Software Requirements Specification

for

<Traffic Management System>

Version 1.0 approved

Prepared by <Harsh(23FE10ITE00336>

<Manipal University,Jaipur>

<10th August 2025>

Table of Contents

Table of Contents 2
Revision History 2

1. Introduction 1
1.1 Purpose 1
1.2 Document Conventions
.... 1
1.3 Intended Audience and Reading Suggestions.. 1
1.4 Product Scope 1
1.5 References 1
2. Overall Description 2
2.1 Product Perspective 2
2.2 Product Functions 2
2.3 User Classes and Characteristics
.... 2 2.4 Operating Environment
.... 2 2.5 Design and
Implementation Constraints 2
2.6 User Documentation 2
2.7 Assumptions and Dependencies 3
3. External Interface Requirements 3
3.1 User Interfaces 3
3.2 Hardware Interfaces 3
3.3 Software Interfaces 3
3.4 Communications Interfaces 3
4. System Features 4
4.1 System Feature 1 4
4.2 System Feature 2 (and so on) 4
5. Other Nonfunctional Requirements 4
5.1 Performance Requirements 4
5.2 Safety Requirements 5
5.3 Security Requirements 5
5.4 Software Quality Attributes... 5
5.5 Business Rules 5
6. Other Requirements 5
Appendix A: Glossary 5
Appendix B: Analysis Models 5
Appendix C: To Be Determined List 6

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

This SRS defines the software requirements for the Traffic Management System (TMS). The goal of the TMS is to provide city authorities with tools to monitor, manage, and optimize urban traffic flow in real time, reduce congestion, improve safety, and provide actionable analytics for planning.

1.2 Document Conventions- 1.3 Intended Audience and Reading Suggestions

- Project Sponsors and City Authorities - Read the Executive Summary, Product Scope, and Business Rules.
- Developers and Architects - Read Sections 2, 3, 4, and Appendix B for implementation guidance. - Testers - Review functional and non-functional requirements in Sections 4 and 5.

1.4 Product Scope

The TMS will provide real-time traffic sensing, adaptive signal control, incident detection and response, emergency vehicle priority, reporting and analytics, and public-facing traveler information services.

1.5 References

See Appendix for link.

2. Overall Description

2.1 Product Perspective

The TMS is a cloud-enabled software platform that integrates with roadside sensors (loops, cameras), traffic signal controllers, third-party data sources (e.g., vehicle GPS, mapping services), and operator dashboards. It may replace or augment existing legacy traffic control systems.

2.2 Product Functions

- Ingest and normalize sensor and camera feeds.

- Provide real-time dashboard for traffic operators.

- Automatically compute and apply adaptive signal timing plans.

- Detect incidents (accidents, stalled vehicles, congestion anomalies) and alert stakeholders.

- Offer APIs for third-party apps and publish commuter information.

2.3 User Classes and Characteristics

- Traffic Operators: Expert users, frequent interaction, need detailed control and monitoring tools.

- System Administrators: Configure system, manage users, policies, and perform maintenance.

- Commuters: Casual users of public app or web portal for travel updates.

- Emergency Responders: Use priority request features and live routing support.

- Data Analysts / City Planners: Use analytics and historical reports.

2.4 Operating Environment

- Backend: Cloud-hosted services (AWS/GCP/Azure) or municipal data center.

- Frontend: Modern web browsers (Chrome, Firefox, Edge) and mobile apps (iOS/Android).

- Edge: Field devices (controllers, sensors, cameras) using standard communication protocols.

2.5 Design and Implementation Constraints

- Must interoperate with legacy controllers using NTCIP or vendor-specific protocols.

- Must adhere to local laws for surveillance and data retention.

- Hardware limitations at field devices (CPU, memory) may constrain feature deployment.).>

2.6 User Documentation

Operator manuals, administration guides, API documentation, quick-start tutorials, and training workshops.

2.7 Assumptions and Dependencies

- Continuous network connectivity for central control; offline modes for local controllers.

- Availability of sensors and camera feeds in required locations.

- Third-party data providers (map, weather) provide accessible APIs.

3. External Interface Requirements

3.1 User Interfaces

- Operator Dashboard: Real-time map, signal status, incident queue, control panel for manual overrides.
- Mobile / Commuter App: ETA, congestion alerts, alternative routes.
- Admin Console: User management, system logs, configuration.

3.2 Hardware Interfaces

- Traffic Signal Controllers: NTCIP/serial/Ethernet connections for timing commands.
- Roadside Sensors: Inductive loops, radar sensors, BLE/Wi-Fi trackers for vehicle counts.
- Cameras: RTSP/ONVIF streams for video analytics.

3.3 Software Interfaces

- Databases: Use PostgreSQL (or similar) for persistent storage; time-series DB (InfluxDB) for telemetry.
- External APIs: Map provider (routing), weather API, emergency services feeds.
- Message Brokers: MQTT/Kafka for telemetry ingestion.

3.4 Communications Interfaces

- Protocols: HTTPS/REST for web APIs, MQTT for lightweight telemetry, secure VPN or TLS for field communications.
- Performance: Typical upstream bandwidth of field devices should support periodic telemetry and occasional video frames.

4. System Features

This section enumerates the major functional features. Each feature includes description, priority, stimulus/response sequences, and functional requirements.

4.1 Real-Time Traffic Monitoring

4.1.1 Description and Priority

Collect live data from road sensors, cameras, and connected vehicles to create a real-time view of traffic conditions. Priority: High

4.1.2 Stimulus/Response Sequences

- Stimulus: Sensor reports vehicle count. Response: System updates map layer and aggregates flow metrics.

4.1.3 Functional Requirements

REQ-001: System shall ingest telemetry from roadside sensors at a configurable interval (e.g., 1s - 60s).

REQ-002: System shall normalize varied sensor formats into a common data model.

REQ-003: System shall present a live map with vehicle density heatmap for operators.

4.2 Intelligent Traffic Signal Control

4.2.1 Description and Priority

Automatically compute and apply optimized signal timings to minimize delays and queue lengths.

Priority: High

4.2.2 Stimulus/Response Sequences

- Stimulus: Surge in traffic on arterial road. Response: System computes and pushes updated timing to controllers.

4.2.3 Functional Requirements

REQ-010: System shall compute optimal phase durations using configurable optimization criteria (e.g., minimize average delay).

REQ-011: System shall support fallback to fixed-time plans in case of communication failure.

4.3 Incident Detection and Alerting

4.3.1 Description and Priority

Detect incidents (accidents, stopped vehicles, sudden slowdowns) via analytics on sensor and video feeds and notify operators.

4.4 Emergency Vehicle Priority

4.4.1 Description and Priority

Provide green corridor creation and dynamic signal pre-emption to facilitate rapid emergency vehicle movement. Priority: High

4.4.2 Stimulus/Response Sequences - Stimulus: Emergency responder requests corridor via app or dispatch. Response: System reserves green waves and updates signals

4.4.3 Functional Requirements

REQ-030: System shall accept authenticated priority requests from authorized emergency services.

REQ-031: System shall compute and enact minimal-disruption signal plans to create a corridor.

REQ-032: System shall log priority activations for audit and accountability

4.5 Integration and API Services

4.8.1 Description and Priority

Expose secure APIs for integration with city systems, navigation providers, and thirdparty services.

Priority: Medium

4.8.2 Stimulus/Response Sequences

- Stimulus: Third-party requests traffic snapshot. Response: Authenticated API returns JSON payload with requested data.

4.8.3 Functional Requirements

REQ-070: System shall provide RESTful APIs with OAuth2-based authentication and rate limiting.

REQ-071: System shall provide webhook subscriptions for event notifications (e.g., incidents).

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- Real-time telemetry ingestion latency must be < 2s end-to-end for critical sensors. - Dashboard updates should reflect new data within 2-5 seconds under normal load.

5.2 Safety Requirements

- The system must have manual override for signals; automated changes must never create conflicting phases.
- In event of system failure, controllers must revert to safe default plans

5.3 Security Requirements

- All control commands must be authenticated and authorized.

- Communications must use TLS 1.2+; sensitive data stored encrypted at rest.

5.4 Software Quality Attributes

- Availability: Target 99.9% uptime for central services.

- Scalability: System must scale horizontally for new intersections and cities. - Maintainability: Modular microservices architecture and documented APIs.

5.5 Business Rules

- Emergency vehicle requests override regular timing but are subject to safety constraints.

- Only authorized roles can perform production timing changes.

6. Other Requirements

- Database requirements: Use ACID-compliant DB for transactional metadata and time-series DB for telemetry.
- Internationalization: UI must support multilingual text (English + local language) and time zone handling.
- Legal & Compliance: Comply with local privacy laws; keep audit trails for at least 1 year.
- Backup & Recovery: Daily backups and RTO (Recovery Time Objective) of 2 hours for critical services.

Appendix A: Glossary

TMS: Traffic Management System

IoT: Internet of Things

NTCIP: National Transportation Communications for ITS Protocol

RSU: Roadside Unit

API: Application Programming Interface

RTO: Recovery Time Objective

Appendix B: Analysis Models

B.1 Data Flow Overview
- Level 0: Field devices -> Ingestion -> Processing/Analytics -> Dashboard / API / Archive

B.2 Entity Relationship (summary)
- Entities: Sensor, Intersection, Controller, Incident, User, Report
- Example relationships: Intersection 1---* Sensor, Controller 1---* Intersection B.3 State/Sequence Notes

- Signal controller supports modes: AUTO, MANUAL, FAILSAFE.

Appendix C: To Be Determined List

TBD-1: Exact hardware vendor list and device models for roadside equipment.

TBD-2: Final SLA numbers, including allowable downtime per month and support windows.

TBD-3: Exact data retention periods for raw video feeds due to privacy considerations.

TBD-4: Budget constraints and deployment phasing schedule.