

code: Camera Rental System

```
package Project;
```

```
class Camera {  
    private int id;  
    private String brand;  
    private String model;  
    private double rentalAmount;  
    private boolean rented;
```

```
    public Camera(int id, String brand, String model, double rentalAmount) {  
        this.id = id;  
        this.brand = brand;  
        this.model = model;  
        this.rentalAmount = rentalAmount;  
        this.rented = false;  
    }
```

```
    public int getId() {  
        return id;  
    }
```

```
    public String getBrand() {  
        return brand;  
    }
```

```
    public String getModel() {  
        return model;  
    }
```

```
    public double getRentalAmount() {  
        return rentalAmount;  
    }
```

```
    public boolean isRented() {  
        return rented;  
    }
```

```
    public void setRented(boolean rented) {  
        this.rented = rented;  
    }  
}
```

```
package Project;

public class User {
    private String username;
    private String password;

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

    public String getUsername() {
        return this.username;
    }

    public void setUsername(String username) {
        this.username = username;
    }

    public String getPassword() {
        return this.password;
    }

    public void setPassword(String password) {
        this.password = password;
    }
}
```

```
package Project;

public class Wallet {
    private double balance;

    public Wallet() {
        balance = 60000;
    }

    public double getBalance()
    {
        return balance;
    }

    public void deposit(double amount) {
        balance += amount;
    }
    public void setBalance(double amount)
    {
        balance=amount;
    }
}
```

```
package Project;
```

```

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

public class CameraRentalApp
{
    private List<Camera> cameraList;
    private Scanner scanner;
    private int cameraIdCounter;
    private List<Camera> MycameraList;
    private Wallet wallet;
    public CameraRentalApp() {
        cameraList = new ArrayList<>();
        MycameraList=new ArrayList<>();
        scanner = new Scanner(System.in);
        cameraIdCounter = 7;
        wallet=new Wallet();
    }

    public void run()
    {
        List<User> users = new ArrayList<>();
        users.add(new User("Prasad", "Prasad78"));
        users.add(new User("Kiran", "Kiran123"));
        users.add(new User("Kaveri", "Wonderful"));
        users.add(new User("seetha", "Sc567"));
        users.add(new User("Ganesh", "12345678"));
        users.add(new User("mahesh", "king1245"));
        users.add(new User("dev", "dialogue"));
        users.add(new User("Lahari", "World12"));
        users.add(new User("Reddy", "Reddy567"));
        users.add(new User("bhargav", "Man1234"));
        cameraList.add(new Camera(1,"Sony", "4k",15000));
        cameraList.add(new Camera(2,"Kodak", "panaroma",5000));
        cameraList.add(new Camera(3,"Panasonic", "UHD",6000));
        cameraList.add(new Camera(4,"Samsung", "HD",3000));
        cameraList.add(new Camera(5,"Nikon", "Z5",5000));
        cameraList.add(new Camera(6,"Canon", "Portrait",10000));
        cameraList.add(new Camera(7,"Sigma", "Uhd56",7000));
        Scanner scanner = new Scanner(System.in);
        System.out.println("+____+____+____+____+");
        System.out.println("|WELCOME TO CAMERA RENTAL APP|");
        System.out.println("+____+____+____+____+");
        System.out.println("Please login to continue");
        System.out.print("Enter your username: ");
        String username = scanner.nextLine();
        System.out.print("Enter your password: ");
        String password = scanner.nextLine();

        User currentUser = null;
        for (User user : users)
        {
            if (user.getUsername().equals(username) &&
                user.getPassword().equals(password))
            {
                currentUser = user;
            }
        }
    }
}

```

```

break;
}
}

if (currentUser == null)
{
System.out.println("Invalid username or password.");
System.out.println("Please make sure that you have entered the correct
Credentials, Try Again!");
return;
}
displayMenu();
}
public void displayMenu()
{

boolean loggedIn=true;
if(loggedIn)
{
while (true)
{
System.out.println("=== MAIN MENU ===");
System.out.println("\n1. 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 = Integer.parseInt(scanner.nextLine());
switch(choice)
{
case 1:goMyCamera();
break;
case 2:goRentCamera();
break;
case 3:goViewAllCameras();
break;
case 4:goMyWallet();
break;
case 5:System.exit(0);
break;
default:
System.out.println("Wrong choice. Please try again.");
break;
}
}
}
}

//MY WALLET
private void goMyWallet() {
Scanner scanner = new Scanner(System.in);

System.out.println("YOUR WALLET BALANCE IS INR." + wallet.getBalance());

System.out.print("DO YOU WANT TO DEPOSIT AMOUNT TO YOUR WALLET?(yes/no):
");

```

```

String depositChoice = scanner.nextLine();

if (depositChoice.equalsIgnoreCase("yes"))
{
    System.out.println("ENTER THE AMOUNT TO DEPOSIT(INR)- ");
    double depositAmount = scanner.nextDouble();

    if (depositAmount > 0)
    {
        wallet.deposit(depositAmount);
        System.out.println("Deposit successful.");
    }
    else
    {
        System.out.println("Invalid amount. Deposit failed.");
    }
}

System.out.println("YOUR WALLET BALANCE UPDATED SUCCESSFULLY.CURRENT WALLET
BALANCE-INR " + wallet.getBalance());
}
}
//MY CAMERA
private void goMyCamera() {
    // TODO Auto-generated method stub
    Scanner scanner=new Scanner(System.in);
    while(true)
    {
        System.out.println("=== SUB MENU ===");
        System.out.println("\n1. 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: ");

        int choice = Integer.parseInt(scanner.nextLine());
        switch(choice)
        {
            case 1:goAddCamera();
            break;
            case 2:goRemoveCamera();
            break;
            case 3:goViewMyCameras();
            break;
            case 4:break;
            default:
                System.out.println("Wrong choice. Please try again.");
                break;
        }
        if(choice==4)
        {
            return;
        }
    }
}

//VIEW MY CAMERAS
private void goViewMyCameras() {

```

```

// TODO Auto-generated method stub
if (MycameraList.isEmpty())
{
System.out.println("YOU HAVE NO CAMERAS.");
}
else {
System.out.println("Cameras List:");
System.out.printf("%-5s %-10s %-10s %-15s %-10s\n", "ID", "Brand", "Model",
"Rental Amount", "Status");
for (Camera camera : MycameraList)
{
String status = camera.isRented() ? "Rented" : "Available";
System.out.printf("%-5d %-10s %-10s $%-15.2f %-10s\n", camera.getId(),
camera.getBrand(), camera.getModel(),
camera.getRentalAmount(), status);
}
}
}

//REMOVE
private void goRemoveCamera() {
Scanner scanner=new Scanner(System.in);
// TODO Auto-generated method stub
System.out.print("ENTER THE CAMERA ID TO REMOVE");
int id=scanner.nextInt();
Iterator<Camera> iterator = cameralist.iterator();
while (iterator.hasNext()) {
Camera camera = iterator.next();
if (camera.getId() == id) {
cameralist.remove(camera);
cameraIdCounter--;
System.out.println("CAMERA SUCCESSFULLY REMOVED FROM THE LIST");
return;
}
}
System.out.println("CAMERA NOT FOUND IN THE LIST");
}

//ADD
private void goAddCamera() {
// TODO Auto-generated method stub
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): ");
int rentalAmount = Integer.parseInt(scanner.nextLine());
Camera camera = new Camera(cameraIdCounter++, brand, model, rentalAmount);
cameralist.add(camera);
System.out.println("YOUR CAMERA HAS BEEN SUCCESSFULLY ADDED TO THE LIST.");
}

//VIEW ALL CAMERAS
private void goViewAllCameras() {
// TODO Auto-generated method stub
if (cameralist.isEmpty()) {
System.out.println("YOU HAVE NO CAMERAS.");
}
else {

```

```

System.out.println("Cameras List:");
System.out.printf("%-5s %-10s %-10s %-15s %-10s\n", "ID", "Brand", "Model",
"Rental Amount", "Status");
for (Camera camera : cameraList)
{
String status = camera.isRented() ? "Rented" : "Available";
System.out.printf("%-5d %-10s %-10s $%-15.2f %-10s\n", camera.getId(),
camera.getBrand(), camera.getModel(),
camera.getRentalAmount(), status);
}
}
}

```

```

//RENT A CAMERA
private void goRentCamera()
{
// TODO Auto-generated method stub
System.out.println("FOLLOWING IS THE LIST OF AVAILABLE CAMERA(S)");
System.out.printf("%-5s %-10s %-10s %-15s %-10s\n", "ID", "Brand", "Model",
"Rental Amount", "Status");
for (Camera camera : cameraList)
{
if (!camera.isRented()) {
String status = "Available";
System.out.printf("%-5d %-10s %-10s $%-15.2f %-10s\n", camera.getId(),
camera.getBrand(), camera.getModel(),
camera.getRentalAmount(), status);
}
}
System.out.println("ENTER THE CAMERA ID YOU WANT TO RENT-");
Scanner scanner=new Scanner(System.in);
int id=scanner.nextInt();
for (Camera camera : cameraList)
{
if(camera.getId()==id )
{
if(camera.getRentalAmount()<=wallet.getBalance())
{
camera.setRented(true);
MycameraList.add(camera);
wallet.setBalance(wallet.getBalance()-camera.getRentalAmount());
System.out.println("YOUR TRANSACTION FOR CAMERA-"+camera.getBrand()+
"+camera.getModel()+" with rent INR."+camera.getRentalAmount()+" HAS
SUCCESSFULLY COMPLETED");
}
else
{
System.out.println("ERROR:TRANSACTION FAILED DUE TO INSUFFICIENT WALLET
BALANCE. PLEASE DEPOSIT THE AMOUNT TO YOUR WALLET");
}
}
}
}
//MAIN METHOD
public static void main(String[] args) {
CameraRentalApp app = new CameraRentalApp();
app.run();
}

```

