

```
# All Requirements Master Document
```

```
---
```

```
## Source: EPIC_Authority_Operations_Dashboard.md
```

```
# EPIC: Authority Operations Dashboard & Zone Management
```

```
**Generated Copy:** This file is generated from `authority_dash_spec.md`. Do not edit directly.
```

```
## Executive Summary
```

A web operations console for Government Admins to manage the full lifecycle of issues within their jurisdiction, from map-based triage to worker assignment and resolution review.

```
## Stakeholders & Value
```

```
### User Personas
```

- **Primary Persona:** Govt Admin (manages a jurisdiction, assigns work, reviews fixes).
- **Stakeholders:** Field Workers, Citizens, System Admin.

```
### User Value
```

Admins can see what is happening in their zone, prioritize by impact, assign efficiently, and enforce quality control.

```
## Goal & Vision
```

Provide a clear, accountable command center that reduces time-to-fix while keeping jurisdiction boundaries explicit and immutable.

```
## Scope
```

```
### In Scope
```

- Geospatial operations map with jurisdiction overlay.
- Zone-of-interest filter using GHMC areas (Hyderabad only for now).
- Kanban workflow with bulk assignment and priority setting.
- Active load view (worker task counts).
- Worker onboarding, approval, and deactivation (no deletion).
- Resolution review with audit logging.
- Public audit log integrations.
- Invitation-based authority registration (via System Admin onboarding).
- Authority profile management.
- Email OTP login and Google OAuth only (no SMS/phone OTP).
- OpenStreetMap base tiles (Mapbox optional).
- Time estimates captured from workers and displayed to authorities.

```
### Out of Scope
```

- Category management (System Admin only).
- Zone creation or editing (System Admin only).
- AI-based priority scoring.
- Push notifications.

```
## Success Metrics
```

- Reduced time from REPORTED to ASSIGNED using bulk assignment.
- Fast review turnaround for RESOLVED tickets.
- Visible reasons for dismissed tickets.

```
## Stories Under This Epic
```

1. [USER STORY 1: Geospatial Operations Map](#user-story-1-geospatial-operations-map)
2. [USER STORY 2: Kanban Dashboard & Bulk Assignment](#user-story-2-kanban-dashboard--bulk-assignment)
3. [USER STORY 3: Worker Onboarding & Team Management](#user-story-3-worker-onboarding--team-management)

4. [USER STORY 4: Resolution Review & Quality Control](#user-story-4-resolution-review--quality-control)
5. [USER STORY 5: Authority Settings & Profile Management](#user-story-5-authority-settings--profile-management)

USER STORY 1: Geospatial Operations Map

Executive Summary

Render the Admin's jurisdiction as a read-only boundary, overlay active issues as pins, and allow focused viewing by GHMC sub-areas.

User Persona & Problem Statement

Who: As a Govt Admin...

Why: I need to see where issues are relative to my official boundary and focus on specific sub-areas quickly.

Scope (In & Out)

In Scope

- Zone boundary overlay (read-only).
- GHMC zone-of-interest filter (dropdown).
- Color-coded pins with report_count and priority.
- Quick-view actions from map pins.

Out of Scope

- Drawing/editing zones.
- Heatmap density analysis (handled in analytics dashboard).
- Real-time worker tracking.

Features & Acceptance Criteria

Feature: Zone Context Overlay

User Story: As a Govt Admin, I want to see my zone clearly marked on the map so I know my operational limits.

Acceptance Criteria:

- [] Verify the map auto-zooms to the assigned zone boundary at login.
- [] Verify the zone is rendered as a read-only polygon (no edits).
- [] Verify the zone name is displayed in the header.
- [] Verify the zone boundary source is GHMC KML/GeoJSON for Hyderabad (no custom zones in Phase 1).

Feature: Zone-of-Interest Filter (GHMC)

User Story: As a Govt Admin, I want to filter the map to a GHMC area so I can focus on hotspots.

Acceptance Criteria:

- [] Verify a dropdown lists GHMC areas within the assigned jurisdiction.
- [] Verify selecting an area zooms the map to that boundary and filters pins to that area.
- [] Verify clearing the selection resets the view to the full zone.

Feature: Operational Issue Visualization

User Story: As a Govt Admin, I want to see all active issues as pins so I can identify clusters.

Acceptance Criteria:

- [] Verify pins show all active tickets for this zone.
- [] Verify pin colors reflect status (REPORTED, ASSIGNED/IN_PROGRESS, RESOLVED).
- [] Verify pins are not clustered and represent unique issue IDs.
- [] Verify each pin includes report_count and priority in its quick-view.
- [] Verify filters by category/status update the map pins.

```

### Feature: Map-to-Action
**User Story:** As a Govt Admin, I want to click a pin to see details and take action.

**Acceptance Criteria:**
- [ ] Verify clicking a pin opens a quick-view with photo, status, report_count, priority, and assigned worker.
- [ ] Verify quick-view shows worker ETA and time since issue creation when available.
- [ ] Verify quick-view links to the full issue detail modal.
- [ ] Verify quick-view includes an "Open Ticket" action.

## UI/UX Design & User Flow
**Flow:** Login -> Map loads -> Zone overlay -> Optional GHMC area filter -> Select pin -> Quick view -> Open issue.

## Functional Requirements
**Zone Data:**
- `GET /zones/{assigned_zone_id}` returns GeoJSON derived from GHMC KML.
- Zone polygons are read-only in the UI.

**Issue Data:**
- `GET /issues?zone_id={id}&status=active` returns active issues for pins.
- Pin payload includes: `issue_id`, `lat`, `long`, `status`, `category_icon`, `report_count`, `priority`, `created_at`, `assigned_worker`, `estimated_minutes`.

**Map Provider:**
- Use OpenStreetMap tiles by default; Mapbox optional.

**Viewport Logic:**
- Fit map to zone bounding box on load.

**Deduplication Assumption:**
- Backend aggregates duplicates within 5m and increments `report_count`.

**Filtering:**
- Map responds to global category/status filters from the dashboard.

## RBAC & Permissions
| Role | Can View Map | Can See Zone | Can Edit Zone |
| --- | --- | --- | --- |
| Govt Admin | Yes | Yes | No |
| System Admin | Yes | Yes | Yes |

## Dependencies
- System Admin zone data must exist before map loads.

## Non-Functional Requirements (Minimal)
- Map interaction should remain responsive at expected pin volumes.

## Edge Cases
- Zone missing returns a "No Zone Configured" message.
- Issues outside the polygon are shown with a warning icon.

---

# USER STORY 2: Kanban Dashboard & Bulk Assignment

## Executive Summary
A Kanban board for high-volume triage with bulk assignment, priority management, and visibility into worker estimates.

## User Persona & Problem Statement

```

Who: As a Govt Admin...

Why: I need to assign many issues quickly and prioritize the most reported ones.

Scope (In & Out)

In Scope

- Kanban board with drag-and-drop.
- Bulk select and assignment.
- Priority setting per issue.
- Sorting by report_count, priority, and created_at.
- Worker ETA visibility.

Out of Scope

- AI-based sorting.
- Push notifications.

Features & Acceptance Criteria

Feature: Kanban Board

User Story: As a Govt Admin, I want a column view of issue status.

Acceptance Criteria:

- [] Verify cards display issue_id, category, photo thumbnail, report_count, priority, and days open.
- [] Verify columns support sorting by report_count (default), priority, or created_at.
- [] Verify drag-and-drop from REPORTED to ASSIGNED triggers assignment modal.

Feature: Smart Assignment (Single & Bulk)

User Story: As a Govt Admin, I want to assign multiple issues to a worker efficiently.

Acceptance Criteria:

- [] Verify bulk selection of multiple REPORTED cards.
- [] Verify assignment modal lists workers with active task counts.
- [] Verify assignment updates cards to ASSIGNED and stores assigned worker.
- [] Verify re-assignment is possible even when a task is IN_PROGRESS.

Feature: Priority Management

User Story: As a Govt Admin, I want to set priority per issue so crews focus on critical work.

Acceptance Criteria:

- [] Verify each card has a priority selector (P1-Critical to P4-Low).
- [] Verify priority changes are saved immediately and visible on map and Kanban.
- [] Verify priority sorting is available in the Kanban header.

Feature: Time Estimate Visibility

User Story: As a Govt Admin, I want to see worker-provided estimates so I can track timelines.

Acceptance Criteria:

- [] Verify cards show estimated completion time when provided by worker.
- [] Verify the UI shows time since issue creation and time remaining vs estimate.

UI/UX Design & User Flow

Flow: Load board -> Select cards -> Assign -> Monitor priority and ETA fields.

Workflow & Entity State Lifecycle

Lifecycle: REPORTED -> ASSIGNED -> IN_PROGRESS -> RESOLVED -> CLOSED.

Functional Requirements

****Data Fetching:****
- `GET /issues?zone_id={id}&status=active` for board data.

****Card Fields:****
- `issue_id`, `category_icon`, `created_at`, `report_count`, `priority`, `thumbnail_url`.
- `accepted_at`, `estimated_minutes`, `assigned_worker` (if available).

****Sorting Logic:****
- Default sort by `report_count` desc.
- Toggle to `priority` or `created_at` as needed.

****Priority Updates:****
- `PATCH /issues/{id}/priority` with payload `{ priority: P1|P2|P3|P4 }`.

****Bulk Assignment:****
- `POST /assignments/bulk` accepts issue_ids[] and worker_id, transactional.

****Worker Load Tracking:****
- Assignment modal displays active task counts per worker.

****Validation:****
- Prevent assignment for RESOLVED/CLOSED issues.

****Audit Logging:****
- Log assignment and priority changes with actor, timestamp.
- Track previous_worker and new_worker for re-assignments.

RBAC & Permissions

Role	Can Assign	Can Re-Assign	Can View Board
---	---	---	---
Govt Admin	Yes	Yes	Yes (own zone only)
Field Worker	No	No	No

Dependencies
- Worker list from Story 3.

Non-Functional Requirements (Minimal)
- Kanban interactions remain responsive for expected ticket volumes.

Edge Cases
- Deactivated workers are hidden from assignment list.
- Concurrent updates may cause conflicts and require refresh.

USER STORY 3: Worker Onboarding & Team Management

Executive Summary
Invite, manage, and deactivate workers while preserving historical work records.

User Persona & Problem Statement

****Who:**** As a Govt Admin...

****Why:**** I must control who can access tasks and ensure workers who quit are disabled without losing history.

Scope (In & Out)

In Scope

- Invite workers by email.
- Auto-activation when invited email registers.
- Active staff list.
- Deactivate worker access (no deletion).

Out of Scope

- Bulk user import.
- Push notifications.

Features & Acceptance Criteria

Feature: Invite-Only Worker Onboarding

User Story: As a Govt Admin, I want to add worker emails so only approved staff can join my org.

Acceptance Criteria:

- [] Verify Admin can add worker emails from the dashboard.
- [] Verify the system sends an invite email with registration link.
- [] Verify workers can only register with an invited email.
- [] Verify first login auto-links the worker to the org and sets status ACTIVE.
- [] Verify invite revocation blocks registration.

Feature: Worker Deactivation (No Deletion)

User Story: As a Govt Admin, I want to disable workers who leave without deleting their ID.

Acceptance Criteria:

- [] Verify each active worker has a "Deactivate" action.
- [] Verify deactivation revokes login access immediately.
- [] Verify the worker record is retained (no hard delete).
- [] Verify historical tasks remain linked to the worker_id.
- [] Verify tasks assigned to a deactivated worker are unassigned and reset to REPORTED.

Functional Requirements

- `GET /workers?org_id={id}&status=ACTIVE` for active list.
- `POST /workers/invite` adds invited emails for the org.
- `DELETE /workers/invite/{id}` revokes an invite.
- On first login, invited email auto-links org_id and sets status ACTIVE.
- `PATCH /workers/{id}/deactivate` sets status INACTIVE and `deactivated_at`.
- Hard deletes are disallowed for workers.
- A worker cannot be in INVITED state for multiple organizations simultaneously.
 - Invite lifecycle: INVITED -> ACTIVE -> INACTIVE.
 - Invites expire after a configured window (default 7 days) and can be re-sent.
 - Optional allowed-domain restriction can be enforced per authority (e.g., *@authority.gov.in).

Workflow & Entity State Lifecycle

Lifecycle: INVITED -> ACTIVE -> INACTIVE.

Success Metrics

- High verification rate for active workers.

Edge Cases

- Deactivation during an active upload results in an "Unauthorized" error.

USER STORY 4: Resolution Review & Quality Control

Executive Summary

Require before/after photo review before closure and enforce transparent rejection reasons.

User Persona & Problem Statement

Who: As a Govt Admin...

Why: I need to verify fix quality and document reasons for rejection.

Scope (In & Out)

In Scope

- Side-by-side photo comparison.
- Reject fix with reason.
- Close ticket.
- Dismiss issue with reason.

Out of Scope

- Automated image quality detection.

Features & Acceptance Criteria

Feature: Resolution Review

User Story: As a Govt Admin, I want to compare before/after photos before closing.

Acceptance Criteria:

- [] Verify before and after photos are shown side-by-side.
- [] Verify reject requires a reason and returns status to IN_PROGRESS.
- [] Verify close sets status to CLOSED and finalizes analytics.

Feature: Dismissal Transparency

User Story: As a Govt Admin, I want to dismiss spam with a reason visible to citizens.

Acceptance Criteria:

- [] Verify dismissal requires a reason code or custom note.
- [] Verify dismissed tickets are removed from active board but preserved in DB.
- [] Verify rejection reason is exposed to the citizen app.

Feature: Citizen Feedback Visibility

User Story: As a Govt Admin, I want to see citizen like/dislike counts for closed issues.

Acceptance Criteria:

- [] Verify the issue detail view displays aggregated like/dislike counts.

Functional Requirements

- `POST /issues/{id}/reject_fix` requires reason.
- `POST /issues/{id}/dismiss` requires reason_code.
- `public_status_message` stores citizen-visible rejection reason.
- Closing a ticket triggers analytics update jobs.
- Dismissal reason_code enum includes: DUPLICATE, SPAM, OUT_OF_ZONE.
- Audit logs capture ISSUE_CLOSED and FIX_REJECTED with timestamps and reasons.

Dependencies

- Citizen app must display rejection reasons and collect feedback votes.

Non-Functional Requirements (Minimal)

- Once CLOSED, photos are immutable.

Edge Cases

- Corrupted "After" photo requires rejection with reason.

USER STORY 5: Authority Settings & Profile Management

Executive Summary

Allow admins to manage profile details with email/Google login only.

User Persona & Problem Statement

Who: As a Govt Admin...

Why: I need to keep my contact details updated without breaking login access.

```
## Scope (In & Out)
#### In Scope
- Edit name and phone (contact only).
- Email OTP login or Google OAuth for authentication.

#### Out of Scope
- Zone editing.
- Category management.

## Features & Acceptance Criteria
#### Feature: Profile Management
**User Story:** As a Govt Admin, I want to update profile details safely.

**Acceptance Criteria:**
- [ ] Verify email (login ID) is read-only.
- [ ] Verify restricted fields (zone_id, role) cannot be updated.
- [ ] Verify profile edits require a recent authenticated session.

## Functional Requirements
- `PATCH /admin/me` updates profile fields.
- Profile changes require a recent authenticated session (email OTP or Google OAuth).

## Edge Cases
- Stale sessions require re-authentication before profile updates.

---  

## Source: admin_dashboard.md
# EPIC: Platform Administration & Governance

## Executive Summary
A centralized control layer for onboarding authorities, configuring system-wide metadata, and monitoring platform health.

## Stakeholders & Value
#### User Personas
- **Primary Persona:** System Admin (platform owner).
- **Stakeholders:** Government Authorities, Field Workers.

#### User Value
Ensures the platform stays secure, configurable, and scalable without code changes.

## Goal & Vision
Provide reliable governance and configuration of the platform's core entities and jurisdictions.

## Scope
#### In Scope
- Authority onboarding and jurisdiction assignment.
- Role management and RBAC.
- Issue and category configuration (priority defaults, SLA).
- System monitoring and audit logs.
- Manual issue creation.
- Email OTP login and Google OAuth only (no SMS/phone OTP).
- GHMC jurisdiction selection (Hyderabad only for now).

#### Out of Scope
- Direct manipulation of citizen user data.
- Billing/financials.
- Custom zone drawing or editing (Phase 1).
```

```
## Success Metrics
- Reduced time to onboard new authorities.
- Consistent audit trail coverage.

## Stories Under This Epic
1. [USER STORY 1: Authority & Role Management](#user-story-1-authority--role-management)
2. [USER STORY 2: Issue & Category Configuration](#user-story-2-issue--category-configuration)
3. [USER STORY 3: System Monitoring & Control](#user-story-3-system-monitoring--control)

---
# USER STORY 1: Authority & Role Management

## Executive Summary
Enable secure onboarding of authorities with predefined jurisdiction boundaries.

## User Persona & Problem Statement
**Who:** As a System Admin...
**Why:** I need to add new authorities and assign their GHMC jurisdiction without manual scripts.

## Scope (In & Out)
#### In Scope
- Authority creation.
- Jurisdiction assignment from GHMC list.
- Role-based access control.

#### Out of Scope
- External SSO integration.

## Features & Acceptance Criteria
#### Feature: Authority Onboarding
**User Story:** As a System Admin, I want to onboard new authorities so they can start using the platform.

**Acceptance Criteria:**
- [ ] Verify Admin can add government departments/municipal bodies.
- [ ] Verify Admin can assign jurisdiction from a GHMC dropdown (Hyderabad only).
- [ ] Verify custom polygon drawing is not available in Phase 1.
- [ ] Verify the selected GHMC area is stored as the authority's zone_id.
- [ ] Verify authority accounts are created via invite and log in using Email OTP or Google OAuth only.

#### Feature: Role Management
**User Story:** As a System Admin, I want to manage roles so access is restricted correctly.

**Acceptance Criteria:**
- [ ] Verify Admin can create/manage roles: Government Admin, Field Worker, NGO Viewer.
- [ ] Verify strict RBAC enforcement across the platform.

## Functional Requirements
- GHMC KML/GeoJSON must be loaded into the zone catalog.
- Authority records must reference zone_id from GHMC boundaries.

## Dependencies
- GHMC KML file and boundary list required before onboarding.
```

```
# USER STORY 2: Issue & Category Configuration

## Executive Summary
Manage issue categories and workflow metadata without code changes.

## User Persona & Problem Statement
**Who:** As a System Admin...
**Why:** I need to add categories and configure defaults without developer support.

## Scope (In & Out)
### In Scope
- Category CRUD.
- Priority defaults and expected resolution time.
- Workflow state configuration.

## Features & Acceptance Criteria
### Feature: Issue Type Management
**User Story:** As a System Admin, I want to configure categories so the citizen app stays relevant.

**Acceptance Criteria:**
- [ ] Verify Admin can add, edit, or deactivate categories.
- [ ] Verify deactivated categories are hidden from the citizen app.
- [ ] Verify Admin can configure metadata (priority default, expected resolution time).

### Feature: Workflow Configuration
**User Story:** As a System Admin, I want to define valid ticket states.

**Acceptance Criteria:**
- [ ] Verify Admin can define ticket states (REPORTED, ASSIGNED, IN_PROGRESS, RESOLVED, CLOSED).
- [ ] Verify Admin can toggle "Admin Approval Required" for resolution.

---

# USER STORY 3: System Monitoring & Control

## Executive Summary
Provide visibility into platform health and data integrity.

## User Persona & Problem Statement
**Who:** As a System Admin...
**Why:** I need to see if the system is healthy and audits are complete.

## Scope (In & Out)
### In Scope
- Health stats.
- Duplicate detection monitoring.
- Manual issue creation.
- Audit logs.

## Features & Acceptance Criteria
### Feature: Platform Health Overview
**User Story:** As a System Admin, I want to see high-level stats so I can monitor usage.

**Acceptance Criteria:**
- [ ] Verify dashboard shows total reports, active users, storage usage.
- [ ] Verify duplicate detection performance indicators are visible.
- [ ] Verify flagged reporting patterns are visible.
```

```
### Feature: Manual Issue Creation
**User Story:** As a System Admin, I want to create issues manually.

**Acceptance Criteria:**
- [ ] Verify Admin can create issues on behalf of authorities.
- [ ] Verify these issues enter the standard workflow.

### Feature: Audit Trail
**User Story:** As a System Admin, I want a log of changes.

**Acceptance Criteria:**
- [ ] Verify role updates, category changes, and onboarding events are logged.
- [ ] Verify logs are immutable and viewable.

## Functional Requirements
- Central config store for permissions/workflows.
- Admin-only access to storage metadata.

--- 

## Source: authority_dash_spec.md

# EPIC: Authority Operations Dashboard & Zone Management

**Source of Truth:** This file is authoritative.
`EPIC_Authority_Operations_Dashboard.md` is a generated copy and should not be edited directly.

## Executive Summary
A web operations console for Government Admins to manage the full lifecycle of issues within their jurisdiction, from map-based triage to worker assignment and resolution review.

## Stakeholders & Value
### User Personas
- **Primary Persona:** Govt Admin (manages a jurisdiction, assigns work, reviews fixes).
- **Stakeholders:** Field Workers, Citizens, System Admin.

### User Value
Admins can see what is happening in their zone, prioritize by impact, assign efficiently, and enforce quality control.

## Goal & Vision
Provide a clear, accountable command center that reduces time-to-fix while keeping jurisdiction boundaries explicit and immutable.

## Scope
### In Scope
- Geospatial operations map with jurisdiction overlay.
- Zone-of-interest filter using GHMC areas (Hyderabad only for now).
- Kanban workflow with bulk assignment and priority setting.
- Active load view (worker task counts).
- Worker onboarding, approval, and deactivation (no deletion).
- Resolution review with audit logging.
- Public audit log integrations.
- Invitation-based authority registration (via System Admin onboarding).
- Authority profile management.
- Email OTP login and Google OAuth only (no SMS/phone OTP).
- OpenStreetMap base tiles (Mapbox optional).
- Time estimates captured from workers and displayed to authorities.
```

```

### Out of Scope
- Category management (System Admin only).
- Zone creation or editing (System Admin only).
- AI-based priority scoring.
- Push notifications.

## Success Metrics
- Reduced time from REPORTED to ASSIGNED using bulk assignment.
- Fast review turnaround for RESOLVED tickets.
- Visible reasons for dismissed tickets.

## Stories Under This Epic
1. [USER STORY 1: Geospatial Operations Map](#user-story-1-geospatial-operations-map)
2. [USER STORY 2: Kanban Dashboard & Bulk Assignment](#user-story-2-kanban-dashboard--bulk-assignment)
3. [USER STORY 3: Worker Onboarding & Team Management](#user-story-3-worker-onboarding--team-management)
4. [USER STORY 4: Resolution Review & Quality Control](#user-story-4-resolution-review--quality-control)
5. [USER STORY 5: Authority Settings & Profile Management](#user-story-5-authority-settings--profile-management)

---
# USER STORY 1: Geospatial Operations Map

## Executive Summary
Render the Admin's jurisdiction as a read-only boundary, overlay active issues as pins, and allow focused viewing by GHMC sub-areas.

## User Persona & Problem Statement
**Who:** As a Govt Admin...
**Why:** I need to see where issues are relative to my official boundary and focus on specific sub-areas quickly.

## Scope (In & Out)
### In Scope
- Zone boundary overlay (read-only).
- GHMC zone-of-interest filter (dropdown).
- Color-coded pins with report_count and priority.
- Quick-view actions from map pins.

### Out of Scope
- Drawing/editing zones.
- Heatmap density analysis (handled in analytics dashboard).
- Real-time worker tracking.

## Features & Acceptance Criteria
### Feature: Zone Context Overlay
**User Story:** As a Govt Admin, I want to see my zone clearly marked on the map so I know my operational limits.

**Acceptance Criteria:**
- [ ] Verify the map auto-zooms to the assigned zone boundary at login.
- [ ] Verify the zone is rendered as a read-only polygon (no edits).
- [ ] Verify the zone name is displayed in the header.
- [ ] Verify the zone boundary source is GHMC KML/GeoJSON for Hyderabad (no custom zones in Phase 1).

### Feature: Zone-of-Interest Filter (GHMC)
**User Story:** As a Govt Admin, I want to filter the map to a GHMC area so I can focus on hotspots.

```

Acceptance Criteria:
- [] Verify a dropdown lists GHMC areas within the assigned jurisdiction.
- [] Verify selecting an area zooms the map to that boundary and filters pins to that area.
- [] Verify clearing the selection resets the view to the full zone.

Feature: Operational Issue Visualization

User Story: As a Govt Admin, I want to see all active issues as pins so I can identify clusters.

Acceptance Criteria:

- [] Verify pins show all active tickets for this zone.
- [] Verify pin colors reflect status (REPORTED, ASSIGNED/IN_PROGRESS, RESOLVED).
- [] Verify pins are not clustered and represent unique issue IDs.
- [] Verify each pin includes report_count and priority in its quick-view.
- [] Verify filters by category/status update the map pins.

Feature: Map-to-Action

User Story: As a Govt Admin, I want to click a pin to see details and take action.

Acceptance Criteria:

- [] Verify clicking a pin opens a quick-view with photo, status, report_count, priority, and assigned worker.
- [] Verify quick-view shows worker ETA and time since issue creation when available.
- [] Verify quick-view links to the full issue detail modal.
- [] Verify quick-view includes an \"Open Ticket\" action.

UI/UX Design & User Flow

Flow: Login -> Map loads -> Zone overlay -> Optional GHMC area filter -> Select pin -> Quick view -> Open issue.

Functional Requirements

Zone Data:

- `GET /zones/{assigned_zone_id}` returns GeoJSON derived from GHMC KML.
- Zone polygons are read-only in the UI.

Issue Data:

- `GET /issues?zone_id={id}&status=active` returns active issues for pins.
- Pin payload includes: `issue_id`, `lat`, `long`, `status`, `category_icon`, `report_count`, `priority`, `created_at`, `assigned_worker`, `estimated_minutes`.

Map Provider:

- Use OpenStreetMap tiles by default; Mapbox optional.

Viewport Logic:

- Fit map to zone bounding box on load.

Deduplication Assumption:

- Backend aggregates duplicates within 5m and increments `report_count`.

Filtering:

- Map responds to global category/status filters from the dashboard.

RBAC & Permissions

Role	Can View Map	Can See Zone	Can Edit Zone
---	---	---	---
Govt Admin	Yes	Yes	No
System Admin	Yes	Yes	Yes

Dependencies

- System Admin zone data must exist before map loads.

Non-Functional Requirements (Minimal)

- Map interaction should remain responsive at expected pin volumes.

Edge Cases

- Zone missing returns a "No Zone Configured" message.
- Issues outside the polygon are shown with a warning icon.

USER STORY 2: Kanban Dashboard & Bulk Assignment

Executive Summary

A Kanban board for high-volume triage with bulk assignment, priority management, and visibility into worker estimates.

User Persona & Problem Statement

Who: As a Govt Admin...

Why: I need to assign many issues quickly and prioritize the most reported ones.

Scope (In & Out)

In Scope

- Kanban board with drag-and-drop.
- Bulk select and assignment.
- Priority setting per issue.
- Sorting by report_count, priority, and created_at.
- Worker ETA visibility.

Out of Scope

- AI-based sorting.
- Push notifications.

Features & Acceptance Criteria

Feature: Kanban Board

User Story: As a Govt Admin, I want a column view of issue status.

Acceptance Criteria:

- [] Verify cards display issue_id, category, photo thumbnail, report_count, priority, and days open.
- [] Verify columns support sorting by report_count (default), priority, or created_at.
- [] Verify drag-and-drop from REPORTED to ASSIGNED triggers assignment modal.

Feature: Smart Assignment (Single & Bulk)

User Story: As a Govt Admin, I want to assign multiple issues to a worker efficiently.

Acceptance Criteria:

- [] Verify bulk selection of multiple REPORTED cards.
- [] Verify assignment modal lists workers with active task counts.
- [] Verify assignment updates cards to ASSIGNED and stores assigned worker.
- [] Verify re-assignment is possible even when a task is IN_PROGRESS.

Feature: Priority Management

User Story: As a Govt Admin, I want to set priority per issue so crews focus on critical work.

Acceptance Criteria:

- [] Verify each card has a priority selector (P1-Critical to P4-Low).
- [] Verify priority changes are saved immediately and visible on map and Kanban.
- [] Verify priority sorting is available in the Kanban header.

```

### Feature: Time Estimate Visibility
**User Story:** As a Govt Admin, I want to see worker-provided estimates so I can track timelines.

**Acceptance Criteria:**
- [ ] Verify cards show estimated completion time when provided by worker.
- [ ] Verify the UI shows time since issue creation and time remaining vs estimate.

## UI/UX Design & User Flow
**Flow:** Load board -> Select cards -> Assign -> Monitor priority and ETA fields.

## Workflow & Entity State Lifecycle
**Lifecycle:** REPORTED -> ASSIGNED -> IN_PROGRESS -> RESOLVED -> CLOSED.

## Functional Requirements
**Data Fetching:**
- `GET /issues?zone_id={id}&status=active` for board data.

**Card Fields:**
- `issue_id`, `category_icon`, `created_at`, `report_count`, `priority`, `thumbnail_url`.
- `accepted_at`, `estimated_minutes`, `assigned_worker` (if available).

**Sorting Logic:**
- Default sort by `report_count` desc.
- Toggle to `priority` or `created_at` as needed.

**Priority Updates:**
- `PATCH /issues/{id}/priority` with payload `{ priority: P1|P2|P3|P4 }`.

**Bulk Assignment:**
- `POST /assignments/bulk` accepts `issue_ids[]` and `worker_id`, transactional.

**Worker Load Tracking:**
- Assignment modal displays active task counts per worker.

**Validation:**
- Prevent assignment for RESOLVED/CLOSED issues.

**Audit Logging:**
- Log assignment and priority changes with actor, timestamp.
- Track previous_worker and new_worker for re-assignments.

## RBAC & Permissions
| Role | Can Assign | Can Re-Assign | Can View Board |
| --- | --- | --- | --- |
| Govt Admin | Yes | Yes | Yes (own zone only) |
| Field Worker | No | No | No |

## Dependencies
- Worker list from Story 3.

## Non-Functional Requirements (Minimal)
- Kanban interactions remain responsive for expected ticket volumes.

## Edge Cases
- Deactivated workers are hidden from assignment list.
- Concurrent updates may cause conflicts and require refresh.

---
```

```

# USER STORY 3: Worker Onboarding & Team Management

## Executive Summary
Invite, manage, and deactivate workers while preserving historical work records.

## User Persona & Problem Statement
**Who:** As a Govt Admin...
**Why:** I must control who can access tasks and ensure workers who quit are disabled without losing history.

## Scope (In & Out)
#### In Scope
- Invite workers by email.
- Auto-activation when invited email registers.
- Active staff list.
- Deactivate worker access (no deletion).

#### Out of Scope
- Bulk user import.
- Push notifications.

## Features & Acceptance Criteria
#### Feature: Invite-Only Worker Onboarding
**User Story:** As a Govt Admin, I want to add worker emails so only approved staff can join my org.

**Acceptance Criteria:**
- [ ] Verify Admin can add worker emails from the dashboard.
- [ ] Verify the system sends an invite email with registration link.
- [ ] Verify workers can only register with an invited email.
- [ ] Verify first login auto-links the worker to the org and sets status ACTIVE.
- [ ] Verify invite revocation blocks registration.

#### Feature: Worker Deactivation (No Deletion)
**User Story:** As a Govt Admin, I want to disable workers who leave without deleting their ID.

**Acceptance Criteria:**
- [ ] Verify each active worker has a "Deactivate" action.
- [ ] Verify deactivation revokes login access immediately.
- [ ] Verify the worker record is retained (no hard delete).
- [ ] Verify historical tasks remain linked to the worker_id.
- [ ] Verify tasks assigned to a deactivated worker are unassigned and reset to REPORTED.

## Functional Requirements
- `GET /workers?org_id={id}&status=ACTIVE` for active list.
- `POST /workers/invite` adds invited emails for the org.
- `DELETE /workers/invite/{id}` revokes an invite.
- On first login, invited email auto-links org_id and sets status ACTIVE.
- `PATCH /workers/{id}/deactivate` sets status INACTIVE and `deactivated_at`.
- Hard deletes are disallowed for workers.
- A worker cannot be in INVITED state for multiple organizations simultaneously.
  - Invite lifecycle: INVITED -> ACTIVE -> INACTIVE.
  - Invites expire after a configured window (default 7 days) and can be re-sent.
  - Optional allowed-domain restriction can be enforced per authority (e.g., *@authority.gov.in).

## Workflow & Entity State Lifecycle
**Lifecycle:** INVITED -> ACTIVE -> INACTIVE.

## Success Metrics
- High verification rate for active workers.

```

```
## Edge Cases
- Deactivation during an active upload results in an "Unauthorized" error.

---
# USER STORY 4: Resolution Review & Quality Control

## Executive Summary
Require before/after photo review before closure and enforce transparent rejection reasons.

## User Persona & Problem Statement
**Who:** As a Govt Admin...
**Why:** I need to verify fix quality and document reasons for rejection.

## Scope (In & Out)
### In Scope
- Side-by-side photo comparison.
- Reject fix with reason.
- Close ticket.
- Dismiss issue with reason.

### Out of Scope
- Automated image quality detection.

## Features & Acceptance Criteria
### Feature: Resolution Review
**User Story:** As a Govt Admin, I want to compare before/after photos before closing.

**Acceptance Criteria:**
- [ ] Verify before and after photos are shown side-by-side.
- [ ] Verify reject requires a reason and returns status to IN_PROGRESS.
- [ ] Verify close sets status to CLOSED and finalizes analytics.

### Feature: Dismissal Transparency
**User Story:** As a Govt Admin, I want to dismiss spam with a reason visible to citizens.

**Acceptance Criteria:**
- [ ] Verify dismissal requires a reason code or custom note.
- [ ] Verify dismissed tickets are removed from active board but preserved in DB.
- [ ] Verify rejection reason is exposed to the citizen app.

### Feature: Citizen Feedback Visibility
**User Story:** As a Govt Admin, I want to see citizen like/dislike counts for closed issues.

**Acceptance Criteria:**
- [ ] Verify the issue detail view displays aggregated like/dislike counts.

## Functional Requirements
- `POST /issues/{id}/reject_fix` requires reason.
- `POST /issues/{id}/dismiss` requires reason_code.
- `public_status_message` stores citizen-visible rejection reason.
- Closing a ticket triggers analytics update jobs.
- Dismissal reason_code enum includes: DUPLICATE, SPAM, OUT_OF_ZONE.
- Audit logs capture ISSUE_CLOSED and FIX_REJECTED with timestamps and reasons.

## Dependencies
- Citizen app must display rejection reasons and collect feedback votes.
```

```

## Non-Functional Requirements (Minimal)
- Once CLOSED, photos are immutable.

## Edge Cases
- Corrupted "After" photo requires rejection with reason.

---

# USER STORY 5: Authority Settings & Profile Management

## Executive Summary
Allow admins to manage profile details with email/Google login only.

## User Persona & Problem Statement
**Who:** As a Govt Admin...
**Why:** I need to keep my contact details updated without breaking login access.

## Scope (In & Out)
#### In Scope
- Edit name and phone (contact only).
- Email OTP login or Google OAuth for authentication.

#### Out of Scope
- Zone editing.
- Category management.

## Features & Acceptance Criteria
#### Feature: Profile Management
**User Story:** As a Govt Admin, I want to update profile details safely.

**Acceptance Criteria:**
- [ ] Verify email (login ID) is read-only.
- [ ] Verify restricted fields (zone_id, role) cannot be updated.
- [ ] Verify profile edits require a recent authenticated session.

## Functional Requirements
- `PATCH /admin/me` updates profile fields.
- Profile changes require a recent authenticated session (email OTP or Google OAuth).

## Edge Cases
- Stale sessions require re-authentication before profile updates.

---

## Source: citizen_dashboard.md

# EPIC: Citizen Issue Reporting & Resilience

## Executive Summary
A mobile-first reporting flow that captures accurate, verifiable infrastructure issues with strong location integrity, offline resilience, and clean evidence handling.

## Stakeholders & Value
#### User Personas
- **Primary Persona:** Citizen (concerned resident, daily commuter).
- **Stakeholders:** Government Admins, Field Workers, System Admin.

#### User Value
Citizens can report issues in seconds with high trust that their report is accurate, accepted, and tracked to completion.

```

Goal & Vision
Deliver a seamless, location-locked reporting experience that avoids spam, enforces evidence integrity, and supports low-connectivity conditions.

Scope
In Scope
- GPS-locked reporting with accuracy indicators and a 5m confirmation constraint.
- Photo upload (Camera or Gallery) with EXIF validation (time <= 7 days; location within 5m of device location).
- Silent duplicate detection and report-count aggregation (no duplicate disclosure in reporting UI).
- Offline store-and-forward.
- Dynamic category fetch from backend.
- "My Reports" dashboard with status tracking and resolution feedback (like/dislike).
- Email OTP login and Google OAuth only (no SMS/phone OTP).
- OpenStreetMap base tiles (Mapbox optional for enhanced tiles).

Out of Scope
- SMS/Email notifications (in-app only).
- Video reporting.
- Remote pin placement outside the 5m radius.
- City-wide analytics or heatmaps (handled in analytics spec).

Success Metrics
- High upload success rate in low-bandwidth zones.
- Reduced duplicate ticket creation via backend deduplication.
- Fast report submission time.

Stories Under This Epic
1. [USER STORY 1: Location Confirmation & Silent Duplicate Check](#user-story-1-location-confirmation--silent-duplicate-check)
2. [USER STORY 2: Create New Infrastructure Report](#user-story-2-create-new-infrastructure-report)
3. [USER STORY 3: Offline Reporting & Sync](#user-story-3-offline-reporting--sync)
4. [USER STORY 4: Citizen Dashboard - My Reports & Feedback](#user-story-4-citizen-dashboard---my-reports--feedback)

USER STORY 1: Location Confirmation & Silent Duplicate Check

Executive Summary
Confirm the user's current location within a strict 5m radius and perform duplicate detection in the background without exposing existing issues on the reporting screen.

User Persona & Problem Statement
Who: As a Citizen standing in front of a pothole...
Why: I want to report quickly without seeing other people's reports, while the system still prevents duplicates behind the scenes.

Scope (In & Out)
In Scope
- Location confirmation map with 5m lock.
- GPS accuracy validation.
- Silent duplicate check and report-count incrementing.
- OpenStreetMap base tiles (Mapbox optional).

Out of Scope
- Displaying nearby issue markers or report counts on the reporting screen.

- Editing or commenting on existing reports.

Features & Acceptance Criteria

Feature: Location Confirmation (5m Lock)

User Story: As a Citizen, I want to confirm my exact location, so the report is tied to the correct spot.

Acceptance Criteria:

- [] Verify that when the user opens the report flow, the map centers on current GPS location.
- [] Verify that the UI shows a "Confirm Location" step with address text (Swiggy-like confirmation).
- [] Verify that the pin is fixed to current location and cannot be moved beyond 5m.
- [] Verify that if GPS accuracy is worse than 5m, the user is prompted to retry or wait.

Feature: Silent Duplicate Check

User Story: As a System, I want to detect nearby duplicates without showing them to the user, so reporting remains simple while data stays clean.

Acceptance Criteria:

- [] Verify that the reporting screen does NOT display any existing issue markers or report counts.
- [] Verify that the system checks for existing issues within a 5m radius before creating a new issue.
- [] Verify that if a match is found, the report is appended to the existing issue and the report_count increments by 1.
- [] Verify that the user sees a normal success confirmation without disclosure that the issue already exists.
- [] Verify that if no match is found, a new issue is created.

UI/UX Design & User Flow

Flow: Home -> Report Issue -> Confirm Location -> Capture>Select Photo -> Submit -> Success.

Functional Requirements

Location Services:

- Subscribe to the device GPS provider.
- Ignore cached locations older than a defined freshness threshold.

Duplicate Matching (Silent):

- Call `GET /issues/nearby` with `latitude`, `longitude`, `radius=5`, `status_exclude=CLOSED,RESOLVED`.
- Do not render the response in the UI. Use it only for matching.
- If a match exists, use `POST /issues/{id}/evidence` to append evidence and increment `report_count` (evidence_type=REPORT).

Map Constraints:

- Validate that distance between device GPS and confirmed pin is <= 5m.

Map Provider:

- Use OpenStreetMap tiles by default; Mapbox can be configured as an optional provider.

Edge Cases

- **GPS Drift:** If GPS drift exceeds 5m, the user is blocked from submitting until accuracy improves.
- **Hidden Duplicate:** Users may unknowingly report an existing issue; backend merges into the existing ticket.
- **Permission Denied:** If location permissions are denied, block access to the report flow.

```
# USER STORY 2: Create New Infrastructure Report

## Executive Summary
Capture high-quality evidence for a new report while enforcing EXIF-based validation for time and location integrity.

## User Persona & Problem Statement
**Who:** As a Citizen...
**Why:** I want to report a hazard with a valid photo so it gets accepted and routed correctly.

## Scope (In & Out)
### In Scope
- GPS-only location capture.
- Photo upload from Camera or Gallery.
- EXIF validation (timestamp <= 7 days; location within 5m of device GPS).
- Category selection and optional description.

### Out of Scope
- Video upload.
- Audio notes.

## Features & Acceptance Criteria
### Feature: Evidence & Location Capture
**User Story:** As a Citizen, I want to submit a photo and category so the correct department is notified.

**Acceptance Criteria:**
**Location:**
- [ ] Verify that the issue location is set to device GPS coordinates.
- [ ] Verify that the UI displays an accuracy indicator (e.g., "Accuracy: +/- 5m").
- [ ] Verify that the user cannot submit if the location accuracy is worse than 5m.

**Photo:**
- [ ] Verify the user can select an image from the Gallery or capture via Camera.
- [ ] Verify at least one photo is mandatory.
- [ ] Verify EXIF timestamp is within 7 days of submission.
- [ ] Verify EXIF location is within 5m of current device GPS.
- [ ] Verify photos with missing/invalid EXIF metadata are rejected with a recapture prompt.

**Category:**
- [ ] Verify categories are fetched dynamically from `GET /categories`.
- [ ] Verify the form cannot be submitted without a category selected.

**Assignment:**
- [ ] Verify backend calculates zone based on lat/long and assigns to the correct authority.

## UI/UX Design & User Flow
**Flow:** Capture>Select Photo -> Report Details Form -> Submit -> Success or Error.

## Functional Requirements
**Category Management:**
- Call `GET /categories` on form load.
- Cache the response for a defined duration.
- Fallback to a safe default list if API fails.

**Data Payload Construction:**
```

- Submit a multipart/form-data POST to `/report` with fields:
 - `latitude`, `longitude`, `accuracy_meters`
 - `category_id`, `description`, `photo`
 - `timestamp`, `exif_timestamp`, `exif_lat`, `exif_long`

****Image Processing:****

- Compress images to stay within size limits.
- Device GPS remains the source of truth for issue location.
- EXIF metadata is used only for validation (not for location placement).

****Backend Assignment Logic:****

- Perform point-in-polygon to determine assigned zone.
- Create ticket with status `REPORTED` and assign to organization.

Edge Cases

- ****Gallery Mismatch:**** Gallery photos from another location fail EXIF validation and are rejected.
- ****Offline Capture Delay:**** If upload is delayed beyond 7 days, the backend rejects the report.

USER STORY 3: Offline Reporting & Sync

Executive Summary
 Ensure reports are not lost when connectivity drops by storing payloads locally and syncing later.

User Persona & Problem Statement

****Who:**** As a Citizen in a low-connectivity zone...

****Why:**** I do not want to lose a report when the network fails.

Scope (In & Out)

In Scope

- Local persistence of reports.
- Background retry logic.
- Pending upload status indicators.

Out of Scope

- Peer-to-peer sync.

Features & Acceptance Criteria

Feature: Store-and-Forward

****User Story:**** As a System, I want to queue reports locally if the API is unreachable, so data is not lost.

****Acceptance Criteria:****

- [] Verify failed `POST /report` calls are saved to local storage.
- [] Verify "Pending Upload" status is shown in "My Reports".
- [] Verify background retry occurs on connectivity restore.
- [] Verify local copies are cleared after successful upload.

UI/UX Design & User Flow

****Flow:**** Submit -> Offline Detected -> Saved to Outbox -> Auto Sync -> Status Updated.

Functional Requirements

****Local Persistence:****

- Use a local DB (SQLite/Realm/AsyncStorage).
- Store payload with `retry_count` initialized to 0.

****Network Detection & Retry:****

- Listen for connectivity changes.
- Retry queued uploads on reconnect.

```

**Storage Management (Minimal NFR):**
- Cap local storage and warn user before accepting new offline reports.

## Edge Cases
- **Long Offline Period:** Reports older than 7 days will fail EXIF validation when synced.

---

# USER STORY 4: Citizen Dashboard - My Reports & Feedback

## Executive Summary
Provide a personal dashboard to track issue status and submit feedback on resolved work.

## User Persona & Problem Statement
**Who:** As a Citizen...
**Why:** I want to see progress on my reports and rate the quality of fixes.

## Scope (In & Out)
### In Scope
- List of user reports with status tracking.
- Detail view with before/after photos.
- Like/Dislike feedback on resolved or closed issues.

### Out of Scope
- Direct chat with authorities.
- City-wide analytics maps.

## Features & Acceptance Criteria
### Feature: Status Tracking
**User Story:** As a Citizen, I want to track my reported issues.

**Acceptance Criteria:**
- [ ] Verify the list shows issues submitted by the logged-in user.
- [ ] Verify each item shows status: REPORTED -> ASSIGNED -> IN_PROGRESS -> RESOLVED -> CLOSED.
- [ ] Verify resolved/closed items show the "After" photo.
- [ ] Verify dismissed items show the rejection reason if available.

### Feature: Resolution Feedback (Like/Dislike)
**User Story:** As a Citizen, I want to rate the completed fix so authorities can be evaluated.

**Acceptance Criteria:**
- [ ] Verify Like/Dislike controls appear only for RESOLVED or CLOSED issues.
- [ ] Verify a user can submit only one vote per issue.
- [ ] Verify the vote is stored with user_id and issue_id.
- [ ] Verify aggregated like/dislike counts are available to authority and analytics dashboards.

## UI/UX Design & User Flow
**Flow:** My Reports -> Select Report -> View Timeline -> Submit Like/Dislike.

## Functional Requirements
### Data Retrieval:
- `GET /user/{user_id}/reports` with pagination.
- Sort by `created_at` desc.

### Status Enumeration:
- REPORTED -> "Report Sent"
- ASSIGNED -> "Authority Notified"
- IN_PROGRESS -> "Work Started"

```

- RESOLVED -> "Fix Verified"
- CLOSED -> "Closed"
- DISMISSED -> "Rejected" (show reason)

Feedback Submission:

- `POST /issues/{id}/feedback` with payload `{ user_id, vote: LIKE|DISLIKE }`.
- Prevent duplicate votes per user.

Success Metrics

- Higher citizen re-engagement after resolution.
- Increased feedback submission rate for closed issues.

Source: combined_project_spec.md

Comprehensive Master Specification: Citizen Road Infrastructure Reporting System

1. Project Overview & Strategic Objective

The Citizen Road Infrastructure Reporting System is an end-to-end civic technology platform designed to transform road maintenance into a transparent, accountable, and data-driven process. By connecting citizens, municipal authorities, and field maintenance teams, the system ensures that infrastructure failures (potholes, drainage issues, etc.) are captured with high precision and resolved with verifiable proof.

2. The "Rules of the Road": System-Wide Logic

These foundational rules govern the interaction between all five project modules.

2.1 The 5-Player Ecosystem

- * **Citizen:** Primary reporter of issues; provider of "Before" evidence.
- * **Govt Admin (Zone Manager):** Operational controller; manages triage, assignment, and quality review for a specific ward/zone.
- * **Field Worker:** On-the-ground execution staff; provider of "After" resolution evidence.
- * **System Admin:** Platform owner; manages the onboarding of authorities and global configuration.
- * **Analytics Viewer (NGOs/Public):** External monitors; track performance and city health via read-only metrics.

2.2 The Unified Issue Lifecycle

Every issue must transition through these states. Skipping steps is forbidden.

1. **REPORTED (Start):** Citizen uploads photo + location. System performs a silent 5m radius duplicate check.
2. **ASSIGNED:** Govt Admin selects a verified worker for the task.
3. **IN_PROGRESS:** Worker clicks "Start Work" upon arrival. Ticket is locked to that worker.
4. **RESOLVED:** Worker uploads mandatory "After" photo proof.
5. **CLOSED (End):** Terminal state after Admin approval. Data is archived for analytics.
6. **DISMISSED:** Admin flags the report as spam, duplicate, or out-of-jurisdiction.

2.3 Shared Data Properties (The Data Dictionary)

- * **Issue ID:** Unique UUID tracking number.
- * **Location:** GPS Lat/Long, Zone ID (GeoJSON polygon), and human-readable Address.
- * **Status:** {REPORTED, ASSIGNED, IN_PROGRESS, RESOLVED, CLOSED, DISMISSED}.
- * **Evidence:** `photo_report` (Before) and `photo_resolve` (After).

```

*   **Scoring Logic:**  

*     **report_count:** Number of citizens reporting the same issue (duplicate aggregation).  

*       **Admin Severity:** P1 (Critical) to P4 (Low).  

*       **Priority Score:** Determined by admin priority and report_count.  

---  

## 3. Module 1: Platform Administration (System Admin)  

**Objective:** Centralized control layer for authority onboarding and system-wide governance.  

  

### 3.1 Functional Requirements  

*   **Authority Management:** Add/manage government departments and municipal bodies. Assign jurisdictional scopes (City, Zone, Ward).  

*   **Role-Based Access Control (RBAC):** Create and manage roles for Govt Admins, Field Workers, and NGO Viewers.  

*   **Issue Configuration:** Manage the global category list (Add/Edit/Deactivate Potholes, Streetlights, etc.). Configure category metadata (Priority, expected resolution time).  

*   **Workflow Engine:** Define valid ticket states and toggle mandatory Admin approval before resolution.  

*   **System Monitoring:** View high-level stats (Total reports, active users, storage usage). Monitor duplicate detection performance.  

*   **Data Integrity:** Flag abnormal reporting patterns or system misuse.  

*   **Manual Creation:** Admin-only interface for adding internal inspection issues.  

*   **Audit Trail:** Maintain logs of all platform-level changes (Role updates, category changes, onboarding events).  

  

### 3.2 Technical Requirements  

*   **Configuration Store:** Centralized service for permissions and workflows; changes propagate without downtime.  

*   **Storage Management:** Admin-only access to MinIO metadata (PII redacted). Monitor storage growth trends.  

  

### 3.3 User Flow: Administration Lifecycle  

1. **Authority Onboarding:** Admin adds authority -> defines jurisdiction boundary -> authority becomes available.  

2. **System Configuration:** Admin updates categories and resolution rules -> changes apply to future reports.  

3. **User Setup:** Admin creates/approves accounts -> users gain access to dashboards.  

4. **Monitoring:** Admin tracks system usage and platform health.  

---  

## 4. Module 2: Citizen Issue Reporting (Mobile App)  

**Objective:** High-fidelity data ingestion with duplicate prevention and offline resilience.  

  

### 4.1 Functional Requirements  

*   **Silent Duplicate Check:** System checks within 5m and merges into an existing issue without showing nearby markers.  

*   **Duplicate Aggregation:** If a match is found, append evidence and increment report_count.  

*   **Location Services:** GPS-locked reporting with Accuracy Indicators ( $\pm 5m$ ). Fine location (GNSS) required; cached data older than X seconds ignored.  

*   **Evidence Collection:** Mandatory photo capture (Camera or Gallery). App strips EXIF metadata for privacy but enforces device GPS location.  

*   **Category Fetching:** Dynamic list retrieval (`GET /categories`) with local caching and "General" fallback.  

*   **Offline Support:** Reports queued in local DB (SQLite/AsyncStorage) when signal is lost.

```

- * **Sync Manager:** Connectivity listener triggers background retries; clearing local copies upon success.
- * **Citizen Dashboard:** Status tracking via timeline/stepper; side-by-side view of Before/After photos for resolved tasks.
- * **Resolution Feedback:** Citizens can like/dislike RESOLVED/CLOSED issues only (no upvoting on report creation).

4.2 User Flows

- * **Report New Issue:** Fab/Report button -> Map loads focused on GPS -> No match found -> Select Category -> Capture Photo -> Submit -> Success redirect to Dashboard.
- * **Duplicate Merge:** Report submitted -> Backend detects nearby issue -> Evidence appended -> report_count increments.
- * **Offline Sync:** Submit report -> Network fails -> "Saved to Outbox" toast -> User moves to WiFi -> System detects sync event -> Status updates to "Reported".

4.3 Edge Cases

- * **GPS Drift:** If drift >5m, users may miss existing tickets. Issues plotted slightly outside polygons are flagged visually.
- * **Photo Mismatch:** Gallery photos from different locations are overridden by device GPS source of truth.
- * **Permission Denial:** GPS refusal blocks the reporting flow with a "Permission Required" modal.
- * **Indefinite Offline:** Reports remain pending until connectivity returns. Storage caps (FIFO not permitted) prevent overfilling.
- * **Hidden duplicate:** User unknowingly reports an existing issue; backend merges it into the same ticket.

5. Module 3: Authority Operations (Govt Dashboard)

Objective: Command center for triage, assignment, and quality control.

5.1 Functional Requirements

- * **Geospatial Operations Map:** Zone boundary overlay (GeoJSON) with color-coded pins (Red: Reported, Yellow: Assigned/In-Progress, Green: Resolved).
- * **Map Interaction:** Click pin for "Quick View" card (Photo, ID, Status) with "Open Ticket" modal trigger.
- * **Kanban Dashboard:** Status columns (Reported, Assigned, In-Progress, Resolved). Sort by Date (Oldest First) or report_count (Urgency).
- * **Bulk Assignment:** Select multiple "Reported" cards -> Assign to Worker -> Modal shows Worker Load (active task count) -> Transnational (Atomic) backend update.
- * **Team Management:** Invite workers by email; auto-activate on first login. Invites expire after a configured window (default 7 days), can be re-sent, and can enforce allowed domains. Remove/Deactivate active staff; triggers immediate session revocation and task reset (Assigned -> Reported).
- * **Resolution Review:** Mandatory photo comparison. "Reject Fix" requires reason text (Pushed to Citizen). "Close Ticket" finalizes analytics.
- * **Spam Dismissal:** Standard reason codes (Duplicate, Fake, Out-of-Zone) + custom notes.
- * **Profile:** Edit name, phone (contact only), and password (email users only).

5.2 Technical Requirements

- * **Performance:** Map interaction maintains {TARGET_FPS}; Zone rendering < {MAX_MS}.
- * **Scalability:** Kanban DOM virtualization for large backlogs.
- * **Integrity:** Bulk assignments are atomic; rollback on subset failure.

5.3 User Flows

- * **Bulk Triage:** Dashboard load -> Select cards in Reported -> Click Bulk Assign -> Select worker from dropdown (seeing load) -> Confirm -> Cards move to

Assigned.

- * **Resolution Review:** Click "Resolved" card -> Inspect Photos side-by-side -> Click Reject -> Enter "Debris left" -> Submit -> Ticket returns to In-Progress.
- * **Team Setup:** Notification on "Team" tab -> Review pending request -> Click Approve -> Worker added to dropdown.

5.4 Edge Cases

- * **Concurrent Updates:** card updated by another Admin during bulk action; requires optimistic UI and error handling.
- * **No Zone Configured:** API 404 for Zone GeoJSON shows error overlay.
- * **Broken Evidence:** Corruption in resolution photos forces "Reject Fix" with "Corrupted Photo" reason.
- * **Worker Deactivation:** If worker removed mid-upload, action fails with "Unauthorized".

6. Module 4: Field Force Mobility (Worker App)

Objective: Digital task queue and verifiable resolution submission.

6.1 Functional Requirements

- * **Onboarding:** Invite-only registration via pre-approved email. Verified status required for task access (HTTP 403 fallback).
- * **Task Queue:** List/Map views with distance sorting. Card displays Issue ID, Category Icon, Photo Thumbnail, and "Days Open".
- * **Navigation:** "Get Directions" triggers external map apps to the specific GPS pin.
- * **Status Lifecycle:** Manual "Start Work" button locks ticket and updates status/timestamp.
- * **Evidence Submission:** Resolution requires in-app camera capture (Gallery forbidden). Resolution notes optional.
- * **Performance History:** Profile counts of total resolved tasks; historical list of Resolved/Closed items.

6.2 User Flow: Task Execution

1. **Acceptance:** Worker opens "My Tasks" -> Selects card -> Navigates to site.
2. **Start:** Worker arrives -> Clicks "Start Work" -> Status Badge updates to In-Progress.
3. **Resolve:** Repairs finished -> Taps "Resolve" -> Capture photo -> Submit -> Ticket moves to Resolved; Citizen notified.

6.3 Edge Cases

- * **Misleading Coordinates:** Citizen reported GPS may be inaccurate; worker relies on "Before" photo for visual search.
- * **Data Limitations:** High-res photo loading may fail on limited data plans.
- * **Forgetfulness:** Worker forgets to click "Start Work", impacting performance metrics.
- * **Accidental Tap:** "Start Work" clicked by mistake; requires resolution or Admin help (no undo).
- * **Upload Failure:** Resolve fails mid-upload; state remains In-Progress with "Retry" option.

7. Module 5: Aggregated Public Analytics (Visualization Dashboard)

Objective: End-to-end transparency and performance visualization.

7.1 Functional Requirements

- * **Issue Pipeline:** Funnel showing live volume transition (Reported -> In-Progress -> Resolved).
- * **Transparency Logs:** Public access to side-by-side Before/After photos for all closed tickets.

- * **Advanced Scoping:** Filters for Issue Type, Geography (Ward/Zone/GPS Radius), and Temporal Range (Historical/Seasonal).
- * **NGO Features:**
 - * **Heatmap of Neglect:** Geospatial visualization of unresolved vs resolved concentrations.
 - * **Average Resolution Time (ART):** Neighborhood-level performance metrics.
 - * **Data Export:** CSV/JSON downloads for advocacy research.

7.2 Technical Requirements

- * **Aggregation Engine:** Background jobs calculate ART and volume trends for instant loading.
- * **Separation:** Read-only architecture physically or logically separate from the Write DB.
- * **Privacy:** Strict data masking (PII redaction) for citizens and worker names.
- * **Accessibility:** Multilingual support (Bhashini integration).

7.3 User Flow: Citizen Reporting & Aggregation

1. **Ingestion:** Citizen identifies issue -> uploads photo/GPS -> ticket created.
2. **Operations:** Admin reviews -> system runs de-duplication -> Admin assigns to worker.
3. **Resolution:** Worker arrives -> updates status -> submits resolution photo.
4. **Analytics:** Background engine pulls resolved data -> Heatmap and Funnel update instantly.

8. Integrated Failure Recovery & Edge Case Master List

Category	Issue	System Response / Mitigation
:--- :--- :---		
GPS	Accuracy/Drift	Accuracy indicators ($\pm 15m$); ignore cached data; flag out-of-bounds pins.
Connectivity	Signal Loss	Store-and-forward (Local DB); Background retry; "Cloud with Slash" status icon.
Operations	Concurrent Edit	Optimistic UI; Server-side status validation before state change.
Evidence	Poor Quality	Admin "Reject Fix" loop; side-by-side photo audit; mandatory reason input.
Personnel	Invite Delays	Dashboard indicators for outstanding invites.
Security	Access Revocation	Immediate session token invalidation; automatic task reset to unassigned.
Data	Duplicates	Silent 5m dedup; append evidence and increment report_count.
Infrastructure	Missing Config	Error overlay if Zone GeoJSON or Categories are not defined by System Admin.

Source: rules_of_the_road.md

Project Coordination Spec: Rules of the Road

Objective: Ensure all modules (Citizen, Authority, Worker, System Admin, Analytics) follow the same lifecycle, data rules, and validation standards.

1. The 5 Players

Role	Description	Key Function/Needs
Citizen	Reports issues.	Accurate location + verified photo.
Govt Admin	Manages a zone.	Assigns work, sets priority.
Field Worker	Fixes issues.	Needs verified evidence capture.
System Admin	Platform owner.	Onboards authorities, configures categories.
Analytics Viewer	Public/NGO.	Read-only transparency data.

2. Issue Lifecycle (Single Source of Truth)

1. **REPORTED (Start)**
 - **Action:** Citizen submits location + photo.
 - **Rule:** System performs silent duplicate check within 5m.
 - If duplicate -> append evidence and increment report_count.
 - If new -> create ticket.
2. **ASSIGNED**
 - **Action:** Govt Admin assigns a worker.
3. **IN_PROGRESS**
 - **Action:** Worker taps "Start Work".
4. **RESOLVED**
 - **Action:** Worker uploads verified "After" photo.
5. **CLOSED (End)**
 - **Action:** Admin closes after review.

3. Core Properties of Issues

- ### A. Identity & Location
- **Issue ID:** Unique identifier.
 - **Category:** Type of issue.
 - **Status:** REPORTED, ASSIGNED, IN_PROGRESS, RESOLVED, CLOSED, DISMISSED.
 - **GPS Location:** Exact latitude/longitude.
 - **Zone ID:** GHMC area (Hyderabad only for now).
 - **Address:** Human-readable address.

B. Evidence & Verification

- **Report Photo:** Required at creation.
- **Resolve Photo:** Required to resolve.
- **EXIF Rules:**
 - Timestamp must be within 7 days of submission.
 - EXIF location must be within 5m of device GPS / issue location.
 - Missing/invalid EXIF blocks submission.

C. People & Ownership

- **Reporter ID:** Citizen who reported.
- **Assigned Worker ID:** Worker handling the issue.
- **Organization ID:** Govt authority responsible.

D. Priority & Metrics

- **report_count:** Aggregated reports for the same issue.
- **Admin Priority:** P1-Critical to P4-Low.
- **Estimated Completion:** Provided by worker on acceptance.
- **Days Open:** Time since REPORTED.
- **Citizen Feedback:** Like/Dislike counts for resolved/closed issues.

4. Authentication & Access

- Email OTP login and Google OAuth only (no SMS/phone OTP) for Citizen, Worker, Authority, and System Admin logins.
- Role-based access control applies across all dashboards.

5. Worker Onboarding (Invite-Only)

- Govt Admin adds worker emails from the Authority dashboard.
- Workers can only register with an invited email.
- First login auto-links the worker to the org and sets status ACTIVE.
- Invites can be revoked to block registration.
- Invite lifecycle: INVITED -> ACTIVE -> INACTIVE.
- Invites expire after a configured window (default 7 days) and can be re-sent.
- Optional allowed-domain restriction (e.g., *@authority.gov.in) can be enforced.

6. Jurisdiction & Zones

- Use GHMC boundaries for Hyderabad (current target).
- Zone creation/editing is System Admin only.
- No custom zone drawing in the initial release.

7. Duplicate Detection & Report Aggregation

- Duplicate check is silent (no display to citizen during reporting).
- Match radius: 5m.
- If duplicate found, append evidence and increment report_count.
- Future phase: add computer vision matching to reduce false positives.

8. Maps & Base Tiles

- OpenStreetMap is the default base layer.
- Mapbox can be configured optionally for enhanced tiles.

9. Worker Lifecycle Rules

- Workers can be DEACTIVATED but never hard-deleted.
- Deactivated workers lose access, but historical tasks remain linked.

10. Citizen Feedback Loop

- Citizens can like/dislike resolved/closed issues only.
- Feedback is visible in authority and analytics dashboards.
- No upvoting on report creation; report_count is only from duplicate aggregation.

Source: visualisation_dashboard.md

EPIC: Aggregated Public Analytics & Transparency

```
## Executive Summary
A read-only analytics portal that visualizes issue lifecycle data, hotspot density, and resolution performance for public accountability.

## Stakeholders & Value
### User Personas
- **Primary Persona:** NGO / Public Viewer.
- **Stakeholders:** Government Officials.

### User Value
Enables transparent, city-wide visibility into infrastructure performance and unresolved hotspots.

## Goal & Vision
Turn government operations into transparent, data-backed public insights.

## Scope
### In Scope
- Lifecycle funnel visualization.
- Before/After photo gallery.
- Filtering by location, time, and issue type.
- Heatmaps, hotspots, and redzone overlays.
- Data export (CSV/JSON).
- Zone-of-interest selection (GHMC areas, Hyderabad only for now).
- Kepler.gl-based map layers.
- OpenStreetMap base tiles (Mapbox optional).

### Out of Scope
- Real-time worker tracking.
- Write access to issues.

## Success Metrics
- High traffic on transparency portal.
- Increased downloads of open data exports.

## Stories Under This Epic
1. [USER STORY 1: End-to-End Lifecycle Visualization](#user-story-1-end-to-end-lifecycle-visualization)
2. [USER STORY 2: Advanced Filtering & Scoping](#user-story-2-advanced-filtering--scoping)
3. [USER STORY 3: NGO & Advocacy Features](#user-story-3-ngo--advocacy-features)

---
# USER STORY 1: End-to-End Lifecycle Visualization

## Executive Summary
Show the journey of issues from REPORTED to CLOSED with verifiable photos and counts.

## User Persona & Problem Statement
**Who:** As a Citizen/NGO...
**Why:** I want evidence and numbers, not just summaries.

## Scope (In & Out)
### In Scope
- Pipeline funnel.
- Before/After side-by-side.
- Report_count visibility (aggregated).

## Features & Acceptance Criteria
### Feature: Issue Pipeline
**User Story:** As a Viewer, I want to see the volume of issues at each stage.
```

Acceptance Criteria:
- [] Verify funnel shows REPORTED -> IN_PROGRESS -> RESOLVED -> CLOSED.
- [] Verify counts are accurate and near real-time.

Feature: Before/After Transparency

User Story: As a Viewer, I want to compare before and after photos.

Acceptance Criteria:

- [] Verify public access to before/after photos for closed tickets.
- [] Verify photos are displayed side-by-side.
- [] Verify sensitive data is redacted if required.

UI/UX Design & User Flow

Flow: Landing -> Funnel -> Select Resolved -> Gallery -> Compare.

USER STORY 2: Advanced Filtering & Scoping

Executive Summary

Provide multi-dimensional filtering for detailed analysis of problem areas.

User Persona & Problem Statement

Who: As a Researcher...

Why: I need to compare performance across wards and time windows.

Scope (In & Out)

In Scope

- Type filter.
- Geo filter by GHMC areas.
- Time filter.

Features & Acceptance Criteria

Feature: Multi-Dimensional Filtering

User Story: As a Viewer, I want to filter by type, location, and time.

Acceptance Criteria:

- [] Verify filter by issue type (potholes, drainage).
- [] Verify drill-down by GHMC area via dropdown.
- [] Verify temporal filters (custom ranges, seasons).

UI/UX Design & User Flow

Flow: Dashboard -> Filters -> Select GHMC area + type + date range -> Charts update.

USER STORY 3: NGO & Advocacy Features

Executive Summary

Provide hotspot and redzone analytics plus exportable datasets for advocacy.

User Persona & Problem Statement

Who: As an NGO...

Why: I need hard data to advocate for neglected areas.

Scope (In & Out)

In Scope

- Hotspot and redzone overlays.
- Average resolution time metrics.
- Data export.

Features & Acceptance Criteria

Feature: Hotspots & Redzones

User Story: As an NGO, I want to see hotspot and redzone clusters of unresolved issues.

Acceptance Criteria:

- [] Verify hotspot layer highlights dense clusters of REPORTED/IN_PROGRESS issues.
- [] Verify redzones show areas with high unresolved ratios.
- [] Verify layers can be toggled independently.

Feature: Average Resolution Time (ART)

User Story: As an NGO, I want to see how fast issues are fixed.

Acceptance Criteria:

- [] Verify ART is shown by GHMC area and issue type.
- [] Verify dismissed tickets are excluded from ART.

Feature: Data Export

User Story: As an NGO, I want to download filtered data.

Acceptance Criteria:

- [] Verify export supports CSV and JSON.
- [] Verify exports respect active filters.

Functional Requirements

- Use Kepler.gl for map layers (hotspots, redzones, heatmaps).
- Read-only access to analytics DB views.
- Default base tiles from OpenStreetMap (Mapbox optional).

Source: worker_epic.md

EPIC: Field Force Mobility & Task Execution Suite

Executive Summary

A mobile workflow for field workers to accept tasks, verify location, execute repairs, and submit verified evidence of completion.

Stakeholders & Value

User Personas

- **Primary Persona:** Field Worker (on-the-ground staff).
- **Stakeholders:** Govt Admin, Citizens, System Admin.

User Value

Workers get a clear task queue, directions, and a simple way to prove work completion with verified evidence.

Goal & Vision

Enable a high-accountability, low-friction field execution process with strong location integrity.

Scope

In Scope

- Worker onboarding and org linking.
- Task queue with map view and location verification.
- Task acceptance with worker-provided time estimates.
- Status transitions and timestamps.
- Mandatory camera capture for resolution evidence with EXIF validation.
- Email OTP login and Google OAuth only (no SMS/phone OTP).
- OpenStreetMap base tiles for map/directions (Mapbox optional).

Out of Scope

- Native navigation.

- Payroll/inventory.
- Live GPS tracking.

Success Metrics

- Faster transitions from ASSIGNED to IN_PROGRESS.
- Lower fix rejection rates.

Stories Under This Epic

1. [USER STORY 1: Invite-Only Onboarding & Org Linking](#user-story-1-invite-only-onboarding--org-linking)
2. [USER STORY 2: Task Queue, Acceptance & Location Verification](#user-story-2-task-queue-acceptance--location-verification)
3. [USER STORY 3: In-Field Execution (Status Lifecycle)](#user-story-3-in-field-execution-status-lifecycle)
4. [USER STORY 4: Resolution Evidence & Mandatory Validation](#user-story-4-resolution-evidence--mandatory-validation)
5. [USER STORY 5: Personal Performance & Work History](#user-story-5-personal-performance--work-history)

USER STORY 1: Invite-Only Onboarding & Org Linking

Executive Summary

Allow workers to register using a pre-invited email and auto-join their organization.

User Persona & Problem Statement

Who: As a Field Worker...

Why: I need a verified way to access my department's tasks.

Scope (In & Out)

In Scope

- Email OTP login or Google OAuth (invite-only).
- Invite-only registration with pre-approved email list.
- Auto-link to organization on first login.

Out of Scope

- Bulk import.

Features & Acceptance Criteria

Feature: Invite-Only Org Linking

User Story: As a Field Worker, I want to register using my invited email so I can access my department's tasks.

Acceptance Criteria:

- [] Verify worker can only register with an email that was invited by the authority.
- [] Verify first login auto-links the worker to the correct org and sets status ACTIVE.
- [] Verify workers using non-invited emails are blocked.
- [] Verify login uses Email OTP or Google OAuth only.

Functional Requirements

- Store `org_id` on worker profile at first login.
- Return HTTP 403 for workers without an invited/active status.
- Invite lifecycle: INVITED -> ACTIVE -> INACTIVE.
- Invites expire after a configured window (default 7 days) and can be re-sent.
- Optional allowed-domain restriction can be enforced (e.g., *@authority.gov.in).

Edge Cases

- If an invite is expired or revoked, registration is blocked.

USER STORY 2: Task Queue, Acceptance & Location Verification

Executive Summary

Provide a centralized task list with map context, plus a required acceptance step that captures worker ETA.

User Persona & Problem Statement

Who: As a Field Worker...

Why: I need clear location context and a way to estimate completion time when accepting tasks.

Scope (In & Out)

In Scope

- List view and map view.
- Before photo visibility.
- Accept task with time estimate.
- Location verification within 5m.

Out of Scope

- Task re-assignment.

Features & Acceptance Criteria

Feature: Location Verification

User Story: As a Field Worker, I want to verify location accuracy before starting work.

Acceptance Criteria:

- [] Verify "Get Directions" opens external map apps.
- [] Verify the original "Before" photo is visible.
- [] Verify the app shows distance to issue and warns when beyond 5m.

Feature: Task Acceptance with ETA

User Story: As a Field Worker, I want to provide an estimated completion time when I accept a task.

Acceptance Criteria:

- [] Verify an "Accept" action is required before starting work.
- [] Verify acceptance requires an estimated duration (minutes/hours).
- [] Verify ETA is saved to the issue and visible to Govt Admins.
- [] Verify acceptance sets `accepted_at` on the issue.

UI/UX Design & User Flow

Flow: My Tasks -> Open Task -> Accept + ETA -> Navigate -> Start Work.

Functional Requirements

- `POST /issues/{id}/accept` with payload `{ worker_id, estimated_minutes }`.
- Store `accepted_at` server timestamp.
- Expose ETA and elapsed time to authority dashboard.

Edge Cases

- ETA changes require re-confirmation by worker.

USER STORY 3: In-Field Execution (Status Lifecycle)

Executive Summary

Allow workers to mark tasks as IN_PROGRESS once they begin work.

User Persona & Problem Statement

Who: As a Field Worker...

Why: I need to show I am actively working on a task.

```

## Scope (In & Out)
#### In Scope
- Manual "Start Work" action.
- Status transition to IN_PROGRESS.

#### Out of Scope
- Manual time entry.

## Features & Acceptance Criteria
### Feature: Status Transition
**Acceptance Criteria:**
- [ ] Verify tapping "Start Work" changes status to IN_PROGRESS.
- [ ] Verify the button is disabled after starting work.
- [ ] Verify a "Working" badge is shown.

## Workflow & Entity State Lifecycle
**Lifecycle:** ASSIGNED -> IN_PROGRESS.

## Functional Requirements
- Record server-side timestamp on status change.
- Lock task to current worker during IN_PROGRESS.

## Edge Cases
- Accidental tap cannot be undone without admin intervention.

---

# USER STORY 4: Resolution Evidence & Mandatory Validation

## Executive Summary
Enforce verified evidence submission with EXIF validation and location proximity checks.

## User Persona & Problem Statement
**Who:** As a Field Worker...
**Why:** I need to prove I finished the job with valid evidence.

## Scope (In & Out)
#### In Scope
- Mandatory camera capture (no gallery).
- EXIF validation for time and location.
- GPS coordinate capture on submit.

#### Out of Scope
- Video evidence.
- Multi-photo uploads.

## Features & Acceptance Criteria
### Feature: Evidence Submission
**User Story:** As a Field Worker, I want to upload an "After" photo to resolve the ticket.

**Acceptance Criteria:**
- [ ] Verify Resolve is disabled until a photo is captured.
- [ ] Verify EXIF timestamp is within 7 days of submission.
- [ ] Verify EXIF location is within 5m of issue location.
- [ ] Verify submission is blocked if EXIF data is missing or invalid.
- [ ] Verify GPS coordinates are captured on submission.

## UI/UX Design & User Flow
**Flow:** Active Task -> Capture Photo -> Add Notes -> Submit Resolution -> Success.

```

```
## Workflow & Entity State Lifecycle
**Lifecycle:** IN_PROGRESS -> RESOLVED.

## Functional Requirements
- Update status to RESOLVED.
- Submit `exif_timestamp`, `exif_lat`, `exif_long`, and `device_location`.

## Non-Functional Requirements (Minimal)
- Optimize image upload size.

## Edge Cases
- Upload failure keeps status as IN_PROGRESS with retry.

---

# USER STORY 5: Personal Performance & Work History

## Executive Summary
Provide workers with a history of completed work and performance totals.

## User Persona & Problem Statement
**Who:** As a Field Worker...
**Why:** I need a record of my completed tasks for reviews.

## Scope (In & Out)
### In Scope
- Historical list of RESOLVED/CLOSED tasks.
- Date-based filtering.
- Total resolved count.

### Out of Scope
- Public leaderboards.

## Features & Acceptance Criteria
### Feature: Work History
**User Story:** As a Field Worker, I want to view my completed tasks.

**Acceptance Criteria:**
- [ ] Verify a History tab shows RESOLVED and CLOSED tasks only.
- [ ] Verify total resolved count is displayed on profile.

## UI/UX Design & User Flow
**Flow:** Profile -> My History -> Filter by Date -> View Details.

## Success Metrics
- Increased daily access to the task history view.
```