Assessment: Group Project – High Level Design

Course & Section: CST8288\_012

Group Members:

Kaur, Prabhsimran

Msiah, Nabila

Purnima, Purnima

Xie, Zhiru

Table of Contents

[1 VERSION HISTORY 2](#_Toc666280799)

[2 INTRODUCTION 3](#_Toc2005631670)

[3 TARGETED AUDIENCE 3](#_Toc889235779)

[4 SCOPE 4](#_Toc250250363)

[5 APPLICATION ARCHITECTURE 5](#_Toc588764561)

[6 BUSINESS ARCHITECTURE 5](#_Toc516230284)

[7 DETAILED DESIGN 7](#_Toc2063369299)

[8 DATA ARCHITECTURE 9](#_Toc790135269)

[9 SECURITY ARCHITECTURE 16](#_Toc59614147)

[10 DEPLOYMENT ARCHITECTURE 16](#_Toc1912686272)

[11 TESTING MODEL 18](#_Toc1167011262)

[12. REFERENCE LIST 20](#_Toc621613132)

[13. ACRONYMS & ABBREVIATIONS 20](#_Toc1361643143)

[14 LIST OF FIGURES 21](#_Toc584783117)

# 1 VERSION HISTORY

|  |  |  |  |
| --- | --- | --- | --- |
| **Version Number** | **Date** | **Description** | **Source of Ideas** |
| 1.0 | July 12, 2025 | Add documentation introduction and stakeholder analysis. | Xie, Zhiru |
| 1.1 | July 13, 2025 | Add application and business architecture, and Detailed Design | Msiah, Nabila |
| 1.2 | July 13, 2025 | Add scope analysis for those within and outside of this documentation’s content. | Xie, Zhiru |
| 1.3 | July 15, 2025 | Added Detailed Deployment Architecture with explanation | Kaur, Prabhsimran |
| 1.4 | July 15, 2025 | Added detailed Data Structure of tables along with ERD and logical diagrams. Also added the Security Architecture. | Purnima, Purnima |
| 1.5 | July 16, 2025 | Added Testing Models and their test cases. | Purnima, Purnima |
| 1.6 | July 16, 2025 | Add contents to Section 7 Detailed Design regarding overall project view (7.1) and MVC implementation (7.2). | Xie, Zhiru |
| 1.7 | July 18, 2025 | Format the document and add section 13 Acronyms & Abbreviations | Zhiru Xie |
| 1.8 | July 18, 2025 | Added Section 14 Table of Figures | Zhiru Xie |

# 2 INTRODUCTION

The **Public Transit Fleet Management System (PTFMS)** is a web-based solution intended to help public transit agencies manage and optimize their vehicle operations. It provides modules for user authentication, vehicle management, real-time **GPS** tracking, component and energy/fuel monitoring, predictive maintenance alerts, and performance reporting for various vehicle types, including diesel buses (**DB**), electric light rail (**ELR**), and diesel-electric trains (**DET**). This High-Level Design (**HLD**) document outlines the architectural, business, data, and design models that form the blueprint for building the system.

# 3 TARGETED AUDIENCE

This solution document is intended for various stakeholders that include but not limited to:

* **Business owners**: Reference to this document to assess how the solution meets operational needs and supports cost-efficiency goals.
* **System developer team**: To understand system structure, patterns, implementation, and maintenance strategies.
* **System Architects**: Review architecture choices to ensure scalability, separation of concerns, and adherence to design best practices.
* **System quality assurance team**: To plan and design test scenarios based on architecture and use cases.
* **Transit managers and operators**: To learn about the system functions as well as the proper approaches to use them.
* **Marketing Staff**: Leverage insights from the reporting and analytics modules to shape promotional campaigns or funding proposals.
* **Students and Learners**: Use High-Level Design (HLD) as a case study to understand software architecture, design patterns, and full-stack development in a real-world scenario.

# 4 SCOPE

**4.1 In Scope**

This documentation elaborates the following content as they are the primary focus of the business objectives:

* Role-based user management system
* Vehicle registration and configuration
* Real-time GPS tracking and manual break logging
* Monitoring and alerting for energy/fuel usage
* Maintenance scheduling based on predictive metrics
* Reporting dashboards for performance, usage, and costs
* Database schema, data retrieval and basic operations (**CRUD**)
* Web application deployment on Apache Tomcat server

**4.2 Out Of Scope**

Although some other aspects also play important roles in software development, such as trans-platform compatibility, **SDK**, performance and structural optimization,they will not (or at least not in large paragraphs) be present in this documentation. Specifically, these include:

* Mobile application support
* Encryption methods
* Compound design patterns
* Third-party API and external services

# 5 APPLICATION ARCHITECTURE

**5.1 Overview**

The application follows a 3-tier architecture, ensuring separation of concerns and scalability. Each tier is responsible for a distinct part of the system’s operation:

**5.2** **Presentation Layer**

* Built using HTML and Java Servlets
* Handles user interaction (login, registration, vehicle entry, etc.)

**5.3 Business Logic Layer**

* Java Servlets and Controllers
* Validates user inputs, performs logic, and acts as a mediator.

**5.4 Data Access Layer (DAO)**

* Uses JDBC with MySQL 8.0
* Handles CRUD operations with prepared statements to prevent SQL injection

# 6 BUSINESS ARCHITECTURE

### **6.1 Use Case Diagram**

**Actors:**

* **Operator**: Views dashboard, logs GPS and fuel
* **Manager**: Manages vehicles, reviews reports, receives alerts

**Use Cases:**

* Register/Login
* Register Vehicle
* Log GPS Location
* Log Fuel Consumption
* Trigger/View Maintenance Alerts
* View Reports

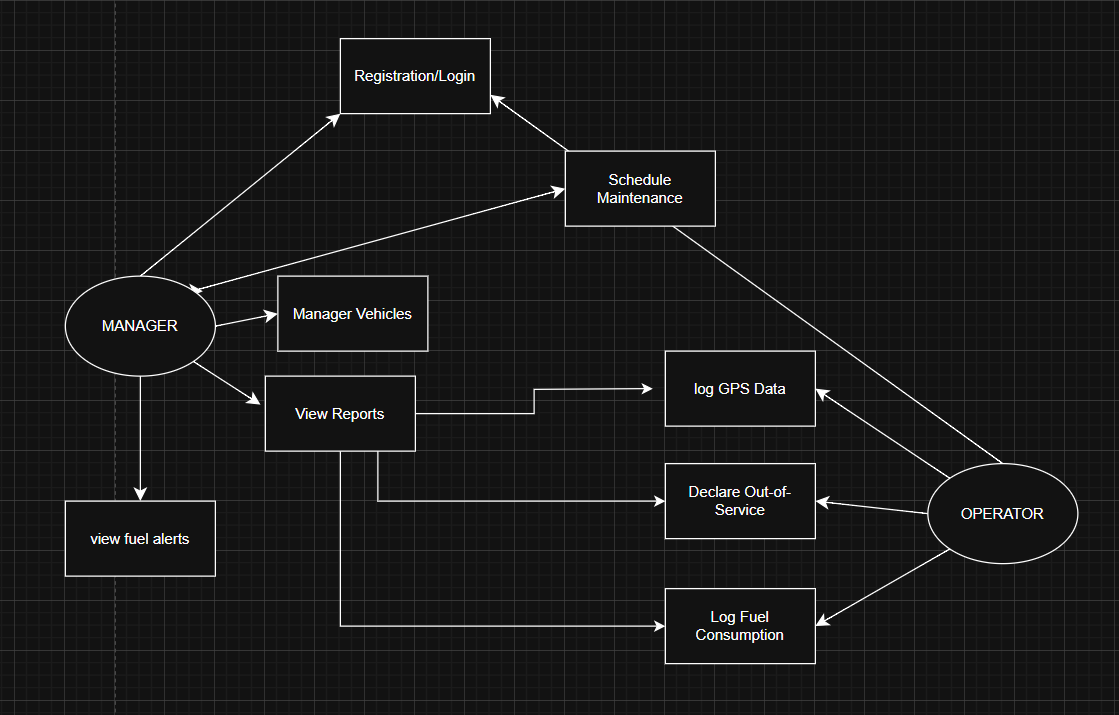


Figure 1 – Use Case Diagram

### **6.2 Use Case Descriptions**

**Use Case: Register/Login**

* **Actor:** Manager, Operator
* **Flow:**
  + User fills form
  + Servlet hashes password
  + DAO stores/compares credentials
  + Session started

**Use Case: Register Vehicle**

* **Actor:** Manager
* **Flow:**
  + Form with vehicle details
  + DAO saves to DB
  + Route assigned

# 7 DETAILED DESIGN

**7.1 Logic Flow Overview**

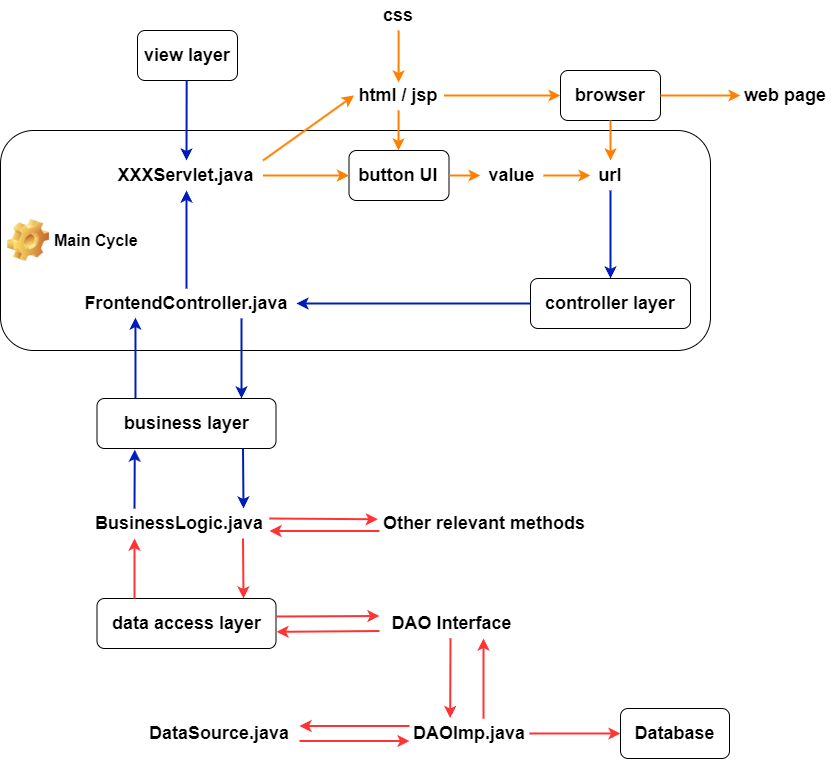


Figure 2 – Detailed Design

**7.2 Core Components**

**7.2.1 Project View**

* Configuration File

Application heavily relies on how the web page URL, Java servlet files, and HTML/JSP files are associated with each other. web.xml in this case is the key to create and maintain these associations.

* HTML/JSP File

The presentation tasks are mostly done by HTML/JSP files and embedded HTML/JSP content inside Java servlet files. The web browser renders the contents inside those files so that the users may visualize them in the form of web pages.

* Java Servlet File (Including the controller)

The Java servlet files are responsible for dispatching web requests to the respective sub-servlet and processing data that is sent, requested, and received during the transition.

**7.2.2 Beneath The White Box**

This project strictly enforces the MVC pattern for higher modularization, easier maintenance, and extension. The MVC pattern consists of three main parts: the model for data, the view for presentation, and the controller for dispatching tasks.

* Model

Java DTO (Data Transfer Object) class represents the model since itself in nature is an imitation of database instance that contains the same attributes while offering accessor and mutator at the same time.

* View

The HTML and JSP files are the primary source of the view since they are the rendering target of the browser. In addition, most Java servlet classes also play essential roles in view formulating because they contain embedded HTML and JSP contents.

* Controller

The FrontControllerServlet.java is the general and the only controller in this project. Its responsibility is to dispatch web requests and forward them to the proper Java servlet class for processing and presenting. The controller itself does not contain any view elements.

**7.3 Component Diagram**

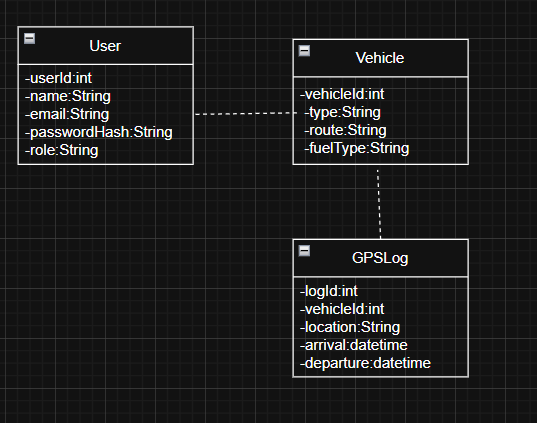


Figure 3 – Component Diagram

[register.html] ---> [RegisterServlet] ---> [UserDAO] ---> [MySQL DB]

[login.html] ---> [LoginServlet] ---> [UserDAO] ---> [MySQL DB]

[VehicleServlet] ---> [VehicleDAO] ---> [MySQL DB]

# 8 DATA ARCHITECTURE

**8.1 ERD Model**

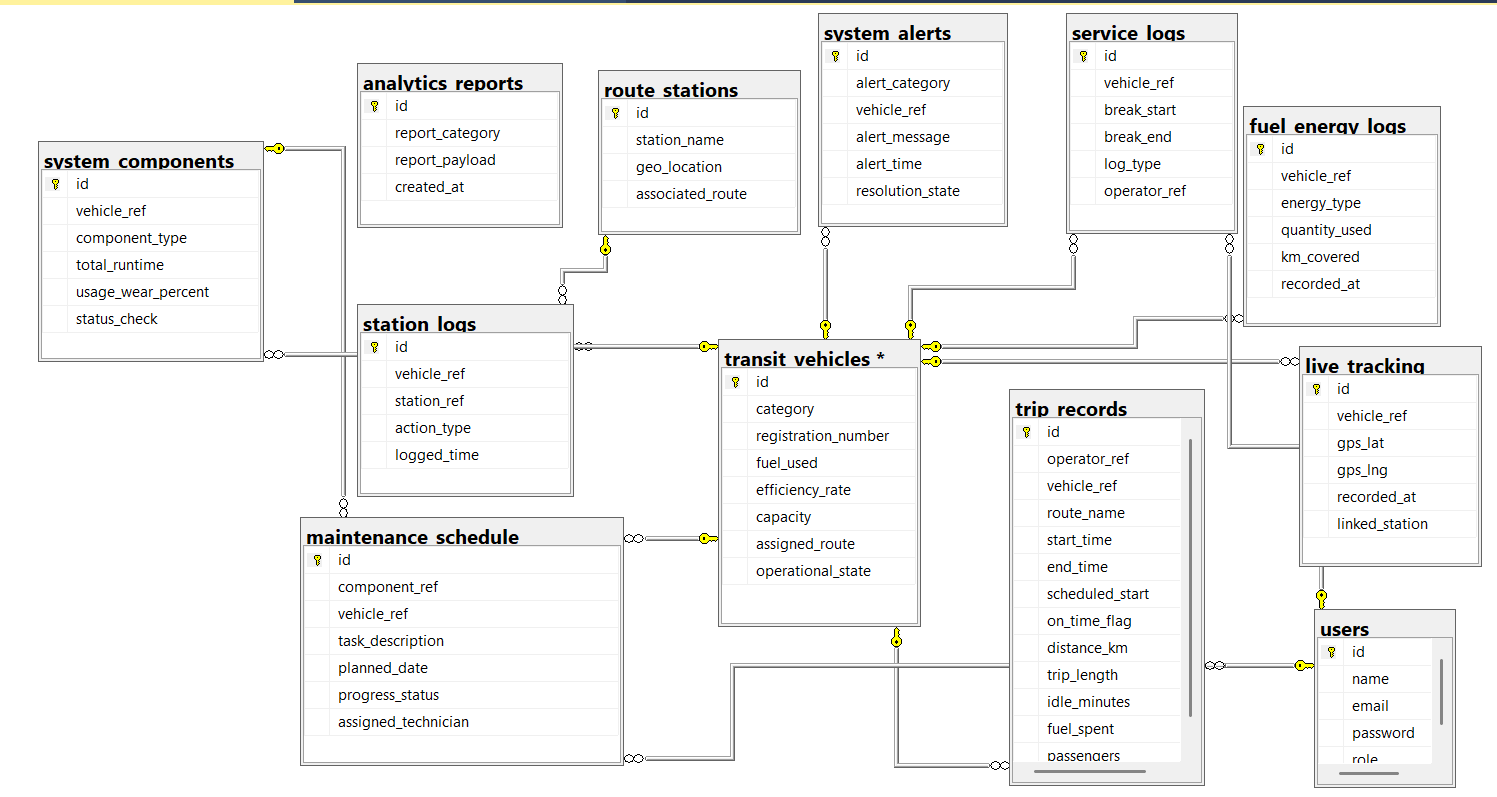


Figure 4 – Logical Model

**8.2 Physical / Logical Model**

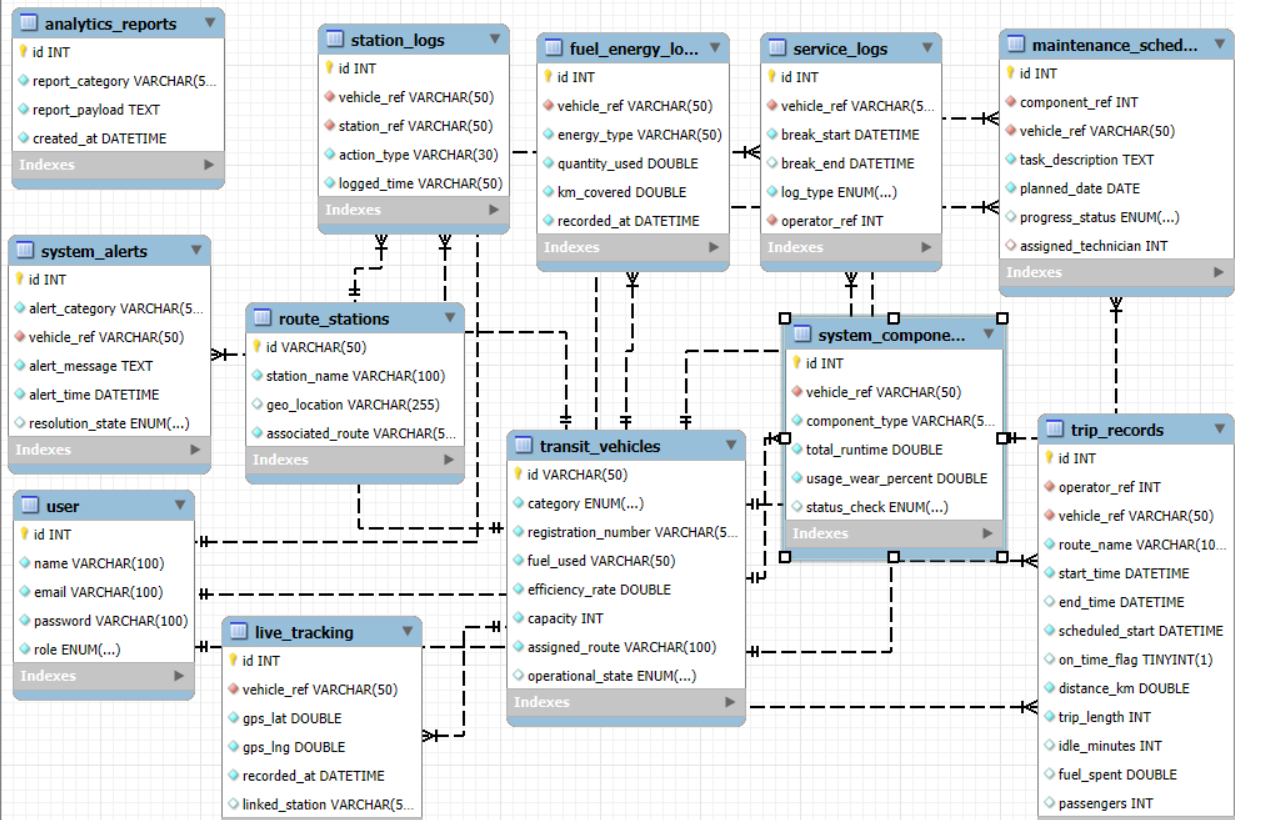


Figure 5 – Physical Model

* User table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Key** | **Description** | **Constraints** |
| id | INT | PK | It's the Primary Key | Auto\_Increment |
| name | VARCHAR (100) |  |  | Not Null |
| email | VARCHAR (100) | Unique | Every user must have a unique email ID | Not Null, Unique |
| password | VARCHAR (100) |  |  | Not Null |
| role | ENUM () |  | (Operator/Manager) | Not Null |

* Transit\_vehicles table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Key** | **Description** | **Constraints** |
| id | INT | PK | It's the Primary Key | **PK** |
| Registration Number | VARCHAR (50) | Unique | LICENCE\_PLATE. Every Vehicle must have a unique License Plate number. | Not Null,Unique |
| category | ENUM () |  | DIESEL BUS, ELECTRIC LIGHT RAIL, DIESEL ELECTRIC TRAIN. | Not Null |
| Fuel\_used | VARCHAR (50) |  |  | Not Null |
| capacity | INT |  |  | Not Null |
| Assigned\_route | VARCHAR (100) |  |  | Not Null |
| Operational\_stage | ENUM | Default:Active | Route assigned to vehicle | Not Null |
| efficiency\_rate | DOUBLE |  | Km/l or km/kWh | Not Null |

* **Live Tracking Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Key** | **Constraints** | **Description** |
| id | INT | PK | Auto\_Increment | Primary Key |
| vehicle\_ref | VARCHAR(50) | FK | NOT NULL | Foreign Key |
| gps\_lat | DOUBLE |  | NOT NULL | Latitude |
| gps\_lng | DOUBLE |  | NOT NULL | Longitude |
| recorded\_at | DATETIME |  | NOT NULL | Timestamp of location Record |
| linked\_station | VARCHAR(50) |  |  | Station Name |

* **Route\_stations table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Key** | **Description** | **Constraints** |
| id | INT | PK | It's the Primary Key | PK |
| Station\_name | VARCHAR (100) |  |  | NOT NULL |
| Geo\_location | VARCHAR (255) |  | Geographic coordinates of station |  |
| Associated\_route | VARCHAR (50) |  |  | NOT NULL |

* STATION\_LOG Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Key** | **Description** | **Constraints** |
| id | INT | PK | It's the Primary Key | Auto Increment |
| Vehicle\_ref | VARCHAR (50) | FK | Foreign Key References  Transit\_vehicles | NOT NULL |
| Station\_ref | VARCHAR (50) | FK | Foreign Key References  Route\_stations | NOT NULL |
| Action\_type | VARCHAR (30) |  | Arrival or Departure | NOT NULL |
| Logged\_time | VARCHAR (50) |  | Time of arrival or departure | NOT NULL |

* *Service Logs* table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Key** | **Description** | **Constraints** |
| id | INT | PK | It's the Primary Key | AUTO\_INCREMENT |
| Vehicle\_ref | VARCHAR(50) | FK | It's the Foreign Key References transit\_tables | NOT NULL |
| Break\_start | DATETIME |  |  | NOT NULL |
| Break\_end | DATETIME |  |  |  |
| Log\_type | ENUM |  | Break or OutOfService | NOT NULL |
| Operator\_ref | INT | FK | References user(id) | NOT NULL |

* Fuel Energy Logs table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Key** | **Description** | **Constraints** |
| id | INT | PK | It's the Primary Key | Auto\_Increment |
| Vehicle\_ref | VARCHAR (50) | FK | Foreign Key References transit\_vehicles(id) | NOT NULL |
| energy\_type | VARCHAR (50) |  |  | NOT NULL |
| quantity\_used | DOUBLE |  |  | NOT NULL |
| Km\_covered | DOUBLE |  | Distance covered during consumption | NOT NULL |
| Recorded\_at | DATTIME |  | Timestamp of energy consumption record | NOT NULL |

* System Components table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Key** | **Description** | **Constraints** |
| id | INT | PK | It's the Primary Key | Auto\_Increment |
| Vehicle\_ref | VARCHAR (50) | FK | It's the Foreign Key | NOT NULL |
| Component\_type | VARCHAR (50) |  |  | NOT NULL |
| Total\_runtime | Double |  |  | NOT NULL |
| Usage\_wear\_percent | DOUBLE |  |  | NOT NULL |
| Status\_check | ENUM |  | Normal, Warning, Critical | Default:Normal |

* System Alert table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Key** | **Description** | **Constraints** |
| id | INT | PK | It's the Primary Key | Auto\_Increment |
| Alert\_category | VARCHAR (50) | - |  | NOT NULL |
| Vehicle\_ref | VARCHAR (50) | FK | It's the Foreign Key | NOT NULL |
| Alert\_message | TEXT |  |  | NOT NULL |
| Alert\_time | DATETIME |  |  | NOT NULL |
| Resolution\_state | ENUM |  |  | DEFAULT: ‘PENDING’ |

**Maintenance Schedule Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Key** | **Constraints** | **Description** |
| id | INT | PK | Auto\_Increment | Primary Key |
| component\_ref | INT | FK | NOT NULL | Foreign Key References syste  m\_components |
| vehicle\_ref | VARCHAR(50) | FK | NOT NULL | Foreign Key References transit  \_vehicles |
| task\_description | TEXT |  | NOT NULL | Description of Maintenance ta  sk |
| planned\_date | DATE |  | NOT NULL | Scheduled Maintenance Date |
| progress\_status | ENUM |  | Default: 'Scheduled' | Scheduled, InProgress, Comp  leted. |
| assigned\_technician | INT | FK |  | Foreign Key References user(id) |

**Analytics Reports Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Key** | **Constraints** | **Description** |
| ID | INT | PK | Auto\_Increment | Primary Key |
| report\_category | VARCHAR(50) |  | NOT NULL | Category/type of report |
| report\_payload | TEXT |  | NOT NULL | JSON format |
| created\_at | DATETIME |  | NOT NULL | Report Generation Timestamp |

**Trips Records Table(Performance Tracking)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Key** | **Constraints** | **Description** |
| id | INT | PK | Auto\_Increment | Trip Record |
| operator\_ref | INT | FK | NOT NULL | References user(id) |
| vehicle\_ref |  |  |  |  |

**8.3 SQL DDL Script**

**CREATE DATABASE ptfms;**

-- Recreate and select the transit database

DROP DATABASE IF EXISTS ptfms\_db;

**-- Users Table: Stores login credentials and roles (Feature Ref: FR-01)**

CREATE TABLE user (

id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(100) NOT NULL,

email VARCHAR(100) NOT NULL UNIQUE,

password VARCHAR(100) NOT NULL, -- Store hashed password in production role ENUM('MANAGER', 'OPERATOR') NOT NULL

);

**-- Vehicles Table: Transit vehicle metadata (Feature Ref: FR-02)**

CREATE TABLE transit\_vehicles (

id VARCHAR(50) PRIMARY KEY,

category ENUM('DieselBus', 'ElectricLightRail', 'DieselElectricTrain') NOT NULL, registration\_number VARCHAR(50) UNIQUE NOT NULL,

fuel\_used VARCHAR(50) NOT NULL,

efficiency\_rate DOUBLE NOT NULL,

capacity INT NOT NULL, assigned\_route VARCHAR(100) NOT NULL,

operational\_state ENUM('Active', 'Inactive', 'Maintenance') DEFAULT 'Active'

);

-- **Real-Time Location Tracking (Feature Ref: FR-03)**

CREATE TABLE live\_tracking (

id INT AUTO\_INCREMENT PRIMARY KEY,

vehicle\_ref VARCHAR(50) NOT NULL,

gps\_lat DOUBLE NOT NULL,

gps\_lng DOUBLE NOT NULL,

recorded\_at DATETIME NOT NULL,

linked\_station VARCHAR(50),

FOREIGN KEY (vehicle\_ref) REFERENCES transit\_vehicles(id)

);

-- **Transit stations**

CREATE TABLE route\_stations (

id VARCHAR(50) PRIMARY KEY,

station\_name VARCHAR(100) NOT NULL,

geo\_location VARCHAR(255),

associated\_route VARCHAR(50) NOT NULL

);

-**- Arrival/Departure logs at stations**

CREATE TABLE station\_logs (

id INT AUTO\_INCREMENT PRIMARY KEY,

vehicle\_ref VARCHAR(50) NOT NULL,

station\_ref VARCHAR(50) NOT NULL,

action\_type VARCHAR(30) NOT NULL,

logged\_time VARCHAR(50) NOT NULL,

FOREIGN KEY (vehicle\_ref) REFERENCES transit\_vehicles(id),

FOREIGN KEY (station\_ref) REFERENCES route\_stations(id) );

-**- Operator Break and Out-of-Service Log (FR-03)**

CREATE TABLE service\_logs (

id INT AUTO\_INCREMENT PRIMARY KEY,

vehicle\_ref VARCHAR(50) NOT NULL,

break\_start DATETIME NOT NULL,

break\_end DATETIME,

log\_type ENUM('Break', 'OutOfService') NOT NULL,

operator\_ref INT NOT NULL,

FOREIGN KEY (vehicle\_ref) REFERENCES transit\_vehicles(id),

FOREIGN KEY (operator\_ref) REFERENCES user(id)

);

-- **Fuel and Energy Monitoring Table (FR-04)**

CREATE TABLE fuel\_energy\_logs (

id INT AUTO\_INCREMENT PRIMARY KEY,

vehicle\_ref VARCHAR(50) NOT NULL,

energy\_type VARCHAR(50) NOT NULL,

quantity\_used DOUBLE NOT NULL,

km\_covered DOUBLE NOT NULL,

recorded\_at DATETIME NOT NULL,

FOREIGN KEY (vehicle\_ref) REFERENCES transit\_vehicles(id)

);

-- **Vehicle Components and Diagnostics (FR-05)**

CREATE TABLE system\_components (

id INT AUTO\_INCREMENT PRIMARY KEY,

vehicle\_ref VARCHAR(50) NOT NULL,

component\_type VARCHAR(50) NOT NULL,

total\_runtime DOUBLE NOT NULL,

usage\_wear\_percent DOUBLE NOT NULL,

status\_check ENUM('Normal', 'Warning', 'Critical') DEFAULT 'Normal',

FOREIGN KEY (vehicle\_ref) REFERENCES transit\_vehicles(id) );

-- **Maintenance Alerts (FR-05)**

CREATE TABLE system\_alerts (

id INT AUTO\_INCREMENT PRIMARY KEY,

alert\_category VARCHAR(50) NOT NULL,

vehicle\_ref VARCHAR(50) NOT NULL,

alert\_message TEXT NOT NULL,

alert\_time DATETIME NOT NULL,

resolution\_state ENUM('Pending', 'Resolved') DEFAULT 'Pending',

FOREIGN KEY (vehicle\_ref) REFERENCES transit\_vehicles(id)

);

-- **Maintenance Task Scheduler (FR-05)**

CREATE TABLE maintenance\_schedule (

id INT AUTO\_INCREMENT PRIMARY KEY,

component\_ref INT NOT NULL,

vehicle\_ref VARCHAR(50) NOT NULL,

task\_description TEXT NOT NULL,

planned\_date DATE NOT NULL,

progress\_status ENUM('Scheduled', 'InProgress', 'Completed') DEFAULT 'Scheduled', assigned\_technician INT,

FOREIGN KEY (component\_ref) REFERENCES system\_components(id),

FOREIGN KEY (vehicle\_ref) REFERENCES transit\_vehicles(id),

FOREIGN KEY (assigned\_technician) REFERENCES user(id) );

-- **Analytics and Summary Reports (FR-06)**

CREATE TABLE analytics\_reports (

id INT AUTO\_INCREMENT PRIMARY KEY,

report\_category VARCHAR(50) NOT NULL,

report\_payload TEXT NOT NULL,

created\_at DATETIME NOT NULL

);

-- **Trips & Operator Performance** Log

CREATE TABLE trip\_records (

id INT AUTO\_INCREMENT PRIMARY KEY,

operator\_ref INT NOT NULL,

vehicle\_ref VARCHAR(50) NOT NULL,

route\_name VARCHAR(100) NOT NULL,

start\_time DATETIME NOT NULL,

end\_time DATETIME,

scheduled\_start DATETIME NOT NULL,

on\_time\_flag BOOLEAN DEFAULT FALSE,

distance\_km DOUBLE NOT NULL,

trip\_length INT NOT NULL, -- minutes idle\_minutes INT DEFAULT 0,

fuel\_spent DOUBLE,

passengers INT,

FOREIGN KEY (operator\_ref) REFERENCES user(id),

FOREIGN KEY (vehicle\_ref) REFERENCES transit\_vehicles(id)

);

-- **Sample Users**

INSERT INTO user (name, email, password, role)

VALUES

('John Doe', '[john.doe@transit.com](mailto:john.doe@transit.com)', 'pass123', 'Manager'),

('Jane Smith', '[jane.smith@transit.com](mailto:jane.smith@transit.com)', 'pass456', 'Operator'),

('Alice Brown', '[alice.brown@transit.com](mailto:alice.brown@transit.com)', 'pass789', 'Operator');

-- **Sample Vehicles**

INSERT INTO transit\_vehicles (id, category, registration\_number, fuel\_used, efficiency\_rate, capacity, assigned\_route, operational\_state)

VALUES

('BUS001', 'DieselBus', 'B123', 'Diesel', 0.5, 50, 'Route A', 'Active'),

('RAIL001', 'ElectricLightRail', 'R456', 'Electricity', 2.0, 100, 'Route B', 'Active'),

('TRAIN001', 'DieselElectricTrain', 'T789', 'Diesel', 1.5, 200, 'Route C', 'Maintenance');

**-- Sample GPS logs**

INSERT INTO live\_tracking (vehicle\_ref, gps\_lat, gps\_lng, recorded\_at, linked\_station)

VALUES

('BUS001', 45.4215, -75.6972, '2025-03-27 08:00:00', 'Station1'),

('RAIL001', 45.4250, -75.6900, '2025-03-27 09:00:00', 'Station2'),

('TRAIN001', 45.4300, -75.6800, '2025-03-27 10:00:00', 'Station3');

-- **Sample Operator Logs**

INSERT INTO service\_logs (vehicle\_ref, break\_start, break\_end, log\_type, operator\_ref)

VALUES

('BUS001', '2025-03-27 12:00:00', '2025-03-27 12:30:00', 'Break', 2),

('RAIL001', '2025-03-27 13:00:00', NULL, 'OutOfService', 3);

-- **Sample Trip Data**

INSERT INTO trip\_records (operator\_ref, vehicle\_ref, route\_name, start\_time, end\_time, scheduled\_start, on\_time\_flag, distance\_km, trip\_length, idle\_minutes, fuel\_spent, passengers) VALUES

(2, 'BUS001', 'Route A', '2025-03-27 07:55:00', '2025-03-27 08:55:00', '2025-03-27 08:00:00', TRUE, 50.5, 60, 5, 12.5, 40),

(3, 'RAIL001', 'Route B', '2025-03-27 08:10:00', '2025-03-27 09:20:00', '2025-03-27 08:00:00', FALSE, 45.0, 70, 10, 55.0, 65),

(2, 'BUS001', 'Route A', '2025-03-28 07:50:00', '2025-03-28 08:50:00', '2025-03-28 08:00:00', TRUE, 52.0, 60, 3, 11.8, 45);

**-- Sample Energy Usage**

INSERT INTO fuel\_energy\_logs (vehicle\_ref, energy\_type, quantity\_used, km\_covered, recorded\_at) VALUES

('BUS001', 'Diesel', 10.5, 20.0, '2025-03-27 08:30:00'),

('RAIL001', 'Electricity', 50.0, 25.0, '2025-03-27 09:30:00'),

('TRAIN001', 'Diesel', 30.0, 40.0, '2025-03-27 10:30:00');

-- S**ample Components**

INSERT INTO system\_components (vehicle\_ref, component\_type, total\_runtime, usage\_wear\_percent, status\_check)

VALUES

('BUS001', 'Brakes', 500.0, 60.0, 'Warning'),

('RAIL001', 'Pantograph', 300.0, 40.0, 'Normal'),

('TRAIN001', 'Engine', 1000.0, 80.0, 'Critical');

-- **Sample Alerts**

INSERT INTO system\_alerts (alert\_category, vehicle\_ref, alert\_message, alert\_time, resolution\_state) VALUES

('Maintenance', 'BUS001', 'Brakes need inspection', '2025-03-27 09:00:00', 'Pending'),

('Fuel', 'TRAIN001', 'Excessive fuel consumption detected', '2025-03-27 10:00:00', 'Pending');

-- **Sample Maintenance Tasks**

INSERT INTO maintenance\_schedule (component\_ref, vehicle\_ref, task\_description, planned\_date, progress\_status, assigned\_technician)

VALUES

(1, 'BUS001', 'Inspect and replace brakes', '2025-03-28', 'Scheduled', 1),

(3, 'TRAIN001', 'Engine overhaul', '2025-03-29', 'Scheduled', NULL);

-- **Sample Reports**

INSERT INTO analytics\_reports (report\_category, report\_payload, created\_at)

VALUES

('Fuel Usage', 'BUS001: 10.5L', '2025-03-27 11:00:00'),

('Fuel Usage', 'RAIL001: 50kWh', '2025-03-28 17:00:00'),

('Maintenance Cost', 'BUS001: $500', '2025-03-31 12:00:00'),

('Maintenance Cost', 'RAIL001: $2000', '2025-03-31 12:00:00');

-- **Sample Stations**

INSERT INTO route\_stations (id, station\_name, geo\_location, associated\_route)

VALUES ('Station01', 'Test Station', '45.4215,-75,6972', 'Route A');

-- **Customized -- Maintenance Task 1: BUS001 - Brakes (Completed)**

INSERT INTO maintenance\_schedule (component\_ref, vehicle\_ref, task\_description, planned\_date, progress\_status, assigned\_technician)

VALUES

(1, 'BUS001', 'Replace brake pads and perform brake system check', '2025-07-31', 'Completed', 1);

-**- Maintenance Task 2: RAIL001 - Pantograph (In Progress)**

INSERT INTO maintenance\_schedule (component\_ref, vehicle\_ref, task\_description, planned\_date, progress\_status, assigned\_technician)

VALUES

(2, 'RAIL001', 'Clean and inspect pantograph contacts', '2025-08-01', 'InProgress', 1);

-- **Maintenance Task 3: TRAIN001 - Engine (Scheduled)**

INSERT INTO maintenance\_schedule (component\_ref, vehicle\_ref, task\_description, planned\_date, progress\_status, assigned\_technician)

VALUES

(3, 'TRAIN001', 'Run diagnostic and tune engine performance', '2025-08-02', 'Scheduled', 1);

# 9 SECURITY ARCHITECTURE

* SQL Injection protection via prepared statements.
* Login details authentication especially ID and Password via HTML and Servlet. Passwords will be encrypted while entering, for security reasons.
* Application access will be according to the role of Operator (Example Manager or Operator)

# 10 DEPLOYMENT ARCHITECTURE

This section describes how the PTFMS application is hosted and delivered on your local device.

**10.1 Infrastructure**

* Hardware and host environment:
  + Client PC: Windows 11, disk space to run IDE, Apache Tomcat or TomEE, MySQL Workbench
  + Application Server Host: Same physical machine as Client PC (localhost) and runs Apache Tomcat 9.0.86
  + Database Server: Same physical machine (localhost) and runs MySQL Workbench 8.0
* Operating System and Middleware
  + Java JDK 21: Added in the java dependencies
  + Apache Netbeans IDE 25: Manages the Maven projects
  + Apache Tomcat 9.0.86: http connector on port 8080 (default port for HTTP requests)
  + MySQL Workbench 8.0: MySQL server at localhost 3306 which contains the database
* Network and Connectivity:
  + All services are running on localhost
  + Browser to Tomcat (Vice – Versa): <http://localhost:8080/PTFMS/>
  + Tomcat to MySQL (Vice – Versa): jdbc:mysql://localhost:3306/PTFMS

**10.2 Deployment Model**

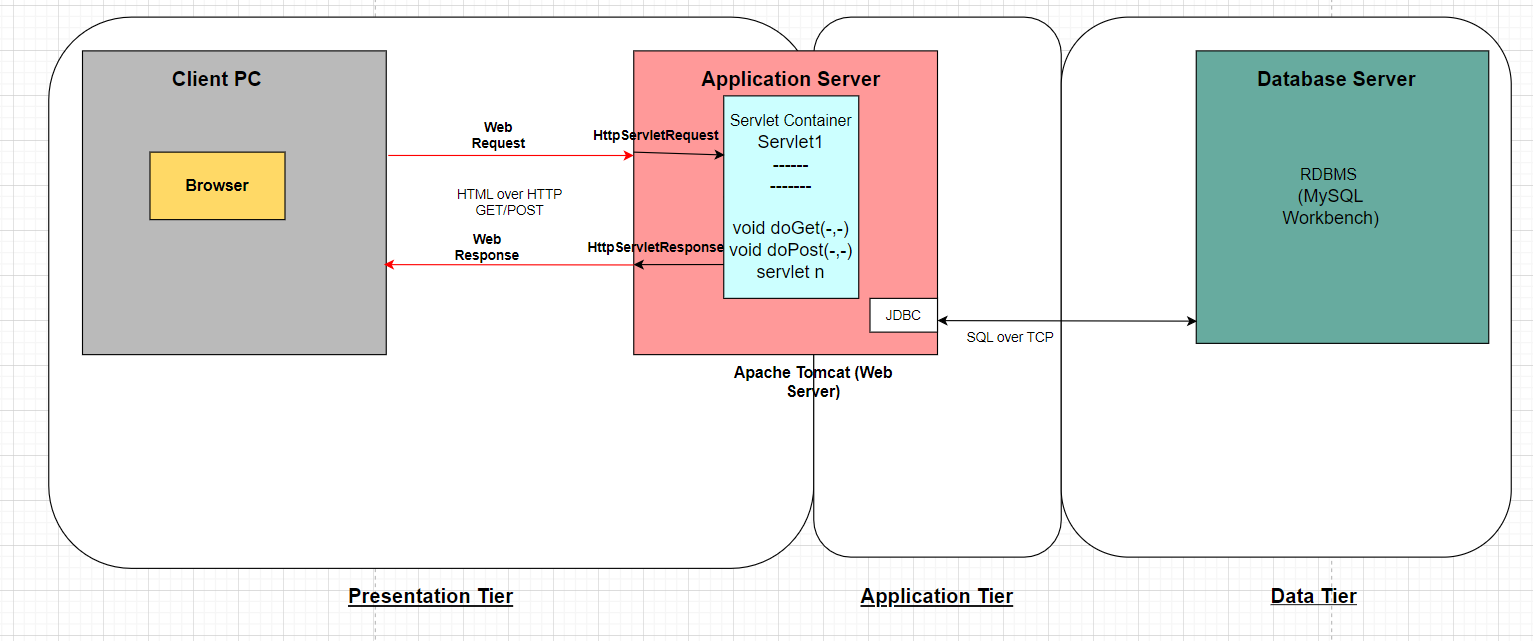


Figure 6 – Deployment Model

**10.3 End-to-End HTTP Request/Response Flow**

Below is the detailed trace of a user login request flow. Each step shows what Tomcat and code are doing behind the scenes.

1. Browser (Presentation Tier):
   1. User submits POST request <http://localhost:8080/PTFMS/login> , where form fields are username, password.
2. Tomcat HTTP Connector:
   1. Listens on port 8080 and receives raw HTTP
   2. Constructs HttpServletRequest and HttpServletResponse objects
3. Tomcat Engine:
   1. Scans the web.xml file and maps request URI ‘/login’ to LoginServlet class
4. Servlet Container:
   1. Calls LoginServlet.init() once on first load
   2. Invokes LoginServlet.service(request, response) and delegates to doPost(request, response)
5. LoginServlet#doPost(request, response):
   1. request.getParameter("username"), .getParameter("password")
   2. Validate non-empty, basic format checks
   3. Invoke UserDAO.authenticate(username, password)
6. DAO Layer (Data Tier)
   1. UserDAO:  
       Connection conn = DriverManager.getConnection(  
       "jdbc:mysql://localhost:3306/PTFMS", dbUser, dbPass);  
       PreparedStatement ps = conn.prepareStatement(  
       "SELECT user\_id, role FROM users WHERE email=? AND password\_hash=?"  
       );  
       ps.setString(1, email);  
       ps.setString(2, hash(password));  
       ResultSet rs = ps.executeQuery();
7. Back in LoginServlet:
   1. If User found:  
       HttpSession session = request.getSession(true);  
       session.setAttribute("currentUser", User);  
       if (User.isManager())  
       response.sendRedirect("managerDashboard.jsp");  
       else  
       response.sendRedirect("operatorDashboard.jsp");
   2. Else:  
       request.setAttribute("errorMsg", "Invalid credentials");  
       request.getRequestDispatcher("login.html")  
       .forward(request, response);
8. View Response:
   1. If redirect: Tomcat sends HTTP 302 with Location header
   2. If forward: JSP / HTML rendered, HttpServletResponse.getWriter().write()
9. Browser Receives Response:
   1. Browser recieves either the error message or the Dashboard page.

# 11 TESTING MODEL

Junit Tests will be used for our Model.

Manual Testing will be performed directly on the User Interface to check for errors.

**11.1 JUNIT Test Case List**

FR-01: User Registration & Authentication.

* UserRegistration with all Valid credentials=> Regd\_ValidCred(),
* UserRegistration with all InValid credentials => Regd\_InValidCred()

FR-02: Vehicle Management

* Add Vehicle with unique ID=> addNewVeh(),
* Get Vehicle by id when it already exists => getVehById()

FR-03: GPS Tracking

* Vehicle gps logs must be tested = > Gps\_ValidData(),
* Track vehicle gps report =>Track\_Vehicle \_GPSReport()

FR-04: Fuel Consumption

* Calculate Fuel Consumption Rate => FuelConsumption\_rate(),
* Fuel alert trigger should be tested > FuelAlert\_Trigger()

FR-05: Maintenance

* Maintenance Entry will be tested => Maintenance\_Entry(),
* Maintenance alert trigger hours exceed the limit= Maintenance\_AlertTrigger()

**11.2 Manual Test Case List**

FR-01: User Registration & Authentication.

* User Registration with all Valid credentials:
* Input: Valid (NAME, EMAIL, Password, User\_Type)
* Action: Fill Registration Form and submit
* Expected: Registration Success with a successful message.
* GUI: Registration form submission followed by success message.
* User Registration with all Invalid credentials:
* Input: Valid (NAME, EMAIL) but wrong Password.
* Expected: Receive a login failure message.
* GUI: Error message is shown in a dialog box and no redirect option.

FR-02: Vehicle Management

* Add Vehicle with unique ID:
* Input: Valid (VEHICLE\_ID, VEHICLE\_NUMBER, ROUTE\_ID)
* Expected: Registration Success with a successful message.
* GUI: Registration form loads successfully, and data is saved.
* Add Vehicle when it already exists:
* Input: Valid VEHICLE\_NUMBER that already exists in the system
* Expected: Must see an error that Vehicle already exists on the system.
* GUI: A dialog box showing error details and saying “Please fill a different VEHICLE\_NUMBER.

FR-03: GPS Tracking

* Vehicle gps logs must be tested:
* Input: Valid details of current Station are logged in manually by the driver
* Expected: Get a Confirmation of entering the details
* GUI: Autogenerated details of Station List and timestamp appear.
* Track vehicle gps report:
* Input: Valid VEHICLE\_ID
* Expected: All route logs along with timestamps are visible.
* GUI: Must show a log table with the stations covered at their respective time in and time out.

FR-04: Fuel Consumption

* Calculate Fuel Consumption Rate
* Input: Put a Valid VEHICLE\_ID and click Generate Fuel Consumption Report.
* Expected: A table with the daily average consumption of fuel of the concerned vehicle is visible.
* GUI: A detailed Report of average daily consumption is presented.
* Fuel alert trigger should be tested:
* Input: Try to fake the fuel rate as above the threshold.
* Expected: A fuel alert message must be triggered.
* GUI: An alert message is displayed in red saying "Fuel out of range”.

FR-05: Maintenance

* Maintenance Schedule will be tested:
* Input: Valid VEHICLE\_ID, Component, Date, Remarks.
* Expected: Confirmation of Maintenance status to be changed to “Scheduled”.
* GUI: Can see the Maintenance status as ‘Scheduled”.
* Maintenance alert trigger hours exceed the limit:
* Input: Vehicle exceeds the hours\_used (wear) threshold.
* Expected: A Maintenance alert message must be triggered.
* GUI: An alert dialog box appears in red displaying “Maintenance Required Asap”.

# 12. REFERENCE LIST

# 13. ACRONYMS & ABBREVIATIONS

|  |  |  |
| --- | --- | --- |
| **Acronyms / Abbreviations** | **Whole Name** | **Description** |
| HLD | High-Level Design | A document that outlines the system architecture and component interactions. |
| PTFMS | Public Transit Fleet Management System | The objective application developed to manage transit vehicles and operations. |
| UI | User Interface | The visual part of an application that users interact with. |
| IDE | Integrated Development Environment | A software application used for programming. |
| HTML | HyperText Markup Language | Standard language for creating web pages. |
| JSP | Java Server Pages | Technology that is used to create dynamic web content using Java. |
| DAO | Data Access Object | Design pattern for abstracting and encapsulating database access. |
| DTO | Data Transfer Object | An object that carries data between processes. |
| GPS | Global Positioning System | Technology for tracking geographical location. |
| CRUD | Create, Read, Update, Delete | Basic operations for interacting with a database. |
| SQL | Structured Query Language | Language used to manage and query relational databases. |
| JDBC | Java Database Connectivity | API for connecting Java applications to databases. |
| MVC | Model-View-Controller | Design pattern that separates application logic into three interconnected components. |
| JDK | Java Development Kit | Software development kit required for developing Java applications. |
| HTTP | HyperText Transfer Protocol | Protocol used for transmitting web pages over the internet. |
| URL | Uniform Resource Locator | The address that is used to access resources on the web. |
| PK | Primary Key | A unique identifier for a record in a database table. |
| FK | Foreign Key | A field in one table that refers to the primary key in another table. |
| DET | Diesel-Electric Train | A hybrid vehicle using diesel and electric power. |
| XML | eXtensible Markup Language | A markup language used for storing and transporting data. |
| ELR | Electric Light Rail | Type of electric-powered transit vehicle. |
| DB | Diesel Bus | Diesel-powered public transportation vehicles. |

# 14 LIST OF FIGURES

[Figure 1 – Use Case Diagram 6](#_Toc203763490)

[Figure 2 – Detailed Design 7](#_Toc203763491)

[Figure 3 – Component Diagram 9](#_Toc203763492)

[Figure 4 – Logical Model 10](#_Toc203763493)

[Figure 5 – Physical Model 10](#_Toc203763494)

[Figure 6 – Deployment Model 17](#_Toc203763495)