Camera Rental Application

Source code for camera rental application.

```
package rentCam;

import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;
import java.util.Collections;
import java.util.Comparator;


public class Camera {

	private int id;
	private String brand;
	private String model;
	private double price;
	private boolean available;

	public Camera(int id, String brand, String model, double price, boolean available) {
	    this.id = id;
	    this.brand = brand;
	    this.model = model;
	    this.price = price;
	    this.available = available;
	}

	// Getters and setters

	public int getId() {
	    return id;
	}

	public String getBrand() {
	    return brand;
	}

	public String getModel() {
	    return model;
	}

	public double getPrice() {
```

```
        return price;
    }

    public boolean isAvailable() {
        return available;
    }

    public void setAvailable(boolean available) {
        this.available = available;
    }
}

class CameraRentalApp {
    private List<Camera> cameraList;
    private double walletBalance;

    public CameraRentalApp() {
        cameraList = new ArrayList<>();
        walletBalance = 10000.0;

        //Added the camera list
        cameraList.add(new Camera(1, "Samsung", "DS123", 500.0, true));
        cameraList.add(new Camera(2, "Sony", "HD214", 500.0, true));
        cameraList.add(new Camera(3, "Panasonic", "XC", 500.0, true));
        cameraList.add(new Camera(4, "Canon", "XLR", 500.0, false));
        cameraList.add(new Camera(5, "Fujitsu", "J5", 500.0, true));
        cameraList.add(new Camera(6, "Sony", "HD226", 500.0, true));
        cameraList.add(new Camera(7, "Samsung", "DS246", 500.0, true));
        cameraList.add(new Camera(8, "LG", "L123", 500.0, true));
        cameraList.add(new Camera(9, "Canon", "XPL", 500.0, true));
        cameraList.add(new Camera(10, "Chroma", "CT", 500.0, true));
        cameraList.add(new Camera(11, "Something", "some", 200.0, false));
        cameraList.add(new Camera(12, "Some", "Another", 100.0, true));
        cameraList.add(new Camera(13, "Canon", "Digital", 123.0, false));
        cameraList.add(new Camera(14, "Nikon", "DSLR-D7500", 500.0, true));
        cameraList.add(new Camera(15, "Sony", "DSLR12", 200.0, true));
        cameraList.add(new Camera(17, "Samsung", "SM123", 200.0, true));
        cameraList.add(new Camera(19, "SONY", "SONY1234", 123.0, true));

    }
//User Info and login Credentials
    public void login() {
        System.out.println("WELCOME TO CAMERA RENTAL APP ");
        System.out.println("PLEASE LOGIN TO CONTINUE - ");
```

```
    Scanner scanner = new Scanner(System.in);
    System.out.print("USERNAME - ");
    String username = scanner.nextLine();
    System.out.print("PASSWORD - ");
    String password = scanner.nextLine();

    if (username.equals("RRR") && password.equals("@RRR123")) {
        showMainMenu();
    } else {
        System.out.println("Invalid credentials. Exiting the app...");
    }
}

public void showMainMenu() {
    Scanner scanner = new Scanner(System.in);
    int option;

    do {

        System.out.println("\nMAIN MENU");
        System.out.println("1. MY CAMERA");
        System.out.println("2. RENT A CAMERA");
        System.out.println("3. VIEW ALL CAMERAS");
        System.out.println("4. MY WALLET");
        System.out.println("5. EXIT");

        option = scanner.nextInt();
        scanner.nextLine();
        switch (option) {
            case 1:
                showMyCameraMenu();
                break;
            case 2:
                rentCamera();
                break;
            case 3:
                viewAllCameras();
                break;
            case 4:
                showWalletMenu();
                break;
            case 5:
                System.out.println("Exiting the app...");
```

```
                break;
            default:
                System.out.println("Invalid choice. Please try again.");
        }
    } while (option != 5);
}

public void showMyCameraMenu() {
    Scanner scanner = new Scanner(System.in);
    int option;

    do {

        System.out.println("1. ADD");
        System.out.println("2. REMOVE");
        System.out.println("3. VIEW MY CAMERAS");
        System.out.println("4. GO TO PREVIOUS MENU");

        option = scanner.nextInt();
        scanner.nextLine();

        switch (option) {
            case 1:
                addCamera();
                break;
            case 2:
                removeCamera();
                break;
            case 3:
                viewMyCameras();
                break;
            case 4:
                return;
            default:
                System.out.println("Invalid choice. Please try again.");
        }
    } while (option != 4);
}

public void addCamera() {
    Scanner scanner = new Scanner(System.in);

    System.out.print("Enter the camera brand: ");
    String brand = scanner.nextLine();
```

```
            System.out.print("Enter the model: ");
            String model = scanner.nextLine();
            System.out.print("Enter the per day price (INR): ");
            double price = scanner.nextDouble();
            scanner.nextLine();

            int id = cameraList.size() + 1;
            Camera camera = new Camera(id, brand, model, price, true);
            cameraList.add(camera);

            System.out.println("YOUR CAMERA HAS BEEN SUCCESSFULLY ADDED TO
THE LIST.");
        }

        public void removeCamera() {
            Scanner scanner = new Scanner(System.in);

            System.out.println("\nCAMERA LIST");
            displayCameraList();

            System.out.print("ENTER THE CAMERA ID TO REMOVE: ");
            int id = scanner.nextInt();
            scanner.nextLine();

            boolean found = false;
            for (Camera camera : cameraList) {
                if (camera.getId() == id) {
                    cameraList.remove(camera);
                    found = true;
                    break;
                }
            }

            if (found) {
                System.out.println("CAMERA SUCCESSFULLY REMOVED FROM THE LIST.");
            } else {
                System.out.println("CAMERA NOT FOUND.");
            }
        }

        public void viewMyCameras() {
            System.out.println("\nMY CAMERA LIST");
            displayCameraList();
        }
```

```
    public void rentCamera() {
        Scanner scanner = new Scanner(System.in);

        System.out.println("\nAVAILABLE CAMERAS");
        displayAvailableCameras();

        System.out.print("ENTER THE CAMERA ID YOU WANT TO RENT: ");
        int id = scanner.nextInt();
        scanner.nextLine();

        Camera selectedCamera = null;
        for (Camera camera : cameraList) {
            if (camera.getId() == id && camera.isAvailable()) {
                selectedCamera = camera;
                break;
            }
        }

        if (selectedCamera != null) {
            double rentPrice = selectedCamera.getPrice();
            if (walletBalance >= rentPrice) {
                selectedCamera.setAvailable(false);
                walletBalance -= rentPrice;
                System.out.println("YOUR TRANSACTION FOR CAMERA  " +
selectedCamera.getBrand() + " " +
                        selectedCamera.getModel() + " with rent INR " + rentPrice + " HAS
SUCCESSFULLY COMPLETED.");
            } else {
                System.out.println("TRANSACTION FAILED DUE TO INSUFFICIENT
WALLET BALANCE. " +
                        "PLEASE DEPOSITE THE AMOUNT TO YOUR WALLET.");
            }
        } else {
            System.out.println("CAMERA NOT FOUND OR NOT AVAILABLE.");
        }
    }

    public void viewAllCameras() {

         // Sort the camera list by ID in ascending order
        Collections.sort(cameraList, Comparator.comparingInt(Camera::getId));

        System.out.println("\nAVAILABLE CAMERAS:");
```

```
            displayCameraList();

            showMainMenu();
            System.out.println("\nALL CAMERAS");
            displayCameraList();
        }

        public void showWalletMenu() {
            Scanner scanner = new Scanner(System.in);
            int option;

            do {
                System.out.println("\nMY WALLET");
                System.out.println("1. VIEW WALLET BALANCE");
                System.out.println("2. DEPOSIT MONEY");
                System.out.println("3. GO TO PREVIOUS MENU");
                System.out.print("Enter your choice: ");
                option = scanner.nextInt();
                scanner.nextLine();

                switch (option) {
                    case 1:
                        viewWalletBalance();
                        break;
                    case 2:
                        depositMoney();
                        break;
                    case 3:
                        return;
                    default:
                        System.out.println("Invalid choice. Please try again.");
                }
            } while (option != 3);
        }

        public void viewWalletBalance() {
            System.out.println("\nYOUR CURRENT WALLET BALANCE IS INR - " +
walletBalance);
        }

        public void depositMoney() {
            Scanner scanner = new Scanner(System.in);
```

```
            System.out.println("\nYOUR CURRENT WALLET BALANCE IS INR - " +
walletBalance);

            System.out.print("DO YOU WANT TO DEPOSIT MORE AMOUNT TO YOUR
WALLET? (1.YES 2.NO): ");
            int choice = scanner.nextInt();
            scanner.nextLine();
            if (choice == 1) {
                System.out.print("ENTER THE AMOUNT (INR): ");
                double amount = scanner.nextDouble();
                scanner.nextLine();

                walletBalance += amount;
                System.out.println("YOUR WALLET BALANCE UPDATED SUCCESSFULLY.
CURRENT WALLET BALANCE - INR. " + walletBalance);
            }
        }

        public void displayCameraList() {
            System.out.printf("%-10s%-15s%-15s%-15s%-10s\n", "CAMERA ID", "BRAND",
"MODEL", "PRICE (PER DAY)", "STATUS");
            for (Camera camera : cameraList) {
                System.out.printf("%-10d%-15s%-15s%-15.2f%-10s\n", camera.getId(),
camera.getBrand(), camera.getModel(),
                        camera.getPrice(), camera.isAvailable() ? "Available" : "Rented");
            }
        }

        public void displayAvailableCameras() {
            System.out.printf("%-10s%-15s%-15s%-15s%-10s\n", "CAMERA ID", "BRAND",
"MODEL", "PRICE (PER DAY)", "STATUS");
            for (Camera camera : cameraList) {
                if (camera.isAvailable()) {
                    System.out.printf("%-10d%-15s%-15s%-15.2f%-10s\n", camera.getId(),
camera.getBrand(), camera.getModel(),
                            camera.getPrice(), "Available");
                }
            }
        }
    }
```

```
package rentCam;

public class Main {

	public static void main(String[] args) {
		CameraRentalApp app = new CameraRentalApp();
		app.login();
	}
}
```