture: Phase 1

Router Layer

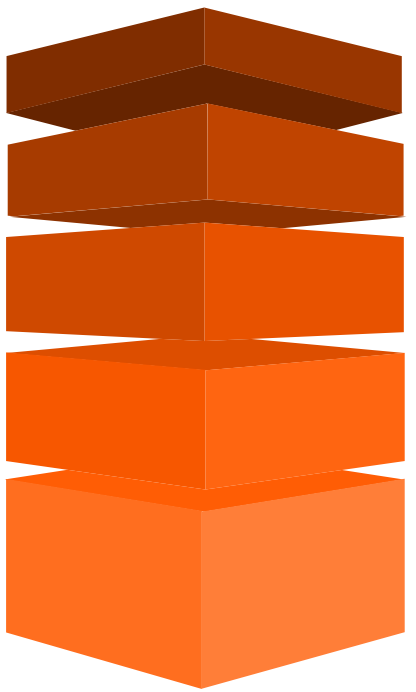
Handle HTTP requests and route to controllers.

Service Layer

Enforce business rules and application logic.

Outputs

Emit audit logs and daily export files.



Controller Layer

Orchestrate inputs and call service operations.

Repository Layer

Persist data to JSON file storage.

Simple Layers, Clear Responsibilities

The architecture separates concerns into distinct layers. Routers handle HTTP requests, controllers orchestrate operations, services enforce business logic, and repositories manage data persistence.

Business rules live exclusively in the service layer, making them easy to test and modify. Phase 1 uses JSON file storage for simplicity, with audit logs and daily exports generated automatically.

Phase 1 Scope and Next Steps

Included in Phase 1

- Backend API with complete workflow enforcement
- Input validation and error handling
- Comprehensive audit logging
- Automated daily reporting and exports
- Status transition business rules

Deferred to Future Phases

- User authentication and role-based permissions
- Frontend user interface and mobile optimization
- Email and push notifications
- Advanced reporting dashboards
- Integration with existing property management systems

Phase 1 establishes a solid foundation for future expansion with proven workflow logic and reliable data tracking.