Project: Utility Asset Maintenance Tracker

1. Introduction

This document outlines the Low-Level Design for the **Utility Asset Maintenance Tracker**, a system developed to manage the lifecycle and maintenance schedules of critical assets (e.g., transformers, pipelines, substations) in a utility infrastructure.

This design supports both **Java (Spring Boot)** and **.NET (ASP.NET Core)** frameworks.

2. Functional Modules

1. **Asset Registration & Hierarchy Management**
   Enables registration and structured organization of utility assets across locations.
2. **Maintenance Schedule Configuration**
   Allows setup of recurring preventive maintenance plans for different asset types.
3. **Work Order Management**
   Manages creation, tracking, and status updates of asset maintenance work orders.
4. **Technician Assignment & Tracking**
   Facilitates assignment of technicians to work orders based on skills and availability.
5. **Reporting and Compliance Logs**
   Provides detailed reports and logs for maintenance history, technician performance, and regulatory compliance.

3. Technology Stack

- **Frontend**: Angular or React
- **Backend**: REST API-based microservices
- **Database**: Relational Database (MySQL / SQL Server)

4. Module Details

4.1 Asset Registration & Hierarchy Management

Entities

- Asset: AssetID, Name, Type, InstallationDate, Status
- Location: LocationID, AssetID, Region, SiteCode

APIs

- POST /api/assets – Register a new asset
- GET /api/assets – List assets

- PUT /api/assets/{id} – Update asset details
- GET /api/assets/location?region= – Filter assets by location

4.2 Maintenance Schedule Configuration

Entities

- MaintenancePlan: PlanID, AssetID, Frequency (Monthly/Quarterly), TaskList
- Task: TaskID, Description, EstimatedHours

APIs

- POST /api/maintenance-plans – Define a plan
- GET /api/maintenance-plans?assetId= – View plans
- PUT /api/maintenance-plans/{id} – Update plan

4.3 Work Order Management

Entities

- WorkOrder: WorkOrderID, PlanID, ScheduledDate, Status (Open/In Progress/Completed)
- WorkLog: LogID, WorkOrderID, StartTime, EndTime, TechnicianID

APIs

- POST /api/work-orders – Generate work order
- GET /api/work-orders?status= – Filter work orders
- PUT /api/work-orders/{id}/status – Update status

4.4 Technician Assignment & Tracking

Entities

- Technician: TechnicianID, Name, SkillSet, Region
- Assignment: AssignmentID, WorkOrderID, TechnicianID

APIs

- POST /api/assignments – Assign technician to work order
- GET /api/technicians?region= – Filter technicians
- GET /api/assignments?technicianId= – View assignments

4.5 Reporting and Compliance Logs

Reports

- Asset maintenance history
- Technician performance reports
- Upcoming maintenance plans

APIs

- GET /api/reports/asset-history?assetId=
- GET /api/reports/technician-summary?technicianId=
- GET /api/reports/schedule-overview?month=

5. Database Schema (Simplified)

Table Name	Primary Key	Foreign Key
Asset	AssetID	–
Location	LocationID	AssetID
MaintenancePlan	PlanID	AssetID
Task	TaskID	PlanID (optional)
WorkOrder	WorkOrderID	PlanID
WorkLog	LogID	WorkOrderID
Technician	TechnicianID	–
Assignment	AssignmentID	WorkOrderID, TechnicianID

6. Security

- Role-Based Access Control (RBAC): Admin, Supervisor, Technician
- Authentication via OAuth 2.0 with JWT Tokens
- Input validation and logging via middleware

7. Assumptions & Constraints

- Real-time GPS or map tracking is not required
- No alerting or notification service is integrated
- Technicians update statuses manually post-maintenance