**EXECUTABLE PROJECT INFORMATION**

**Project Title:** Car Rental System

**Document Owner:** Sujana D

**Date:** 26-06-2025

**1. PROJECT OVERVIEW**

This is a console-based Car Rental System written in Python 3.13 (or later), developed in VS Code, and connected to a MySQL database via mysql-connector-python. It supports CRUD operations for customers, vehicles, leases, and payments, enforces business rules (e.g., only available cars can be leased), and includes custom exception handling and Excel-based test reporting.

**2. PROJECT STRUCTURE**

CarRentalSystem-CASESTUDY/

│ db.properties # Database connection settings

│ SQL\_SCRIPT.sql # DDL & sample DML

│ generate\_test\_report.py # Excel unit-test report script

│ README.md # Project overview & run instructions

│

├── app/ # Application entry point

│ └── main.py # Menu-driven CLI

│

├── entity/ # Domain model classes

│ ├── customer.py

│ ├── vehicle.py

│ ├── lease.py

│ └── payment.py

│

├── dao/ # Data access interface & implementation

│ ├── icar\_lease\_repository.py

│ └── icar\_lease\_repository\_impl.py

│

├── exception/ # Custom exception classes

│ └── custom\_exceptions.py

│

├── util/ # Utility modules

│ ├── db\_property\_util.py

│ └── db\_conn\_util.py

│

└── tests/ # Unit tests (PyUnit)

├── test\_customer.py

├── test\_vehicle.py

├── test\_lease.py

└── test\_payment.py

**3. TECHNOLOGY STACK**

* **Language & Runtime:** Python 3.13+
* **Database:** MySQL (via MySQL Workbench)
* **Connector:** mysql-connector-python
* **Excel Reporting:** openpyxl
* **Testing:** Python unittest (PyUnit)
* **IDE:** Visual Studio Code

**4. DATABASE CONFIGURATION**

File: **db.properties**

host=localhost

user=root

password=your\_password

database=CarRentalDB

Ensure you replace your\_password with your MySQL root password.

**5. KEY COMPONENTS**

**5.1 DAO (Data Access Layer)**

* ICarLeaseRepository.py – Abstract interface defining methods for customer, vehicle, lease, and payment operations.
* ICarLeaseRepositoryImpl.py – Implements the interface, executes parameterized SQL, commits transactions, closes resources, and raises custom exceptions on “not found” conditions.

**5.2 Entity (Model Classes)**

* customer.py, vehicle.py, lease.py, payment.py – Plain data classes with attributes and \_\_str\_\_ methods; no business logic.

**5.3 Exception**

* custom\_exceptions.py – Defines CustomerNotFoundException, VehicleNotFoundException, and LeaseNotFoundException.

**5.4 Util**

* db\_property\_util.py – Reads db.properties for connection parameters.
* db\_conn\_util.py – Provides get\_connection() to open a MySQL connection.

**5.5 Main**

* main.py – Menu-driven CLI that prompts the clerk, invokes DAO methods, handles exceptions, and displays results.

**6. UNIT TESTING**

**File: tests/**

* **Test Files:**
  + **test\_customer.py –** Tests add, find, list, remove (including exception).
  + **test\_vehicle.py –** Tests add, list, remove, find (including exception).
  + **test\_lease.py –** Tests create lease, list active/history, return (including exception).
  + **test\_payment.py –** Tests record payment (placeholder asserts) and can be extended to verify history.
* **Execution:**

python -m unittest discover -s tests

* **Reporting:**

python generate\_test\_report.py

**7. AVAILABLE FEATURES (VIA CONSOLE)**

|  |  |
| --- | --- |
| **OPTION** | **DESCRIPTION** |
| 1 | Add a new customer |
| 2 | Remove an existing customer (if no leases exist) |
| 3 | Add a new vehicle |
| 4 | View available vehicles |
| 5 | Create a lease (daily or monthly) |
| 6 | Return a car (by lease ID) |
| 7 | Record a payment for a lease |
| 8 | View active leases |
| 9 | View lease history (completed leases) |
| 10 | View payment history & calculate total revenue |
| 11 | Generate Excel unit-test report |
| 12 | Exit the application |

**8. DATABASE SCHEMA**

Ensure MySQL schema includes these tables:

CREATE TABLE Vehicle (

vehicleID INT AUTO\_INCREMENT PRIMARY KEY,

make VARCHAR(50),

model VARCHAR(50),

year INT,

dailyRate DECIMAL(10,2),

status VARCHAR(20),

passengerCapacity INT,

engineCapacity DECIMAL(5,2)

);

CREATE TABLE Customer (

customerID INT AUTO\_INCREMENT PRIMARY KEY,

firstName VARCHAR(50),

lastName VARCHAR(50),

email VARCHAR(100),

phoneNumber VARCHAR(15)

);

CREATE TABLE Lease (

leaseID INT AUTO\_INCREMENT PRIMARY KEY,

vehicleID INT,

customerID INT,

startDate DATE,

endDate DATE,

type VARCHAR(20),

FOREIGN KEY (vehicleID) REFERENCES Vehicle(vehicleID),

FOREIGN KEY (customerID) REFERENCES Customer(customerID)

);

CREATE TABLE Payment (

paymentID INT AUTO\_INCREMENT PRIMARY KEY,

leaseID INT,

paymentDate DATE,

amount DECIMAL(10,2),

FOREIGN KEY (leaseID) REFERENCES Lease(leaseID)

);

**9. USAGE TIPS**

* Start MySQL before running the app.
* Populate the database by running SQL\_SCRIPT.sql in MySQL Workbench.
* Configure db.properties with correct credentials.
* Run the CLI:

python app/main.py

* Run all tests and generate the report:

python -m unittest discover -s test

python generate\_test\_report.py