import java.io.\*;

import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;

car rental system - I am using here as system name

public static final String CAR\_FILE = "cars. txt";

private static final String RENTAL\_FILE = "rentals. txt";

private List customers;

public CarRentalSystem() {

this.customers = new ArrayList<>();

}

public void addCar(Car car) {

try (BufferedWriter writer = new BufferedWriter(new FileWriter(CAR\_FILE, true))) {

writer. write(car. //Instead car. getLicensePlate() + "," + car. getMake() + "," + car. getModel() + "," + car. isAvailable());

writer.newLine();

System. out. System.out.println( "Car added successfully!" ); );

} catch (IOException e) {

System. out. System.out.println(""Error adding car: "" + e.getMessage());

}

}

public void deleteCar(String licensePlate) {

List cars = readCars();

Car car = findCar(licensePlate, cars); Predictable Car car = findCar(licensePlate, cars);

if (car!= null) {

cars.remove(car);

writeCars(cars);

System. out. println("Car removed successfully!" );

} else {

System. out. println("Car not found!" );

}

}

void modifyCar(String licensePlate, String newMake, String newModel) {

List cars = readCars();

Car car = findCar(licensePlate, cars);

if (car!= null) {

car.setMake(newMake);

car. setModel(newModel);

writeCars(cars);

}

System.out.println("\t\t-----------------------------------------------------------------");

System.out.println("\t\t\t\t CAPALOT CAR RENTAL AGENCY MENU");

System.out.println("\t\t-----------------------------------------------------------------");

System.out.println("\t\t\n1. Add Car");

System.out.println("\t\t\n2. Remove Car");

System.out.println("\t\t\n3. Rent Car");

System.out.println("\t\t\n4. Return Car");

System.out.println("\t\t\n5. Modify Car");

System.out.println("\t\t\n6. Display Cars");

System.out.println("\t\t\n0. Exit");

System.out.print("\n\t\t\nEnter your choice: ");

choice = scanner.nextInt();

scanner.nextLine(); // Consume newline

switch (choice) {

case 1:

System.out.print("Enter license plate: ");

String licensePlate = scanner.nextLine();

System.out.print("Enter make: ");

String make = scanner.nextLine();

System.out.print("Enter model: ");

String model = scanner.nextLine();

Car newCar = new Car(licensePlate, make, model);

addCar(newCar);

break;

case 2:

System.out.print("Enter license plate of the car to remove: ");

String plateToRemove = scanner.nextLine();

removeCar(plateToRemove);

break;

case 3:

System.out.print("Enter customer ID: ");

int customerId = scanner.nextInt();

scanner.nextLine(); // Consume newline

System.out.print("Enter license plate of the car to rent: ");

String plateToRent = scanner.nextLine();

System.out.print("Enter number of days: ");

int days = scanner.nextInt();

scanner.nextLine(); // Consume newline

Customer customerToRent = new Customer(customerId, "CustomerName"); // Simplified customer creation

Car carToRent = findCar(plateToRent, readCars());

if (carToRent != null) {

rentCar(customerToRent, carToRent, days);

} else {

System.out.println("Invalid car!");

}

break;

case 4:

System.out.print("Enter customer ID: ");

int customerIdReturn = scanner.nextInt();

scanner.nextLine(); // Consume newline

System.out.print("Enter license plate of the car to return: ");

String plateToReturn = scanner.nextLine();

Customer customerToReturn = new Customer(customerIdReturn, "CustomerName"); // Simplified customer creation

Car carToReturn = findCar(plateToReturn, readCars());

if (carToReturn != null) {

returnCar(customerToReturn, carToReturn);

} else {

System.out.println("Invalid car!");

}

break;

case 5:

System.out.print("Enter license plate of the car to modify: ");

String plateToModify = scanner.nextLine();

System.out.print("Enter new make: ");

String newMake = scanner.nextLine();

System.out.print("Enter new model: ");

String newModel = scanner.nextLine();

modifyCar(plateToModify, newMake, newModel);

break;

case 6:

displayCars();

break;

case 0:

System.out.println("Exiting...");

break;

default:

System.out.println("Invalid choice! Please try again.");

}

} while (choice != 0);

scanner.close();

}

private List<Car> readCars() {

List<Car> cars = new ArrayList<>();

try (BufferedReader reader = new BufferedReader(new FileReader(CAR\_FILE))) {

String line;

while ((line = reader.readLine()) != null) {

String[] parts = line.split(",");

if (parts.length == 4) {

cars.add(new Car(parts[0], parts[1], parts[2], Boolean.parseBoolean(parts[3])));

}

}

} catch (IOException e) {

System.out.println("Error reading cars: " + e.getMessage());

}

return cars;

}

private void writeCars(List<Car> cars) {

try (BufferedWriter writer = new BufferedWriter(new FileWriter(CAR\_FILE))) {

for (Car car : cars) {

writer.write(car.getLicensePlate() + "," + car.getMake() + "," + car.getModel() + "," + car.isAvailable());

writer.newLine();

}

} catch (IOException e) {

System.out.println("Error writing cars: " + e.getMessage());

}

}

private List<Rental> readRentals() {

List<Rental> rentals = new ArrayList<>();

try (BufferedReader reader = new BufferedReader(new FileReader(RENTAL\_FILE))) {

String line;

while ((line = reader.readLine()) != null) {

String[] parts = line.split(",");

if (parts.length == 3) {

Customer customer = new Customer(Integer.parseInt(parts[0]), "CustomerName"); // Simplified customer creation

Car car = findCar(parts[1], readCars());

if (car != null) {

rentals.add(new Rental(customer, car, Integer.parseInt(parts[2])));

}

}

}

} catch (IOException e) {

System.out.println("Error reading rentals: " + e.getMessage());

}

return rentals;

}

private void writeRentals(List<Rental> rentals) {

try (BufferedWriter writer = new BufferedWriter(new FileWriter(RENTAL\_FILE))) {

for (Rental rental : rentals) {

writer.write(rental.getCustomer().getId() + "," + rental.getCar().getLicensePlate() + "," + rental.getDays());

writer.newLine();

}

} catch (IOException e) {

System.out.println("Error writing rentals: " + e.getMessage());

}

}

public static void main(String[] args) {

CarRentalSystem rentalSystem = new CarRentalSystem();

rentalSystem.menu();

}

}

class Car {

private String licensePlate;

private String make;

private String model;

private boolean available;

public Car(String licensePlate, String make, String model) {

this.licensePlate = licensePlate;

this.make = make;

this.model = model;

this.available = true;

}

public Car(String licensePlate, String make, String model, boolean available) {

this.licensePlate = licensePlate;

this.make = make;

this.model = model;

this.available = available;

}

public String getLicensePlate() {

return licensePlate;

}

public String getMake() {

return make;

}

public void setMake(String make) {

this.make = make;

}

public String getModel() {

return model;

}

public void setModel(String model) {

this.model = model;

}

public boolean isAvailable() {

return available;

}

public void setAvailable(boolean available) {

this.available = available;

}

}

class Customer {

private int id;

private String name;

public Customer(int id, String name) {

this.id = id;

this.name = name;

}

public int getId() {

return id;

}

public String getName() {

return name;

}

}

class Rental {

private Customer customer;

private Car car;

private int days;

public Rental(Customer customer, Car car, int days) {

this.customer = customer;

this.car = car;

this.days = days;

}

public Customer getCustomer() {

return customer;

}

public Car getCar() {

return car;

}

public int getDays() {

return days;

}

}