**CAMERA RENTAL APPLICATION SOURCE CODE**

1. **Welcome Page Code**

**package** main;

**public** **class** CamaraRentalApplication {

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

System.***out***.println();

System.***out***.println("|======================================

==============================|");

System.***out***.println("|WELCOME TO CAMERA

RENTAL APP |");

System.***out***.println("|================================

====================================|");

Methods camara = **new** Methods();

**if**(camara.login()) {

camara.DisplayMenu();

}**else** {

System.***out***.println("Enter Username and password");

}

}

}

1. **Application Methods**

package main;

import java.util.InputMismatchException;

import java.util.NoSuchElementException;

import java.util.Scanner;

import camara.Camera;

import exception.InSufficientAmount;

import wallet.Wallet;

public class Methods {

private Camera cam = new Camera();

private Wallet wall = new Wallet();

/\*Login Method\*/

public Boolean login() {

Scanner Input = new Scanner(System.in);

System.out.println("Enter The Username ");

String username = Input.next();

System.out.println("Enter The Password");

String password =Input.next();

if(username !=null && password != null) {

return true;

}else {

System.out.println("Fail To Login");

return false;

}

}

/\* Display the methods \*/

public void DisplayMenu() {

String[] optionlist = {"1. My Camera",

"2. Rent A Camera",

"3. View All Camera",

"4. My Wallet",

"5. Exit"

};

for(int i=0; i<optionlist.length; i++) {

System.out.println(optionlist[i]);

//display the list of options

}

int[] arr = {1,2,3,4,5};

Scanner Input = new Scanner(System.in);

System.out.println("Select The Option");

int option = 0;

try {

option= Input.nextInt();

}catch(InputMismatchException e) {

System.err.println("Error Invalid Input.");

}catch(NoSuchElementException e) {

System.err.println("Error Invalid Input.");

}

//Taking input for user options

for(int j=1; j<arr.length; j++) {

if(j==option) {

switch(option) {

case 1:

String suboption[] = {"1. Add", "2. Remove",

"3. View My Camera", "4. Go To Previous menu"

};

for(int i=0; i< suboption.length; i++) {

System.out.println(suboption[i]);

}

System.out.println("Enter The option");

int Secoundoption = 0;

try {

Secoundoption= Input.nextInt();

}catch(InputMismatchException e) {

System.err.println("Error Invalid Input");

}

//taking value from the user

int[] aOption = {1,2,3,4};

for(int a =1; a< aOption.length; a++) {

if(a == Secoundoption) {

switch(Secoundoption) {

case 1:

System.out.println("Enter The Id");

int id = Input.nextInt();

System.out.println("Enter The Brand");

String brand =Input.next();

System.out.println("Enter The Model");

String Model = Input.next();

System.out.println("Enter The Prise");

int prise = Input.nextInt();

String status = "Avaliable";

cam.addCamera(id, brand, Model, prise, status);

DisplayMenu();

break;

case 2:

cam.ViewCameraList();

System.out.println("Enter The Camera Id To Remove");

int Id = Input.nextInt();

cam.RemoveCamara(Id);

cam.ViewCameraList();

DisplayMenu();

break;

case 3:

cam.ViewCameraList();

DisplayMenu();

break;

case 4:

DisplayMenu();

break;

}

}

}

DisplayMenu();

break;

case 2:

cam.ViewCameraList();

System.out.print("Enter The Camera Id ");

int id = Input.nextInt();

int amount =cam.rentcamera(id);

try {

if(amount != -1) {

if(wall.reduceamount(amount)) {

cam.setStatus(id);;

}

}else {

System.out.println("The Camera already in rent");

}

} catch (InSufficientAmount e) {

System.out.println(e.getMessage());

}

cam.ViewCameraList();

DisplayMenu();

break;

case 3:

cam.ViewCameraList();

DisplayMenu();

break;

case 4:

String[] wallet = {"1. View Balance", "2. Add Balance", "3. Main Menu"};

for(int a=0; a<wallet.length; a++) {

System.out.println(wallet[a]);

}

int n[] = {1,2,3};

System.out.println("Enter The Option");

int walletoption = 0;

try {

walletoption = Input.nextInt();

}catch(InputMismatchException e) {

System.err.println("Error Input");

}

for(int a=1; a< n.length; a++) {

if(walletoption == a) {

switch(walletoption) {

case 1:

System.out.println("Balance amount"+ wall.DisplayAmount());

break;

case 2:

System.out.println("Enter The Balence amout ");

try {

int addamount = Input.nextInt();

wall.addamount(addamount);

}catch(InputMismatchException e) {

System.err.println("Error Input");

}

break;

case 3:

DisplayMenu();

break;

}

}

}

DisplayMenu();

break;

case 5:

System.out.println("Exit");

return;

}

}

}

Input.close();

}

}

1. **Camera Data And Methods Of Printing, Add And Deleting**

package camara;

import java.util.ArrayList;

import java.util.Collections;

import java.util.Comparator;

class CameraDetails{

private int id;

private String brandName;

private String model;

private int prise;

private String status;

//constructer;

public CameraDetails(int id, String brandName, String model, int prise,

String status)

{

this.id = id;

this.brandName = brandName;

this.model = model;

this.prise = prise;

this.status = status;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getBrandName() {

return brandName;

}

public void setBrandName(String brandName) {

this.brandName = brandName;

}

public String getModel() {

return model;

}

public void setModel(String model) {

this.model = model;

}

public int getPrise() {

return prise;

}

public void setPrise(int prise) {

this.prise = prise;

}

public String getStatus() {

return status;

}

public void setStatus(String status) {

this.status = status;

}

}

public class Camera {

public Camera() {

// CameraList.add(new CameraDetails(123, "sony", "dslr678", 23000,

"Avaliable"));

            // CameraList.add(new CameraDetails(125, "vivo", "dslr78", 53000,

"Avaliable"));

            // CameraList.add(new CameraDetails(1, "oppo", "dslr68", 23000, "Avaliable"));

            // CameraList.add(new CameraDetails(2, "mi", "dslr6", 13000, "Avaliable"));

            // CameraList.add(new CameraDetails(120, "Apple", "dslr000", 30020,

"Avaliable"));

            // CameraList.add(new CameraDetails(128, "realme", "dslroo", 300440,

"Avaliable"));

            // CameraList.add(new CameraDetails(121, "samsung", "dslr7834", 34000,

"Avaliable"));

            // CameraList.add(new CameraDetails(12, "nokhia", "dslr10101", 30000,

"Avaliable"));

            // CameraList.add(new CameraDetails(90, "color", "dslr420", 40003,

"Avaliable"));

            // CameraList.add(new CameraDetails(45, "milkey", "dslr640", 70030,

"Avaliable"));

            /\* uncomment code while using the application it is abour testing  \*/

}

public ArrayList<CameraDetails> CameraList = new ArrayList<>();

//Adding Camera To the list

public void addCamera(int id, String brandName, String model, int prise,

String status)

{

if(id > 0 && brandName != null && model != null && prise > 0 && status

!= null )

{

CameraList.add(new CameraDetails(id, brandName, model, prise,

status));

System.out.println(" Camera Has Been Sucessfully Added To The

List");

}else {

System.out.println("Error Incomplete Filds");

}

}

//printing the camera to the list

public void ViewCameraList() {

//sorting

Collections.sort(CameraList,

Comparator.comparingInt(CameraDetails::getId));

System.out.println();

System.out.println("|=======================================================

=============|");

System.out.println("ID Brand Model Prise

Status");

System.out.println("|=======================================================

=============|");

//Printing the list

for(int i=0; i < CameraList.size(); i++) {

System.out.println(CameraList.get(i).getId()+"

"+CameraList.get(i).getBrandName()+"

"+CameraList.get(i).getModel()

+" "+

CameraList.get(i).getPrise()+”

"+CameraList.get(i).getStatus());

}

System.out.println();

System.out.println("|========================================

=============================|");

System.out.println();

}

public void RemoveCamara(int id) {

if(CameraList == null) {

System.out.println("No Data Present at This Moment.");

return;

}

int index =-1;

for(int i=0; i < CameraList.size(); i++) {

if(id == CameraList.get(i).getId()) {

CameraDetails delete = CameraList.get(i);

CameraList.remove(delete);

index = 1;

break;

}

}

if(index ==1) {

System.out.println("Camera Successfully Removed From The List");

}else {

System.out.println("No Camera Found In The List With Id "+id);

}

}

public int rentcamera(int id) {

int rentamount =-1;

int index = -1;

for(int i=0; i < CameraList.size(); i++) {

if(id == CameraList.get(i).getId()) {

if(CameraList.get(i).getStatus() == "Rented") {

return -1;

}

rentamount = CameraList.get(i).getPrise();

index =1;

break;

}

}

if(index ==1) {

return rentamount;

}else {

return rentamount;

}

}

public void setStatus(int id) {

for(int i=0; i < CameraList.size(); i++) {

if(id == CameraList.get(i).getId()) {

CameraList.get(i).setStatus("Rented");

System.out.printf("YOUR TRANSACTION FOR CAMERA -%s %s with rent INR %s HAS SUCCESSFULLY COMPLETED FOR ONE DAY",CameraList.get(id).getBrandName(),CameraList.get(id).getModel(),CameraList.get(id).getPrise()+"\n");

break;

}

}

}

}

1. **Wallet Class and Methods Adding Amount Depicting Amount**

**package** wallet;

**import** exception.InSufficientAmount;

**public** **class** Wallet {

//amount

**private** **int** amount = 50000;

//adding amount to wallet

**public** **void** addamount(**int** a) {

amount +=a;

System.***out***.println("Amount Added To Your Wallet "+amount);

}

//reduce amount from the Wallet

**public** **boolean** reduceamount(**int** a) **throws** InSufficientAmount {

**if**(a>amount) {

**throw** **new** InSufficientAmount("In sufficient Amount In

Your Wallet "+amount);

}**else** {

amount -=a;

System.***out***.println("Total Amount "+amount);

**return** **true**;

}

}

**public** **int** DisplayAmount() {

**return** amount;

}

}

1. **Insufficient Amount Exception class**

**package** exception;

**public** **class** InSufficientAmount **extends** Exception{

/\*\*

\*

\*/

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**public** InSufficientAmount(String e) {

**super**(e);

}

}