

Documentation Gap Analysis: Incident Management System

1. User Entity & Access Control (RBAC)

There is a significant disconnect between the backend user types and the frontend visibility.

1. **Role Definitions:** The backend contains four distinct roles (Admin, Security Analyst, SOC Analyst, Incident Responder). Are these roles intended to have hierarchical permissions (e.g., an "Incident Responder" seeing fewer menu items than an "Admin")?
2. **User Lifecycle (CRUD):** There is currently no UI for creating, editing, or deactivating users.
 - a. Is user management intended to be an Admin-only frontend feature?
 - b. What are the mandatory data attributes for a user (Email, Phone, Department, Avatar)?
3. **Deletion Logic:** API tests show that a **SOC Analyst** (Sarah) cannot delete an **Admin** incident. We need a defined **Permission Matrix** to confirm if users can delete incidents they didn't create.

2. Functional Inconsistencies (UI vs. API)

The frontend does not currently support the full capability of the API.

- **Incident Updates:** The API allows updating all fields of an incident, but the frontend UI only allows the user to update the Status field. Should the frontend be updated to match the API capability, or is it an API only feature?
- **Field Constraints:** The requirements mention testing for **Max Lengths**, but specific character limits (e.g., Title: 255 chars, Description: 2000 chars) are not defined.

3. API & Security Specifications

Several security tests cannot be "Pass/Fail" validated without clear benchmarks.

- **Token Lifecycle:** The task requires testing for **Token Expiration**.
 - What is the defined TTL (Time-to-Live) for a JWT?
 - Is there a refresh token strategy in place?
- **Error Handling (403 Forbidden):** The documentation lists a "Forbidden" error for incident deletion but does not specify the business rule.
 - *Requirement needed:* "Only the Creator or an Admin may delete an incident."

4. High-Risk Areas (Impact on Testing)

Due to the lack of documentation in the areas above, the following testing activities are currently **blocked** or incomplete:

1. **Security Sensitivity:** We cannot validate if a "Low Level" user is successfully restricted from "High Level" data without a defined Permission Matrix.
2. **Data Integrity:** Without max-length constraints, we cannot perform proper "Stress/Negative" testing on the database inputs.