NeuroFleetX AI-Driven Urban Mobility Optimization

**1. Introduction**

**1.1 Purpose**

The purpose of this project is to design and implement a system that optimizes urban mobility and fleet management using traffic prediction, route optimization, and real-time monitoring. The system aims to reduce congestion, improve fleet efficiency, lower operational costs, and support decision-making with KPIs and reports.

**1.2 Scope**

The system will:

* Predict traffic congestion and travel times for urban roads.
* Optimize routes for individual vehicles and entire fleets.
* Manage fleets including vehicle allocation, scheduling, and tracking.
* Provide real-time dashboards, KPIs, and reports for decision-makers.
* Improve sustainability by reducing fuel consumption and CO₂ emissions.

**2. Functional Requirements**

**2.1 Traffic Prediction**

* Collect historical and real-time traffic data.
* Predict congestion levels and travel times per road segment.
* Provide prediction results via REST APIs.

**2.2 Route Optimization**

* Represent the urban road network using OpenStreetMap or similar sources.
* Calculate optimal routes considering predicted traffic conditions.
* Support Vehicle Routing Problems (VRP) for fleet-wide optimization.
* Incorporate constraints like capacity, delivery time windows, fuel/energy levels, and driver shifts.

**2.3 Fleet Management**

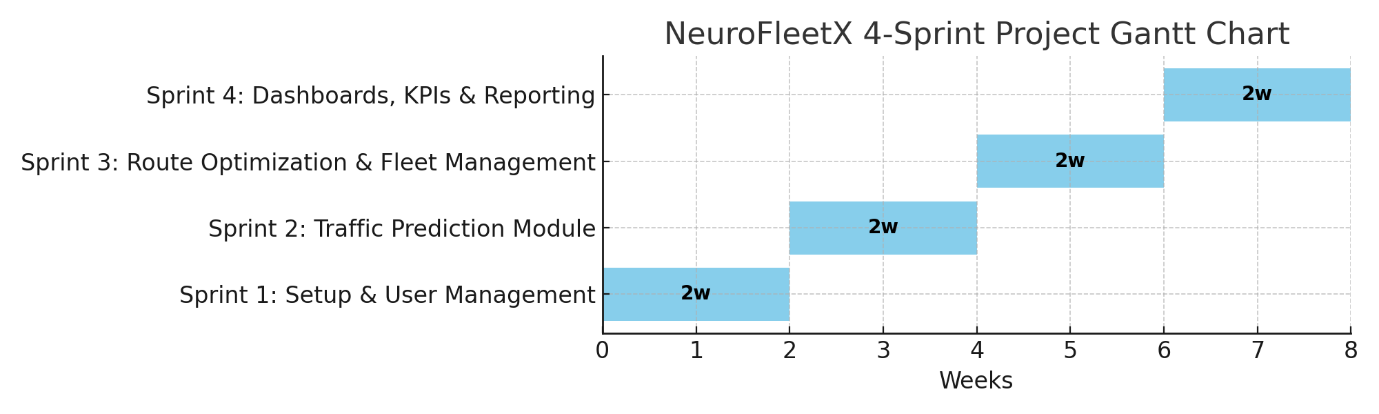
* Register and manage vehicles with details (type, capacity, fuel/EV, current status).
* Assign vehicles to trips dynamically based on route optimization results.
* Track vehicle locations, speed, and fuel/energy usage.
* Predict maintenance schedules (optional advanced feature).

**2.4 Dashboards and KPIs**

* **Real-time Dashboard**:
  + Map visualization of fleet positions and traffic conditions.
  + Highlight congestion hotspots.
* **KPIs**:
  + Fleet utilization rate.
  + Average vs. predicted travel times.
  + Fuel/energy consumption and efficiency.
  + On-time delivery percentage.
* **Reports**:
  + Daily, weekly, and monthly performance reports.
  + Cost savings from optimization.
  + Environmental impact (e.g., CO₂ emissions saved).

**3. System Workflow**

|  |  |  |  |
| --- | --- | --- | --- |
| Sprint | Duration | Tasks | Deliverables |
| Sprint 1 – Setup & User Management | 2 weeks | - Setup Java (Spring Boot), React, MongoDB - User authentication (Login, Register, JWT) - Basic UI for login/register - Database schema design - Unit tests for authentication | - Working login & registration - MongoDB schema - Authentication tests |
| Sprint 2 – Traffic Prediction Module | 2 weeks | - Collect & preprocess traffic data (India sources) - Build ML model for traffic prediction - Develop traffic prediction APIs - UI for traffic charts/maps - Test with sample data | - Traffic prediction API - Dashboard section showing live/predicted traffic |
| Sprint 3 – Route Optimization & Fleet Management | 2 weeks | - Route optimization (shortest + real-time) - Fleet management APIs (add, assign, track) - Integrate Google Maps/OpenStreetMap APIs - UI for fleet monitoring - Backend integration testing | - Fleet dashboard with routes & status - Optimized route suggestion system |
| Sprint 4 – Dashboards, KPIs & Reporting | 2 weeks | - KPI calculation (avg. travel time, utilization, congestion %) - Real-time dashboards in React - Report generation (PDF/Excel) - Final integration testing - Deployment setup (Docker optional) - Documentation (User & Technical) | - Full KPI dashboard - PDF/Excel reports - Final integrated system - Deployment-ready application |

****

1. Collect and store traffic and fleet data.
2. Use machine learning models to predict traffic conditions.
3. Apply route optimization algorithms for fleet operations.
4. Assign optimized routes to vehicles.
5. Track fleet and traffic in real-time.
6. Visualize data on dashboards and generate performance reports.

**4. Technology Stack (Java-based Backend)**

* **Backend Framework**: Spring Boot (REST APIs, services).
* **Database**: MongoDB (for both structured and unstructured data such as GPS traces).
* **Traffic Prediction**: Deeplearning4j / Tribuo / Weka.
* **Route Optimization**: OptaPlanner, GraphHopper, or JGraphT.
* **Visualization**: React frontend with APIs; JasperReports or BIRT for reports.
* **Security**: Spring Security + JWT.

**5. Analysis and Test Cases**

## 1. System Analysis

### 1.1 Problem Analysis

- Rapid urbanization leads to heavy traffic congestion.  
- Inefficient fleet utilization increases cost and pollution.  
- Lack of predictive traffic systems causes delays.

### 1.2 Feasibility Analysis

- Technical Feasibility: Achievable using Java backend, ML libraries, MongoDB, React.  
- Operational Feasibility: Fleet operators and city planners can use dashboards.  
- Economic Feasibility: Reduces fuel and operational costs, improves delivery efficiency.

### 1.3 System Architecture (High-level)

- Data Layer: MongoDB (stores traffic, vehicle, route, and reports data).  
- Application Layer: Java (Spring Boot) for traffic prediction, route optimization, fleet management APIs.  
- Presentation Layer: React dashboards, KPIs, and reports.

## 2. Test Cases

### 2.1 User Authentication (Login & Register)

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case ID | Description | Input | Expected Output |
| UA-01 | Successful registration | Valid username, email, password | User registered successfully, success message |
| UA-02 | Registration with existing email | Already used email | Error: “Email already exists” |
| UA-03 | Registration with weak password | Password < 8 chars | Error: “Password too weak” |
| UA-04 | Successful login | Correct email & password | Redirect to dashboard |
| UA-05 | Login with invalid password | Correct email + wrong password | Error: “Invalid credentials” |
| UA-06 | Login with unregistered email | Email not in DB | Error: “User not found” |
| UA-07 | Empty fields in login | Empty email/password | Error: “All fields are required” |
| UA-08 | Logout | Active session | User logged out, redirected to login page |

### 2.2 Traffic Prediction Module

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case ID | Description | Input | Expected Output |
| TP-01 | Predict congestion for given road segment | Historical + real-time traffic data | Congestion level (Low/Medium/High) |
| TP-02 | Predict travel time | Road segment, time of day | Predicted travel time (in minutes) |
| TP-03 | Handle missing traffic data | Empty/partial input | Default prediction or error message |

### 2.3 Route Optimization Module

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case ID | Description | Input | Expected Output |
| RO-01 | Generate shortest path | Source & destination | Route with minimum distance |
| RO-02 | Optimize considering traffic | Source, destination, traffic data | Fastest route |
| RO-03 | Vehicle routing problem (VRP) | Multiple delivery points & fleet | Optimized routes per vehicle |

### 2.4 Fleet Management

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case ID | Description | Input | Expected Output |
| FM-01 | Register new vehicle | Vehicle details | Vehicle added to DB |
| FM-02 | Assign trip | Vehicle ID, trip details | Trip assigned successfully |
| FM-03 | Track vehicle | GPS input | Updated vehicle location |

### 2.5 Dashboard & KPIs

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case ID | Description | Input | Expected Output |
| DB-01 | Show live fleet status | Real-time data | Map with fleet positions |
| DB-02 | Generate daily report | Date range | PDF/Excel report |
| DB-03 | Display KPIs | Traffic + fleet data | Utilization %, Avg. travel time, etc. |

**5. Deliverables**

* Traffic prediction module (API-based).
* Route optimization service (API-based).
* Fleet management module (vehicles, trips, assignments).
* Real-time dashboard with KPIs.
* Reports (PDF/Excel summaries).

**6. Conclusion**

This project provides an end-to-end system for traffic prediction, route optimization, fleet management, and decision support in urban mobility. It leverages a Java-based backend for scalability, security, and integration, while ensuring real-time insights and analytics to optimize operations and reduce congestion.