**Automotive System Management**

**1. Project Overview**

This project is a console-based C++14 application designed to manage and monitor vehicle-related data. It includes functionality for vehicle data management, diagnostics through DTC codes, real-time monitoring via simulated OBD-II, alert generation, and basic data analysis. SQLite is used for persistent data storage.

**2. System Features**

* Manage vehicle records (CRUD operations)
* Manage Diagnostic Trouble Codes (DTCs)
* Simulate real-time engine data (e.g., RPM, speed, temperature)
* Generate alerts when thresholds are exceeded
* Analyze performance data and usage patterns
* Console-based user interface for interaction

**3. Technical Specifications**

* **DatabaseManager**: Handles SQLite database connection and queries.
* **Vehicle**: Stores vehicle details like Make, Model, Year, VIN.
* **DTC**: Stores diagnostic trouble codes and severity levels.
* **RealTimeData**: Tracks real-time data like engine speed and temperature.
* **Alert**: Stores alert types triggered by diagnostic or data thresholds.

**4. User Guide**

To use the system:

* Build the project using CMake and make
* Launch the app from the terminal
* Select options from the console menu
* Insert, update, or analyze data as needed
* Monitor alerts and diagnostics
* Exit safely from the menu

**5. Build & Test Instructions**

**Requirements**:

* C++14 compiler
* SQLite3
* CMake

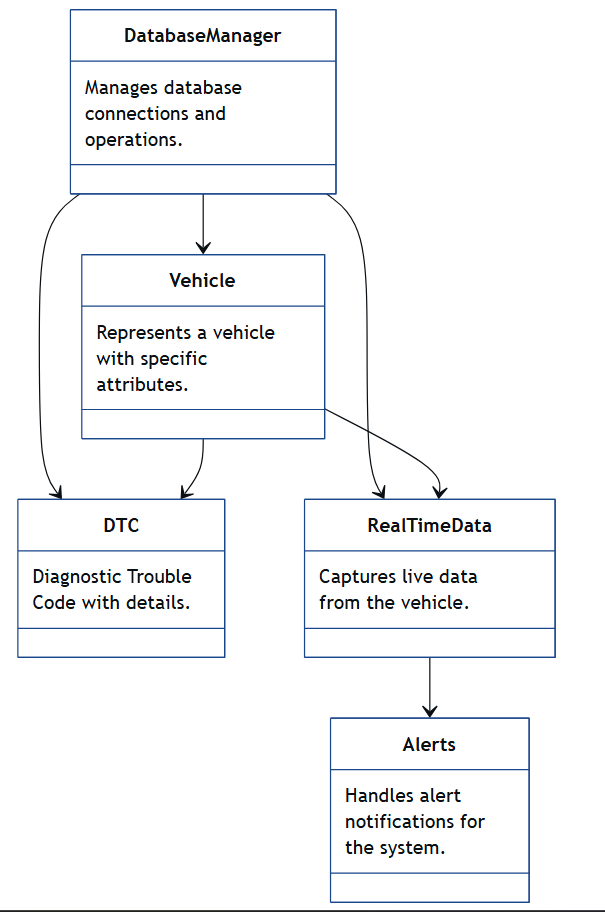
**Steps**:

1. Install dependencies
2. Run cmake . in the root directory
3. Run make to build the project
4. Run the output binary
5. For testing, compile and run the test\_cases.cpp

**6. Test Cases**

| **Test Case Name** | **Description** | **Input** | **Expected Output** |
| --- | --- | --- | --- |
| testDBConnection | Validate DB connection | DB init | Successful connection |
| testVehicleTable | Check Vehicle table creation | SQLite schema query | Table exists |
| testInsertVehicle | Insert vehicle record | Make, Model, VIN | Record inserted |
| testDTCInsert | Insert a DTC | DTCCode, VehicleID | Record inserted |
| testOBDSimulation | Simulate OBD data | Random engine data | RealTimeData entries |
| testAlerts | Trigger alert based on thresholds | RPM > 5000 | Alert created |

**7. UML Diagram**

****

**8. Notes**

* All real-time data is simulated
* Alert thresholds can be configured
* VINs must be unique
* Data is stored in SQLite for portability