



Centurion
UNIVERSITY

*Shaping Lives...
Empowering Communities...*

Software Requirements Specification

Project: Smart City Complaint & Governance System

Supervisor

Mr. Rakesh kumar Ray

By

Vikas Das (230301120326)

Shrabanee Routray (230301120295)

Table of Contents

1. Introduction

- 1.1 Project Overview
- 1.2 Purpose
- 1.3 Scope
- 1.4 Target Audience

2. Overall Description

- 2.1 Product Perspective
- 2.2 Tools and Technologies
- 2.3 User Classes and Characteristics
- 2.4 Operating Environment
- 2.5 Design and Implementation Constraints

3. System Features and Requirements

- 3.1 Home Page Module
 - 3.1.1 Header and Navigation Bar
 - 3.1.2 Main Banner / Hero Section
 - 3.1.3 Footer Section
- 3.2 Future Scope

4. External Interface Requirements

- 4.1 User Interface (UI)
- 4.2 Software Interfaces

5. Non-Functional Requirements

- 5.1 Performance
- 5.2 Usability
- 5.3 Reliability
- 5.4 Security

1. Introduction

1.1 Project Overview

The Smart City Complaint & Governance System is a web-based application designed to digitalize the process of managing civic complaints within a city. The system enables citizens to report issues such as road damage, garbage overflow, water leakage, and streetlight malfunction. The system integrates GPS-based location tracking, Google Maps visualization, heatmap analytics, real-time updates, and automated escalation mechanisms to enhance transparency and efficiency in city governance.

1.2 Purpose

The purpose of this project is to:

- Provide an online platform for complaint registration.
- Enable role-based access control (Admin, Officer, Citizen).
- Integrate GPS-based automatic location capture.
- Visualize complaints on Google Maps.
- Provide heatmap-based complaint density analysis.
- Improve communication between citizens and authorities.

1.3 Scope

The system covers:

- User registration and authentication.
- Complaint submission with automatic location detection.
- Complaint assignment and tracking.
- Map visualization of complaints.
- Heatmap analysis for administrators.
- Real-time status update system.
- Automated escalation for unresolved complaints.

This project is intended for municipal governance environments.

1.4 Target Audience

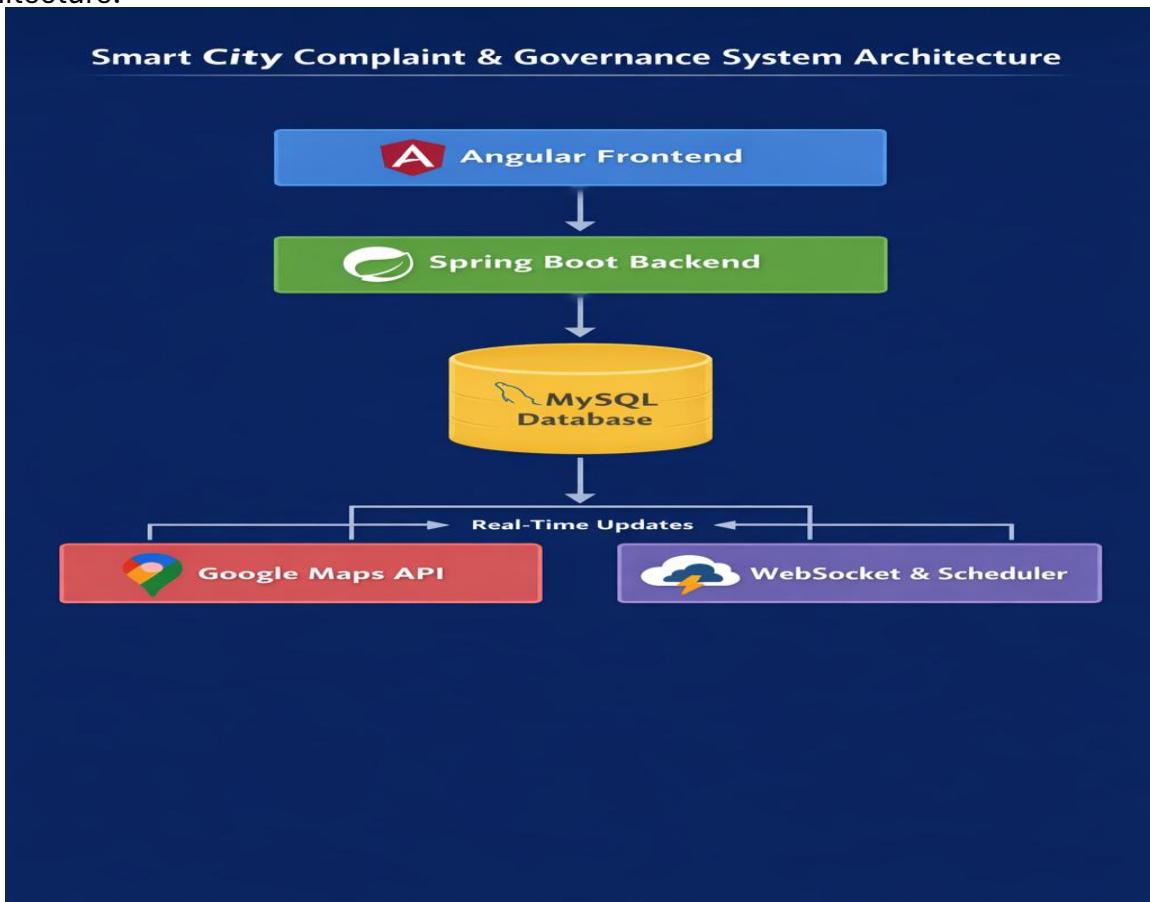
- Municipal Authorities
- City Administrators
- Government Officers
- Citizens
- Academic Evaluators (for project submission)

2. Overall Description

2.1 Product Perspective

The Smart City Complaint & Governance System is a client-server based web application.

Architecture:



2.2 Tools and Technologies

Frontend

- Angular
- HTML, CSS, TypeScript
- Google Maps API
- WebSocket Client

Backend

- Spring Boot
- Spring Security (Session-Based)
- Spring Data JPA
- WebSocket
- Scheduler

Database

- MySQL

2.3 User Classes and Characteristics

Citizen

- Basic computer knowledge
- Can register and submit complaints
- Can track complaint status

Officer

- Handles assigned complaints
- Updates complaint status
- Views complaints on map

Admin

- Full system access

- Assigns officers
- Views heatmap analytics
- Monitors escalation reports

2.4 Operating Environment

- Operating System: Windows / Linux / macOS
- Browser: Chrome, Edge, Firefox
- Backend Server: Spring Boot Embedded Tomcat
- Database: MySQL Server
- Internet connection required

2.5 Design and Implementation Constraints

- Requires internet connectivity
- Requires GPS permission from user
- Google Maps API key required
- System performance depends on server configuration

3. System Features and Requirements

3.1 Home Page Module

This is the core module developed in the video tutorial.

3.1.1 Header and Navigation Bar

- **Description:** The header provides branding and primary navigation for the site.
- **Requirements:**
 - Logo and project name
 - Navigation links:
 - Home
 - Login
 - Register
 - About
 - Role-based navigation after login

3.1.2 Main Banner / Hero Section

- **Description:** A large, visually engaging section below the header to attract user attention.
- **Requirements:**
 - Display project title
 - “Raise Complaint” button
 - Information about smart governance
 - Call-to-action message

3.1.3 Features section

- Display project title
- “Raise Complaint” button
- Information about smart governance
- Call-to-action message

3.1.4 Footer Section

- **Description:** The section at the bottom of the page containing supplementary information.
- **Requirements:**
 - Contact details
 - About city governance
 - Social links
 - Copyright information

3.2 User Management Module

- User Registration
- Login Authentication
- Role-based access
- Password encryption
- Session management

3.3 Complaint Management Module

- Raise complaint

- Automatic GPS location capture
- Store latitude & longitude
- Upload description and details
- Track complaint status
- View complaint history

3.4 Map Visualization Module

- Display complaints as markers
- Filter by status
- Show complaint details on click

3.5 Heatmap Module

- Display complaint density
- Highlight high-complaint zones
- Available to Admin only

3.6 Real-Time Update Module

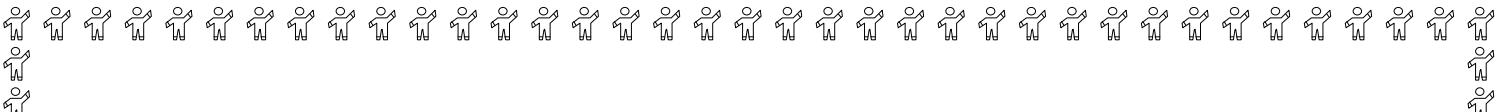
- Instant status update using WebSocket
- Notification to citizen
- Dashboard auto-refresh

3.7 Escalation Module

- Monitor complaint time duration
- Escalate after 48 hours
- Notify admin
- Change status to “Escalated”

3.2 Future Scope: User and Book Management

- AI-based complaint classification
- SMS notifications
- Mobile application
- Data analytics dashboard
- Government API integration



4. External Interface Requirements

4.1 User Interface (UI)

- Responsive design
- Map-based visualization
- Role-based dashboard
- Clean and simple navigation

4.2 Software Interfaces

- Google Maps API
- MySQL Database
- WebSocket Server
- REST APIs

5. Non-Functional Requirements

5.1 Performance

- The home page should load within 3-5 seconds on a standard internet connection.
- Images should be optimized for the web to reduce load times.
- System shall support multiple users simultaneously.

5.2 Usability

- The navigation should be logical and predictable.
- All interactive elements (buttons, links) must be clearly identifiable and functional.
- The color scheme and typography should ensure readability.

5.3 Reliability

- The application should be available 24/7, with minimal downtime.
- Links should not be broken, and all page elements should render correctly.



5.4 Security

- Encrypted passwords (BCrypt)
- Role-based access control
- Session-based authentication
- Input validation

Thank You !