SMART CITY MANAGEMENT PLATFORM (SCMP)

School of Computer Science

UNIVERSITY OF PETROLEUM AND ENERGY STUDIES

SESSION 2025

SRS DOCUMENTATION

Submitted by:

- Name Ayush Gupta SapId: 500125537 Batch: 7
- Yash Srivastava Sap Id: 500125840 Batch: 8
- Shashank Dimri Sap Id: 500124397 Batch: 7
- Madhav Sharma Sap Id: 500125858 Batch: 7
- Arsh Rathi Sap Id: 500127591 Batch: 7

Submitted To:

Mr. Ayush Gujjar

Introduction

Purpose:

The purpose of this document is to define the Software Requirements Specification (SRS) for the Smart City Management Platform (SCMP) for Dehradun. This platform aims to provide real-time information and utilities for the city's residents and visitors, including traffic management, weather updates, waste management services, tourism information, optimal route suggestions, and local news updates.

Scope:

SCMP is a web-based platform developed using Python (Flask), HTML/CSS, JavaScript, and public APIs such as Google Maps API, OpenWeatherMap API, and NewsAPI. The platform includes modules:

- Traffic Management – Live traffic updates using Google Maps API.
- Weather Management – Live weather data from OpenWeatherMap API.
- Waste Management – Reporting and admin control for bin status.
- Optimal Route Suggestions – Using Directions API for best routes.
- Tourism Guide – Static info on attractions, hotels, cafes, with embedded maps.
- News Updates – Latest news from NewsAPI.org for Dehradun.

Intended Audience:

- Software Engineering Instructors and Evaluators
- Smart City Developers and Planners
- Citizens of Dehradun
- Future Developers/Contributors

Functional Requirements:

1. **Traffic Module**
   - Display real-time traffic conditions in Dehradun using Google Maps Traffic Layer.
   - Use embedded maps to show congested areas.
2. **Weather Module**
   - Fetch current weather data (temperature, humidity, wind speed, etc.).
   - Update UI dynamically every 10 minutes using OpenWeatherMap API.
3. **Waste Management Module**

   User Panel:
   - Report bin locations with description.
   - View status of nearby bins.

Admin Panel:

- Update bin collection status.
- View all user reports in table format.

4. **Tourism Module**
   - Display top attractions using local images.
   - Show hotels, restaurants, and map links.
   - Each location has a "Read More" button linking to external pages.
5. **News Module**
   - Fetch Dehradun-specific news from NewsAPI.
   - Auto-refresh news every minute.

User Interface Requirements:

The Smart City Management Platform (SCMP) provides a clean, modern, and responsive user interface designed for ease of use and accessibility across devices. The interface is built using HTML, CSS, and inline JavaScript to ensure fast loading times and smooth navigation.

- **Responsive Design**: The entire platform is designed to be mobile-friendly, adapting seamlessly to different screen sizes including desktops, laptops, tablets, and smartphones. This ensures that users and administrators can access the platform anytime, anywhere.
- **Colour Themes**: Each module in the SCMP project uses a distinct and intuitive color theme to enhance clarity and visual separation:
  - ➢ News Module: Features a bold red theme, emphasizing urgency and relevance, with neatly designed cards for each news article.
  - ➢ Weather Module: Uses a light blue theme with a modern card layout, presenting temperature, weather conditions, and other details in an easily digestible format.
  - ➢ Waste Management Module: The page is divided into two clear sections—User Panel and Admin Panel—with color and layout differences to distinguish them visually. The user panel allows the public to report bin locations, while the admin panel allows status updates.
  - ➢ Traffic and Optimal Routes Modules: Integrated using Google Maps with live traffic indicators and route suggestions using traffic-aware Directions API.
  - ➢ Tourism Module: Uses a gallery-style layout with cards for each location, embedded maps, and a light professional theme focused on clarity and attraction.

- **Map Embeds**: Google Maps is embedded at multiple points across the application. These include:
    - ➢ Live Traffic Maps to visualize congestion in real-time.
    - ➢ Optimal Route Suggestions using the Directions API.
    - ➢ Nearby Attractions, such as cafes, parks, and temples in the Tourism Module.
    - ➢ Hotel and Restaurant Recommendations are also embedded with location pins via Maps API.
- **Card-Based Layouts**:
    - ➢ A consistent card design is used to display discrete units of information across modules:
    - ➢ News Cards show title, description, source, and date.
    - ➢ Weather Cards present temperature, humidity, wind, and description with icons.
    - ➢ Tourist Spot Cards include an image, title, and brief description with a "Read More" link for details.
- **Font and Typography**: Readable, modern fonts like "Segoe UI" are used throughout, with proper contrast and spacing to reduce eye strain and enhance readability.
- **Minimalist Layout**: The overall interface avoids clutter by keeping each module focused on its core functionality, promoting ease of navigation without overwhelming the user with information.

Non-Functional Requirements:

Non-functional requirements define the overall attributes and constraints of the Smart City Management Platform (SCMP) that ensure reliability, usability, maintainability, and performance. These are essential for user satisfaction and system stability across real-world conditions.

Performance Requirements:

- **Speed**: The platform should load individual modules (such as weather, news, tourism, etc.) within 2–3 seconds on a standard internet connection. Optimized HTML, CSS, and lightweight JavaScript are used to ensure fast response times.
- **API Response Time**: External API calls (e.g., OpenWeatherMap, NewsAPI, Google Maps APIs) should return results within 1 second, ensuring near real-time data updates for weather, traffic, and news modules.

- **Auto Refresh Capabilities:** The News Module is designed to auto-refresh every minute, pulling the latest data without needing manual intervention. This ensures that information is current and relevant at all times.
- **Scalability:** The platform can be hosted on cloud platforms like Render.com, and can be scaled up easily by increasing backend instance sizes or horizontally deploying additional app servers with a load balancer.

Reliability Requirements

- **System Uptime**: The hosted application must maintain a 99.9% uptime to be considered reliable. It should remain accessible for end-users (public) and admins during all hours of the day.
- **Backup and Failover**: While the current system is designed for academic purposes, it can be extended with regular backups and failover mechanisms using cloud services like Render, AWS, or Azure in future deployments.

Usability Requirements

- **Intuitive Interface**: The layout is clean and minimal, ensuring that even non-technical users can access and understand the features (e.g., submitting waste bin reports or reading the news).
- **Language and Readability**: The content is presented in **clear, concise English** with good color contrast, making it accessible to a broad audience. Tooltips or descriptions are minimal but can be added for guidance.
- **No Login Barrier**: There is no login system, which makes access immediate and removes friction for users and admins alike—ideal for public or demo environments.

Maintainability Requirements

- **Modular Codebase**: The code is organized into **separate HTML templates and Python routes**, allowing for easy debugging, updates, and addition of new features or pages in the future.
- **API Key Management**: External services are cleanly integrated using API keys, which can be easily updated from a central configuration without modifying core logic.
- **Documentation & Comments**: All critical functions are commented, and documentation (like this SRS) is available to help new developers understand and maintain the system.

Security Requirements

- **API Key Protection**: API keys should ideally be stored in environment variables or server configs when deployed to a live server. In the current academic setup, they are embedded but can be moved to secure storage in production.
- **Form Validation**: User inputs (like bin locations) are validated on the frontend to prevent malformed data submission. Further backend validation can be added for robustness.

External Interfaces / APIs Used

1. **Google Maps JavaScript API**

- Used in: Traffic module, tourism map sections.
- API Key: AIzaSyBOtXBTnWqPNOejz88xNNxLEqHI0FV3Wgw
- Services Enabled:
  - ➢ Maps JavaScript API
  - ➢ Maps Static API
  - ➢ Directions API

2. **OpenWeatherMap API**

- Used in: Weather module.
- API Key: ec07a498b22fdbf9dd4eba902a527b0c
- Services Used:
- Current Weather Endpoint:
  - ➢ https://api.openweathermap.org/data/2.5/weather?q=Dehradun

3. **Google Directions API**

- Used in: Optimal Routes module.
- API Key: Same as above (Google Maps key).
- Used to fetch best and alternate routes between user-selected points.

4. **NewsAPI.org**

- Used in: News module.
- API Key: be6fcde483ad483e91ab40baa4696fbd
- Endpoint: https://newsapi.org/v2/everything?q=Dehradun&sortBy=publishedAt
- Fetches latest articles specific to Dehradun.

Future Scope

The Smart City Management Platform (SCMP) is currently designed as a prototype system for academic demonstration and public utility awareness. However, its modular and extensible architecture allows for future enhancements and full-scale deployment. The following outlines the future potential of SCMP:

1. Expansion to New Cities: While the current implementation focuses on Dehradun, the platform can be easily adapted for other cities by:

- Changing the static assets and data references (images, maps, locations).
- Modifying API query parameters (e.g., weather, news, traffic).
- Replacing or enhancing local governance data modules for city-specific regulations.

2. Integration of Real-time Data and Sensors

- In a fully deployed smart city infrastructure, SCMP can be extended to pull real-time data from:
- IoT-based Waste Bins that automatically send fill-level data.
- Live Traffic Cameras or City Surveillance Systems for congestion and safety updates.
- Air Quality Sensors for environmental modules.

These integrations will require backend systems to handle continuous data streams and possibly AI for predictive analytics.

3. Mobile App Development : To increase public reach and real-time interaction, a mobile version of SCMP can be developed with:

- Push notifications (e.g., weather warnings, garbage pickup delays).
- Offline support for viewing maps and past data.
- GPS integration for personal navigation and reporting.

4. User Feedback and Grievance Redressa:l A feedback module can allow:

- Citizens to report civic issues (potholes, broken street lights, etc.).
- Upload images or short descriptions.
- Admins to mark status as “Under Review”, “Resolved”, etc.

Limitations:

Despite its strong potential, the current system faces several practical limitations, mostly due to reliance on third-party APIs and budget constraints.

1. API Usage Costs:

While APIs like Google Maps, OpenWeatherMap, and News API offer limited free tiers, their advanced features or higher usage levels require paid plans. As students, it becomes difficult to afford sustained usage of these premium services, which limits features like unlimited route requests, hourly forecasts, or high-frequency map updates.

2. **Real-world Data Accuracy**

Since this project is developed in an academic environment, some modules rely on manually added or demo data (e.g., tourism places, cafes, admin bin status). In a real deployment, data would need to be dynamically updated from sensors, public sources, or city databases to reflect true conditions, which is currently not feasible in our setup.

3. Limited Access to Official Government Data

Many smart city services depend on official municipal data (like live garbage truck locations, verified incident reports, and citizen feedback systems). As students, we do not have access to secured government APIs or municipal databases, which prevents us from creating a fully real-time or authenticated system.

Conclusion

The Smart City Management Platform (SCMP) for Dehradun is a practical and innovative solution aimed at simplifying urban challenges through technology. With interactive modules for traffic, waste, weather, news, and tourism, the platform offers an engaging and informative user experience. Leveraging modern APIs and intuitive UI design, this project serves as a strong foundation for smart city initiatives and demonstrates the potential of digital transformation at a city level. With future enhancements and real-time data integration, SCMP can evolve into a full-fledged system to support smart governance.