Camera Rental Source Code

import java.util.\*;

public class Camerarental {

public static double wallet;

public static int cameraid;

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

System.out.println("LOG IN TO THE CAMERA RENTAL APP");

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

int exit1 = 2;

wallet = 1500.00;

cameraid=22;

for (; exit1 >= 0; exit1--) {

System.out.print("User ID: ");

String userid = sc.nextLine();

System.out.print("Password: ");

String password = sc.nextLine();

if (userid.equalsIgnoreCase("admin") &&

password.equalsIgnoreCase("admin123")) {

System.out.println("----------------\nLogIn Successful\n----------------");

break;

} else {

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

System.out.println("Incorrect ID or Password," + exit1 + " more attempt remaining");

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

}

}

if (exit1 > -1) {

ArrayList<Camera> allCameraList = new ArrayList<>();

allCameraList.add(new Camera(5, "canon", "EOS d5300", 500.00, "available"));

allCameraList.add(new Camera(6, "Nikon", "coolpix 130sx", 200.00, "available"));

allCameraList.add(new Camera(9, "canon", "eos 200D", 1800.00, "available"));

allCameraList.add(new Camera(7, "sony", "alpha a6600", 2800.00, "available"));

allCameraList.add(new Camera(15, "nikon", "d5600", 1600.00, "available"));

allCameraList.add(new Camera(16, "sony", "alpha a6400", 2000.00, "available"));

allCameraList.add(new Camera(17, "fujifilm", "x-t20", 1700.00, "available"));

allCameraList.add(new Camera(11, "nikon", "d7500", 2500.00, "available"));

allCameraList.add(new Camera(19, "panasonic", "lumix GH5", 2500.00, "available"));

allCameraList.add(new Camera(20, "canon", "eos 90D", 2200.00, "available"));

allCameraList.add(new Camera(2, "Nikon", "D850", 600.00, "available"));

allCameraList.add(new Camera(3, "Sony", "Alpha A7 III", 700.00, "available"));

allCameraList.add(new Camera(21, "nikon", "z50", 1800.00, "available"));

ArrayList<Camera> cameraList = new ArrayList<>();

// Add some initial cameras to the list

cameraList.add(new Camera(1, "Canon", "EOS 5D Mark IV", 1000.50, "available"));

cameraList.add(new Camera(4, "red", "Hydrogen 1", 2000.00, "available"));

cameraList.add(new Camera(8, "nikon", "d3500", 1500.00, "available"));

cameraList.add(new Camera(10, "sony", "alpha a6000", 1200.00, "available"));

cameraList.add(new Camera(12, "sony", "alpha a7 III", 3000.00, "available"));

cameraList.add(new Camera(14, "canon", "eos M50", 1400.00, "available"));

cameraList.add(new Camera(18, "olympus", "om-d e-m10 Mark III", 1500.00, "available"));

allCameraList.addAll(cameraList);

allCameraList.sort(Comparator.comparingInt(c -> c.id));//sort the arraylist

while (true) {

try{

Scanner scanner=new Scanner(System.in);

System.out.println("1: MY CAMERAS\n2: RENT A CAMERA\n3: VIEW ALL CAMERAS\n4: MY WALLET\n5: EXIT");

int choice = scanner.nextInt();

scanner.nextLine();

//my camera portal

if (choice == 1) {

while (true) {

try {

Scanner sc1=new Scanner(System.in);

System.out.println("1:ADD\n2:REMOVE\n3:VIEW MY CAMERAS\n4:GO TO PREVIOUS MENU");

int innerChoiceOf1 = sc1.nextInt();

if (innerChoiceOf1 == 4) {

break;

} else if (innerChoiceOf1 == 1) {

System.out.println("Enter the details for the new camera:");

int id = cameraid;

cameraid++;

sc1.nextLine();

System.out.print("Brand: ");

String brand = sc1.nextLine();

System.out.print("Model: ");

String model = sc1.nextLine();

System.out.print("Rent per Day: ");

double rentPerDay = sc1.nextDouble();

sc1.nextLine();

String status = "available";

Camera newCamera = new Camera(id, brand, model, rentPerDay, status);

cameraList.add(newCamera);

allCameraList.add(newCamera);

displayCameraTable(cameraList);

} else if (innerChoiceOf1 == 2) {

displayCameraTable(cameraList);

System.out.println("Enter the ID of the camera you want to remove:");

int idToRemove = sc1.nextInt();

sc1.nextLine();

boolean removed = false;

for (Camera camera : cameraList) {

if (camera.id == idToRemove) {

cameraList.remove(camera);

allCameraList.remove(camera);

removed = true;

break;

}

}

if (removed) {

System.out.println("Camera with ID " + idToRemove + " is removed");

// Display the updated camera table

displayCameraTable(cameraList);

} else {

System.out.println("Camera with ID " + idToRemove + " not found");

}

} else if (innerChoiceOf1 == 3) {

displayCameraTable(cameraList);

} else

System.out.println("PLEASE ENTER A CORRECT CHOICE");

}

catch (Exception e){

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

System.out.println("OOPS! Something went wrong, Try again");

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

}

}

//rent a camera

} else if (choice == 2) {

displayCameraTableOnlyAvailable(allCameraList);

boolean checkavailability ;

int rentedcheck ;

do {

checkavailability = true;

rentedcheck = 0;

System.out.println("Enter ID of the camera you need in rent");

int inputRentID = scanner.nextInt();

scanner.nextLine();

for (Camera camera : allCameraList) {

if (camera.id == inputRentID) {

if (wallet >= camera.rentPerDay) {

if (camera.status.equalsIgnoreCase("available")) {

camera.status = "rented";

wallet = wallet - camera.rentPerDay;

System.out.println("Your wallet is deducted for INR "+camera.rentPerDay+

"\nThe remaining balance is "+wallet);

rentedcheck = 1;

break;

} else {

System.out.println("THIS CAMERA IS ALREADY ON RENT, choose another one");

checkavailability = false;

}

} else {

System.out.println("Your wallet balance is Insufficient");

System.out.println(walletTopUp());

checkavailability= false;

}

}

else {

rentedcheck=2;

}

}

if (rentedcheck == 1)

break;

} while (!checkavailability);

if (rentedcheck == 1) {

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

System.out.println("Camera rented successfully");

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

} else if(rentedcheck == 2){

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

System.out.println("Camera ID not found in the list");

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

}

//view all cameras

} else if (choice == 3) {

displayCameraTable(allCameraList);

} else if (choice == 4) {

System.out.println("Your wallet balance is INR " + wallet);

System.out.println(walletTopUp());

} else if (choice == 5) {

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

System.out.println("THANK YOU SEE YOU AGAIN SOON");

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

break;

}

}

catch (Exception e){

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

System.out.println("OOPS! Something went wrong, Try again");

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

}

}

}

else {

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

System.out.println("Try running the program again after some time");

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

}

}

public static String walletTopUp() {

while (true) {

try {

Scanner scanner=new Scanner(System.in);

System.out.print("Do you want to Topup your wallet balance? (1:yes/2:no)");

int topupchoice = scanner.nextInt();

if (topupchoice == 1) {

System.out.print("\nEnter the TOPUP Amount");

Camerarental.wallet = Camerarental.wallet + scanner.nextInt();

return "Wallet balance updated successfully, BALANCE: INR" + Camerarental.wallet;

} else if (topupchoice == 2) {

return "Transaction canceled";

}

} catch (Exception e) {

System.out.println("Invalid input, Try again");

}

}

}

public static void displayCameraTable (ArrayList<Camera> cameraList) {

System.out.println("Camera Table:");

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

System.out.printf("%-10s %-15s %-20s %-18s %-12s%n", "Camera ID", "Brand", "Model", "Rent per Day", "Status");

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

for (Camera camera : cameraList) {

System.out.printf("%-10d %-15s %-20sINR %-14.2f %-15s%n", camera.id, camera.brand,

camera.model, camera.rentPerDay, camera.status);

}

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

}

public static void displayCameraTableOnlyAvailable (ArrayList<Camera> cameraList) {

System.out.println("Camera Table:");

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

System.out.printf("%-10s %-15s %-20s %-18s %-12s%n", "Camera ID", "Brand", "Model", "Rent per Day", "Status");

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

for (Camera camera : cameraList) {

if(camera.status.equalsIgnoreCase("available")) {

System.out.printf("%-10d %-15s %-20sINR %-14.2f %-15s%n", camera.id, camera.brand,

camera.model, camera.rentPerDay, camera.status);

}

}

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

}

static class Camera {

int id;

String brand;

String model;

double rentPerDay;

String status;

public Camera(int id, String brand, String model, double rentPerDay, String status) {

this.id = id;

this.brand = brand;

this.model = model;

this.rentPerDay = rentPerDay;

this.status=status;

}

}

}