Autonomous Predictive Maintenance and Proactive Service Scheduling System

---

1. Introduction

The automotive aftersales ecosystem in India faces major challenges:

- Unexpected vehicle breakdowns
- Poor service planning
- Low customer retention
- Recurring manufacturing defects
- No proactive maintenance engagement

This project presents an **Agentic AI-based Autonomous Predictive Maintenance System** that continuously monitors vehicle health, predicts failures, autonomously schedules service appointments, engages customers via voice assistant, and generates manufacturing quality insights using RCA/CAPA data — all secured with UEBA-based anomaly detection.

---

2. Business Context

A leading automotive OEM aims to:

- Improve vehicle uptime
- Reduce unplanned breakdowns
- Optimize service center utilization
- Improve product quality
- Increase customer satisfaction

The system acts as a digital intelligent layer between:

- Vehicle telematics
- Service network
- Customers
- Manufacturing team

---

3. Problem Statement

Traditional vehicle maintenance is reactive. Customers visit service centers only after breakdown or warning signs, leading to:

- High downtime

- Increased repair costs
- Overloaded service centers
- Poor experience
- Recurring component defects

The objective is to design an **Agentic AI System** where a Master Agent orchestrates multiple Worker Agents to:

1. Analyze real-time vehicle data
2. Predict failures proactively
3. Engage customers via voice
4. Autonomously schedule appointments
5. Generate RCA/CAPA insights
6. Secure all agent interactions using UEBA

---

4. System Architecture

4.1 High-Level Architecture

Frontend (React Dashboard)
↓
Master Agent (Node.js Backend)
↓
Worker Agents:

- Data Analysis Agent
- Diagnosis Agent
- Customer Engagement Agent
- Scheduling Agent
- Manufacturing Insights Module
- UEBA Monitoring Layer

Database: PostgreSQL

---

5. Agentic AI Design

5.1 Master Agent

- Orchestrates all agents
- Monitors vehicle
- Initiates conversation

- Ensures UEBA compliance

API:

POST /api/orchestrate/:vehicleId

5.2 Data Analysis & Diagnosis Agent

- Evaluates:
    - Engine temperature
    - Battery health
    - Brake wear

Risk Scoring:

- LOW
- MEDIUM
- HIGH
- CRITICAL

Output:

- Risk score
- Priority
- Recommended service window
- Failure explanation

5.3 Customer Engagement Agent

Generates persuasive conversation:

- Opening greeting
- Issue explanation
- Consequence
- Benefit
- Service proposal

Frontend supports:

- Chat UI
- Voice playback (SpeechSynthesis API)

5.4 Scheduling Agent

- Fetches available slots
- Suggests earliest slots for critical vehicles
- Confirms booking
- Logs action to UEBA

5.5 Manufacturing Quality Insights Module

Fetches RCA/CAPA records:

- Component
- Root cause
- Corrective action
- Recurrence count

Purpose:

- Identify recurring defects
- Improve product design
- Feed insights back to OEM manufacturing team

5.6 UEBA (User & Entity Behaviour Analytics)

Monitors:

- Agent name
- Action performed
- Resource accessed
- Risk score
- Status (OK / ANOMALY)

Example:
If SchedulingAgent accesses telematics → marked as anomaly.

---

6. Database Design

Tables:

1. vehicles
2. service_slots

3. ueba_logs
4. capa_rca

PostgreSQL used for structured relational storage.

---

7. Functional Requirements

FR1: System shall analyze vehicle health data.
FR2: System shall predict potential failures.
FR3: System shall generate personalized service recommendation.
FR4: System shall provide voice-based customer interaction.
FR5: System shall schedule appointments automatically.
FR6: System shall log all agent actions in UEBA.
FR7: System shall generate manufacturing insights using RCA/CAPA.

---

8. Non-Functional Requirements

- Web-based interface
- REST API architecture
- PostgreSQL backend
- Secure logging
- Real-time response < 2 sec
- Scalable modular agent design

---

9. Technologies Used

Frontend:

- React.js
- CSS
- Web Speech API

Backend:

- Node.js
- Express.js

Database:

- PostgreSQL

Security:

- UEBA logging

10. Key Innovations

- Agentic AI orchestration architecture
- Proactive maintenance prediction
- Voice-based service engagement
- Integrated manufacturing feedback loop
- UEBA-based anomaly detection for AI agents

11. Limitations

- Rule-based prediction (no ML model yet)
- No real-time telematics streaming
- No demand forecasting model
- Feedback agent not fully implemented

12. Future Enhancements

- Machine learning-based failure prediction
- Real-time WebSocket telematics
- Service demand forecasting
- Customer feedback scoring system
- Advanced UEBA anomaly profiling

13. Conclusion

This system demonstrates how Agentic AI can:

- Reduce breakdowns
- Improve service planning
- Enhance customer experience
- Improve manufacturing quality
- Secure autonomous agents

It provides a scalable foundation for intelligent automotive aftersales ecosystems.

🔁 Future Scope – Manufacturing Feedback Loop

1. **Automated Failure–Defect Correlation**
   Link predicted vehicle failures with historical CAPA/RCA records to identify recurring manufacturing defects.
2. **Failure Pattern Clustering**
   Use data analytics to detect defect trends by region, model year, supplier, or batch.
3. **Dynamic RCA Recommendation Engine**
   Automatically suggest probable root causes and preventive actions based on failure trends.
4. **Design Improvement Insights**
   Generate actionable recommendations for component redesign, supplier changes, or material upgrades.
5. **Closed-Loop Learning System**
   Continuously feed service outcomes back to manufacturing to reduce future defect recurrence.

🤖 Future Scope – ML-Based Prediction Models

1. **Supervised Failure Prediction Model**
   Use ML algorithms (Random Forest, XGBoost, Neural Networks) to predict probability of component failure.
2. **Time-Series Degradation Forecasting**
   Apply LSTM/ARIMA models to forecast component wear and Remaining Useful Life (RUL).
3. **Service Demand Forecasting**
   Predict service center workload using historical booking and vehicle usage patterns.
4. **Adaptive Risk Scoring System**
   Replace rule-based thresholds with dynamically learned risk weights.
5. **ML-Based UEBA Enhancement**
   Detect abnormal agent behavior using behavior profiling and anomaly detection models.