

*OOP project*

## ***Vehicle Rental Management System***

### **Group Members:**

1. Kalkidan Birhabu UGR/1053/17
2. Kenean Engda UGR/3226/17
3. Lidya Demerw UGR/6153/17
4. Maedot Eskender UGR/9011/17
5. Nardos Nega UGR/8725/17

URL : <https://github.com/lidyademerw/Vehicle-Rental-Management-System>

Submission Date: January 30, 2026.

# Vehicle Rental Management System

- **Problem Statement**

Vehicle rental businesses often rely on manual record-keeping, which leads to errors such as double-booking, inconsistent pricing, and loss of rental history. This project, the Vehicle Rental Management System, provides a robust digital solution. It automates fleet tracking and cost calculation while ensuring data security through defined user roles and persistent storage.

- **Explanation of OOP Concepts Used**

**Encapsulation:** All class attributes (like plateNumber and password) are set to private. Access is controlled via public getters and setters to protect data integrity.

**Abstraction:** The Vehicle and User classes are declared as abstract. This ensures that common code is shared, but no "generic" vehicle or user can be created without being specific (e.g., a Car or a Customer).

**Inheritance:** We used the extends keyword so that Car inherits features from Vehicle. This reduces code duplication.

**Polymorphism:** The calculateTotalCost() method is defined in the parent class but behaves differently depending on the specific vehicle type.

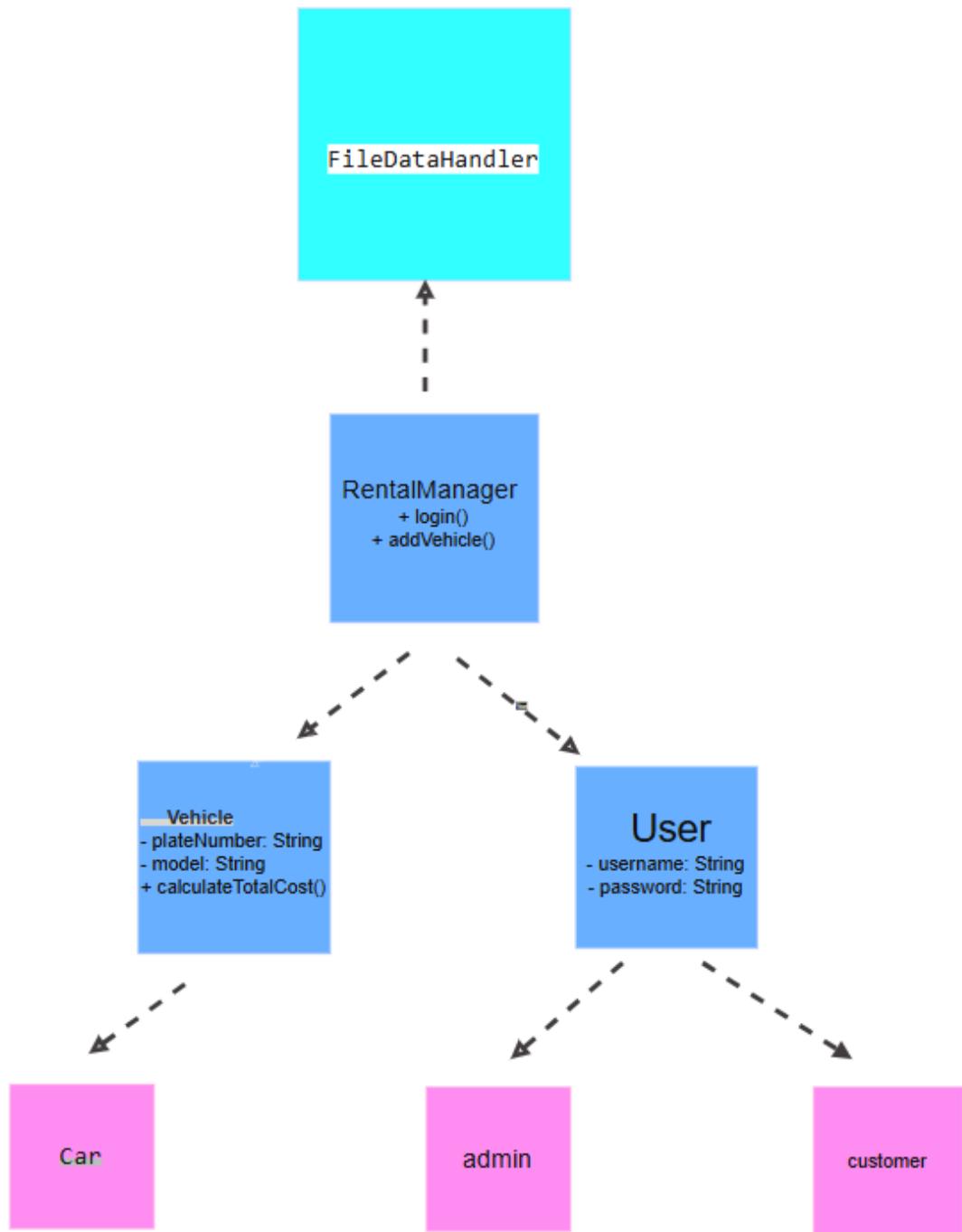
- **SOLID Principles Applied**

**Single Responsibility (SRP):** The FileDataHandler class is solely responsible for file operations, while the RentalManager handles business logic.

**Liskov Substitution (LSP):** Any method that expects a Vehicle can accept a Car without causing errors.

**Dependency Inversion (DIP):** High-level controllers depend on the RentalManager abstraction rather than directly communicating with the raw data file.

- **SystemFeatures**



Multi-Role Access: Separate dashboards for Admins (Inventory Control) and Customers (Booking).

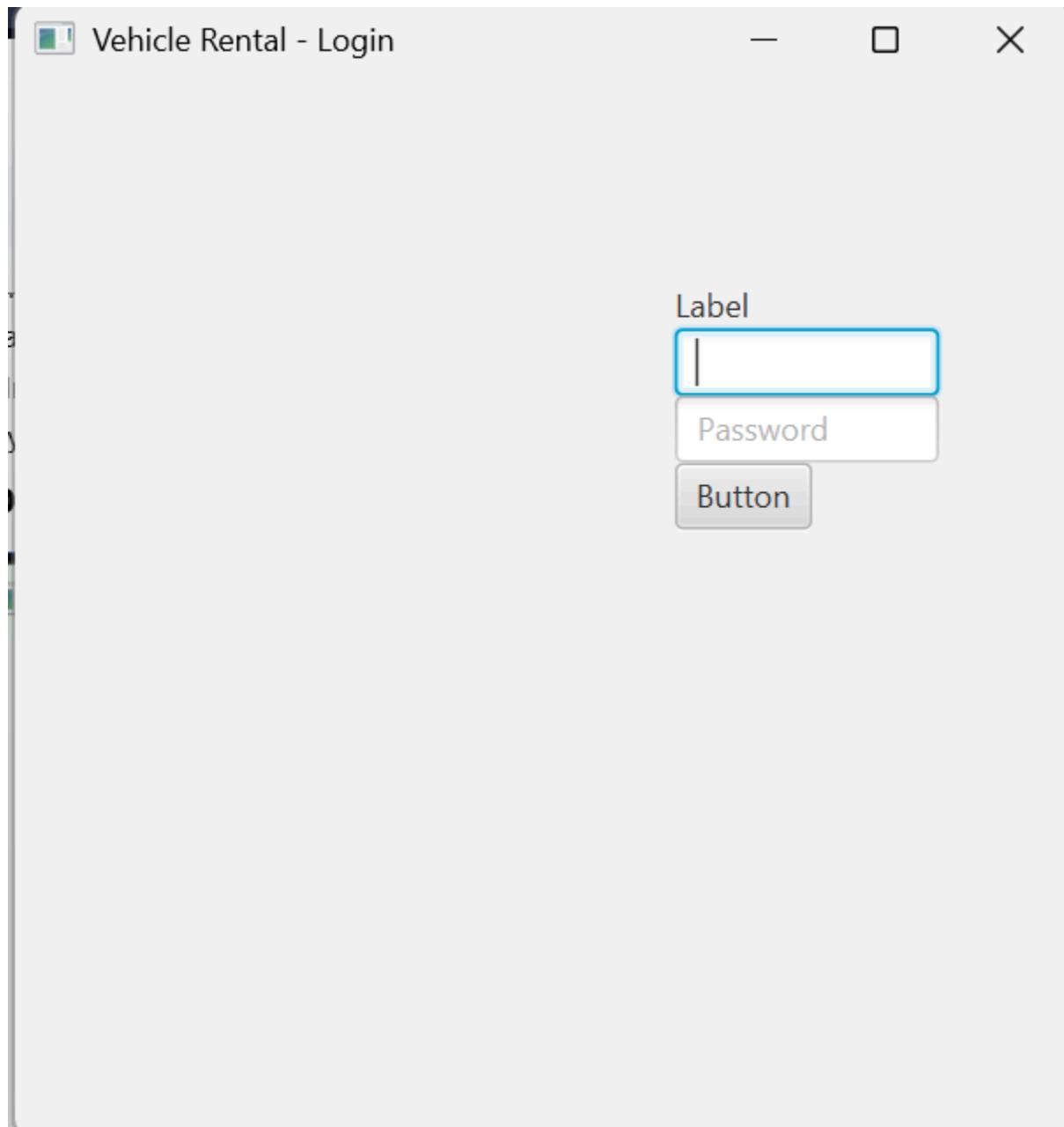
Fleet Management (CRUD): Admins can Create, Read, and Delete vehicle records in real-time.

Booking System: Customers can view only available vehicles and rent them with a single click.

Data Persistence: All data is saved to and loaded from a local vehicles.txt file using Java

I/O. Input Validation: The system prevents errors by validating user inputs during login and data entry.

## Application Screenshots





## Admin Dashboard - Fleet Management

Plate Number	Model	Daily Price
ABC-111	Toyota Corolla	50.0
XYZ-222	Honda Civic	60.0
LUX-333	Mercedes C-Class	120.0
MND-234	Ferrari	400.0
ZMG-345	bajaj	300.0
ASZ-008	Ford Mustang	600000.0
SMC-990	Tesla Model 3	56734.0

Add CarDelete Selected CarLogout

