B.M.S. COLLEGE OF ENGINEERING BENGALURU
Autonomous Institute, Affiliated to VTU

OOMD Mini Project Report

Smart Traffic Management System for Urban Congestion

Submitted in partial fulfillment for the award of degree of

Bachelor of Engineering
in
Computer Science and Engineering

Submitted by:

Bhoomika B G(1BM23CS067)
Bhoomika M(1BM23CS068)
Chethan (1BM23CS074)
Jayanth Vikram(1BM23CS326)

Department of Computer Science and Engineering
B.M.S. College of Engineering
Bull Temple Road, Basavanagudi, Bangalore 560 019
2025-26

B.M.S. COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We, Bhoomika B G(1BM23CS067) ,Bhoomika M(1BM23CS068), Chethan (1BM23CS0), Jayanth Vikram(1BM23CS) students of 5th Semester, B.E, Department of Computer Science and Engineering, BMS College of Engineering, Bangalore, hereby declare that, this OOMD Mini Project entitled " **Smart Traffic Management System for Urban Congestion**" has been carried out in Department of CSE, B.M.S. College of Engineering, Bangalore during the academic semester August 2025- December 2025. I also declare that to the best of our knowledge and belief, the OOMD mini Project report is not from part of any other report by any other students.

Signature of the Candidate

Bhoomika B G(1BM23CS067)
Bhoomika M(1BM23CS068)
Chethan (1BM23CS074)
Jayanth Vikram(1BM23CS326)

B.M.S. COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the OOMD Mini Project titled " **Smart Traffic Management System for Urban Congestion"** has been carried out by Bhoomika B G(1BM23CS067), Bhoomika M(1BM23CS068), Chethan (1BM23CS074), Jayanth Vikram(1BM23CS326) during the academic year 2025-2026.

Signature of the Faculty incharge

Vikranth B M

Assistant Professor

Table of Contents

Chapter 1: Problem Statement

Smart Traffic Management System for Urban Congestion

Urban traffic congestion has become a critical issue due to rapid urbanization, increasing vehicle density, and inefficient static traffic signal systems. Existing traffic control methods do not adapt to real-time traffic fluctuations, resulting in longer commute times, fuel wastage, and increased environmental pollution.

There is a need for an intelligent and adaptive traffic management system that can dynamically optimize signal timings based on live traffic conditions. This project aims to develop an AI-based Smart Traffic Management System that utilizes real-time data from cameras and IoT sensors, along with computer vision and reinforcement learning techniques, to analyze traffic flow, predict congestion, and adjust signal phases accordingly.

The goal of the system is to reduce average commute time by at least 10% in a simulated urban environment and provide traffic authorities with a monitoring dashboard for real-time control and decision-making. The proposed solution is technically feasible through the integration of existing traffic camera networks, OpenCV-based vision processing, and reinforcement learning models.

Chapter 2: Software Requirement Specification

1. Introduction

1.1 Purpose

Provide a validated, testable SRS for an AI-based Smart Traffic Management System (AI-TMS) that optimizes signal timings and reduces congestion in urban areas. This revised SRS adds measurable acceptance criteria, SMART functional and nonfunctional requirements, safety invariants, canonical data schemas, a reproducible simulation test plan, an operations runbook, and a requirements traceability matrix.

1.2 Scope and Expected Outcome

Deliver a prototype that, in a simulated urban environment, reduces average commute time by at least **10%** versus baseline fixed-time or actuated control, and provides a dashboard for traffic authorities to monitor and control signals.

1.3 Acceptance Criteria

- **Primary Acceptance** Average travel time reduction ≥ 10% across the defined evaluation scenarios with statistical significance $p < 0.05$.
- **Secondary Acceptance** Improvements in at least two of: average delay per vehicle (≥ 10% reduction), average queue length (≥ 10% reduction), emergency vehicle delay (≥ 20% reduction).
- **System Acceptance** All high-priority FR and NFR test cases pass; safety invariants hold in all simulated runs.
- **Reproducibility** All experiments must be runnable via provided containerized scripts and configuration files producing identical metrics given the same random seed.

2. Functional Requirements SMART List

Each requirement is labeled FR-x, states priority, measurable success metric, update frequency, test case ID.

FR-1 Real-Time Telemetry Ingestion High

- Description: Collect video streams and IoT telemetry and normalize into time-series at 1–5 second intervals.

- Success Metric 99.9% of ingest messages processed and stored within 2 seconds.
- Update Frequency Ingestion pipeline processes windows every 1 second.
- Test ID TC-INGEST-01

FR-2 Per-Segment Metrics High

- Description Compute per-segment metrics: flow, speed, occupancy, queue length, vehicle classification every 10 seconds.
- Success Metric MAE for flow ≤ 5 vehicles/min on validation traces.
- Update Frequency 10 seconds.
- Test ID TC-METRICS-01

FR-3 Short-Horizon Forecasting High

- Description Provide forecasts for 5, 15, 30 minute horizons per segment with confidence scores.
- Success Metric Forecast MAE ≤ 10% relative to observed flow for 5-min horizon; calibration score for confidence intervals within 5% error.
- Update Frequency Forecasts updated every 1 minute.
- Test ID TC-PRED-01

FR-4 RL Based Signal Optimizer High

- Description Produce phase timing proposals using RL policy that maximizes network throughput while obeying safety constraints.
- Success Metric Average travel time reduction ≥ 10% compared to baseline in evaluation scenarios.
- Update Frequency Recompute control decisions every control cycle (configurable, default 10 seconds).
- Test ID TC-RL-01

FR-5 Safe Application of Timing Changes High

- Description Apply timing changes to controllers with pre-checks and rollback capability.
- Success Metric 100% of applied changes pass safety interlocks; rollback completes within 5 seconds on failure.
- Test ID TC-CONTROL-01

FR-6 Emergency Preemption High

- Description Authenticate emergency vehicle requests and reserve green corridor ensuring pedestrian minima.
- Success Metric Emergency vehicle delay reduction ≥ 20% in routed runs; authentication

latency ≤ 500 ms.

- Test ID TC-EMERG-01

FR-7 Mobile Alerts High

- Description Notify subscribed users of incidents and route advisories localized to user route.
- Success Metric 95% of critical alerts delivered within 5 seconds; false positive alert rate ≤ 2%.
- Test ID TC-MOBILE-01

FR-8 Manual Override and Audit High

- Description Allow operator manual overrides with full audit logging and simulated impact preview.
- Success Metric All overrides logged; UI shows predicted impact within 2 seconds.
- Test ID TC-UI-01

FR-9 Model Explainability Medium

- Description Provide per-action justification and confidence score for RL decisions and forecasts.
- Success Metric Explanations generated within 1 second for UI display; operator satisfaction ≥ 80% in user trials.
- Test ID TC-XAI-01

FR-10 Data Retention Controls Medium

- Description Configurable retention and anonymization policies for video and location data.
- Success Metric Raw video retention default ≤ 7 days; anonymized metrics retained per policy.
- Test ID TC-PRIV-01

3. Nonfunctional Requirements and Safety

3.1 Performance NFRs

- **NFR-1 Control Latency High** End-to-end control decision latency ≤ 2 seconds under nominal load. Test ID TC-NFR-PERF-01.
- **NFR-2 Ingestion Throughput High** Support 10,000 events/sec per region with ≤ 1% dropped messages. Test ID TC-NFR-PERF-02.
- **NFR-3 Dashboard Availability High** Dashboard 99.9% uptime; stale-data threshold configurable, default 5 seconds. Test ID TC-NFR-PERF-03.

3.2 Safety Invariants and Constraints High

Safety invariants must be enforced in all automated actions and tested in simulation and staging:

- **SI-1 No Conflicting Greens** Never allow two movements that create a conflicting trajectory simultaneously.

- **SI-2 Minimum Pedestrian Green** Enforce pedestrian green of at least configured minimum per local regulation, default 7 seconds.
- **SI-3 Maximum Phase Extension** Do not extend a green phase beyond configured maximum, default 30 seconds without operator consent.
- **SI-4 Preemption Safety Timeout** Emergency preemption expires if corridor cannot be established within 20 seconds; then system reverts to safe fallback.
- **SI-5 Fail-Safe Default** On loss of connectivity or control errors, edge controllers revert to preprogrammed safe timing and log events.

Test ID TC-SAFETY-01

3.3 Security NFRs High

- **NFR-SEC-1 Authentication** Mutual TLS for control channels; JWT with short lifetime for operator sessions. TC-SEC-01.
- **NFR-SEC-2 Authorization** Role Based Access Control; least privilege enforced. TC-SEC-02.
- **NFR-SEC-3 Audit** Immutable audit logs for control actions and emergency preemptions retained per retention policy. TC-SEC-03.
- **NFR-SEC-4 Data Protection** Encrypt data at rest and in transit; PII anonymized before storage for analytics. TC-SEC-04.

3.4 Maintainability and Extensibility Medium

- Modular microservices, CI/CD pipeline, automated unit and integration tests with code coverage targets ≥ 80%. TC-MAINT-01.

4. External Interfaces Data Schemas and Protocols

Provide canonical JSON schemas and sample control command for developer integration.

4.1 Sensor Telemetry Schema


```
{
  "schema_version": "1.0",
  "timestamp_utc": "2025-10-25T06:00:00Z",
  "source_id": "camera-ny-01",
  "type": "camera_analytics",
  "segment_id": "seg-153",
  "metrics": {
    "vehicle_count": 12,
    "avg_speed_kmph": 28.5,
    "occupancy_pct": 32.4,
    "queue_length_m": 15.2
  },
  "meta": {
    "frame_id": 12345,
    "confidence": 0.92
  }
}
```


4.2 Control Command Schema


```
{
  "schema_version": "1.0",
  "timestamp_utc": "2025-10-25T06:00:10Z",
  "controller_id": "ctrl-45",
  "command_type": "set_phase_timing",
  "payload": {
    "phase_id": 2,
    "green_duration_s": 35,
    "start_offset_ms": 0,
    "mode": "auto",
    "effective_from_utc": "2025-10-25T06:00:15Z",
    "auth_token": "jwt-token"
  },
```

```
  "request_id": "req-98765"
}
```


4.3 Emergency Preemption Request Schema


```
{
  "schema_version": "1.0",
  "timestamp_utc": "2025-10-25T06:10:00Z",
  "vehicle_id": "emerg-ambul-37",
  "vehicle_type": "ambulance",
  "route": ["seg-110","seg-111","seg-112"],
  "eta_per_segment_s": [30, 45, 60],
  "auth_signature": "signed-token"
}
```


4.4 Protocols and Mappings

- Telemetry over MQTT topic structure telemetry/{region}/{source_id} with JSON payload.
- Control via HTTPS REST POST to controller adapter or vendor-specific NTCIP mapping when available.
- WebSocket for low-latency dashboard push updates.

5. Testing Simulation and Evaluation Plan

5.1 Simulation Environment

- Use SUMO microsimulator with a provided city topology and traffic demand files.
- Provide Docker image containing SUMO, RL training stack, and evaluation scripts.
- Each experiment run uses a fixed seed for reproducibility; seeds and dataset versions logged.

5.2 Evaluation Scenarios and Datasets

- **Baseline Fixed-Time** Fixed signal plans derived from historical schedules. 30 runs.
- **Actuated Baseline** Standard actuated controllers using detector inputs. 30 runs.
- **RL Controlled** Trained RL policy deployed in simulation. 30 runs.
- **Incident Scenarios** 10 randomized incidents per scenario type (lane closure, accident) with 30 runs each.

- **Peak Demand Variations** Low, medium, high demand traces testing robustness.

5.3 Metrics Collected

- Average travel time per OD pair.
- Average delay per vehicle.
- Average queue length per segment.
- Throughput vehicles/hour per corridor.
- Emergency vehicle delay.
- Control action frequency and safety violation counts.
- Prediction MAE and calibration metrics.

5.4 Statistical Analysis and Acceptance

- Report mean, standard deviation, and 95% confidence interval for each metric across runs.
- Perform paired t-test between baseline and RL runs for primary metric; require $p < 0.05$ and effect size consistent with $\geq 10\%$ reduction for acceptance.

5.5 Reproducibility Artifacts

- Docker compose and image tags.
- SUMO network and route files with versioning.
- Experiment configs with seeds.
- Scripts to produce evaluation reports and plots.

5.6 Test Case Mapping Example

- TC-RL-01 Run RL policy across 30 runs of defined peak demand topology and compute average travel time reduction versus fixed-time baseline; assert $\geq 10\%$ reduction and $p < 0.05$.

6. Operations Runbook Rollout and Traceability

6.1 Pilot Rollout Plan

- **Phase 0 Pilot Setup** Deploy to a pilot zone of 10 intersections with edge nodes and limited

camera set. Validate telemetry and controller connectivity.

- **Phase 1 Canary** Run RL control in canary mode for 2 weeks with operator approval required for live application. Monitor safety metrics.
- **Phase 2 Gradual Rollout** Add zones incrementally, run A/B testing between RL and baseline in matched corridors.
- **Rollback Criteria** If safety invariant violation occurs, control error rate > 0.5% in 10 min window, or travel time worsens by > 5% for 30 min, then immediate rollback to last safe plan.

6.2 Monitoring and SLOs

- **Control Path SLO** 99.9% successful control application.
- **Telemetry SLO** 99.5% ingestion success.
- **Prediction Service SLO** 99% availability; average inference latency ≤ 500 ms.
- **Alerting** Pager alerts for safety violations, control errors, and degraded SLOs.

6.3 Maintenance and Model Retraining

- **Retrain Cadence** Weekly retrain for short-term models; monthly full retrain for spatial GNN models.
- **Validation Gate** New model must outperform current in simulation on held-out scenarios before deployment.
- **Rollback** Automated model rollback if production metrics degrade beyond configurable thresholds.

6.4 Security and Key Management

- Rotate TLS and sign keys every 90 days.
- Emergency tokens issued per vehicle and revoked after mission completion.
- Admin access via 2FA and audited.

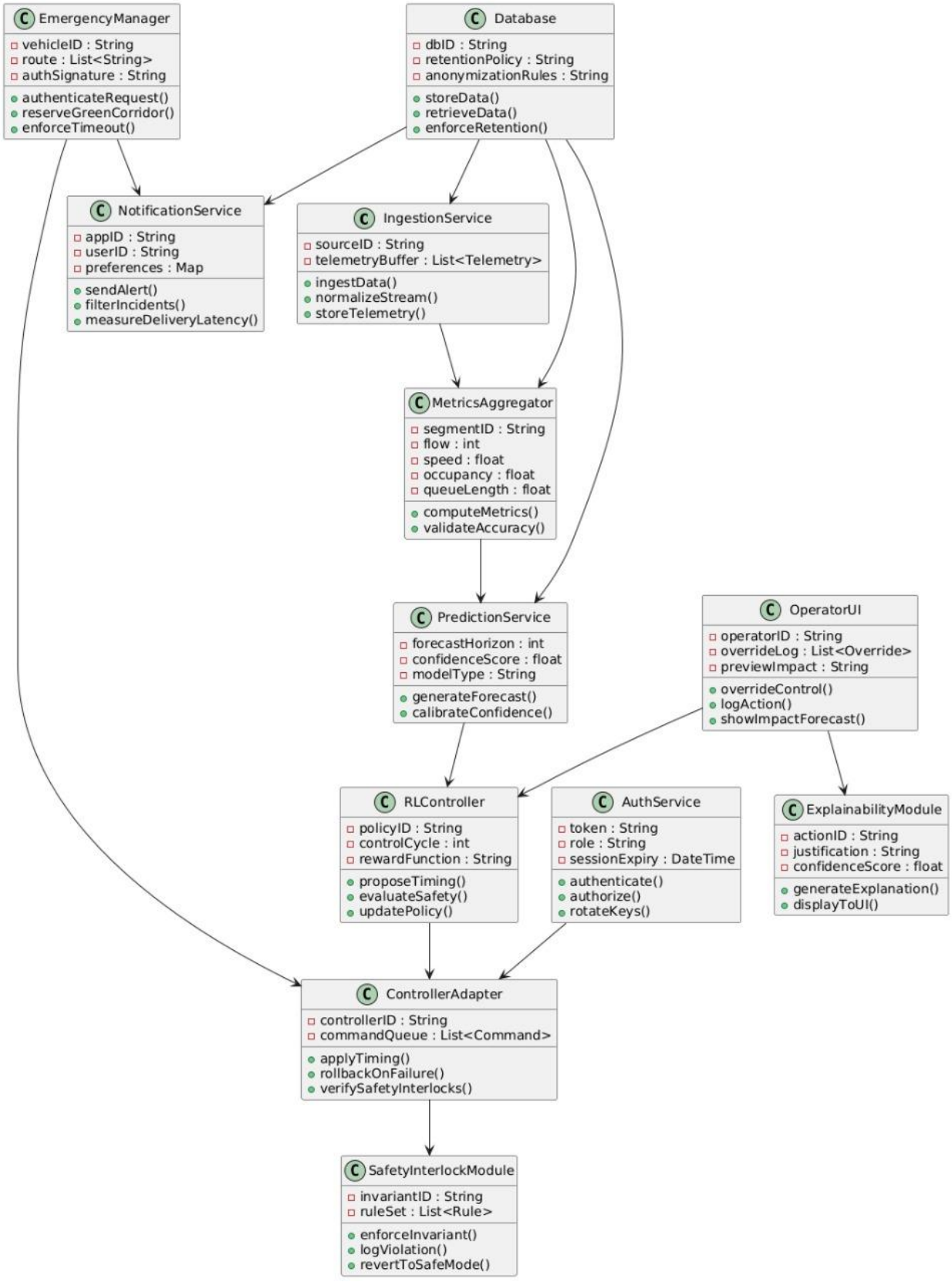
6.5 Traceability Matrix Summary

Requirement ID	Design Module	Test Case ID
FR-1	Ingestion Service	TC-INGEST-01
FR-2	Metrics Aggregator	TC-METRICS-01
FR-3	Prediction Service	TC-PRED-01
FR-4	RL Controller	TC-RL-01
FR-5	Controller Adapter	TC-CONTROL-01
FR-6	Emergency Manager	TC-EMERG-01
FR-7	Notification Service	TC-MOBILE-01
FR-8	Operator UI	TC-UI-01
FR-9	Explainability Module	TC-XAI-01
NFR-SEC-1	Auth Service	TC-SEC-01
SI-1 SI-5	Safety Interlock Module	TC-SAFETY-01

6.6 Deliverables and Documentation Pack

- Updated SRS and traceability matrix.
- API specification with JSON schemas and sample messages.
- Dockerized simulation and evaluation package.
- Operator runbook and safety checklist.
- CI/CD pipelines and model validation scripts.

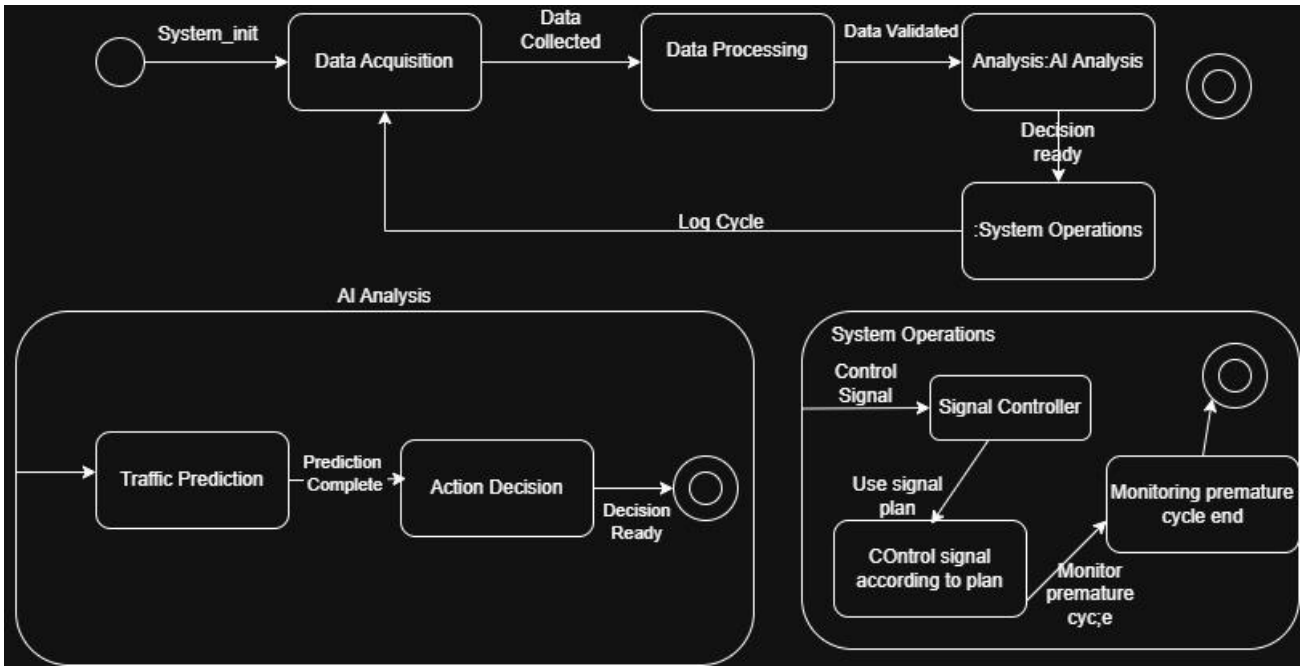
Chapter 3: Class Modeling



Class Name	What the Class Does (Responsibility)	Important Attributes	Important Methods	Related Classes (How They Are Connected)
EmergencyManager	Handles emergency vehicle authentication, green corridor creation, emergency timeout	vehicleID, route, authSignature	authenticateRequest(), reserveGreenCorridor(), enforceTimeout()	• Talks to **IngestionService** to receive emergency data • Uses **NotificationService** to send emergency alerts
Database	Stores telemetry, predictions, logs; enforces retention & anonymization rules	dbID, retentionPolicy, anonymizationRules	storeData(), retrieveData(), enforceRetention()	• Used by **IngestionService**, **MetricsAggregator**, **NotificationService**, **PredictionService** for storage
NotificationService	Sends alerts to users/operators, filters incident messages	appID, userID, preferences	sendAlert(), filterIncidents(), measureDeliveryLatency()	• Receives events from **EmergencyManager** • Displays notifications on **OperatorUI** • Sends user alerts
IngestionService	Collects raw telemetry, normalizes it, writes to DB	sourceID, telemetryBuffer	ingestData(), normalizeStream(), storeTelemetry()	• Sends data to **MetricsAggregator** • Stores raw data in **Database** • Used by **EmergencyManager** for emergency checks
MetricsAggregator	Computes flow, speed, occupancy, queue length	segmentID, flow, speed, occupancy, queueLength	computeMetrics(), validateAccuracy()	• Receives telemetry from **IngestionService** • Sends results to **PredictionService**
PredictionService	Predicts traffic congestion & confidence scores	forecastHorizon, confidenceScore, modelType	generateForecast(), calibrateConfidence()	• Receives metrics from **MetricsAggregator** • Sends predictions to **RLController** • Logs predictions to **Database**
RLController	Reinforcement learning engine that proposes optimal signal timings and evaluates safety	policyID, controlCycle, rewardFunction	proposeTiming(), evaluateSafety(), updatePolicy()	• Uses **PredictionService** inputs • Provides commands to **ControllerAdapter** • Provides explanations via **ExplainabilityModule**
AuthService	Manages authentication, authorization & role permissions for operators	token, role, sessionExpiry	authenticate(), authorize(), rotateKeys()	• Secures access to **OperatorUI** • Interacts with all operator-facing modules

Class Name	What the Class Does (Responsibility)	Important Attributes	Important Methods	Related Classes (How They Are Connected)
OperatorUI	Dashboard for operator monitoring, overrides, logs, and forecast previews	operatorID, overrideLog, previewImpact	overrideControl(), logAction(), showImpactForecast()	• Retrieves explanations from **ExplainabilityModule** • Sends manual overrides to **RLController** • Displays alerts from **NotificationService**
Explainability Module	Produces justifications for RL decisions, displays them in UI	actionID, justification, confidenceScore	generateExplanation(), displayToUI()	• Works with **RLController** to explain decisions • Sends explanation to **OperatorUI**
ControllerAdapter	Sends final commands to physical traffic controllers; handles rollback and interlocks	controllerID, commandQueue	applyTiming(), rollbackOnFailure(), verifySafetyInterlocks()	• Receives optimized timings from **RLController** • Verifies safety via **SafetyInterlockModule**
SafetyInterlock Module	Checks and enforces safety constraints (no conflicting greens, min pedestrian time)	invariantID, ruleSet	enforceInvariant(), logViolation(), revertToSafeMode()	• Works with **ControllerAdapter** to validate commands • Prevents unsafe RL decisions

Chapter 4: State Modeling

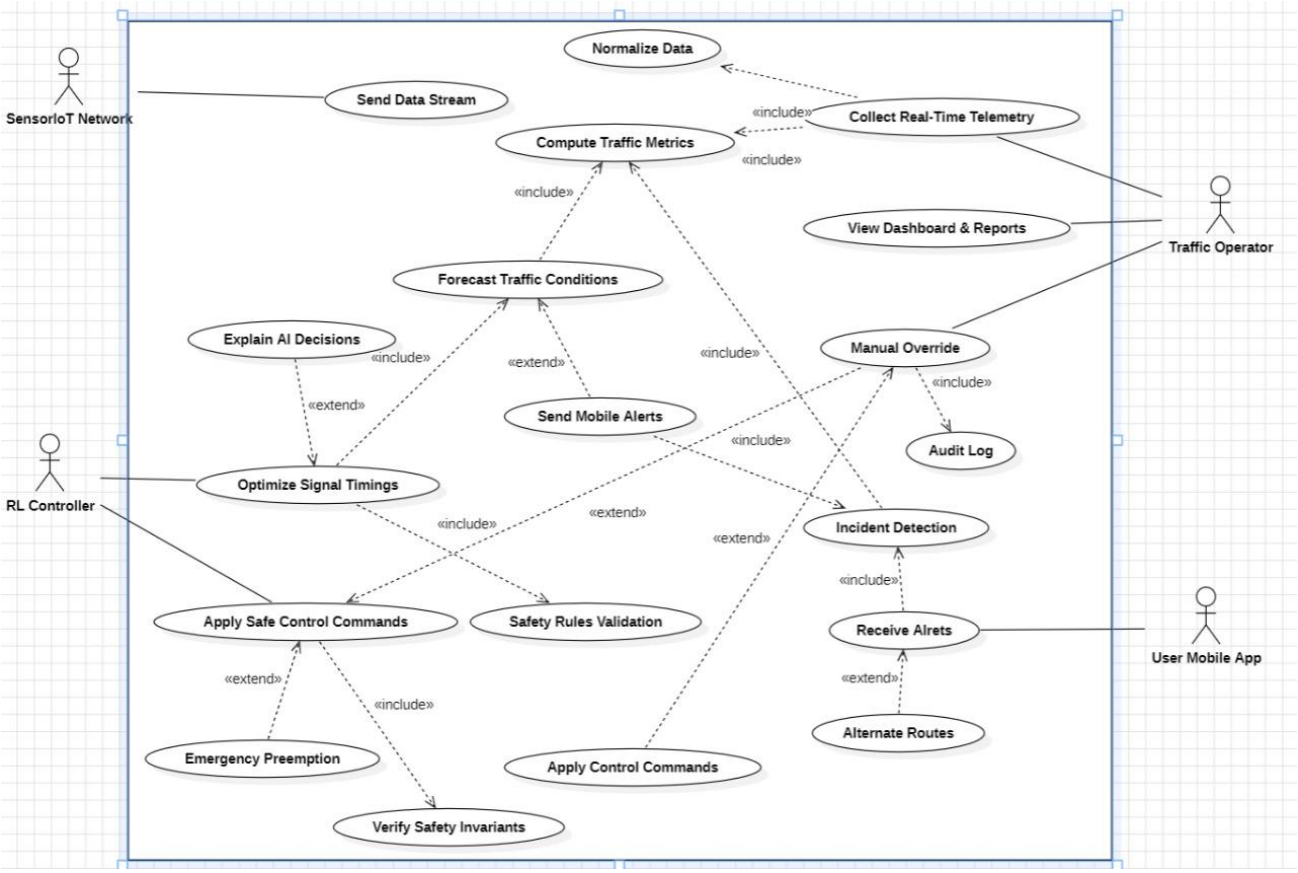


State Name	**What the State Does**	**How It Is Related to System**	**Next State**
System_init	Loads configuration, initializes services, connects to sensors	Starts the whole AI-TMS cycle, prepares system for data ingestion	Data Acquisition
Data Acquisition	Collects real-time telemetry from cameras & IoT sensors	Implements SRS FR-1 (telemetry ingestion)	Data Processing
Data Processing	Normalizes data, checks integrity	Implements FR-1 & FR-2; ensures only valid data reaches AI	If valid → AI Analysis; if invalid → Data Acquisition
AI Analysis – Traffic Prediction	Predicts short-term congestion	Implements FR-3 (5–30 min forecasting)	Action Decision

State Name	What the State Does	How It Is Related to System	Next State
AI Analysis – Action Decision	RL engine computes optimized timing plan	Implements FR-4 (RL optimizer)	System Operations
System Operations	Applies timing plans, monitors signals	Implements FR-5 (Safe control) & FR-6 (Emergency handling)	Normal → Log Cycle; Emergency → Preemption flow
Log Cycle	Stores logs, reports, telemetry snapshots	Supports audit, monitoring, SRS logging requirements	Returns to Data Acquisition (continuous loop)
Fail-Safe (implicit)	Activated on error, connectivity loss	Enforces SRS Safety Invariants SI-4 & SI-5	Data Acquisition after recovery

Chapter 5: Interaction Modeling

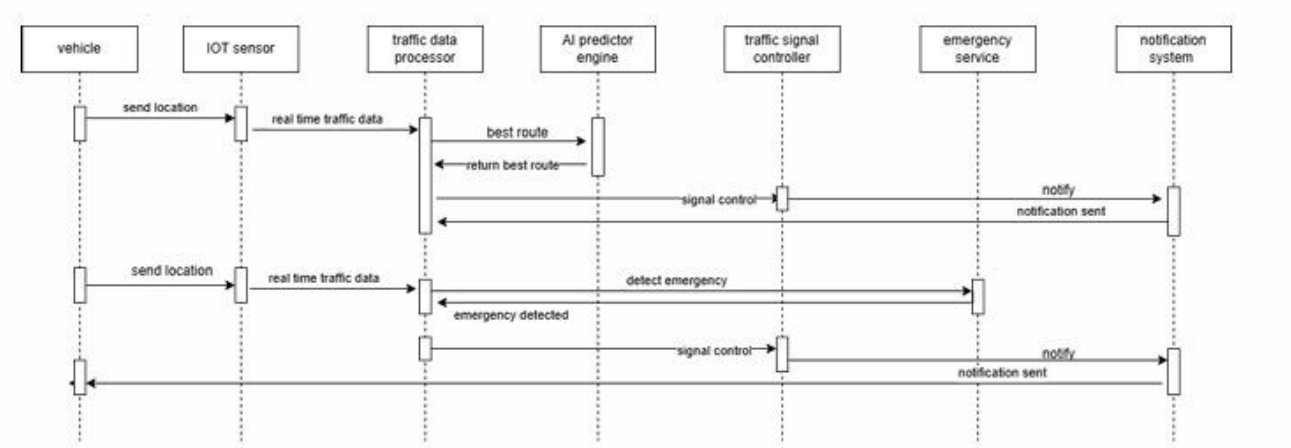
5.1 Use Case Model



Actor	Role in System	Key Related Use Cases	How It Connects to SRS
Sensor / IoT Network	Provides real-time traffic telemetry	Send Data Stream, Collect Telemetry, Normalize Data	FR-1, FR-2
Traffic Operator	Supervises system, performs overrides	View Dashboard, Manual Override, Audit Log, Receive Alerts	FR-8, Security & Audit NFR
RL Controller (AI Engine)	Optimizes signal timing	Forecast Traffic, Optimize Signals, Apply Safe Commands	FR-3, FR-4, FR-5
User Mobile App	Receives alerts & alternate routes	Send Mobile Alerts, Alternate Routes	FR-7
Emergency Vehicle System (implicit)	Triggers emergency preemption	Emergency Preemption, Incident Detection	FR-6, Safety Invariants

Use Case	What It Does	Triggered By	Related SRS Requirement
Send Data Stream	Sends raw data to system	Sensor/IoT Network	FR-1
Normalize Data	Cleans and formats data	System ingestion	FR-1
Compute Traffic Metrics	Calculates flow, speed, occupancy	System	FR-2
Forecast Traffic Conditions	Predicts future traffic	AI Engine	FR-3
Optimize Signal Timings	RL model generates timing plan	RL Controller	FR-4
Apply Control Commands	Sends phase timing changes	AI Engine → Controller	FR-5
Verify Safety Invariants	Checks minimum pedestrian time, no conflicting green	Controller	SI-1 to SI-5
Emergency Preemption	Creates green corridor	Emergency System	FR-6
View Dashboard & Reports	Shows real-time map, congestion, signals	Operator	FR-8
Manual Override	Operator overrides AI	Operator	FR-8
Audit Log	Records all actions	System	NFR-SEC-3
Send Mobile Alerts	Sends alerts to users	Notification Service	FR-7
Alternate Routes	Suggests best path to drivers	System	FR-7
Incident Detection	Detects accidents, breakdowns	AI Engine	FR-3/FR-7

5.2 Sequence Model



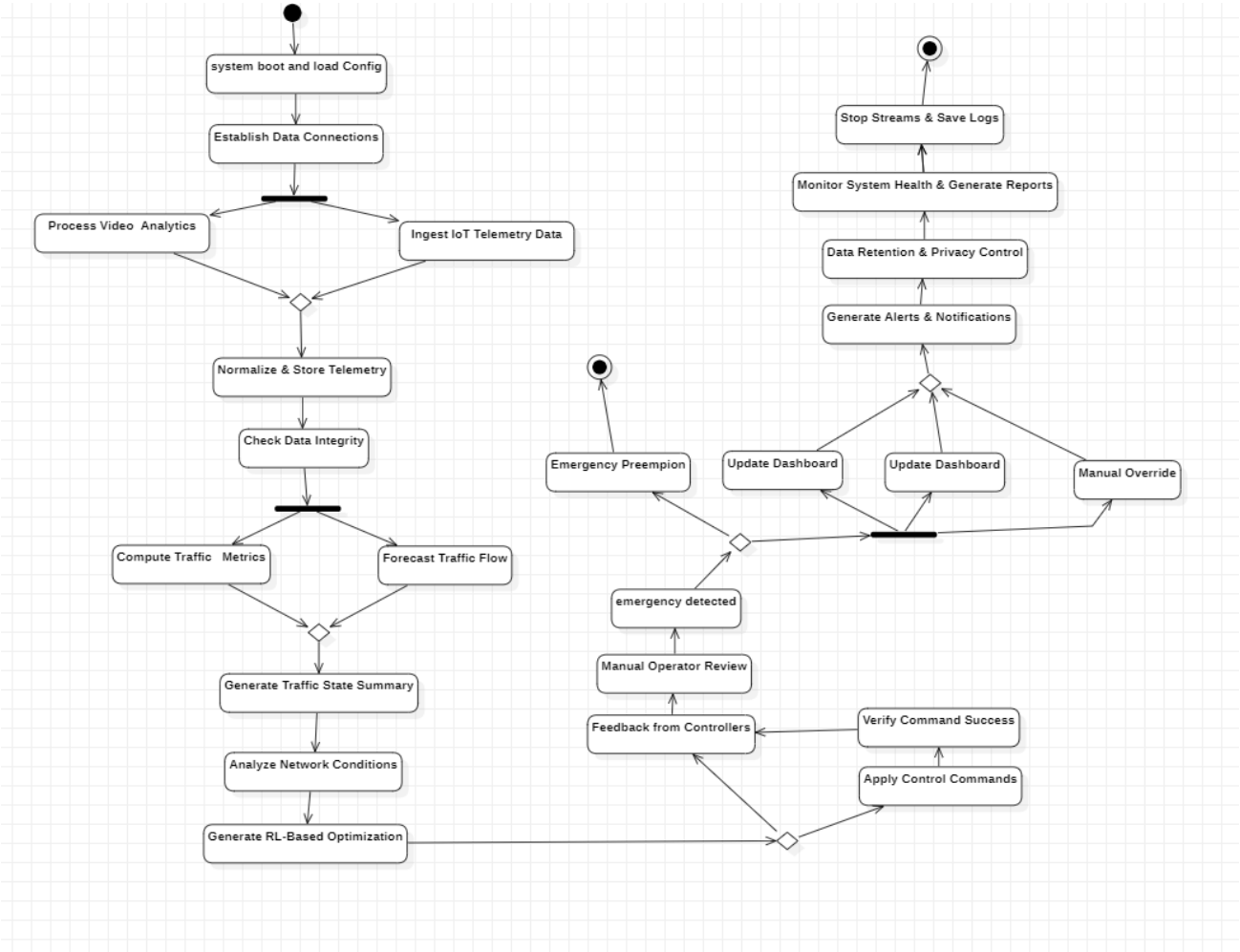
Normal Traffic Scenario

Step	From → To	Message / Action	Meaning
1	Vehicle → IoT Sensor	Send location/speed	Start of telemetry
2	IoT Sensor → Traffic Processor	Send real-time traffic data	Data ingestion (FR-1)
3	Processor → AI Predictor Engine	Send aggregated metrics	Pre-processing
4	AI Predictor → Signal Controller	Send best route/signal plan	RL timing decision (FR-4)
5	Signal Controller → Notification System	Notify users	Alerts (FR-7)

Emergency Scenario

Step	From → To	Action	Meaning
1	Emergency vehicle → IoT sensor	Send emergency request	Emergency detected
2	IoT Sensor → Traffic Processor	Forward request	Processing
3	Processor → AI Predictor	Detect emergency	Trigger preemption
4	AI → Signal Controller	Send emergency signal control	Create green corridor
5	Controller → Notification System	Send alerts	Notify public + emergency service

5.3 Activity Model



Activity	Description	Related SRS Items	Next Activities
System Boot & Load Config	Initializes system	Runbook 6.1	Establish Data Connections
Establish Data Connections	Connect to sensors, cameras	FR-1	Process Video Analytics, Ingest IoT Data (parallel)
Process Video Analytics	Extracts vehicle count, speed	FR-1	Normalize & Store Telemetry
Ingest IoT Telemetry	Gets speed, occupancy, queues	FR-1	Normalize & Store Telemetry

Activity	Description	Related SRS Items	Next Activities
Normalize & Store Telemetry	Clean, format, timestamp	FR-1	Check Data Integrity
Check Data Integrity	Validates input	FR-1, FR-2	Valid → Compute Metrics / Forecast; Invalid → Retry
Compute Traffic Metrics	Flow, occupancy, queue length	FR-2	Generate Traffic Summary
Forecast Traffic Flow	ML predictions	FR-3	Generate Traffic Summary
Generate Traffic Summary	Combine metrics + prediction	FR-2, FR-3	Analyze Network Conditions
Analyze Network Conditions	Detect congestion	FR-3	Generate RL Optimization
Generate RL Optimization	RL timing proposal	FR-4	Emergency Decision
Emergency Preemption	Create green corridor	FR-6	Manual Review / Update Dashboard
Manual Operator Review	Human approval	FR-8	Apply Control Commands
Update Dashboard	Show changes	FR-8	Continue Monitoring
Apply Control Commands	Change signal timings	FR-5	Verify Command Success
Verify Command Success	Check response	FR-5	Generate Alerts / Normal Flow
Generate Alerts & Notifications	Alerts for incidents	FR-7	Data Retention
Data Retention & Privacy	Store logs & anonymize	FR-10	Monitoring

Activity	Description	Related SRS Items	Next Activities
Monitor System Health	Uptime, failures	NFR Performance	Stop Streams & Save Logs
Stop Streams & Save Logs	Controlled shutdown	Runbook	End

Chapter 6: UI Design with Screenshots

Fig 6.1 Home page

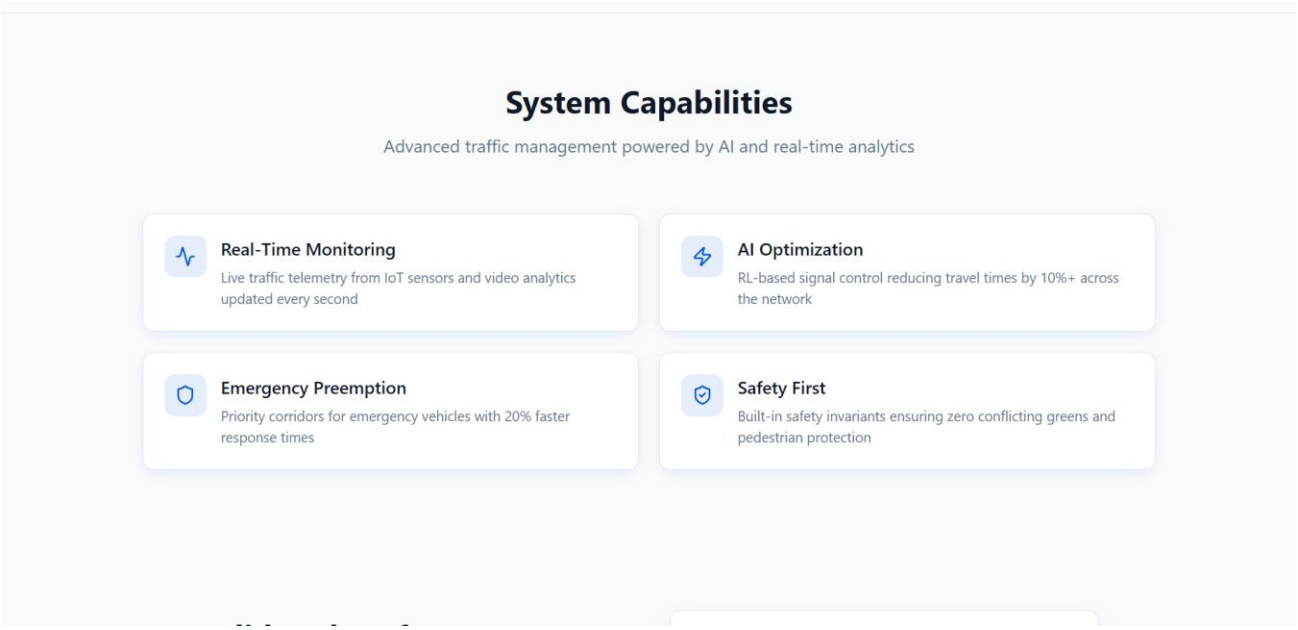


Fig 6.2 Key Features

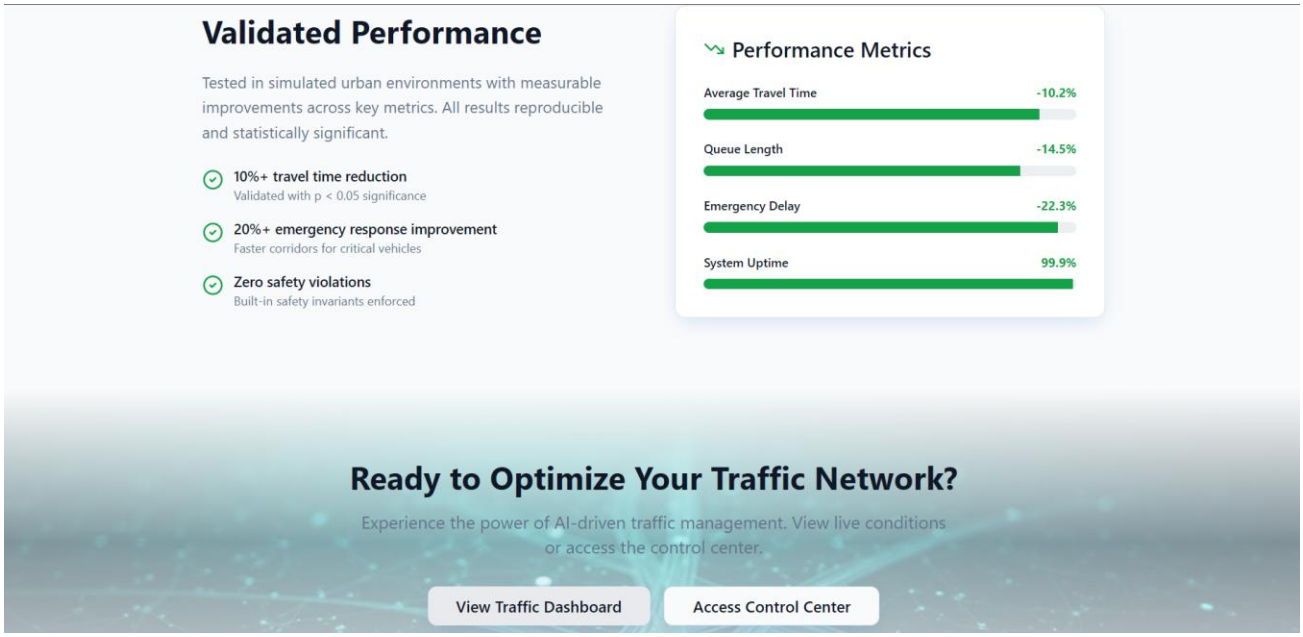


Fig 6.3 Performance

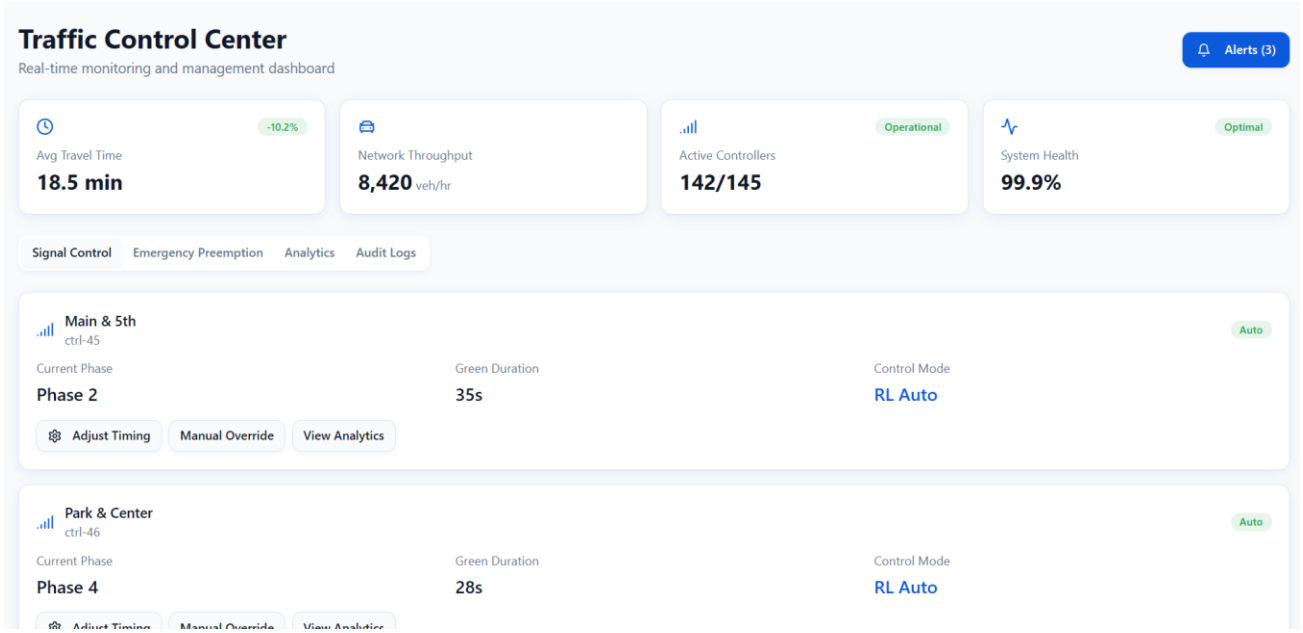


Fig 6.4 Control centre