

Crowd-Powered Smart Complaint Management System

Sprint 1 Concept Poster

Student Details:

Name: Neev Vegada
Student ID: 202301031

Project Overview:

The proposed system provides a digital platform where citizens can conveniently report complaints such as road damage, garbage collection issues, Wi-Fi outages, or noise disturbances. The system not only collects these complaints but also applies machine learning techniques to classify and prioritize them based on urgency, frequency, and location. An important feature is the adoption of Federated Learning (FL), which preserves user privacy while still benefiting from collective data. This enables both scalability and security while ensuring fairness in complaint resolution. The platform aims to increase transparency, accountability, and efficiency in municipal services.

Sprint 1 Goals:

- Identification of Stakeholders & End Users
- Determining elicitation techniques with justification
- Gathering initial requirements (functional, non-functional, domain)
- Writing user stories for different stakeholders
- Creating EPICs for backlog and aligning with agile practices
- Identifying and resolving any conflicts during requirement gathering
- Ensuring that requirements align with both user needs and technical feasibility
- Laying the foundation for Sprint 2 by establishing priority features and dependencies

Stakeholders & End Users:

- End Users: Citizens submitting complaints such as damaged roads, garbage, or Wi-Fi outages. They expect quick response times and transparency.
- Municipal Authorities: Officials who review and address complaints, ensuring accountability and efficient resource allocation.
- System Administrators: Manage and maintain the platform, ensuring uptime, data protection, and smooth operation.
- Data Scientists & Developers: Work on improving the ML model, refining complaint classification, and integrating new features.
- Policy Makers: Use long-term complaint data analytics to improve infrastructure planning and budget allocation.

Elicitation Techniques:

- Interviews (Municipal Authorities) – to gather detailed, context-specific input on workflows and challenges.
- Surveys (End Users) – to collect broad insights from a large and diverse population.
- Observation (System Admins) – to understand operational workflows and system usage challenges.

- Workshops (Data Scientists & Dev Team) – to identify technical constraints and opportunities for ML model training.
- Document Analysis (Government Policies) – to ensure the system complies with legal and administrative guidelines.

Sample Requirements:

- Functional: Submit complaints, Track complaint status, Auto-classification of complaints, Upvote feature, Assign complaints to departments.
- Non-Functional: Data privacy, Response time < 2 seconds, Scalability for thousands of users, Transparency, 24/7 uptime, and system reliability.
- Domain: Types of complaints, location-specific handling, compliance with government regulations, long-term data storage for planning, geotagging.

User Story Samples:

- Citizen: "As a citizen, I want to submit a complaint so that authorities can fix issues."
- Acceptance Criteria: Complaint submission form, confirmation notification, and tracking ID.
- Citizen: "As a citizen, I want to upvote complaints from others so that major issues are prioritized."
- Acceptance Criteria: Upvote option available, visible count on each complaint.
- Citizen: "As a citizen, I want to receive notifications when the complaint status changes so that I am informed of progress."
- Acceptance Criteria: Push notifications or SMS alerts when complaint status is updated.
- Authority: "As an authority, I want to update complaint status so that citizens know the progress."
- Acceptance Criteria: Status updates with remarks visible to users, auto-notifications on changes.
- Authority: "As an authority, I want to assign complaints to relevant departments so that they are resolved efficiently."
- Acceptance Criteria: Assignment option available, record maintained in system logs.
- System Admin: "As a system admin, I want to monitor system performance so that downtime can be minimized."
- Acceptance Criteria: System health dashboard with alerts and logging.
- System Admin: "As a system admin, I want to manage user roles so that only authorized users can access specific functionalities."
- Acceptance Criteria: Role-based access control implemented.

EPIC Creation (Examples):

- EPIC 1: Complaint Submission and Tracking
- EPIC 2: Complaint Classification and Prioritization
- EPIC 3: Privacy-Preserving Learning Module
- EPIC 4: System Monitoring and Administration
- EPIC 5: Notifications and User Engagement Features

Conflict Resolution:

One major conflict arises between user privacy and the need for data collection to improve the system. This is resolved by applying Federated Learning, which ensures that raw user data never leaves their device, while still enabling collaborative model improvement. Additionally, conflicts may occur between municipal authorities and citizens in terms of complaint prioritization. This can be mitigated by providing a transparent prioritization model that combines citizen upvotes, urgency levels, and location-based impact. In this way, both user concerns and official constraints are balanced, ensuring fairness and efficiency.