## UE20CS352:Object Oriented Analysis with Java Mini Project

**Project Title: Car Rental Management System**

# Team Details:

**SRN :** PES1UG20CS119 **Name:** Debanjan Das

**SRN :** PES2UG20CS016 **Name:** Adarsh Kumar

**SRN :** PES1UG20CS109 **Name:** Chetan Reddy Bandi

**SRN :** PES1UG20CS112 **Name:** Chetan Gurram

Brief Synopsis:

Car rental management software is used to help car rental management businesses manage their operations. The software helps companies to keep track of their cars and customers. Customers can use the software to rent cars, view rental rates, and select the car they want to rent. The software also stores customer information, like their contact details and rental history. The car rental system typically includes a database of cars that are available for rental, along with their rental rates and availability. Customers can browse the available cars, select the dates and times they need the car, and make a reservation. The system also lets users make payments after returning the rental vehicle . Users can also see and pay the additional charges incurred during their travel. The software also has a customer interaction feature which helps the customer interact with support staff. Customers can also leave their feedback about the cars, service, etc. after their rental.

1. Project Description:

The transport sector faces numerous challenges in urban areas such as:

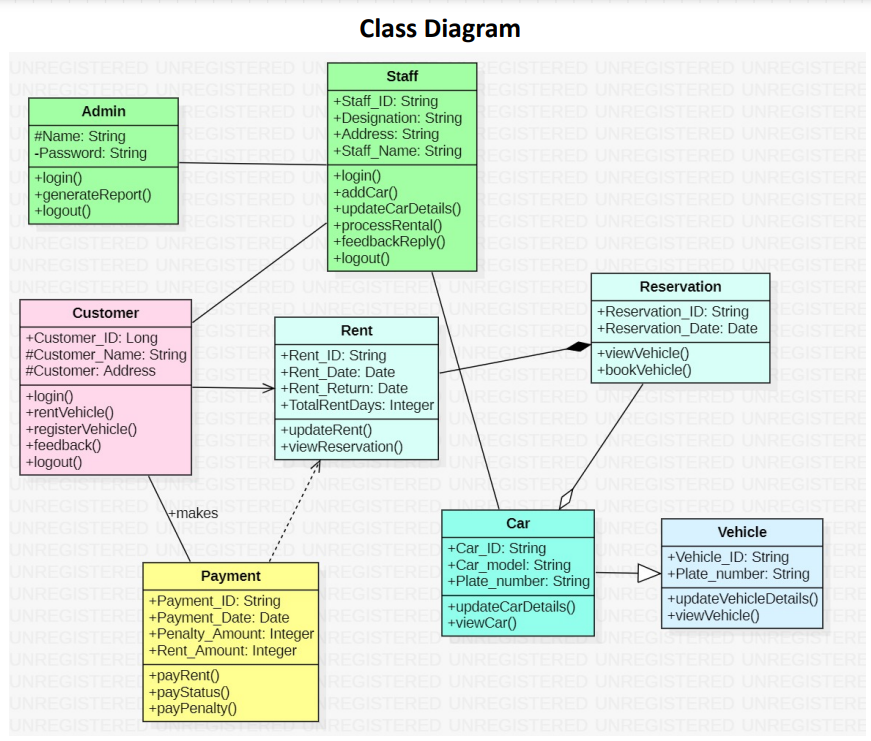
* Traffic Congestion and Parking Difficulties: The lack of adequacy and connectivity offered by public transport has led to extreme growth in the number of automobile owners. However, limited infrastructure has not been able to keep up with the same.
* Environmental Impacts: Air and noise pollution are the products of increasing consumption of traditional, unsustainable fuel driven by urban mobility systems. These impede the quality of life and health of the local population.
* Energy Prices: High demand for energy resources has led to an exorbitant costs of transportation. This is not feasible on a daily basis, in the longer run.
* Liability: Vehicles often require high maintenance and expenditure in terms of repair, insurance, fuel costs and loans.

Our project aims to address the above concerns by proposing a car rental solution. It will provide a platform for people to rent cars for short periods of time.

FUNCTIONALITIES

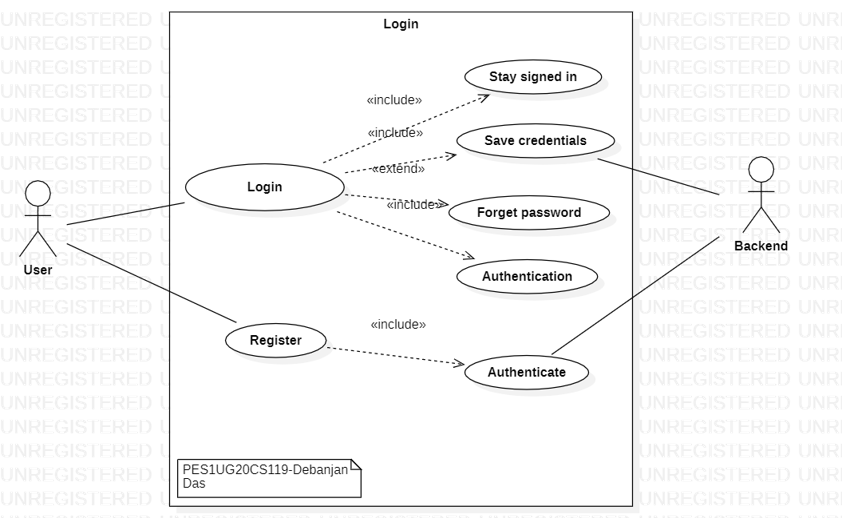
* Car Rental Interface: Clients can visit the website and choose a car of their choice and booking can be done as per their requirements. Feedback can also be provided on avail of the car rental service.
* Payment Portal: Order placing and cancellation are maintained by the admin. Amount is generated based on the type of car and duration of rental.
* Staff: Data about the repair/replacement of any parts of the car are maintained by the staff post inspection on pick up.

1. USE CASE AND CLASS DIAGRAMS:

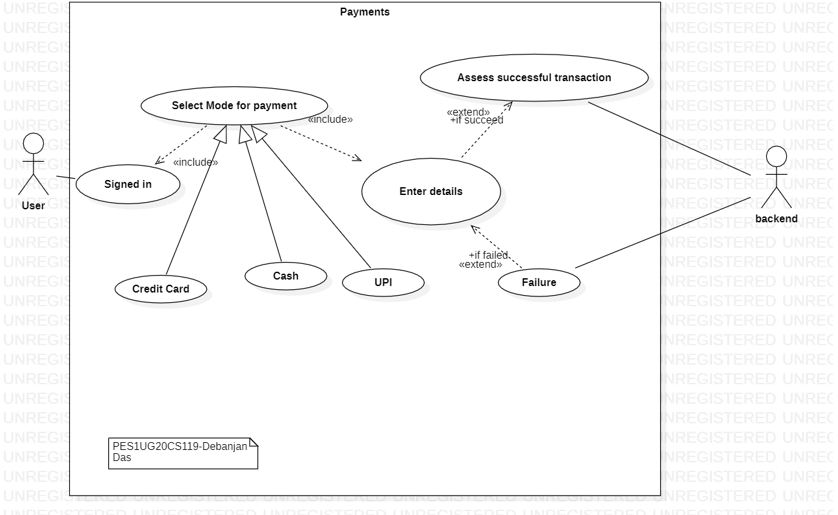
CLASS DIAGRAM: 

USECASE DIAGRAMS:

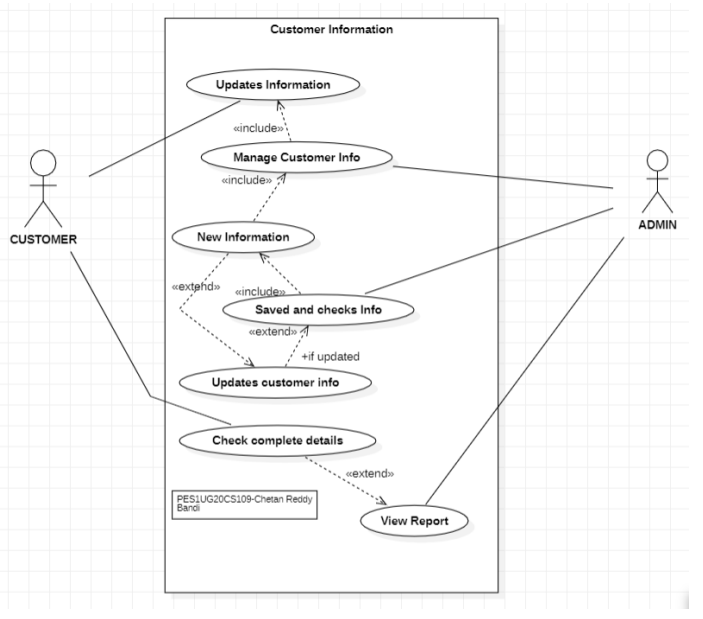
1: Login



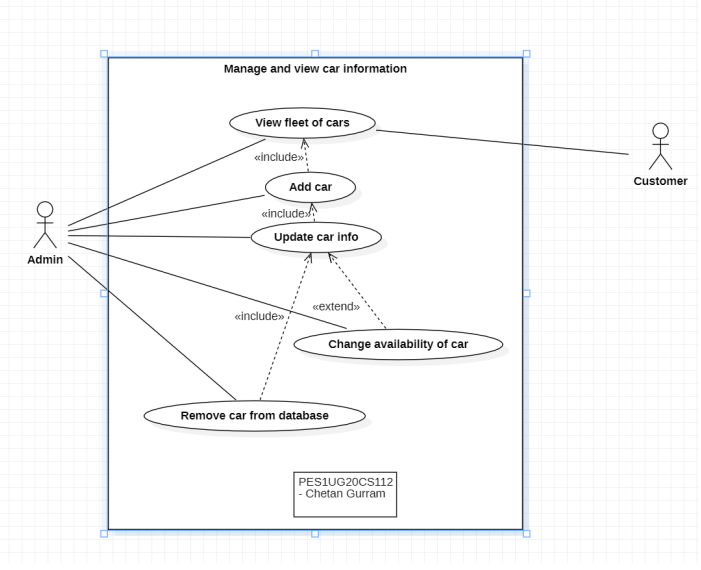
2:Payments



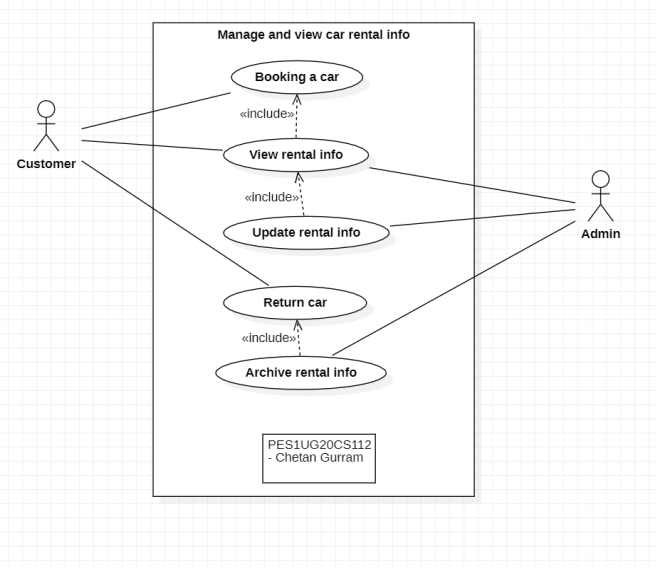
3. Customer Information



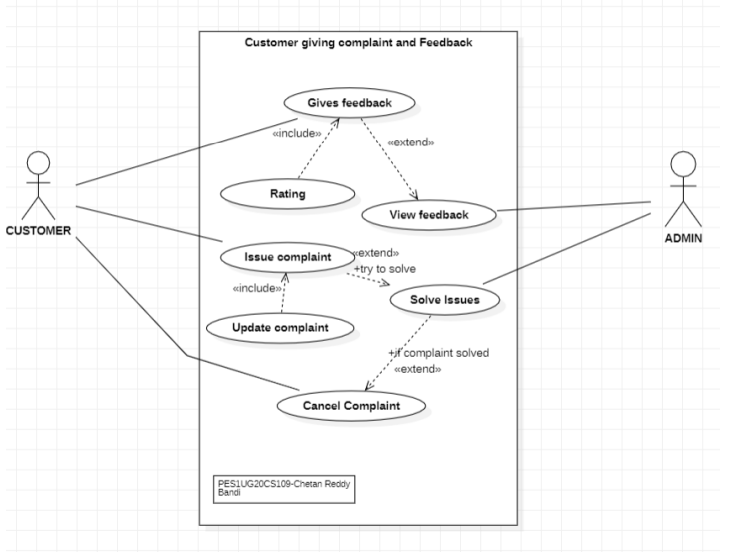
4. Manage and view car information



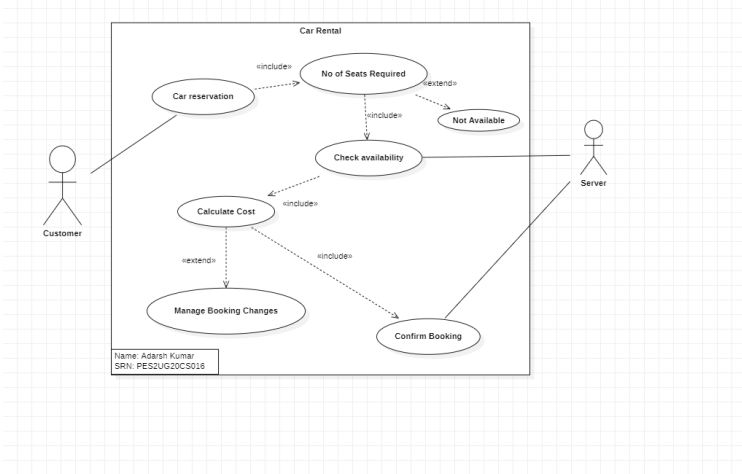
5. Manage and view car rental information



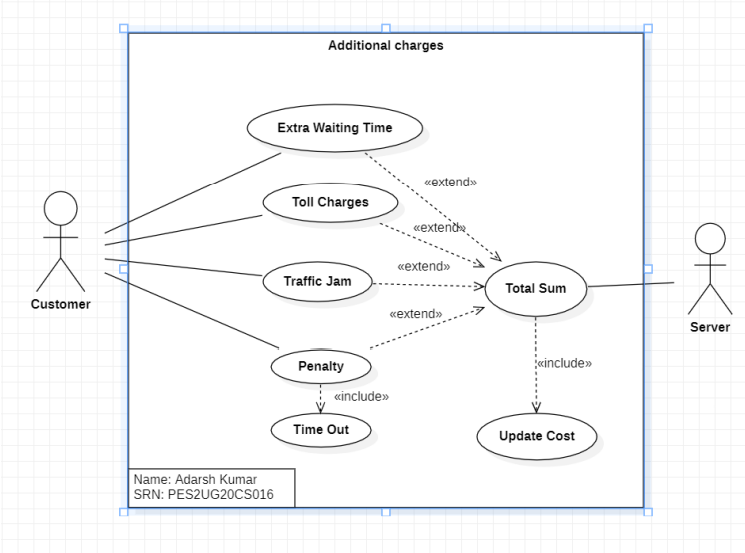
6.Customer Raising Complaint and giving Feedback



7. Car Rental



8. Additional charges



1. Design Principles and Design Patterns Applied

1.Factory Design Pattern - Applied to instantiate different models of rental cars. The Vehicle super-class, which is extended by all Car models sub-classes (Audi, Mercedes, Creta, Innova). This allows us to utilize the common methods of the Vehicle super-class.

2. Decorator Pattern - Add discounts to booking of rental based on run-time criteria like booking time, additional offers for new users and festive discounts.

3. Command Pattern - For user inputs, like submit and cancel buttons.

SOLID Design Principles were followed in the development of different parts of this project. - Single Responsibility Principle: Every class in the design has a single responsibility. For example, the Payment class is responsible only for getting and setting the data of the Payment Object (details like payment id, amount, and payment status). - Using factory design patterns for creating different models of cars, the open-close principle was followed. The Vehicle super-class, was extended by all Car models sub-classes (Audi, Mercedes, Creta, Innova). New car models can extend from the Car super-class without modifying Vehicle class.

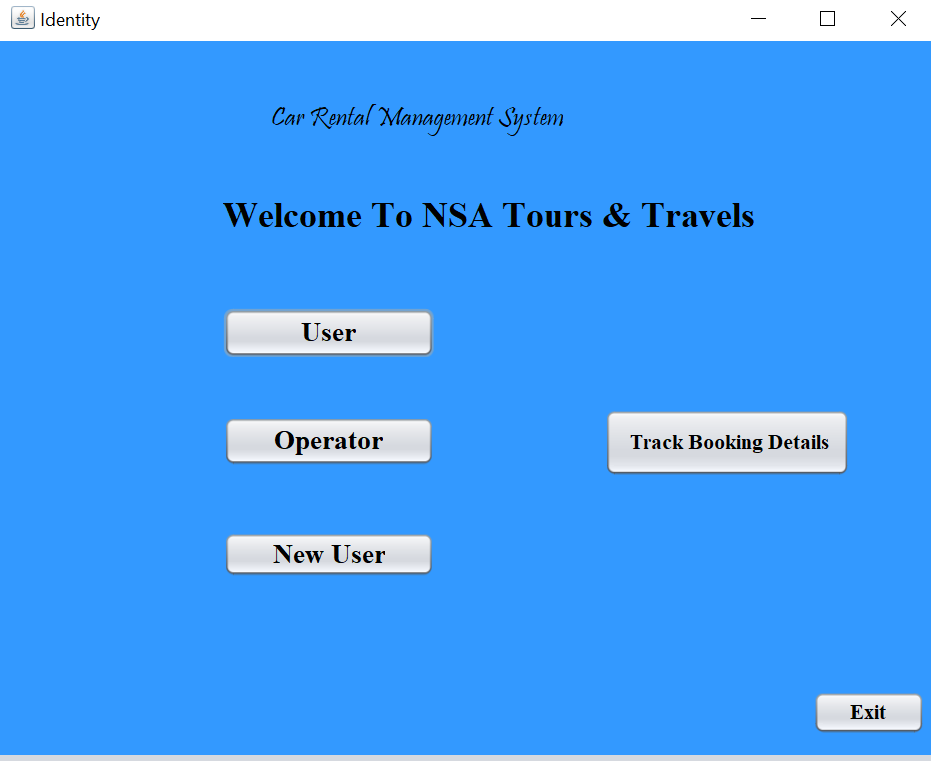
GitHub link to the codebase: <https://github.com/AugeGottes/OOADJ_Project>

Individual contributions of team members

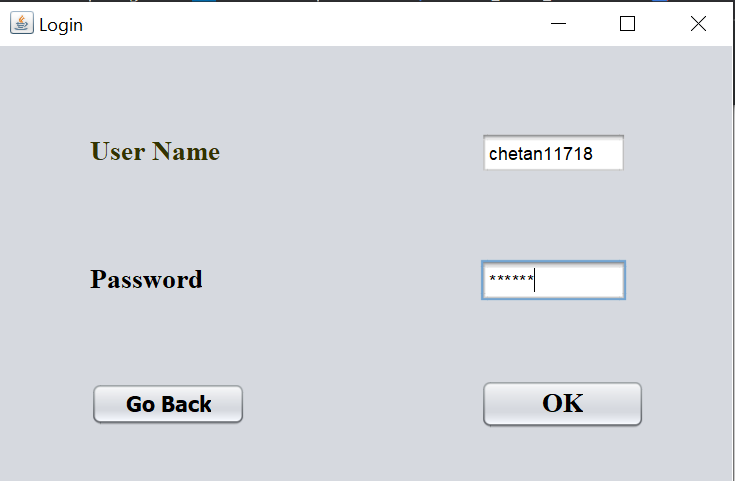
* Debanjan Das: Login and Payment.
* Chetan Gurram: Car Information and Rental Information.
* Chetan Reddy Bandi: Customer Information and Operator details.
* Adarsh Kumar: Booking and new Registration.

**Screenshots of the GUI:**

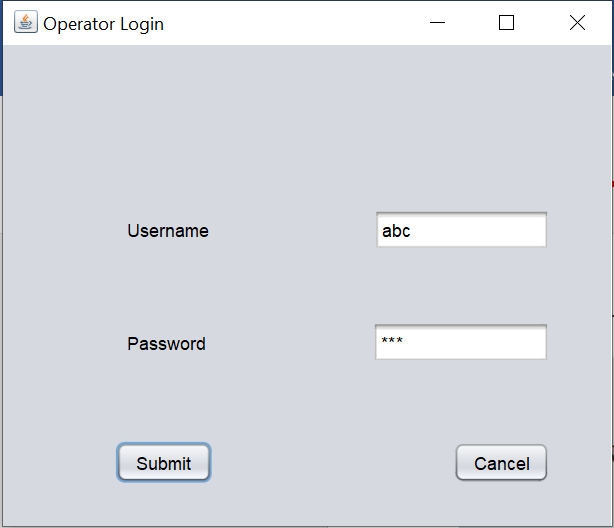
Landing Page:



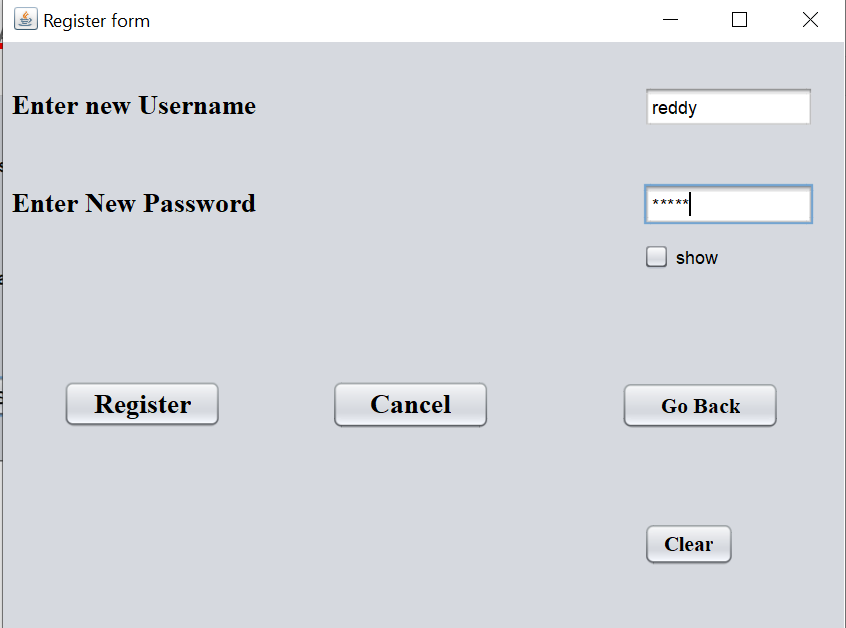
Login page for user:



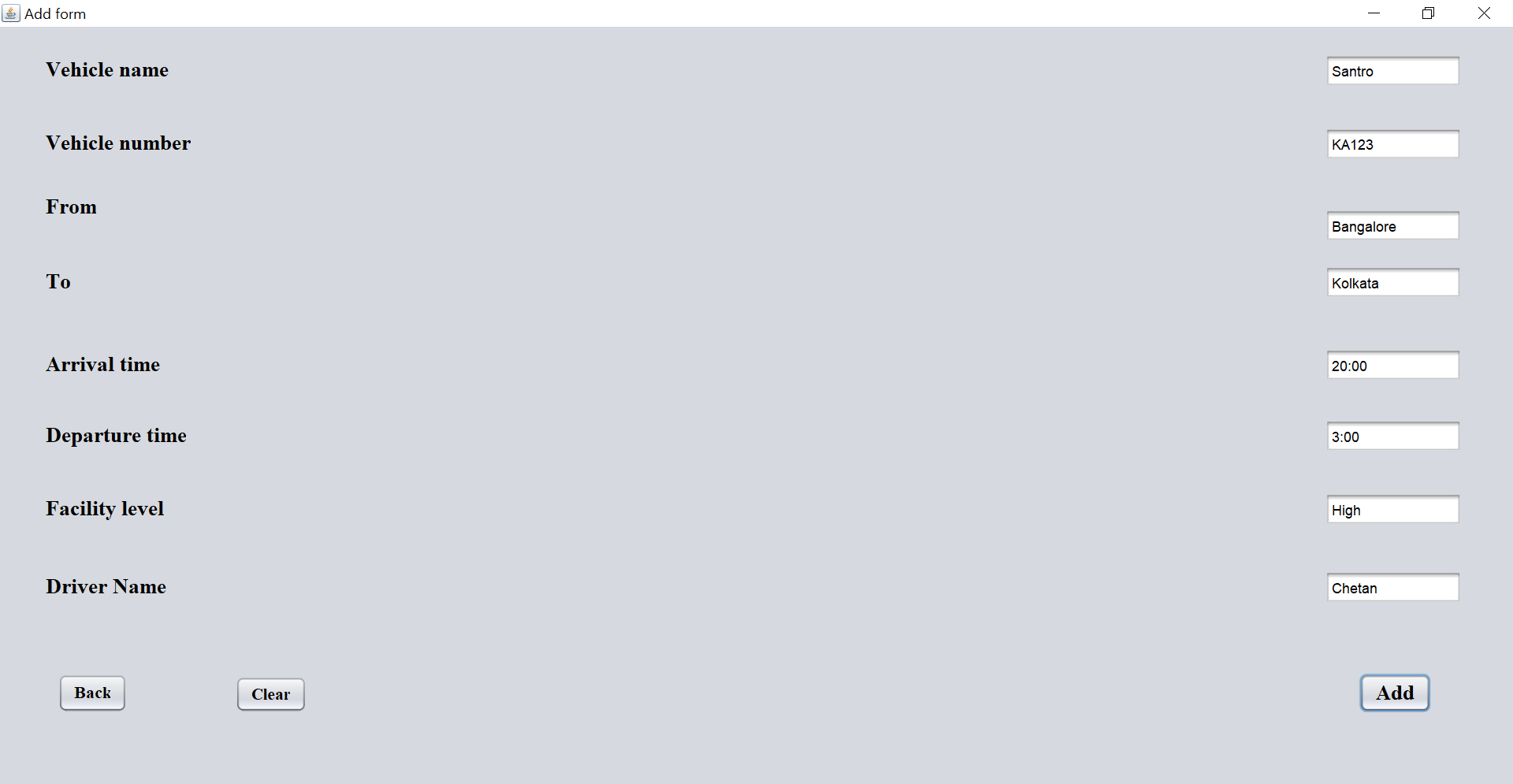
Operator Login Page:



New User Register Page:



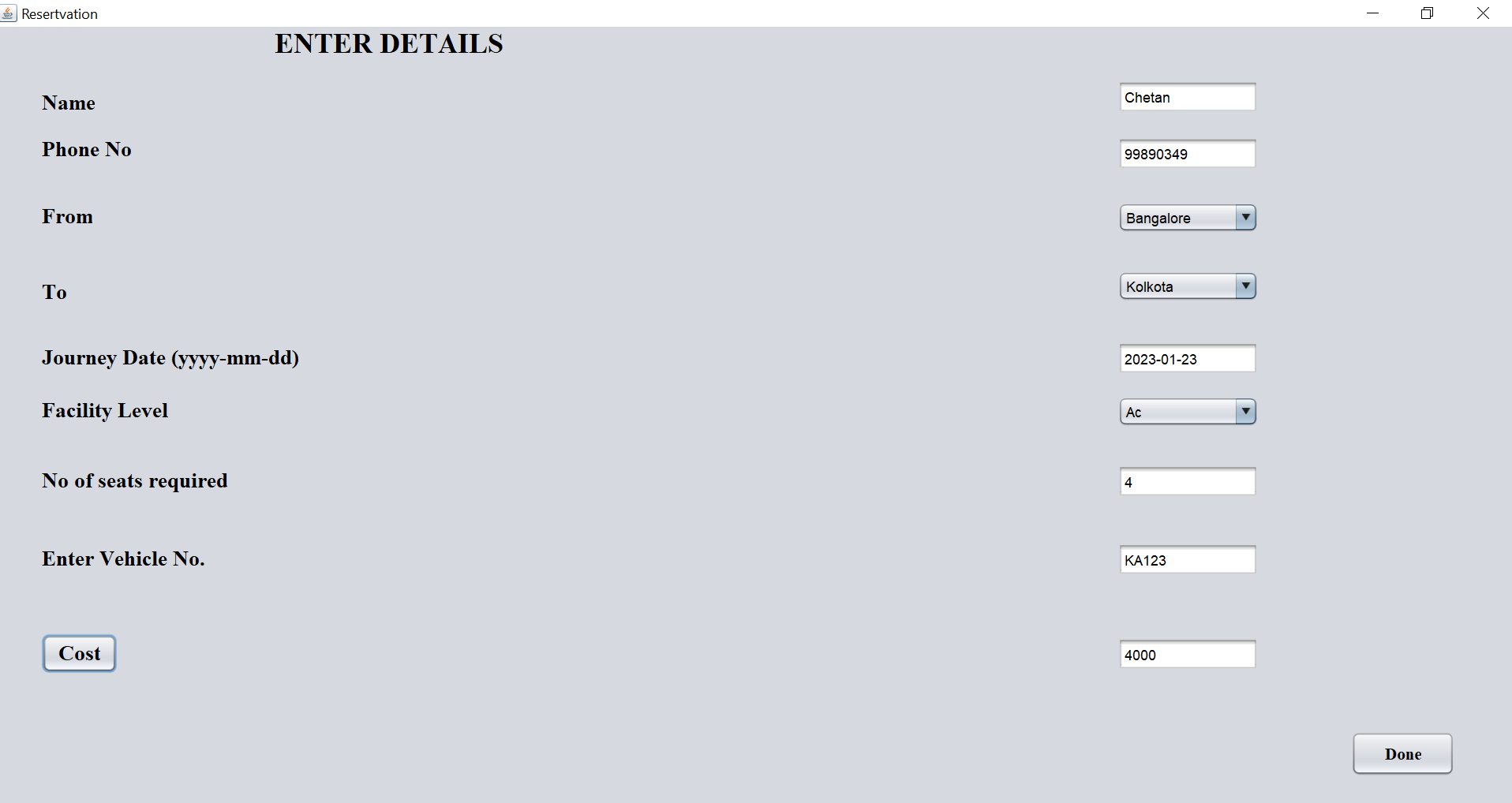
Operator adding Details of Car Page:



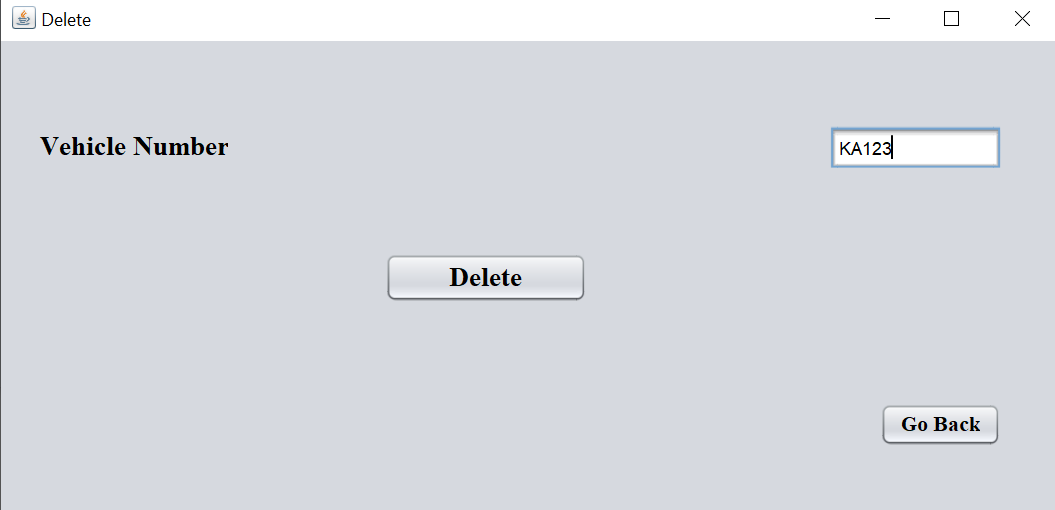
Selection form :



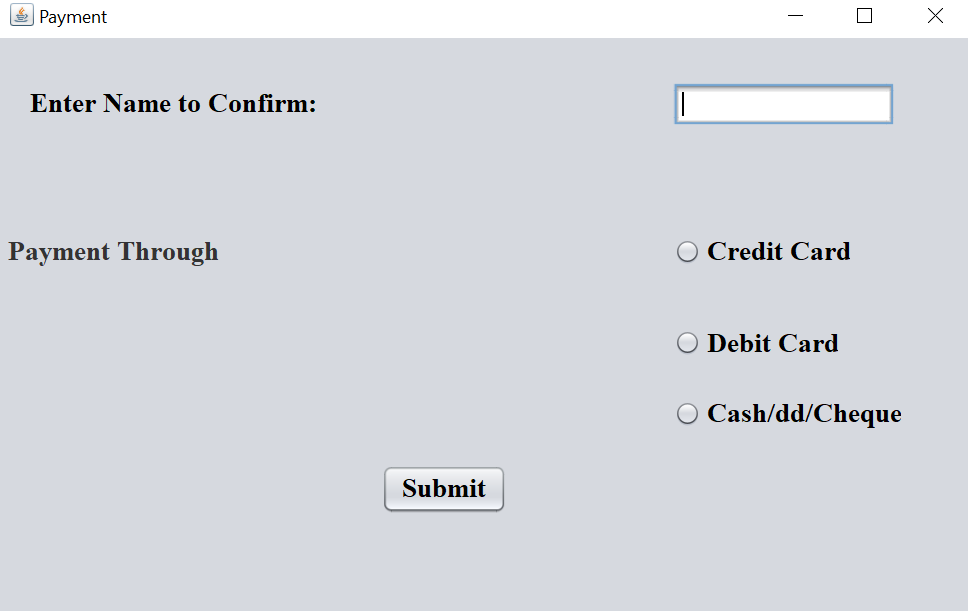
Reservation Page:



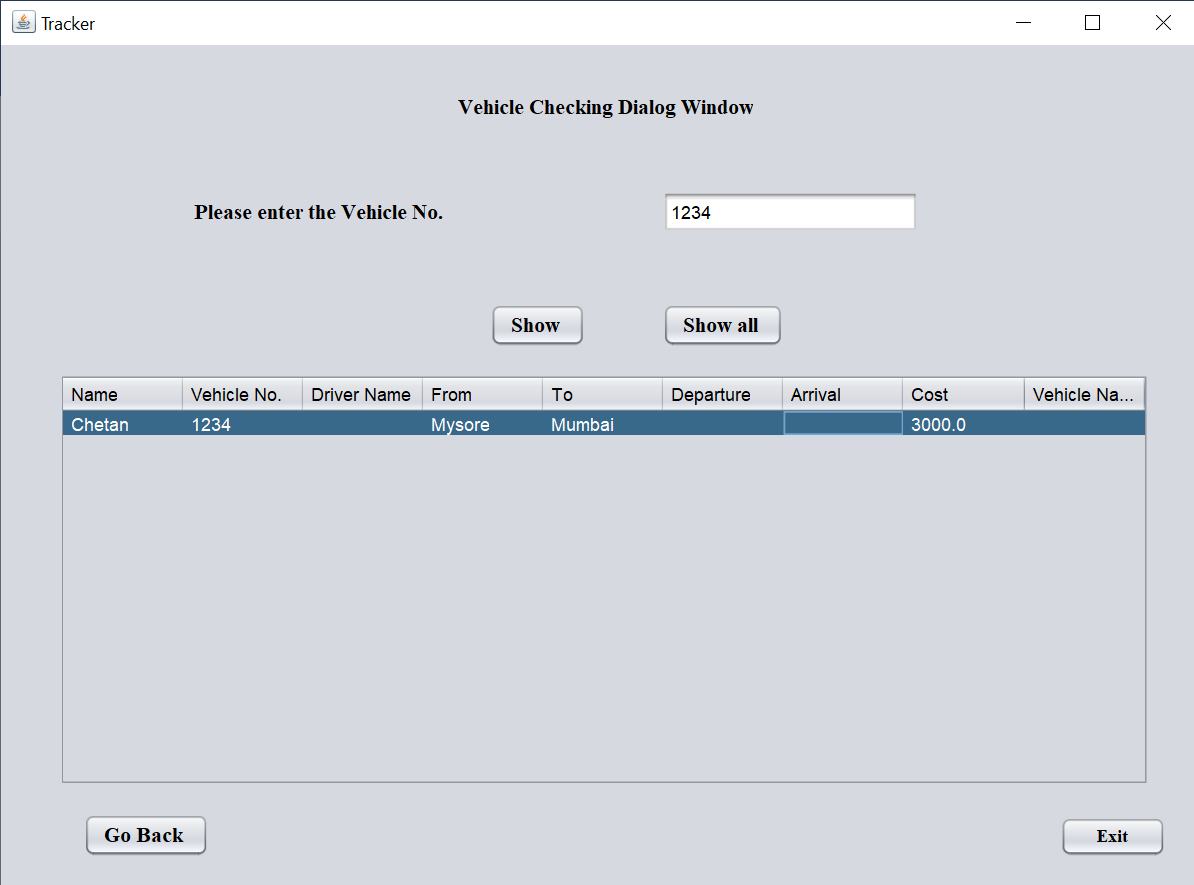
Delete car from operator:



Payment:



Tracker:



Database:

