NeuroFleetX: AI-Powered Bus Fleet Management System

Milestone Report – Project Documentation

---

1. Introduction

NeuroFleetX is an **AI-powered intelligent transportation platform** designed for urban bus fleets. It is built to solve modern, real-time fleet management challenges.

The system provides core features for improving operations and passenger experience:

1. Smart route optimization
2. Real-time bus tracking
3. Passenger demand forecasting
4. AI-driven arrival time prediction (ETA)
5. Automated operational insights

The system is a **Java Full Stack application** utilizing:

- **Frontend:** React.js
- **Backend:** Java Spring Boot
- **Database:** MongoDB
- **Intelligence:** AI/ML Models (for ETA, demand, and route prediction)

---

2. Core Challenges Addressed

NeuroFleetX is designed to directly resolve critical issues found in traditional urban bus systems:

Challenge	Problem	NeuroFleetX Solution
Unpredictable Bus ETA	Traffic jams and delays cause inaccurate arrival estimation, frustrating passengers.	**AI-Based ETA Prediction** using real-time and historical data.
Lack of Real-Time Visibility	Operators cannot instantly view bus location, performance, or breakdowns.	**Live GPS Tracking** and Admin Dashboard for instant status.
Manual Route Planning	Fixed schedules are inefficient during peak traffic or emergencies.	**Smart Route Optimization** based on real-time traffic intelligence.
Poor Passenger Demand Understanding	No forecasting leads to underutilized buses or severe overcrowding.	**Demand Forecasting** estimates expected passenger load based on history/events.

Challenge	Problem	NeuroFleetX Solution
Weak Decision-Making Tools	Operators rely on manual logs, lacking analytics on fuel efficiency or congestion.	**Performance Dashboards** and **AI-powered Analytics** for clear insights.
No Integrated Ticketing/Seat Updates	Data is scattered across different systems, making boarding slow and confusing.	**Centralized Ticketing** with real-time seat availability updates.

---

3. Functional Requirements (What the System Does)

The system is structured around core operational requirements:

A. Fleet Operations

- **Bus Management:** Add/edit/delete buses; assign driver, route, and capacity; track status (running, parked, maintenance).
- **Real-Time Bus Tracking:** Live GPS monitoring of location, speed, and route progress. Includes Geo-fencing alerts.
- **Route Management:** Define routes, list of stops, stop order, and distance. Suggests alternative peak vs. off-peak routes.

B. Passenger & Intelligence

- **Passenger Ticketing:** Online booking, QR-based authentication, and display of seat availability.
- **Seat Availability Monitoring:** Tracks total, occupied, and available seats, updated via QR scan or AI counting.
- **AI-Based ETA Prediction:** Predicts arrival time for each stop based on traffic, speed, and route history.
- **Traffic Intelligence:** Provides a **Congestion Index** per zone, leveraging historical patterns and predictive congestion alerts.
- **Demand Forecasting:** Uses historical ridership to estimate expected passengers, peak timings, and festival/event surges.
- **Admin Dashboard:** Real-time system overview, charts & analytics, and incident management.

---

4. Non-Functional Requirements

Requirement	Metric / Description
High Performance	API response time must be **less than 200ms**.
Scalability	Must support **5,000+ buses** simultaneously.

Requirement	Metric / Description
Reliability	Automatic failover and real-time data syncing.
Security	**JWT authentication** and strict access control levels (Admin / Driver / Passenger).
Maintainability	Layered architecture, ready for conversion to **Microservices**.
Portability	Deployment via **Docker containers** for all components.

---

5. Technology Stack & Architecture

Technology Stack

Layer	Technologies	Key Tools/Concepts
Frontend	React.js	React Hooks, Material UI/Tailwind, Axios, Maps (Leaflet/Google Maps API), Recharts
Backend	Java Spring Boot	Spring Web, **Spring Security (JWT)**, Spring Data, **WebSockets** (for real-time tracking)
Database	MongoDB	**GeoJSON**, Time-series collections, Collection sharding
AI/ML	Python/Java ML Services	Models: Random Forest, LSTM, XGBoost. Connected via REST API.

System Architecture (High-Level)

The architecture follows a distributed model, separating the data processing and intelligence layers.

$$\text{React UI} \rightarrow \text{Spring Boot API} \begin{cases} \rightarrow \text{MongoDB Database} \\ \rightarrow \text{AI/ML Engine} \end{cases}$$

---

6. MongoDB Database Schema (Key Collections)

The database design uses MongoDB's flexible schema and geospatial capabilities.

Collection	Key Fields	Purpose
users	name, email, role, password	User authentication and role-based access control.
buses	registrationNo, driverId, capacity, routeId	Master data for all vehicles.
routes	routeName, stops (array of stopId), totalDistance	Defines the entire route network.
stops	name, **location (GeoJSON)**, order, routeId	Defines physical bus stop locations.

Collection	Key Fields	Purpose
liveTracking	busId, **location (GeoJSON)**, speed, timestamp	Real-time position and velocity updates (high-frequency data).
tickets	userId, busId, seatNo, fare, qrCode	Record of all passenger bookings.
seatStatus	busId, totalSeats, occupiedSeats	Real-time seat occupancy count for passenger visibility.
trafficData	zone, congestionIndex, timestamp	Real-time and historical traffic condition data.
aiPredictions	busId, **predictedETA** (array), **demandForecast**, routeRecommendation	Stores the intelligent output from the AI/ML Engine.

---

7. AI/ML Model Overview

The system's intelligence relies on three core machine learning components:

1. ETA Prediction

- **Inputs:** Current speed, stop distance, real-time traffic, historical patterns.
- **Model: Random Forest Regression**
- **Output:** Accurate time remaining for the bus to reach each upcoming stop.

2. Passenger Demand Prediction

- **Inputs:** Historical ridership, time of day, day of week, weather, local events.
- **Model: LSTM Time Series** (Long Short-Term Memory)
- **Output:** Expected number of passengers at a given time and location.

3. Route Optimization

- **Inputs:** Congestion index, total distance, stop density, scheduled arrival window.
- **Algorithms: A*** (A-Star) and **Dijkstra's Algorithm**
- **Output:** The fastest and least congested route recommendation for the driver.

---

8. Conclusion

NeuroFleetX successfully integrates modern technologies to deliver a comprehensive, AI-powered solution for bus fleet operations. It significantly simplifies daily management, enhances passenger experience through better visibility and information, and provides smart analytics essential for data-driven decision-making by transport authorities.