

## **Lab Assignment: Case Study on Class Modeling**

Useful References -

1. **What is a class diagram? - [Class Diagrams](#)**
2. **[OO-Relationships](#)**

Tools - [StarUML](#) , [draw.io](#), [SmartDraw](#)

Prepare a detailed class diagram including all classes, their methods and attributes (with visibility notation), relevant relationships (dependencies, generalizations, associations) with multiplicities, enumeration, etc. as described by the case study given below.

### **Smart Ride-Sharing and Cab Booking System**

A metropolitan city plans to implement a smart ride-sharing and cab booking system to improve urban mobility, reduce congestion, and provide seamless transportation services. The system is governed by the City Transport Authority, which sets operational policies, oversees the fleet of vehicles, monitors station performance, and generates analytics reports. Administrators operate under this authority to manage users, monitor bookings, and ensure operational efficiency.

The system serves multiple user roles: Commuters, who request rides; Drivers, who operate vehicles; and Admins, who supervise the overall system. Each user maintains a digital identity with personal details, role-based access, and authentication credentials. Users can update their profiles, view ride histories, and access functionalities relevant to their responsibilities. All users are part of a notification system, ensuring real-time alerts for ride confirmations, cancellations, or payment updates.

The city fleet consists of various vehicles, each with a type (e.g., sedan, SUV, electric), status (available, on trip, under maintenance, offline), and optional amenities like WiFi, AC, or child seats. Vehicles are operated by drivers and tracked in real time. Maintenance records are kept for every vehicle, documenting reported issues, repair status, and maintenance history. Drivers can report issues for vehicles they operate, linking operational feedback directly to maintenance records.

Commuters create bookings by specifying pickup and drop-off locations. Each booking is associated with a specific vehicle, a calculated fare, and a route capturing waypoints, distance, and estimated time. Upon ride completion, feedback is collected from commuters and linked to both the booking and the driver. Payments are processed digitally, recorded per booking, and include status indicators (success, failed, pending).

The system ensures smooth coordination by managing relationships among users, vehicles, bookings, maintenance, routes, amenities, feedback, and payments. Administrators monitor usage, reallocate vehicles if needed, and generate reports. The city transport authority can oversee the entire fleet, track maintenance, and ensure optimal ride-sharing efficiency across the metropolitan area.