```java
package endproject;

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

class Camera {
    private int id;
    private String brand;
    private String model;
    private double pricePerDay;
    private boolean available;

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

    public int getId() {
        return id;
    }

    public String getBrand() {
        return brand;
    }

    public String getModel() {
        return model;
    }

    public double getPricePerDay() {
        return pricePerDay;
    }

    public boolean isAvailable() {
        return available;
    }

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

class User {
    private String username;
    private String password;
    private double walletBalance;

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

```
        this.walletBalance = walletBalance;
    }

    public String getUsername() {
        return username;
    }

    public String getPassword() {
        return password;
    }

    public double getWalletBalance() {
        return walletBalance;
    }

    public void depositToWallet(double amount) {
        walletBalance += amount;
    }

    public void deductFromWallet(double amount) {
        walletBalance -= amount;
    }
}

class CameraRentalApp {
    private User currentUser;
    private List<Camera> cameraList;

    public CameraRentalApp() {
        currentUser = null;
        cameraList = new ArrayList<>();

        // Add predefined cameras
        Camera camera1 = new Camera(12, "some", "another", 100.0, true);
        Camera camera2 = new Camera(17, "Samsung", "SM123", 200.0, true);
        Camera camera3 = new Camera(3, "Sony", "Alpha A7 III", 55.0, true);
        Camera camera4 = new Camera(11, "something", "some", 200.0, true);
        Camera camera5 = new Camera(14, "NIKON", "DSLR-D7500", 500.0, true);
        Camera camera6 = new Camera(2, "Nikon", "D850", 60.0, true);

        cameraList.add(camera1);
        cameraList.add(camera2);
        cameraList.add(camera3);
        cameraList.add(camera4);
        cameraList.add(camera5);
        cameraList.add(camera6);
    }


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

        System.out.println("WELCOME TO CAMERA RENTAL APP\n");
        System.out.println("PLEASE LOGIN TO CONTINUE:");
```

```
        while (true) {
            System.out.print("USERNAME - ");
            String username = scanner.nextLine();

            System.out.print("PASSWORD - ");
            String password = scanner.nextLine();
            if (login(username, password)) {
                break;
            } else {
                System.out.println("Invalid username or password. Please try
again.");
            }

        }

        showMainMenu();
    }

    private boolean login(String username, String password) {
        // Check if the username and password are valid
        // For simplicity, we assume the usernames and passwords are stored in a list
        List<User> userList = new ArrayList<>();
        userList.add(new User("thomas", "abcdef", 100.0));
        userList.add(new User("esther", "asha1993", 0.0));
        userList.add(new User("john", "johnabc", 0.0));
        userList.add(new User("cena", "cenaabc", 200.0));
        // Add more users as needed

        for (User user : userList) {
            if (user.getUsername().equals(username) &&
user.getPassword().equals(password)) {
                currentUser = user;
                return true;
            }
        }

        return false;
    }



    private void showMainMenu() {
        Scanner scanner = new Scanner(System.in);

        while (true) {
            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");

            System.out.print("Enter your choice: ");
            int choice = scanner.nextInt();
            scanner.nextLine(); // Consume the newline character
```

```
            switch (choice) {
                case 1:
                    showMyCameraMenu();
                    break;
                case 2:
                    rentCamera();
                    break;
                case 3:
                    viewAllCameras();
                    break;
                case 4:
                    showMyWalletMenu();
                    break;
                case 5:
                    System.out.println("Thank you for using the Camera Rental App.");
                    System.exit(0);
                    break;
                default:
                    System.out.println("Invalid choice. Please try again.");
                    break;
            }
        }
    }

    private void showMyCameraMenu() {
        Scanner scanner = new Scanner(System.in);

        while (true) {
            System.out.println("\nMY CAMERA MENU:");
            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");

            System.out.print("Enter your choice(CameraMenu): ");
            int choice = scanner.nextInt();
            scanner.nextLine(); // Consume the newline character

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

```
    }
}

private 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 pricePerDay = scanner.nextDouble();
    scanner.nextLine(); // Consume the newline character

    int nextCameraId = cameraList.size() + 1;
    Camera newCamera = new Camera(nextCameraId, brand, model, pricePerDay, true);
    cameraList.add(newCamera);

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

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

    System.out.println("\nMY CAMERAS:");
    printCameraTableHeader();
    for (Camera camera : cameraList) {
        System.out.printf("%-10d%-15s%-15s%-15.2f%s%n",
                camera.getId(),
                camera.getBrand(),
                camera.getModel(),
                camera.getPricePerDay(),
                camera.isAvailable() ? "Available" : "Rented");
    }

    System.out.print("ENTER THE CAMERA ID TO REMOVE: ");
    int cameraId = scanner.nextInt();
    scanner.nextLine(); // Consume the newline character

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

    if (cameraRemoved) {
        System.out.println("CAMERA SUCCESSFULLY REMOVED FROM THE LIST.");
    } else {
        System.out.println("Invalid camera ID.");
    }
```

```
    }

    private void viewMyCameras() {
        System.out.println("\nMY CAMERAS:");
        printCameraTableHeader();
        for (Camera camera : cameraList) {
            System.out.printf("%-10d%-15s%-15s%-15.2f%s%n",
                    camera.getId(),
                    camera.getBrand(),
                    camera.getModel(),
                    camera.getPricePerDay(),
                    camera.isAvailable() ? "Available" : "Rented");
        }
    }

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

        System.out.println("\nFOLLOWING IS THE LIST OF AVAILABLE CAMERA(S):");
        printCameraTableHeader();
        for (Camera camera : cameraList) {
            if (camera.isAvailable()) {
                System.out.printf("%-10d%-15s%-15s%-15.2f%s%n",
                        camera.getId(),
                        camera.getBrand(),
                        camera.getModel(),
                        camera.getPricePerDay(),
                        "Available");
            }
        }

        System.out.print("ENTER THE CAMERA ID TO RENT: ");
        int cameraId = scanner.nextInt();
        scanner.nextLine(); // Consume the newline character

        boolean cameraRented = false;
        for (Camera camera : cameraList) {
            if (camera.getId() == cameraId && camera.isAvailable()) {
                System.out.print("ENTER THE NUMBER OF DAYS TO RENT: ");
                int numOfDays = scanner.nextInt();
                scanner.nextLine(); // Consume the newline character

                double rentalAmount = camera.getPricePerDay() * numOfDays;
                if (rentalAmount <= currentUser.getWalletBalance()) {
                    currentUser.deductFromWallet(rentalAmount);
                    camera.setAvailable(false);
                    cameraRented = true;
                    System.out.printf("CAMERA SUCCESSFULLY RENTED FOR %d DAYS. Rental
Amount: %.2f INR%n", numOfDays, rentalAmount);
                    System.out.println("Updated Wallet Balance: " +
currentUser.getWalletBalance() + " INR");
                } else {
                    System.out.println("Insufficient balance in your wallet. Unable
to rent the camera.");
                }
```

```
                break;
            }
        }

        if (!cameraRented) {
            System.out.println("Invalid camera ID or camera is already rented.");
        }
    }


    private void viewAllCameras() {
        System.out.println("\nALL CAMERAS:");
        printCameraTableHeader();
        for (Camera camera : cameraList) {
            System.out.printf("%-10d%-15s%-15s%-15.2f%s%n",
                    camera.getId(),
                    camera.getBrand(),
                    camera.getModel(),
                    camera.getPricePerDay(),
                    camera.isAvailable() ? "Available" : "Rented");
        }
    }

    private void showMyWalletMenu() {
        Scanner scanner = new Scanner(System.in);

        while (true) {
            System.out.println("\nMY WALLET MENU:");
            System.out.println("1. DEPOSIT");
            System.out.println("2. CHECK BALANCE");
            System.out.println("3. GO TO PREVIOUS MENU");

            System.out.print("Enter your choice: ");
            int choice = scanner.nextInt();
            scanner.nextLine(); // Consume the newline character

            switch (choice) {
                case 1:
                    depositToWallet();
                    break;
                case 2:
                    checkWalletBalance();
                    break;
                case 3:
                    return;
                default:
                    System.out.println("Invalid choice. Please try again.");
                    break;
            }
        }
    }

    private void depositToWallet() {
        Scanner scanner = new Scanner(System.in);
```

```
        System.out.print("ENTER THE AMOUNT TO DEPOSIT: ");
        double amount = scanner.nextDouble();
        scanner.nextLine(); // Consume the newline character

        currentUser.depositToWallet(amount);

        System.out.println("AMOUNT DEPOSITED SUCCESSFULLY.");

        while (true) {
            System.out.print("Do you want to deposit more money? (yes/no): ");
            String choice = scanner.nextLine();

            if (choice.equalsIgnoreCase("yes")) {
                System.out.print("ENTER THE AMOUNT TO DEPOSIT: ");
                amount = scanner.nextDouble();
                scanner.nextLine(); // Consume the newline character

                currentUser.depositToWallet(amount);

                System.out.println("AMOUNT DEPOSITED SUCCESSFULLY.");
            } else if (choice.equalsIgnoreCase("no")) {
                return; // Return to the wallet menu
            } else {
                System.out.println("Invalid choice. Please enter 'yes' or 'no'.");
            }
        }

    }


    private void checkWalletBalance() {
        System.out.println("WALLET BALANCE: " + currentUser.getWalletBalance() + "
INR");
    }

    private void printCameraTableHeader() {
        System.out.println("----------------------------------------------------------
-------------");
        System.out.printf("%-10s%-15s%-15s%-15s%s%n",
                "ID", "Brand", "Model", "Price/Day", "Availability");
        System.out.println("----------------------------------------------------------
-------------");
    }
}


package endproject;

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