As the main purpose of this task is to evaluate the object-oriented design principles to implement the requested model, I did not consider the required communication between the client-server aspect.  
Therefore I’ve created a simple application that can add vehicles to inventory, start auctions, have clients bidding in live auction, stop auctions and search for vehicles in inventory.

**Vehicle Class Hierarchy:**

- `Vehicle` is the base class for all types of vehicles (Sedan, SUV, Hatchback and Truck).

- Specific types like `Sedan`, `SUV`, `Hatchback`, and `Truck` inherit from `Vehicle` and add their own unique properties (e.g., `Doors`, `Seats`, `LoadCapacity`).

**Auction Class:**

- `Auction` handles a vehicle’s auction, keeping track of the starting bid, current bid, and whether the auction is active. It also allows the clients to place bids and manager to close the auction.

**AuctionManager Class**

- Manages the auction process, including adding vehicles, starting auctions and closing auctions.

- Uses a dictionary to track active auctions by vehicle ID.

- Uses a list of objects (Vehicle) to track valid inventory.

**BidderClient Class**

- Allows to bid in active auctions by auction ID.

- Allows to search for vehicles in inventory by their characteristics.

**Error Handling**

**Duplicate Vehicle ID:** Ensured by checking if a vehicle with the same ID already exists in the inventory.

**Auction Start & Active Auction Checks:** Verified that no other auction is active for a vehicle when starting an auction and that the vehicle exists in the inventory.

**Bid Validation:** Ensured that the bid amount is higher than the current bid and that the auction is active.

**Assumptions:**

The `Id` field for vehicles is a int that must be unique across all vehicles.

Auctions for vehicles are one-time only. Once the auction is closed, it cannot be reopened, and vehicle is removed from inventory.

Focused only on basic functionality for vehicle management and auction bidding.

Simple program without any external storage like databases, the system is meant for in-memory management.