**Overview**

The Car Rental System is a Java application that allows a rental agency to manage its fleet of cars and customer base. It provides functionalities to add cars and customers, rent cars to customers, view rental transactions, and display rented cars. The system persists data in text files for cars and customers.

**Classes**

**1. RentalAgency**

The main class that manages the rental operations.

**Attributes**

* cars: A Map<String, Car> to store cars, keyed by license plate.
* customers: A Map<String, Customer> to store customers, keyed by customer ID.
* rentals: A List<String> to store rental transactions.
* CUSTOMER\_FILE, CAR\_FILE: Constants for file names to store customer and car data.

**Methods**

* addCar(Car car): Adds a car to the car .
* addCustomer(Customer customer): Adds a customer to the system.
* findCar(String licensePlate): Finds a car by its license plate.
* findCustomer(String id): Finds a customer by ID.
* rentCar(String licensePlate, String customerId): Rents a car to a customer.
* getRentals(): Returns all rental transactions.
* getCars(): Returns all cars.
* getCustomers(): Returns all customers.
* getAvailableCars(): Returns a list of available (not rented) cars.
* saveCustomers(), loadCustomers(): Saves/loads customer data to/from file.
* saveCars(), loadCars(): Saves/loads car data to/from file.
* main(String[] args): The entry point, provides a console-based menu.

**2. Customer**

Represents a customer in the rental system.

**Attributes**

* id: Customer's unique identifier.
* name: Customer's name.

**Methods**

* Getters for id and name.

**3. Car**

Represents a car in the rental fleet.

**Attributes**

* licensePlate: Car's unique identifier.
* model: Car's model name.
* isRented: Boolean indicating if the car is currently rented.

**Methods**

* Getters for licensePlate, model, and isRented.
* rent(): Marks the car as rented.

**Workflow**

1. On startup, the system loads existing customer and car data from files.
2. The user interacts with the system through a console menu with options:
   * Add a customer (with ID and name).
   * Add a car (with license plate and model).
   * Rent a car (displays available cars, then prompts for car and customer).
   * Show rental transactions.
   * Show all rented cars.
   * Exit the system.
3. Data is saved to files after each add or rent operation.

**Data Persistence**

* Customer data is stored in customers.txt in the format: customerId,customerName.
* Car data is stored in cars.txt in the format: licensePlate,model,isRented.

**LIMITATIONS**

1. No rental pricing or payment processing.