This document acts as the definitive **System Requirements Document (SRD)**. It consolidates all functional, technical, and operational requirements discussed, tailored for deployment on **Amazon AWS**. It serves as the "Source of Truth" for the subsequent System Design (LLD) and UX/UI Design phases.

# Tama38- System Requirements Document (SRD): Urban Renewal Tenant Lifecycle Management Platform

Version: 1.0

Date: December 24, 2025

Architect: Gemini (AI Thought Partner)

Target Environment: Amazon AWS

## 1. Introduction

The Tama38 system is a SaaS platform designed to manage the end-to-end lifecycle of tenants in Israeli Urban Renewal projects (*Pinui Binui* / *TAMA 38*)1111. It bridges the gap between the legal/planning process and the tenant experience, replacing traditional tenant apps with a smart **WhatsApp Chatbot** interface while providing a robust command-and-control dashboard for the developer2222.

### 1.1 Core Objectives

1. **Single Source of Truth:** Centralized database for all project data, managing the hierarchy of Project $\rightarrow$ Building $\rightarrow$ Unit $\rightarrow$ Owner3333.
2. **Weighted Majority Engine:** Real-time calculation of the "Required Majority" (*Rov Nidrash*) based on configurable legal thresholds (Headcount/Area)4444.
3. **Hybrid Workflow:** Seamless integration of digital signatures (WhatsApp) and manual overrides (Hard Copies)5.
4. **Manager Approval Gate:** No document is final until validated by a manager.

## 2. System Actors & Role-Based Access Control (RBAC)

The system enforces strict data isolation and permission hierarchies6666.

| **Role** | **Interface** | **Key Responsibilities & Permissions** |
| --- | --- | --- |
| **Super Admin** | Desktop Web | **Global Config:** Define "Required Majority" logic per project. Manage product catalog. Full access to all data and logs7. |
| **Project Manager (PM)** | Desktop Web | **Ops & Approval:** Approve signed contracts. Manage Agent teams. View "Red/Green" dashboards. Handle alerts and override requests8888. |
| **Agent** | Mobile Web/App | **Field Work:** Lead management. "Click-to-Call" logging. Ingest manual docs. Initiate signature flows9999. |
| **Tenant** | WhatsApp Bot | **Passive/Active:** Receive status updates. Request documents. Perform ID verification and digital signing10. |

## 3. Functional Requirements: The Modules

### 3.1 The Management Dashboard (Command & Control)

Designed for the Project Manager and Super Admin to identify bottlenecks immediately11.

* **Project Pulse (Donut Charts):**
  + Visual representation of signature progress per building vs. the Legal Threshold (e.g., 67%)12.
  + **Traffic Light Logic:** Green (On Track), Yellow (Warning), Red (Below Threshold/Stuck)13.
* **Building Matrix View:**
  + A grid visualizing the building physically (Rows = Floors, Cells = Apartments)14.
  + **Color Coding:** 🟩 Signed, 🟨 Negotiation, 🟥 Refusal, ⬜ No Contact 15.
  + **Drill-Down:** Hovering over a cell displays the specific owners and their status (e.g., "Cohen: 50% Signed")16.
* **Alert Center:**
  + **Logic-Based Triggers:** "Building X dropped below 50% due to cancellation" or "Agent Y hasn't logged activity for 3 days"17171717.

### 3.2 The Field Agent App (Mobile Web)

Optimized for mobile use in stairwells and meetings ("Thumb-friendly")18.

* **Lead Feeder:** List of assigned tenants sorted by priority/status19.
* **Action Hub:**
  + **Communication:** Buttons for Call/WhatsApp. Mandatory "Call Summary" popup after every interaction20.
  + **Manual Ingest:** Camera interface to scan physical ID cards or paper contracts signed offline. These enter a "Pending Approval" queue21.
* **Signature Trigger:** Button to "Send Signing Link" to the tenant's WhatsApp22.

### 3.3 The Tenant Interface: WhatsApp Chatbot

A verified Business Account acting as the tenant's primary gateway.

* **Service A: Status Updates (Push/Pull)**
  + *Pull:* Tenant types "Status"; Bot replies: "Your building is at 45% signatures. 22% more to go!"23.
  + *Push:* Automated message when milestones are met (e.g., "Permit Received")24.
* **Service B: Document Retrieval**
  + Tenant requests "My Contract"; Bot generates a secure, time-limited AWS S3 link to the PDF25.
* **Service C: Digital Signing & ID Verification**
  + Bot sends a unique link to a secured mobile web view.
  + **Flow:** Upload ID photo $\rightarrow$ Biometric Liveness Check $\rightarrow$ Review Document $\rightarrow$ Sign on Screen.
* **Service D: Q&A / Support**
  + Simple menu-driven support (1. Legal, 2. Technical) or escalation to a live human agent task26.

## 4. Key Workflows (Main Flows)

### Flow 1: Digital Signature with Manager Approval (The "Happy Path")

1. **Initiation:** Agent meets tenant, agrees on terms. Agent clicks "Start Signing Process" in the App.
2. **Tenant Action:** Tenant receives WhatsApp link.
   * *Step 1:* System validates ID (OCR match against DB record).
   * *Step 2:* Tenant signs digitally.
3. **System State:** Contract status changes to **"Signed - Pending Approval"**. The unit's progress bar *does not* update yet.
4. **Manager Action:** PM receives an alert. Opens the document, verifies the terms and the ID match.
5. **Finalization:** PM clicks "Approve".
   * Status becomes **"Finalized"**.
   * Majority Engine recalculates project percentages.
   * Tenant receives a copy of the countersigned PDF via WhatsApp.

### Flow 2: Manual Override (The "Geriatric" Path)

1. **Scenario:** Elderly tenant refuses smartphones. Signs a paper contract.
2. **Ingestion:** Agent takes a photo of the paper contract via the Agent App.
3. **Indexing:** Agent manually tags the file as "Signed Contract" for Unit 4.
4. **Validation:** PM reviews the uploaded scan.
5. **Override:** PM manually toggles the Unit Status to "Signed" and attaches the scanned file as the proof.
6. **Audit:** System logs this as "Manual Override by [PM Name] on [Date]".

### Flow 3: Ownership Transfer (Lifecycle Event)

1. **Trigger:** Tenant notifies Agent they sold the apartment.
2. **Agent/PM Action:** Enters "Edit Unit" mode.
3. **Archive:** Current owner is moved to "Historical Owners" (audit trail preserved).
4. **New Entry:** New Owner details entered. Status resets to "Not Contacted".
5. **Recalculation:** Building percentage drops immediately (e.g., from 67% to 65%).
6. **Alert:** System generates a "Critical Alert" if the drop crosses a threshold27.

## 5. Data Entities & AWS Infrastructure

### 5.1 Entity Relationships (ERD Concepts)

* **Project:** Configs (Threshold %, Location).
* **Building:** Linked to Project. Aggregates Unit stats.
* **Unit:** The physical asset. Attributes: Floor, Area (SqM), Current Status.
* **Owner (Tenant):** The legal entity. Linked to Unit (M:N relation - one unit can have multiple owners). Attributes: ID, Phone, % Ownership Share.
* **Interaction Log:** Timestamp, Type (Call/Meet), Summary, Agent ID.
* **Document:** Type (Contract, Taba), S3 URL, Status (Draft/Signed/Approved).

### 5.2 AWS Services Mapping

* **Compute:** AWS Lambda (Serverless) for handling WhatsApp webhooks and Majority Engine logic.
* **Database:** Amazon RDS (PostgreSQL) for relational data (Projects, Owners).
* **Storage:** Amazon S3 for storing PDF contracts, scanned IDs, and plans.
  + *Security:* S3 Bucket Policies restricting public access; presigned URLs for WhatsApp delivery.
* **Authentication:** Amazon Cognito for internal users (Admins, Agents).
* **Verification:** Amazon Rekognition (optional) for comparing ID scans vs. Selfie for biometric verification.

## 6. Non-Functional Requirements (NFR)

1. **Security & Privacy:**
   * All PII (ID numbers, Phone) must be encrypted at rest (AWS KMS).
   * Role separation: Agents cannot view projects they are not assigned to28.
2. **Audit Trail:**
   * Immutable logging of every status change (who changed "Refusal" to "Negotiation"?).
   * Manual overrides require a mandatory text field explanation.
3. **Performance:**
   * Majority calculation updates must propagate to the dashboard in <3 seconds29.
4. **Backup & Recovery:**
   * Daily automated snapshots of RDS and S3.
   * Retention policy: 7 years (legal requirement for contracts).

## 7. UX/UI Design Guidelines

* **Visual Hierarchy:** Dashboard must prioritize "Exceptions" (Red items) over "Normalcy" (Green items)30.
* **Accessibility:** WhatsApp messages and Mobile Web signing pages must use high-contrast, large fonts for older demographics31.
* **Frictionless Field Work:** Agent app buttons must be large, with minimal typing required (use dropdowns/selectors)32.