**DEMO AND TESTCASE DOCUMENT**

1. **PROJECT DEMONSTRATION OVERVIEW**

This document walks through the Car Rental System’s console‐based operation and summarizes the key test cases used to verify its core functionality. The application is implemented in Python 3.x, uses MySQL for persistence, follows the DAO pattern and OOP principles, and is covered by automated unittest tests.

1. **DEMO INSTRUCTIONS (CONSOLE USAGE)**

|  |  |  |
| --- | --- | --- |
| **Step** | **Action** | **Description** |
| 1 | Run main.py | Launches the menu-driven CLI in the console. |
| 2 | Select menu option | Choose one of: Customer, Vehicle, Lease, Payment, or Exit. |
| 3 | Input required details | Enter IDs, names, dates, amounts, etc., as prompted by the selected feature. |
| 4 | View output | The console confirms operations (e.g., “Customer added successfully!”) or displays query results. |
| 5 | Exit | Select the “Exit” option to terminate the application gracefully. |

1. **FUNCTIONAL REQUIREMENTS COVERED**

|  |  |
| --- | --- |
| **Feature ID** | **Description** |
| FR1 | Add new customer |
| FR2 | Update existing customer information |
| FR3 | Remove customer (if no leases) |
| FR4 | List all customers / Find customer by ID |
| FR5 | Add new vehicle |
| FR6 | Update vehicle availability |
| FR7 | List available vehicles / List rented vehicles / Find vehicle by ID |
| FR8 | Create daily or monthly lease and calculate cost |
| FR9 | Return car (end lease & mark available) |
| FR10 | List active leases / View lease history |
| FR11 | Record payment for a lease |
| FR12 | View payment history for a customer |
| FR13 | Calculate total revenue from all payments |

1. **TEST CASE SUMMARY**

All tests are implemented in tests/ and executed via python -m unittest. Below is a summary of representative test cases:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | **Description** | **Input** | **Expected Output** | **Result** |
| TC\_01 | Add Customer | Valid Customer object | Customer added to DB | Pass |
| TC\_02 | Remove Nonexistent Customer | customerID = 9999 | Raise CustomerNotFoundException | Pass |
| TC\_03 | Add Vehicle | Valid Vehicle object | Vehicle added to DB | Pass |
| TC\_04 | Find Nonexistent Vehicle | vehicleID = 9999 | Raise VehicleNotFoundException | Pass |
| TC\_05 | Create Lease | customerID=1, vehicleID=2, dates | Lease created; vehicle marked notAvailable | Pass |
| TC\_06 | Return Invalid Lease | leaseID = 9999 | Raise LeaseNotFoundException | Pass |
| TC\_07 | Record Payment | leaseID=1, amount=500.00 | Payment recorded in DB | Pass |

All tests passed successfully.

1. **TESTING TOOLS USED**

* Python unittest (PyUnit) for automated test execution.
* openpyxl for generating the Excel test-report.
* MySQL Workbench for schema setup and manual verification.
* Manual console testing via the menu-driven CLI for real-time demo feedback.