```
package javapract;

import java.util.ArrayList;
import java.util.Scanner;

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;
    }

    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 Admin {
    private String username = "admin";
    private String password = "password";

    public boolean authenticate(String username, String password) {
        return this.username.equals(username) && this.password.equals(password);
    }
}

class User {
    private String username;
    private String password;
    private double walletBalance;
    private ArrayList<Camera> rentedCameras = new ArrayList<>();

    public User(String username, String password) {
        this.username = username;
```

```
        this.password = password;
        this.walletBalance = 0.0;
    }

    public String getUsername() {
        return username;
    }

    public String getPassword() {
        return password;
    }

    public double getWalletBalance() {
        return walletBalance;
    }

    public void setWalletBalance(double walletBalance) {
        this.walletBalance = walletBalance;
    }

    public ArrayList<Camera> getRentedCameras() {
        return rentedCameras;
    }
}

public class P1MainProjCameraRental {
    private static ArrayList<Camera> cameraList = new ArrayList<>();
    private static Admin admin = new Admin();
    private static User currentUser;

    public static void main(String[] args) {
```

```
    initializeCameras();
    welcomepage();

}
private static void welcomepage()
{
        Scanner scanner = new Scanner(System.in);
    boolean exit = false;

    while (!exit) {
         System.out.println(" ");
        System.out.println("Welcome to Camera Rental App");
        System.out.println(" ");
        System.out.println("1. Admin");
        System.out.println("2. User");
        System.out.print("Enter your choice: ");
        int choice = scanner.nextInt();
        scanner.nextLine();  // Consume newline character

        switch (choice) {
            case 1:
                adminLogin(scanner);
                adminMenu(scanner);
                break;
            case 2:
                userLogin(scanner);
                userMenu(scanner);
                break;
            default:
                System.out.println("Invalid choice. Please try again.");
        }
```

```
            System.out.print("Do you want to exit the application? (y/n): ");
            String input = scanner.nextLine();
            exit = input.equalsIgnoreCase("y");
        }

        scanner.close();
    }

    private static void adminLogin(Scanner scanner) {
        System.out.print("Enter admin username: ");
        String username = scanner.nextLine();
        System.out.print("Enter admin password: ");
        String password = scanner.nextLine();

        if (admin.authenticate(username, password)) {
            System.out.println("Login successful!");
        } else {
            System.out.println("Invalid username or password. Login failed!");
            System.exit(0);
        }
    }

    private static void adminMenu(Scanner scanner) {
        boolean backToMenu = false;

        while (!backToMenu) {
            System.out.println("\nAdmin Main Menu");
            System.out.println("1. Add Camera");
            System.out.println("2. Remove Camera");
            System.out.println("3. View All Cameras");
```

```
        System.out.println("4. Go to Previous Menu");
        System.out.print("Enter your choice: ");
        int choice = scanner.nextInt();
        scanner.nextLine();

        switch (choice) {
            case 1:
                addCamera(scanner);
                break;
            case 2:
                removeCamera(scanner);
                break;
            case 3:
                viewAllCameras();
                break;
            case 4:
                welcomepage();
                break;
            default:
                System.out.println("Invalid choice. Please try again.");
        }
    }
}



        private static void addCamera(Scanner scanner) {
    System.out.print("Enter camera ID: ");
    int id = scanner.nextInt();
    scanner.nextLine();
    System.out.print("Enter camera brand: ");
```

```
    String brand = scanner.nextLine();
    System.out.print("Enter camera model: ");
    String model = scanner.nextLine();
    System.out.print("Enter camera price: ");
    double price = scanner.nextDouble();
    scanner.nextLine();
    Camera camera = new Camera(id, brand, model, price, true);
    cameraList.add(camera);
    System.out.println("Camera added to the list.");
}

private static void removeCamera(Scanner scanner) {
    System.out.print("Enter camera ID to remove: ");
    int id = scanner.nextInt();
    scanner.nextLine();

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

    if (cameraToRemove != null) {
        cameraList.remove(cameraToRemove);
        System.out.println("Camera removed from the list.");
    } else {
        System.out.println("Camera not found in the list.");
    }
}
```

```
    private static void viewAllCameras() {
        System.out.println("\nAvailable Cameras:");

System.out.println("================================================================
==");
            System.out.println("ID\t\tBrand\t\tModel\t\tPrice\t\t");

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

        for (Camera camera : cameraList) {

          if (camera.isAvailable()) {
             System.out.println(camera.getId()+"\t\t" + camera.getBrand() +"\t\t" +
camera.getModel()+"\t\t"  + camera.getPrice()+"\t\t" );
          }
        }

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


        System.out.println("\nRented Cameras:");


System.out.println("================================================================
==");
            System.out.println("ID\t\tBrand\t\tModel\t\tPrice\t\t");

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

        for (Camera camera : cameraList) {
```

```
        if (!camera.isAvailable()) {
            System.out.println(camera.getId()+"\t\t" + camera.getBrand() +"\t\t" +
camera.getModel()+"\t\t"  + camera.getPrice()+"\t\t" );
        }
      }

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

  }

  private static void userLogin(Scanner scanner) {
    System.out.print("Enter username: ");
    String username = scanner.nextLine();
    System.out.print("Enter password: ");
    String password = scanner.nextLine();

    currentUser = new User(username, password);
    System.out.println("Login successful!");
  }

  private static void userMenu(Scanner scanner) {
    boolean backToMenu = false;

    while (!backToMenu) {
      System.out.println("\nUser Main Menu");
      System.out.println("1. My Cameras");
      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");
      System.out.print("Enter your choice: ");
```

```
        int choice = scanner.nextInt();
        scanner.nextLine();

        switch (choice) {
            case 1:
                viewRentedCameras();
                break;
            case 2:
                rentCamera(scanner);
                break;
            case 3:
                viewmyCameras();
                break;
            case 4:
                myWallet(scanner);
                break;
            case 5:
                welcomepage();
                break;
            default:
                System.out.println("Invalid choice. Please try again.");
        }
      }
   }
   private static void viewmyCameras()
   {

          System.out.println("============================================================
========");
          System.out.println("ID\t\tBrand\t\tModel\t\tPrice\t\t");
```

```
            System.out.println("============================================================
========");


        for (Camera camera : cameraList) {


            if (camera.isAvailable()) {

                System.out.println(camera.getId()+"\t\t" + camera.getBrand() +"\t\t" +
camera.getModel()+"\t\t"  + camera.getPrice()+"\t\t" );

            }

        }


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




    }

    private static void viewRentedCameras() {

        ArrayList<Camera> rentedCameras = currentUser.getRentedCameras();

        if (rentedCameras.isEmpty()) {

            System.out.println("You haven't rented any cameras yet.");

        } else {

            System.out.println("\nRented Cameras:");


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

            System.out.println("ID\t\tBrand\t\tModel\t\tPrice\t\t");


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

            for (Camera camera : rentedCameras) {

                System.out.println(camera.getId()+"\t\t" + camera.getBrand() +"\t\t" +
camera.getModel()+"\t\t"  + camera.getPrice()+"\t\t" );
```

```
        }

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


    }
  }

  private static void rentCamera(Scanner scanner) {
    System.out.print("Enter camera ID 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 walletBalance = currentUser.getWalletBalance();
      double cameraPrice = selectedCamera.getPrice();
      if (walletBalance >= cameraPrice) {
        currentUser.setWalletBalance(walletBalance - cameraPrice);
        selectedCamera.setAvailable(false);
        currentUser.getRentedCameras().add(selectedCamera);
        System.out.println("Camera rented successfully!");
      } else {
        System.out.println("Transaction failed! Insufficient wallet balance.");
      }
```

```
    } else {
        System.out.println("Camera not available for rent.");
    }
}

private static void myWallet(Scanner scanner) {
    System.out.println("Wallet Balance: $" + currentUser.getWalletBalance());
    System.out.print("Do you want to deposit more amount? (1. Yes / 2. No): ");
    int choice = scanner.nextInt();
    scanner.nextLine();

    if (choice == 1) {
        System.out.print("Enter the amount to deposit: $");
        double amount = scanner.nextDouble();
        scanner.nextLine();

        double walletBalance = currentUser.getWalletBalance();
        currentUser.setWalletBalance(walletBalance + amount);
        System.out.println("Amount deposited successfully!");
        System.out.println("Updated Wallet Balance: $" + currentUser.getWalletBalance());
    }
}

private static void initializeCameras() {
        cameraList.add(new Camera(1, "SONY", "SONY1234", 123.0, true));
        cameraList.add(new Camera(2, "canon", "5050", 500.0, true));
    cameraList.add(new Camera(3, "nikon", "2030", 800.0, true));
    cameraList.add(new Camera(4, "Sony", "DS123", 700.0, true));
    cameraList.add(new Camera(5, "Sony", "HD214", 600.0, true));
    cameraList.add(new Camera(6, "Canon", "EOS R5", 250.0, true));
    cameraList.add(new Camera(7, "Sony", "Alpha", 350.0, true));
```

```
        cameraList.add(new Camera(8, "Nikon", "Z7 II", 550.0, true));
        cameraList.add(new Camera(9, "Samsung", "DS123", 700.0, true));
        cameraList.add(new Camera(10, "Sony", "HD214", 900.0, true));
        cameraList.add(new Camera(11, "Canon", "XPL", 800.0, true));
        cameraList.add(new Camera(12, "Chroma", "cT", 400.0, true));
        cameraList.add(new Camera(13, "Canon", "Digital", 623.0, true));
        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(16, "Panasonic", "XC", 400.0, true));
        cameraList.add(new Camera(17, "Canon", "XLR", 700.0, true));
        cameraList.add(new Camera(18, "Fujitsu", "Is", 600.0, true));
        cameraList.add(new Camera(19, "Sony", "HD226", 800.0, true));
        cameraList.add(new Camera(20, "LG", "L123", 500.0, true));


    }
}
```