

A Mini Project Report
on
Online Shopping Management System

Course: Object Oriented Programming
Sem: III **Sec:** CSE-B

By
P. Ashish
1602-19-733-065

B. Visista
1602-19-733-121



Department of Computer Science & Engineering
Vasavi College of Engineering (Autonomous)

Ibrahimbagh, Hyderabad-31

2020

Acknowledgment

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

I would like to express my gratitude towards M.S.V Sashi Kumar sir for his kind co-operation and encouragement which help me in completion of this project.

My thanks and appreciations also go to my classmate in developing the project and people who have willingly helped me out with their abilities.

TABLE OF CONTENTS

S.NO	CONTENT	PAGE.NO
1.	Abstract	4
2.	Introduction	5
3.	Design-Strategy, Algorithm	6
4.	Code/Implementation	7-18
5.	Outputs/Results	19-20
6.	Test cases	21
7.	Conclusion	22
8.	References	23

ABSTRACT

OBJECTIVE:

The main objective of this code is to make your way of shopping easier. It helps the costumer to shop without any worries because whatever he/she shopped will be saved in their respective account.

Working:

In this code we can do all the things we normally do while shopping. It keeps a record of shopping done by the customer.

For example,

We have a particular budget to shop and we need to calculate the price of all the stuff in the cart, for this, this code is useful when your shopping is over your total price will be shown, if it is over your respective budget you can modify or delete the items in the cart. And also, all your items will be saved in your cart in your respective account so that even if you get logged out by mistakenly your shopping will be saved if you log in again.

Concepts used: Threads, Exceptions, Collections, OOP concepts, i/o, string

Language used: Java

INTRODUCTION

Are you guys being forgetful? Are you guys being unable to make time from your daily schedule for shopping? Are you doing a budget-based shopping?

No worries, Here come a platform where you can shop tension free, within your budget. Its time saving too and also you can shop when ever you like whenever you remember a new item to add you can add anytime using your credentials too. Your cart won't be disappeared like that. Its safe and no can see your cart too.

Problem Statement:

The customers will be able to implement all kinds of operations i.e., adding, displaying, updating, removing items by themselves. And with also login and sign-up option.

Objective:

The main objective is to save the time of the customers and to shop as they like to the fullest without having a regret of forgetting.

Description:

- First, the customer will be shown login/sign up options.
- Next if we are already a member, we can login or else we can sign up.
- We will be showing the items available in the mall, we can add the items we can like.
- After adding if you want the total price, if we choose the option, we will get the price.
- If the price exceeds the budget, we can use modify or delete option to modify the items in the cart according to our budget.
- After all the shopping is done, we can pay using any online method.
- Your shopping history will be saved in your account.

DESIGN

4.1 Strategy:

Firstly, we thought of all sample cases and everyday applications in terms of shopping and added everything to our project. Dynamic programming is used in order to maximize the solution. Optimization over plain recursion is used.

The idea is to simply store the results of sub problems, so that we do not have to re-compute them when needed later. This simple optimization reduces time complexities from exponential to polynomial.

Steps:

Step 1: We split the project code into different methods and classes, coded them accordingly

Step 2: We use all the concepts known to us to make the code more optimised and productive

Step 3: Memorize if there are overlapping subproblems, memoize

4.2 Algorithm:

```
Package cart.*;
Class cartoperations{
//all the items available and their costs
Void addTocart
//adds selected items to cart
Void coupons
//coupon amount is subtracted from final bill
Void removeFromcart
//removes items from cart
Void billpayment
//displays final bill amount and payment mode
Void specialgifts
//special gifts for bills exceeding a limit
}}
```

```
Package onlineshopping.*;
Class customer{
//adds customers details to files and items along with their ids and costs, discounts and
availability
}
Class onlineshopping{
//user can sign in or login here, it coordinates all the methods and exceptions available in
other packages ,uses threads and i/o .
}
Package exceptions.*;
// outofstock
// expired
// itemdamaged
//cancelledorder
// returnProduct
```

IMPLEMENTATION

```
package cartoperations;
import java.util.ArrayList;
import java.util.*;

interface offersAndDeals{
    abstract void specialGifts();
}

public class cart implements offersAndDeals{
    List<String> items = Arrays.asList("kurthi","westernwear","denims","ethnic
wear","nightwear","denims","formals","casuals","nightwear","regulars","ethnic","nightwea
r","cleansers","masks","tablets","dettol","insulin","veggies","cereals","edible
oil","lyzol","surf","shampoo","deodrant","sandals","wedges","sneakers","heels","slippers","
boat shoes","slippers","loafers","espadrilles","serum","matte","face
wash","sunscreen","face mask","laptop","refrigerator","phone","ac");
    List<Integer> rates =
    Arrays.asList(750,800,2000,5000,600,1500,1000,800,500,600,1000,400,250,300,100,100,35
0,200,150,150,140,300,325,325,2500,2800,200,200,600,1500,600,1000,1000,300,850,250,2
00,120,60000,80000,42000,40000);
    ArrayList<String> itemList = new ArrayList<String>();
    List<Integer> offers = Arrays.asList(2,5,15,19,21,25,32,34,36,39,41);
    public int bill = 0;
    Scanner sc = new Scanner(System.in);

    public void addToCart(ArrayList orderID){
        for(int i=0;i < orderID.size();i++){
            int x= (int)orderID.get(i) - 1;
            if(offers.contains(x)){
                this.bill += rates.get(x)*0.8;
            }else{
                this.bill +=rates.get(x);
            }
            itemList.add(items.get(x));
        }
    }

    public void coupons(int b){
        Scanner sc1 = new Scanner(System.in);
        System.out.println("do you have a coupon??\n1.yes\n2.no");
        int x=sc1.nextInt();
        switch(x){
            case 1:System.out.println("Enter your coupon value");
                int discount = sc1.nextInt();
                this.bill -=discount;
                break;
            case 2:System.out.println("Pay Rs.1500/- to get one. Monthly 20/- will credited to
your coupon. Happy Shopping!!");
                break;
        }
    }
}
```

```

    }
}
public List items(){
    return itemList;
}
public void removeFromCart(){
    System.out.println("Enter id of item to be removed from cart");
    int x= sc.nextInt()-1;
    itemList.remove(items.get(x));
    this.bill -= rates.get(x);
    System.out.println("\nYour current bill amount is: "+bill);
}
public void display(){
    System.out.println("\n\nItems in the cart are:");
    for(String i:itemList){
        System.out.println((i));
    }
    System.out.println("Your bill is "+bill+"\n\n");
}
public int billPayment(){
    System.out.println("\nYour total bill amount is: "+bill);
    System.out.println("Choose your mode of payment\n1.pay on delivery\n2.debit
card\n3.Google Pay");
    int x= sc.nextInt();
    switch(x){
        case 1: System.out.println("You chose Cash On Delivery(COD). VISIT AGAIN :)");
        break;
        case 2: System.out.println("Input your card credentials here");
        break;
        case 3: System.out.println("Google Pay your amount to 9347687154. VISIT AGAIN :)");
        break;
    }return bill;
}
public void specialGifts(){
    if(bill > 15000){
        System.out.println("Since your bill crossed 15,000/-, your special gift is ---'ASUS LAPTOP
i3 8th gen'");
        System.out.println("collect it during delivery. WITHOUT FAIL. VISIT AGAIN!!!");}
    else if(bill>6500){
        System.out.println("Since your bill crossed 6,500/-, your special gift is ---'WASHING
MACHINE'");
        System.out.println("collect it during delivery. WITHOUT FAIL. VISIT AGAIN!!!");}
    else if(bill > 5000){
        System.out.println("Since your bill crossed 5000/-, your special gift is ---'MICROWAVE
OVEN'");
        System.out.println("collect it during delivery. WITHOUT FAIL. VISIT AGAIN!!!");}
    else if(bill > 4000){

```



```

        System.out.println("Since your bill crossed 4000/-, your special gift is ---'PRESSURE
COOKER'");
        System.out.println("collect it during delivery. WITHOUT FAIL. VISIT AGAIN!!!");}
    }
    public int bill(){
        return bill;
    }
}
//
package exceptions;

public class cancelException extends Exception{
    public String exceptionDescription;
    public cancelException(String description){
        super(description);
        exceptionDescription = description;
    }
    public String toString(){
        return exceptionDescription;
    }
}
//
package exceptions;

public class expiredException extends Exception{
    public String exceptionDescription;
    public expiredException(String description){
        super(description);
        exceptionDescription = description;
    }
    public String toString(){
        return exceptionDescription;
    }
}
//
package exceptions;

public class itemDamagedException extends Exception{
    public String exceptionDescription;
    public itemDamagedException(String description){
        super(description);
        exceptionDescription = description;
    }
    public String toString(){
        return exceptionDescription;
    }
}

```

```

}
//
package exceptions;

public class outOfStockException extends Exception{
    public String exceptionDescription;
    public outOfStockException(String description){
        super(description);
        exceptionDescription = description;
    }
    public String toString(){
        return exceptionDescription;
    }
}
//
package exceptions;

public class returnException extends Exception {
    public String exceptionDescription;
    public returnException(String description){
        super(description);
        exceptionDescription = description;
    }
    public String toString(){
        return exceptionDescription;
    }
}
//
package onlineexceptions;
import onlineshopping.*;
import exceptions.*;
public class cancelorder extends onlineShopping{
    public void exceptions()
    {
        try {
            throw new outOfStockException("I want to cancel my order");
        }
        catch(outOfStockException e) {
            System.out.println(e);
            System.out.println("Send order code to our mail. thank you. visit again");
        }
    }
}
//
package onlineexceptions;
import java.util.Scanner;
import onlineshopping.onlineShopping;

```

```

import exceptions.*;

public class expired extends onlineShopping{

    public void exceptions()
    {
        System.out.println("Did you receive an \n1.expired good\n2.item with no
expiry date");
        Scanner sc= new Scanner(System.in);
        int exc = sc.nextInt();
        switch(exc)
        {
            case 1 :
                try {
                    throw new expiredException("Product delivered is
expired");
                }
                catch(expiredException e) {
                    String response = e.toString();
                    System.out.println(e);
                    if("Product delivered is expired ".equals(response)) {
                        System.out.println("Sorry for our Negligence,
We will send you a new product along with 45% cashback");
                    }
                }
                break;
            case 2 :
                try {
                    throw new expiredException("Product delivered has no expiry date
over it");
                }
                catch(expiredException e) {
                    String response = e.toString();
                    System.out.println(e);
                    if("Product delivered has no expiry date over
it".equals(response))
                        System.out.println("Sorry for our Negligence, We will
send you the expiry date of the product, please send us the item code");
                    break;
                }
        }
    }
}

//
package onlineexceptions;
import java.util.*;
import onlineshopping.*;
import exceptions.*;
public class itemDamaged extends onlineShopping{
    public void exceptions()

```

```

        {
            System.out.println("Did you recieve an \n1.Product delivered is damaged\n2.Product
delivered is not what I have ordered");
            Scanner sc= new Scanner(System.in);
            int exc = sc.nextInt();
            switch(exc)
            {
            case 1 :
                try {
                    throw new itemDamagedException("Product delivered is damaged");
                }
                catch(itemDamagedException e) {
                    String response = e.toString();
                    System.out.println(e);
                    if("Product delivered is damaged".equals(response)) {
                        System.out.println("Sorry for our Negligence, We will send you a new product
along with 30% cashback");
                    }
                }
                break;
            case 2 :
                try {
                    throw new itemDamagedException("Product delivered is not what I have ordered");
                }
                catch(itemDamagedException e) {
                    System.out.println(e);
                    String response = e.toString();
                    System.out.println(e);
                    if("Product delivered is not what I have ordered".equals(response))
                        System.out.println("Sorry for our Negligence, We will send your desired product
within 5-6 business days, return the product you received");
                }
                break;
            }
        }
    }
}

//
package onlineexceptions;
import onlineshopping.*;
import exceptions.*;
public class outOfStock extends onlineShopping{
    public void exceptions()
    {
        try {
            throw new outOfStockException("Your desired product is out of stock at present");
        }
        catch(outOfStockException e) {
            System.out.println(e);
            System.out.println("Sorry for your Inconvinience, We will update the stock within
2-3 weeks. Stay tuned");
        }
    }
}

```

```

        }}}
//
package onlineexceptions;
import onlineshopping.*;
import exceptions.*;
public class returnProduct extends onlineShopping{
    public void exceptions()
    {
        try {
            throw new outOfStockException("I want to return my Product received");
        }
        catch(outOfStockException e) {
            System.out.println(e);
            System.out.println("Sorry for your Inconvinience,Send us an email of problem
faced by you ,we will resolve it and then you may return");
        }
    }
}
//
package onlineshopping;
import java.util.*;
import onlineexceptions.outOfStock;

import java.io.FileOutputStream;
public class customer {
    public String name,loginID,password;
    public long phonenumber;
    public void userCredentials(List itemList,int bill){
        Scanner sc1 = new Scanner(System.in);
        System.out.println("enter your name, phone number, loginID and password");
        name = sc1.nextLine();
        phonenumber = sc1.nextLong();
        loginID = sc1.next();
        password = sc1.next();
        saveData(itemList,bill);
    }
    public void saveData(List itemList,int bill){
        try{
            FileOutputStream f1 = new FileOutputStream("client.txt");
            String entry = "name: "+name +" \nuserID: "+ loginID+" \npassword: " + password+"
\nphone number: "+ phonenumber + " \ncart items: "+itemList+" \nBill amount: "+bill;
            f1.write(entry.getBytes());
            f1.close();
            System.out.println("Your info is stored successfully");
        }catch(Exception e){}
    }
    public ArrayList items(){

```

```

        outOfStock o1 = new outOfStock();
        Scanner sc = new Scanner(System.in);
        ArrayList<Integer> list1 = new ArrayList<Integer>();
        int category,x,n=0;
        while(n != -1){
            System.out.println("Choose the
category\n1.fashion\n2.medical\n3.pantry\n4.footwear\n5.beauty and skin\n6.electrical
appliances\n");
            category = sc.nextInt();
            switch(category){
                case 1: System.out.println("choose \n1.women\n2.men\n3.children");
                x = sc.nextInt();
                switch(x){
                    case 1: System.out.println("Today's deal: We have 20% off on denims");
                        System.out.println("choose your items\n1.kurthi - 750/-\n2.western tops -
800/-\n3.denims - 2000/-\n4.ethnic wear - 5000/-\n5.nightwear - 600/-\n");
                        x = sc.nextInt();
                        list1.add(x);
                        break;
                    case 2: System.out.println("Today's deal: We have 20% off on denims");
                        System.out.println("choose your items\n6.denims - 1500/-\n7.formals - 1000/-
\n8.casuals - 800/-\n9.nightwear - 500/-");
                        x = sc.nextInt();
                        list1.add(x);
                        break;
                    case 3: System.out.println("choose your items\n10.regular wear - 600/-
\n11.traditional - 1000/-\n12.nightwear - 400/-");
                        x = sc.nextInt();
                        list1.add(x);
                        break;
                }
                break;
            case 2: System.out.println("Today's deal: We have 20% off on tablets from Apollo
Pharmacy");
                System.out.println("choose items\n13.sanitizer and disinfectant spray- 250/-
\n14.masks and gloves - 300/-\n15.tablets - 100/-\n16.dettol and bandages -100/-
\n17.insulin - 350/-\n");
                x = sc.nextInt();
                if(x==16)
                    o1.exceptions();
                else
                    list1.add(x);
                break;
            case 3: System.out.println("Today's deal: We have 20% off on Edible oil and surf
excel");

```

```

        System.out.println("choose your items\n18.vegetables - 200/-\n19.cereals - 150/-\n20.edible oil - 150/-\n21.lyzol - 140/-\n22.surf excel - 300/-\n23.tresemme shampoo - 325/-\n24.deodrant - 325/-");
        x = sc.nextInt();
        if(x==18 || x==23)
            o1.exceptions();
        else
            list1.add(x);
        break;
        case 4: System.out.println("choose \n1.women\n2.men\n");
        x = sc.nextInt();
        switch(x){
            case 1: System.out.println("Today's deal: We have 20% off on Wedges");
                System.out.println("choose your items\n25.sandals - 2500/-\n26.wedges - 2800/-\n27.sneakers - 2000/-\n28.heels - 2000/-\n29.slippers - 600/-\n");
                x = sc.nextInt();
                if(x==25)
                    o1.exceptions();
                else
                    list1.add(x);
                break;
            case 2: System.out.println("Today's deal: We have 20% off on Espadrilles. New in stock, try them!!!");
                System.out.println("choose your items\n30.Boat shoes - 1500/-\n31.slippers - 600/-\n32 loafers - 1800/-\n33.Espadrilles - 1000/-");
                x = sc.nextInt();
                if(x==31)
                    o1.exceptions();
                else
                    list1.add(x);
                break;}
        break;
        case 5: System.out.println("Today's deal: We have 20% off on Matte products and sunscreen");
            System.out.println("choose your fashion\n34.serum - 300/-\n35.matte products - 850/-\n36.face wash - 250/-\n37.sunscren - 200/-\n38.face mask - 120/-\n");
            x = sc.nextInt();
            if(x==36 || x==38)
                o1.exceptions();
            else
                list1.add(x);
            break;
        case 6: System.out.println("Today's deal: We have 20% off on Refrigerators and Air conditioners. Happy Summers :)");
            System.out.println("choose items\n39.laptop - 60000/-\n40.refrigerator - 80000/-\n41.1+ phone - 42000/-\n42.AC - 40000/-\n");
            x = sc.nextInt();

```

```

        if(x==41)
            o1.exceptions();
        else
            list1.add(x);
        break;
    }
    System.out.println("To continue shopping press 0 else press -1");
    n = sc.nextInt();
}return list1;
}}
//
package onlineshopping;
import java.util.*;
import java.io.FileInputStream;
import cartoperations.*;
import onlineexceptions.cancelorder;
import onlineexceptions.expired;
import onlineexceptions.itemDamaged;
import onlineexceptions.returnProduct;
public class onlineShopping extends customer{
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Do you want to \n1.sign in \n2.login");
        int s =sc.nextInt();
        switch(s){
            case 1:
                cart c = new cart();
                Thread t1;
                customer c1 = new customer();
                t1 = new Thread(new Runnable(){
                    public void run(){
                        c.addToCart(c1.items());
                        c.display();
                        System.out.println("Do you want to add or remove any items??\n1.yes\n2.no");
                        int s = sc.nextInt();
                        if(s==1){
                            System.out.println("press\n1.add\n2.remove");
                            int x = sc.nextInt();
                            switch(x){
                                case 1:
                                    c.addToCart(c1.items());
                                    break;
                                case 2:
                                    c.removeFromCart();
                                    break;
                            }
                        }
                    }
                }
            }else{

```



```

        System.out.println("VISIT AGAIN");
    }
    c.coupons(c.bill);
    int b=c.billPayment();
    c1.userCredentials(c.items(),b);
    c.specialGifts();
    });
t1.start();
try{
    t1.join();
}catch(Exception e){}
break;
case 2:System.out.println("enter your username and password");
String username,password;
    username =sc.next();
    password = sc.next();
    try{
        FileInputStream fin = new FileInputStream("client.txt");
        String s1 = null;
        int z=0;
        while((z=fin.read())!=-1){
            s1+=(char)z;
        }
        fin.close();
        if(s1.contains(username) && s1.contains(password)){
            System.out.println("logged in successfully");
            cart cart1 = new cart();
Thread t2;
customer c2 = new customer();
t2 = new Thread(new Runnable(){
    public void run(){
        System.out.println("Do you have any 1.queries or 2.order an item");
        int x = sc.nextInt();
        switch(x){
            case 1:System.out.println("What is your query??\n1.Product expired\n2.Product
damaged\n3.Cancel order\n4.Return product");
                x = sc.nextInt();
                switch(x){
                    case 1:expired e1 = new expired();
                        e1.exceptions();
                        break;
                    case 2:itemDamaged i1 = new itemDamaged();
                        i1.exceptions();
                        break;
                    case 4:returnProduct r1 = new returnProduct();
                        r1.exceptions();
                        break;

```

```

        case 3:cancelorder ca1 = new cancelorder();
        ca1.exceptions();
        break;
    }
    break;
    case 2:
    cart1.addToCart(c2.items());
    cart1.display();
    System.out.println("Do you want to add or remove any items??\n1.yes\n2.no");
    int s = sc.nextInt();
    if(s==1){
        System.out.println("press\n1.add\n2.remove");
        int x1 = sc.nextInt();
        switch(x1){
            case 1:
                cart1.addToCart(c2.items());
                break;
            case 2:
                cart1.removeFromCart();
                break;
        }
    }else{
        System.out.println("VISIT AGAIN");
    }
    cart1.coupons(cart1.bill);
    int b=cart1.billPayment();
    c2.userCredentials(cart1.items(),b);
    cart1.specialGifts();
    break;
} }));
t2.start();
try{
    t2.join();
}catch(Exception e){}
    }else{
        System.out.println("\naccount not found!!! create a new one or try again");
    } }
    catch(Exception e){System.out.println(e);}
}}}

```

OUTPUTS/RESULTS:

```
Do you want to
1.sign in
2.login
1
Choose the category
1.fashion
2.medical
3.pantry
4.footwear
5.beauty and skin
6.electrical appliances

1
choose
1.women
2.men
3.children
3
choose your items
10.regular wear - 600/-
11.traditional - 1000/-
12.nightwear - 400/-
10
To continue shopping press 0 else press -1
0
Choose the category
1.fashion
2.medical
3.pantry
4.footwear
5.beauty and skin
6.electrical appliances

3
Today's deal: We have 20% off on Edible oil and surf excel
choose your items
18.vegetables - 200/-
19.cereals - 150/-
20.edible oil - 150/-
21.lyzol - 140/-
22.surf excel - 300/-
23.tresemme shampoo - 325/-
24.deodrant - 325/-
19
To continue shopping press 0 else press -1
```

```
3
Today's deal: We have 20% off on Edible oil and surf excel
choose your items
18.vegetables - 200/-
19.cereals - 150/-
20.edible oil - 150/-
21.lyzol - 140/-
22.surf excel - 300/-
23.tresemme shampoo - 325/-
24.deodrant - 325/-
24
To continue shopping press 0 else press -1
0
Choose the category
1.fashion
2.medical
3.pantry
4.footwear
5.beauty and skin
6.electrical appliances

4
choose
1.women
2.men

1
Today's deal: We have 20% off on Wedges
choose your items
25.sandals - 2500/-
26.wedges - 2800/-
27.sneakers - 2000/-
28.heels - 2000/-
29.slippers - 600/-

26
To continue shopping press 0 else press -1
-1
```

```
Items in the cart are:
regulars
cereals
matte
deodrant
wedges
Your bill is 3995

Do you want to add or remove any items??
1.yes
2.no
1
press
1.add
2.remove
1
Choose the category
1.fashion
2.medical
3.pantry
4.footwear
5.beauty and skin
6.electrical appliances

5
Today's deal: We have 20% off on Matte products and sunscreen
choose your fashion
34.serum - 300/-
35.matte products - 850/-
36.face wash - 250/-
37.sunscreen - 200/-
38.face mask - 120/-

36
Your desired product is out of stock at present
Sorry for your Inconvinience, We will update the stock within 2-3 weeks. Stay tuned
To continue shopping press 0 else press -1
0
Choose the category
1.fashion
```

```
5
Today's deal: We have 20% off on Matte products and sunscreen
choose your fashion
34.serum - 300/-
35.matte products - 850/-
36.face wash - 250/-
37.sunscreen - 200/-
38.face mask - 120/-

37
To continue shopping press 0 else press -1
-1
do you have a coupon??
1.yes
2.no
1
Enter your coupon value
200

Your total bill amount is: 3955
Choose your mode of payment
1.pay on delivery
2.debit card
3.Google Pay
3
Google Pay your amount to 9347687154. VISIT AGAIN :)
enter your name, phone number, loginID and password
visista
9352689569
visista121
vishi3214
Your info is stored successfully
PS C:\Users\VISISTA\OneDrive\oops project>
```

≡ client.txt

```
1 name: visista
2 userID: visista121
3 password: vishi3214
4 phone number: 9352689569
5 cart items: [regulars, cereals, matte, deodrant, wedges, sunscreen]
6 Bill amount: 3955
```

CONCLUSION

To conclude, this project helps the customers to shop on their own in an easy manner. The use of collections, threads in this project made this program efficient. We conclude that our project makes our customer satisfied as it saves time and effort in it.

REFERENCES:

- [1] Text Book: “Object Oriented Programming in Java” by G T Thampi
- [2] Javatpoint website
- [3] Amazon shopping website for application ideas