```java
package Java;

import java.util.*;
class Camera {
 private int id;
 private String brand;
 private String model;
 private double rentAmount;
 private boolean isRented;
 public Camera(int id, String brand, String model, double rentAmount) {
 this.id = id;
 this.brand = brand;
 this.model = model;
 this.rentAmount = rentAmount;
 this.isRented = false;
 }
 public int getId() {
 return id;
 }
 public String getBrand() {
 return brand;
 }
 public String getModel() {
 return model;
 }
 public double getRentAmount() {
 return rentAmount;
 }
 public boolean isRented() {
 return isRented;
 }
 public void setRented(boolean rented) {
 isRented = rented;
 }
}
class User {
 private String username;
 private String password;
 public User(String username, String password) {
 this.username = username;
 this.password = password;
 }
 public String getUsername() {
 return username;
 }
 public String getPassword() {
 return password;
 }
}
public class CameraRentalApp {
 private static List<Camera> cameraList = new ArrayList<>();
 private static User currentUser;
 private static double walletBalance = 0;
 private static Scanner scanner = new Scanner(System.in);
 public static void main(String[] args) {
 initializeCameras();
 login();
 if (currentUser != null) {
 showMainMenu();
 }
 }
 private static void initializeCameras() {
```

```java
System.out.println("----- Camera Rental App -----");
System.out.println("Developer: Navya");
System.out.println("Welcome to mycam.io\n");
System.out.println("Enter the number of cameras to add: ");
int numCameras = scanner.nextInt();
scanner.nextLine(); // Consume the newline character
for (int i = 0; i < numCameras; i++) {
System.out.println("Enter camera ID: ");
int id = scanner.nextInt();
scanner.nextLine(); // Consume the newline character
System.out.println("Enter camera brand: ");
String brand = scanner.nextLine();
System.out.println("Enter camera model: ");
String model = scanner.nextLine();
System.out.println("Enter per-day rental amount: ");
double rentAmount = scanner.nextDouble();
scanner.nextLine(); // Consume the newline character
Camera camera = new Camera(id, brand, model, rentAmount);
cameraList.add(camera);
}
sortCameraList();
}
private static void sortCameraList() {
cameraList.sort(Comparator.comparingInt(Camera::getId));
}
private static void login() {
System.out.println("\n----- Login -----");
System.out.println("Enter username: ");
String username = scanner.nextLine();
System.out.println("Enter password: ");
String password = scanner.nextLine();
```

```
User user = new User("admin", "password"); // Change the credentials as Paper your requirement

if (username.equals(user.getUsername()) &&
password.equals(user.getPassword())) {
currentUser = user;
System.out.println("Login successful!\n");
} else {
System.out.println("Invalid credentials. Login failed!\n");
}
}
private static void showMainMenu() {
int choice;
do {
System.out.println("----- Main Menu -----");
System.out.println("1. List cameras");
System.out.println("2. Rent a camera");
System.out.println("3. Add/view wallet balance");
System.out.println("4. Exit");
System.out.println("Enter your choice: ");
choice = scanner.nextInt();
switch (choice) {
case 1:
listCameras();
break;
case 2:
rentCamera();
break;
case 3:
walletMenu();
break;
case 4:
```

```
System.out.println("Exiting the application...");
break;
default:
System.out.println("Invalid choice. Please try again.");
}
} while (choice != 4);
}
private static void listCameras() {
if (cameraList.isEmpty()) {
System.out.println("No Data Present at This Moment.");
} else {
System.out.println("----- Camera List -----");
System.out.println("ID Brand Model Rent Amount Status");
System.out.println("--------------------------------------------------------------------");

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

for (Camera camera : cameraList) {
String status = camera.isRented() ? "Rented" : "Available";
System.out.printf("%-6d%-12s%-12s%-15.2f%s%n", camera.getId(),
camera.getBrand(), camera.getModel(), camera.getRentAmount(), status);
}
System.out.println("-------------------------------------------------------------------");

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

}
}
private static void rentCamera() {
if (cameraList.isEmpty()) {
System.out.println("No cameras available for rent.");
```

```
return;
}
if (walletBalance <= 0) {
System.out.println("Insufficient balance in the wallet. Cannot rent a camera.");

return;
}
System.out.println("Enter camera ID to rent: ");
int cameraId = scanner.nextInt();
Camera selectedCamera = null;
for (Camera camera : cameraList) {
if (camera.getId() == cameraId) {
selectedCamera = camera;
break;
}
}
if (selectedCamera != null) {
if (selectedCamera.isRented()) {
System.out.println("Camera is already rented.");
} else {
System.out.println("Camera rented successfully!");
selectedCamera.setRented(true);
walletBalance -= selectedCamera.getRentAmount();
}
} else {
System.out.println("Invalid camera ID.");
}
}
private static void walletMenu() {
int choice;
do {
```

```
System.out.println("----- Wallet Menu -----");
System.out.println("1. View wallet balance");
System.out.println("2. Add amount to wallet");
System.out.println("3. Go back");
System.out.println("Enter your choice: ");
choice = scanner.nextInt();
switch (choice) {
case 1:
System.out.printf("Wallet balance: %.2f%n", walletBalance);
break;
case 2:
System.out.println("Enter amount to add: ");
double amount = scanner.nextDouble();
walletBalance += amount;
System.out.println("Amount added successfully!");
break;
case 3:
System.out.println("Going back to the main menu...");
break;
default:
System.out.println("Invalid choice. Please try again.");
}
} while (choice != 3);
}
}
```