# Car Rental System

## By Group 11

# ABSTRACT

The Car Rental System project presents a comprehensive solution for managing car rentals efficiently. At its core, the system is structured around three key entities: cars, customers, and rentals. Through encapsulated classes and methods, the project ensures modularity and extensibility, allowing for easy integration of additional features in the future. The Car class encapsulates essential attributes of a vehicle, including its identification, brand, model, and rental details. Customers are represented as distinct entities with unique identifiers and personal information. Rentals bridge the gap between cars and customers, facilitating the process of renting and returning vehicles seamlessly.

Complementing the backend logic, the Car Rental System GUI offers an intuitive interface for users to interact with the system. Leveraging Java's Swing framework, the GUI provides functionalities for renting and returning cars, as well as displaying available cars for rental. Through text fields and buttons, users can input necessary information and execute actions with minimal effort. The GUI enhances user experience by providing real-time updates on available cars, streamlining the rental process, and offering immediate feedback through informative message dialogs. Together, the backend logic and GUI frontend form a cohesive solution that empowers users to manage car rentals effectively and effortlessly.

# Methods

In the Car Rental System project, several methods are utilized across various classes to enable functionalities such as renting cars, returning cars, managing customers, and interacting with the user through a graphical user interface (GUI). Here's an overview of the methods used in the project:

1. \*\*Car Class Methods:\*\*

- `calculatePrice(int rentalDays)`: Calculates the total rental price based on the number of rental days.

- `isAvailable()`: Checks if the car is available for rent.

- `rent()`: Marks the car as rented.

- `returnCar()`: Marks the car as available for rent again.

2. \*\*Customer Class Methods:\*\*

- `getCustomerId()`: Retrieves the customer's ID.

- `getName()`: Retrieves the customer's name.

3. \*\*Rental Class Methods:\*\*

- `getCar()`: Retrieves the rented car.

- `getCustomer()`: Retrieves the customer who rented the car.

- `getDays()`: Retrieves the number of days the car was rented for.

4. \*\*CarRentalSystem Class Methods:\*\*

- `addCar(Car car)`: Adds a new car to the system.

- `addCustomer(Customer customer)`: Adds a new customer to the system.

- `rentCar(Car car, Customer customer, int days)`: Processes the rental of a car by a customer for a specified number of days.

- `returnCar(Car car)`: Processes the return of a rented car.

- `menu()`: Displays a menu-driven interface for user interaction.

- `getCars()`: Retrieves the list of cars in the system.

- `getCustomers()`: Retrieves the list of customers in the system.

- `getRentals()`: Retrieves the list of rental transactions in the system.

5. \*\*CarRentalSystemGUI Class Methods:\*\*

- `initialize()`: Initializes the graphical user interface components.

- `updateAvailableCarsList()`: Updates the list of available cars displayed in the GUI.

- `rentCar()`: Processes the rental of a car when triggered by the GUI.

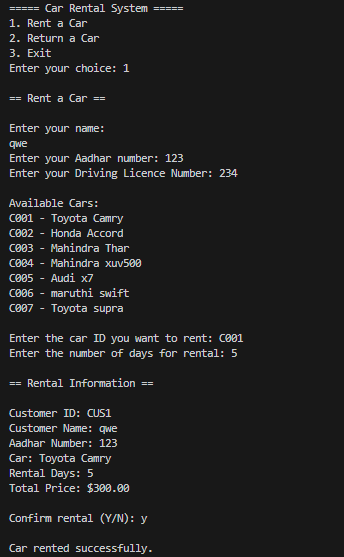
- `returnCar()`: Processes the return of a rented car when triggered by the GUI.

- `setVisible(boolean visible)`: Sets the visibility of the GUI frame.

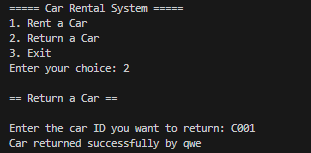
These methods collectively enable the core functionalities of the Car Rental System, from managing cars and customers to facilitating rental transactions and providing a user-friendly interface for interaction.

# OUTPUT

Renting a car



Returning a car



Exceptions

Returning a car that is not rented



Renting a car that is already rented

