GEBZE TECHNICAL UNIVERSITY

DEPARTMENT OF COMPUTER ENGINEERING

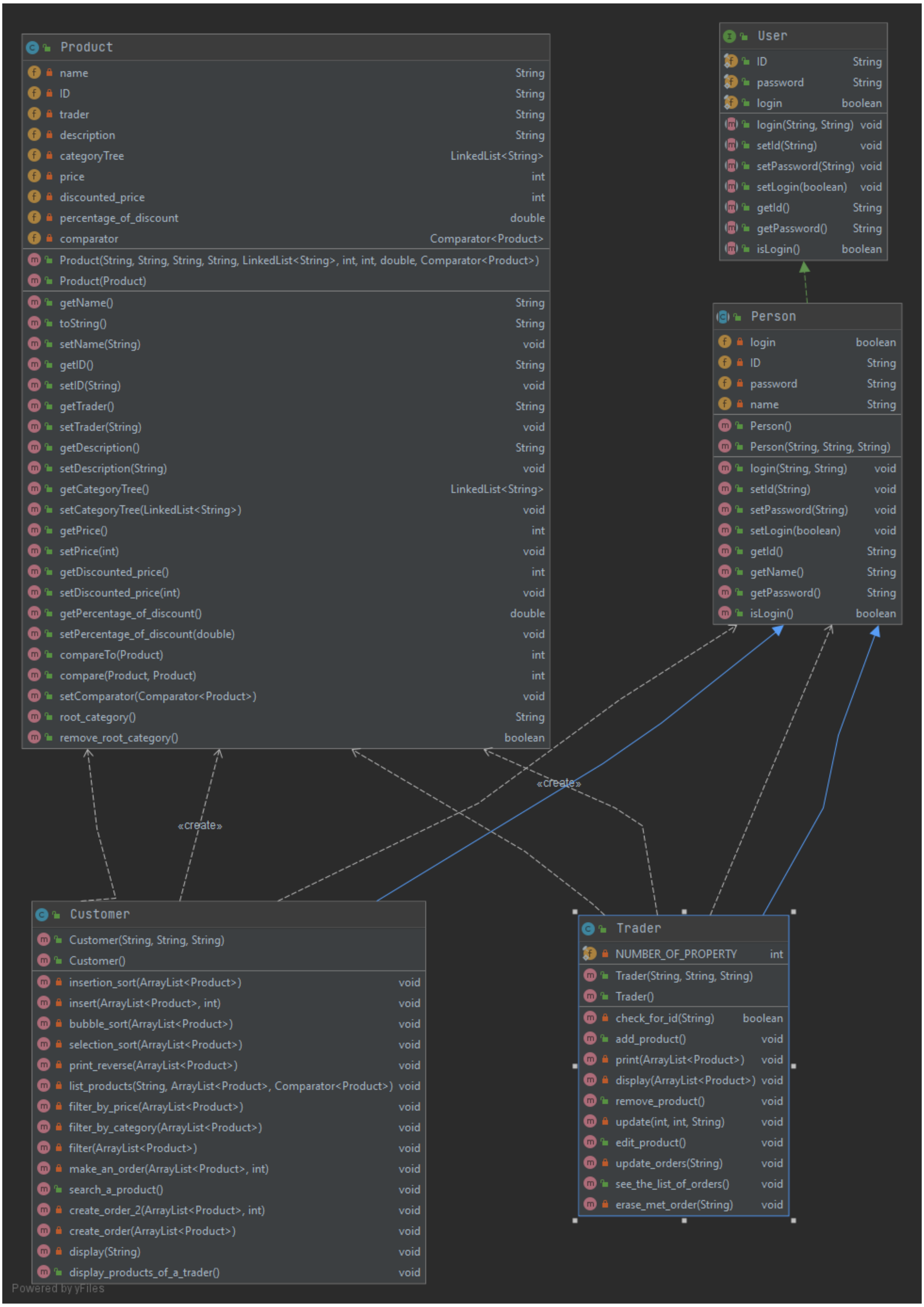
CSE222/505 – Spring 2021

Homework 6 Report

Yakup Talha Yolcu

1801042609

1. CLASS DIAGRAMS

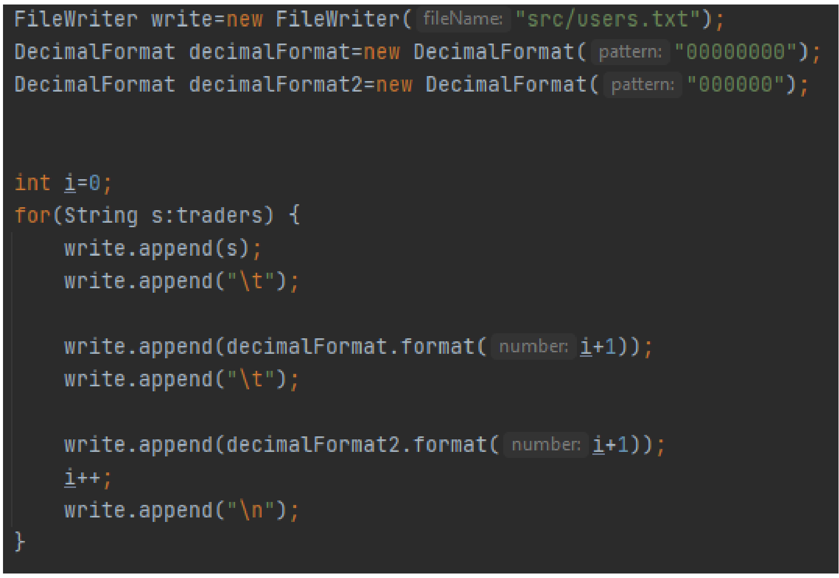


Product
name String
ID String
trader String
description String
categoryTree LinkedList<String>
price int
discounted_price int
percentage_of_discount double
comparator Comparator<Product>
Product(String, String, String, String, LinkedList<String>, int, int, double, Comparator<Product>)
Product(Product)
getName() String
toString() String
setName(String) void
getID() String
setID(String) void
getTrader() String
setTrader(String) void
getDescription() String
setDescription(String) void
getCategoryTree() LinkedList<String>
setCategoryTree(LinkedList<String>) void
getPrice() int
setPrice(int) void
getDiscounted_price() int
setDiscounted_price(int) void
getPercentage_of_discount() double
setPercentage_of_discount(double) void
compareTo(Product) int
compare(Product, Product) int
setComparator(Comparator<Product>) void
root_category() String
remove_root_category() boolean
User
ID String
password String
login boolean
login(String, String) void
setId(String) void
setPassword(String) void
setLogin(boolean) void
getId() String
getPassword() String
isLogin() boolean
Person
login boolean
ID String
password String
name String
Person()
Person(String, String, String)
login(String, String) void
setId(String) void
setPassword(String) void
setLogin(boolean) void
getId() String
getName() String
getPassword() String
isLogin() boolean
«create»
«create»
Customer
Customer(String, String, String)
Customer()
insertion_sort(ArrayList<Product>) void
insert(ArrayList<Product>, int) void
bubble_sort(ArrayList<Product>) void
selection_sort(ArrayList<Product>) void
print_reverse(ArrayList<Product>) void
list_products(String, ArrayList<Product>, Comparator<Product>) void
filter_by_price(ArrayList<Product>) void
filter_by_category(ArrayList<Product>) void
filter(ArrayList<Product>) void
make_an_order(ArrayList<Product>, int) void
search_a_product() void
create_order_2(ArrayList<Product>, int) void
create_order(ArrayList<Product>) void
display(String) void
display_products_of_a_trader() void
Trader
NUMBER_OF_PROPERTY int
Trader(String, String, String)
Trader()
check_for_id(String) boolean
add_product() void
print(ArrayList<Product>) void
display(ArrayList<Product>) void
remove_product() void
update(int, int, String) void
edit_product() void
update_orders(String) void
see_the_list_of_orders() void
erase_met_order(String) void
Powered by yFiles

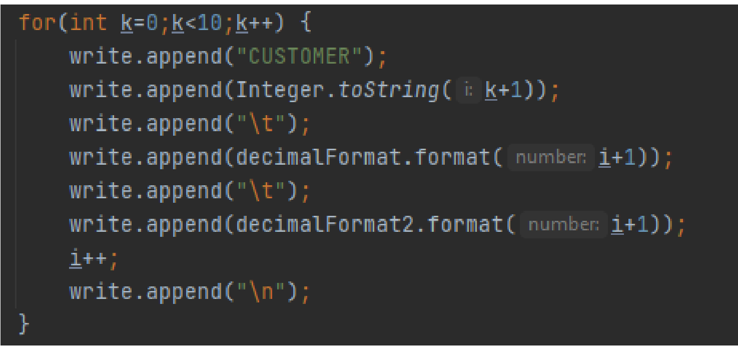
2. PROBLEM SOLUTIONS APPROACH

At first, I read the .csv file. Then I created products.txt . I keep the Trader names in **HashSet.** I used HashSet because I will need to create users.txt and I need to have unique Trader Names. In products.txt file, Each line is like following structure:
Product name ; description ; Trader Name ; Category Tree ; Price ; Discounted Price
I did it like that because I will need to have product name and description of the product when customer searches a product by given text.

After products.txt is done, I created the users.txt file. Each user should have a unique ID. I used HashSet's iterator and each iteration I incremented the counter and I assigned the ID and password. Like traders, I did the same thing to the customers. I have 10 customers.

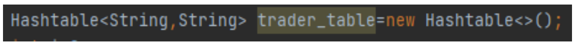


```
FileWriter write=new FileWriter( fileName: "src/users.txt");
DecimalFormat decimalFormat=new DecimalFormat( pattern: "00000000");
DecimalFormat decimalFormat2=new DecimalFormat( pattern: "000000");


int i=0;
for(String s:traders) {
    write.append(s);
    write.append("\t");

    write.append(decimalFormat.format( number: i+1));
    write.append("\t");

    write.append(decimalFormat2.format( number: i+1));
    i++;
    write.append("\n");
}
```

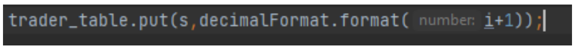


```
for(int k=0;k<10;k++) {
    write.append("CUSTOMER");
    write.append(Integer.toString( i: k+1));
    write.append("\t");
    write.append(decimalFormat.format( number: i+1));
    write.append("\t");
    write.append(decimalFormat2.format( number: i+1));
    i++;
    write.append("\n");
}
```

I used **Hashtable** keep trader names and ID's



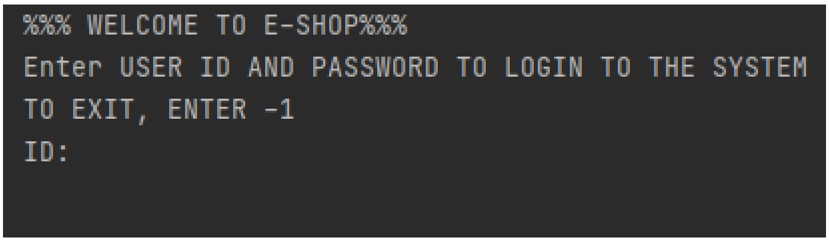
```
Hashtable<String,String> trader_table=new Hashtable<>();
```



```
trader_table.put(s,decimalFormat.format( number: i+1));
```

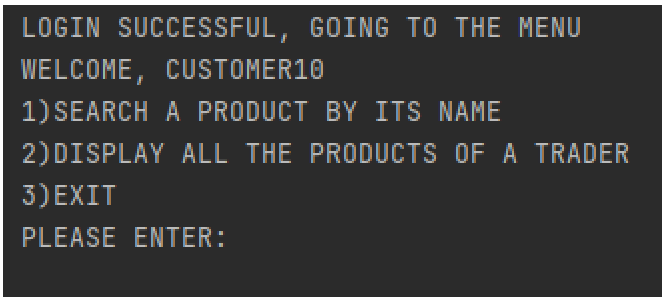
After creation of users.txt, I created the orders.txt file. I just created, I did not write anything to it yet.

After creation of 3 files, User is going to see a menu like that:

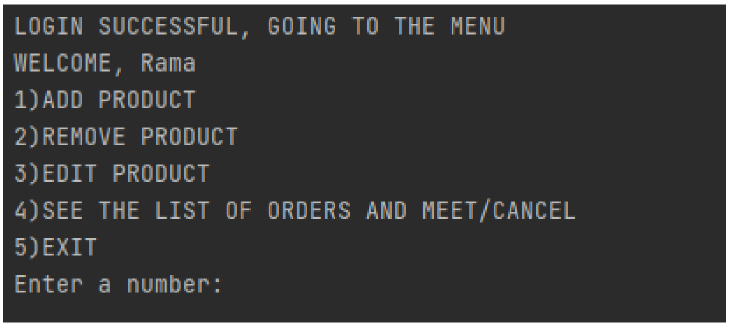


```
%%% WELCOME TO E-SHOP%%%
Enter USER ID AND PASSWORD TO LOGIN TO THE SYSTEM
TO EXIT, ENTER -1
ID:
```

Here, user should enter the ID and then enter the password. After the authentication process done properly (I compared the given ID and password) (Firstly I compared the length of the ID-should 8- and password-should 6- then I started to read file), depend on the user role, menu will be shown.



```
LOGIN SUCCESSFUL, GOING TO THE MENU
WELCOME, CUSTOMER10
1)SEARCH A PRODUCT BY ITS NAME
2)DISPLAY ALL THE PRODUCTS OF A TRADER
3)EXIT
PLEASE ENTER:
```

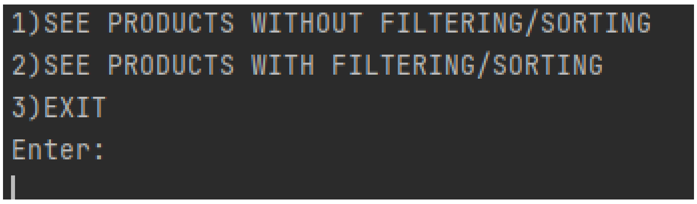


```
LOGIN SUCCESSFUL, GOING TO THE MENU
WELCOME, Rama
1)ADD PRODUCT
2)REMOVE PRODUCT
3)EDIT PRODUCT
4)SEE THE LIST OF ORDERS AND MEET/CANCEL
5)EXIT
Enter a number:
```

CUSTOMER MENU	TRADER MENU

- **CUSTOMER MENU**

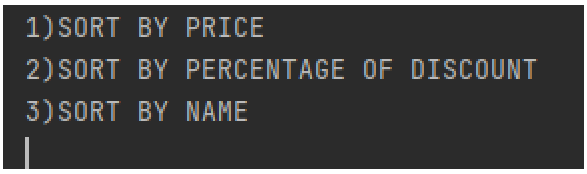
  Customer can search a product by given text. They can also display all the products of a customer. When customer want to search a product, this menu will be shown:



```
1)SEE PRODUCTS WITHOUT FILTERING/SORTING
2)SEE PRODUCTS WITH FILTERING/SORTING
3)EXIT
Enter:
```

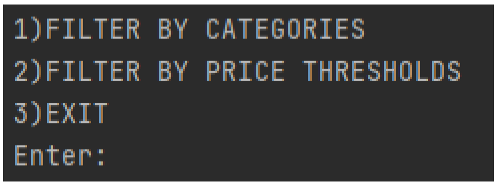
If user selects to not filter, Text will be taken and products will be listed within this text. When products are reading from the file, they will be stored in **ArrayList**. I used ArrayList because customer should be able to access the product at O(1) time. When sorting products (decrasing order), I used **insertion sort**. User can give order from here after products are listed. Index is taken from the customer.

If user selects to filter, text is taken and then this menu will be shown:



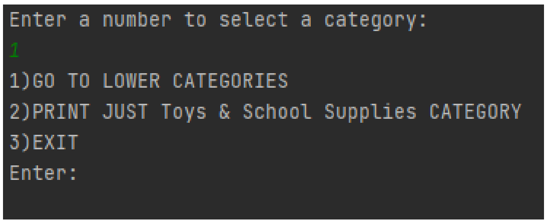
```
1)SORT BY PRICE
2)SORT BY PERCENTAGE OF DISCOUNT
3)SORT BY NAME
```

If user selects the sort by price, Filter menu will be shown. Before filtering, Products are sorted by using **bubble sort.** Filter menu is like that:



```
1)FILTER BY CATEGORIES
2)FILTER BY PRICE THRESHOLDS
3)EXIT
Enter:
```

If user selects the filter by categories, all higher level categories will be shown. Then this menu is shown:



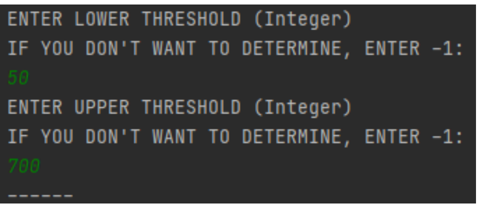
```
Enter a number to select a category:
1
1)GO TO LOWER CATEGORIES
2)PRINT JUST Toys & School Supplies CATEGORY
3)EXIT
Enter:
```

If user selects the going to lower categories, same function will be called **recursively.** Base case for recursive structure is the situation that there is no lower category anymore.

If user selects the other option, All products have the current category will be printed.

Reminder: Products are hold in **ArrayList** in whole sorting/filtering process.

If user selects the Filter by price thresholds, this menu will shown:



```
ENTER LOWER THRESHOLD (Integer)
IF YOU DON'T WANT TO DETERMINE, ENTER -1:
50
ENTER UPPER THRESHOLD (Integer)
IF YOU DON'T WANT TO DETERMINE, ENTER -1:
700
------
```

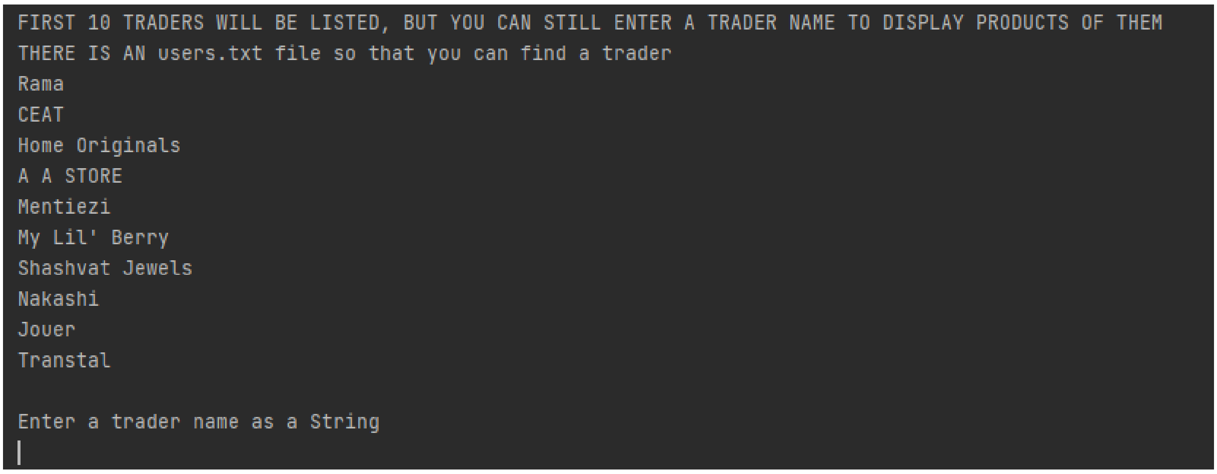
Depend on the choices products will be shown. Again I **used ArrayList** here.

When user selects the SORT BY PERCENTAGE OF DISCOUNT, products will be listed by using **selection sort**. Then filter process will be done as if in the sort by price case.

When user selects the SORT BY NAME, products will be listed by using **insertion sort**. Then filter process will be done as if previous cases.

After sorting/filtering customer can give an order by entering index.

If customer wants to see products of a trader, this will be shown:



```
FIRST 10 TRADERS WILL BE LISTED, BUT YOU CAN STILL ENTER A TRADER NAME TO DISPLAY PRODUCTS OF THEM
THERE IS AN users.txt file so that you can find a trader
Rama
CEAT
Home Originals
A A STORE
Mentiezi
My Lil' Berry
Shashvat Jewels
Nakashi
Jouer
Transtal

Enter a trader name as a String
|
```

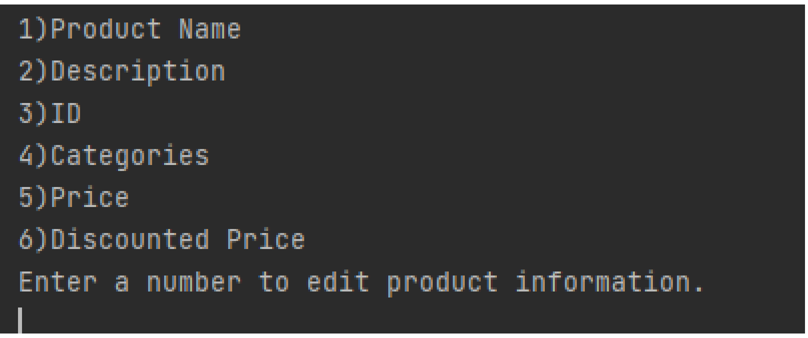
After customer enters the Trader name properly, products of a trader will be listed. I used **TreeSet** when storing first ten traders to show the trader names to the customer. TreeSet stored just Strings as name of the trader. When listing products here, I did not store them anywhere. I just printed them as it needed in the PDF. User can't make an order from here.

- **TRADER MENU**

When trader selects to add a product, They should enter **PRODUCT NAME, DESCRIPTION, UNIQUE ID, CATEGORIES, PRICE, DISCOUNTED PRICE**. After being proper of the product informations, product is added. Categories are stored in **LinkedList** as a String in the Product class. I used LinkedList because I had to store the higher category at the first and I should be able to remove the higher category from here at O(1) time. I used **ArrayList** to hold the products.

When trader selects to remove a product, They should enter an index to select a product.

When trader selects to edit a product information and enters an index to select product, this menu will be shown:



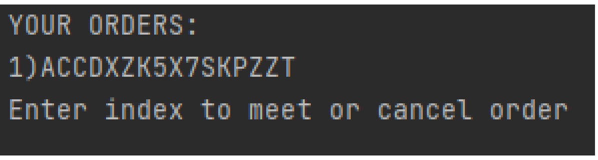
```
1)Product Name
2)Description
3)ID
4)Categories
5)Price
6)Discounted Price
Enter a number to edit product information.
|
```

Trader can edit product information one at a time.

After trader selects a property, they edit this property if it is proper:

If ID is not unique, prices and discounted price is not proper, they will not be edited, they should be taken again.

If trader wants to see the orders, this information will be shown:

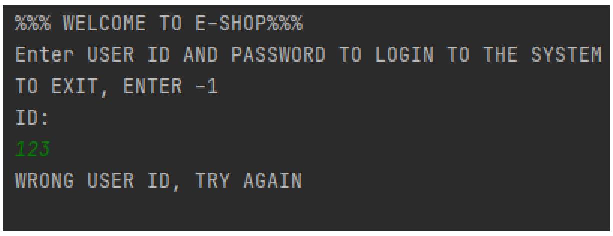


```
YOUR ORDERS:
1)ACCDXZK5X7SKPZZT
Enter index to meet or cancel order
```

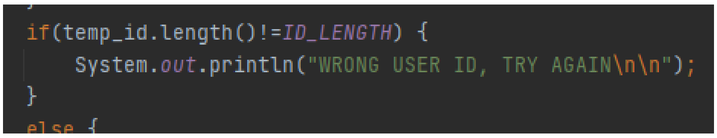
Trader can see orders here. Orders are stored in **Priority Queue**. I used this structure because trader can meet/cancel order one at a time. Trader can meet or cancel the orders by entering index if there any order.

3. TEST CASES

Entered user id's length is not 8.

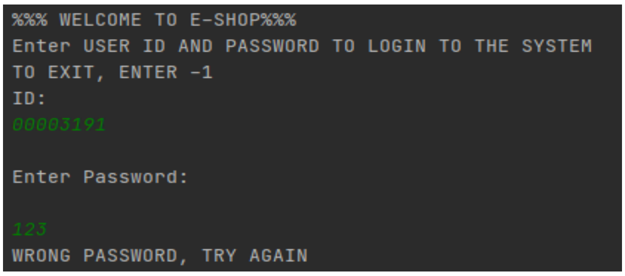


```
%%% WELCOME TO E-SHOP%%%
Enter USER ID AND PASSWORD TO LOGIN TO THE SYSTEM
TO EXIT, ENTER -1
ID:
123
WRONG USER ID, TRY AGAIN
```

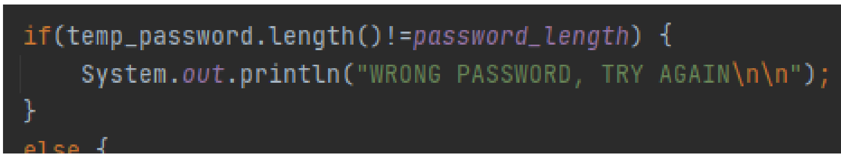


```
if(temp_id.length()!=ID_LENGTH) {
    System.out.println("WRONG USER ID, TRY AGAIN\n\n");
}
else {
```

Entered password's length is not 6

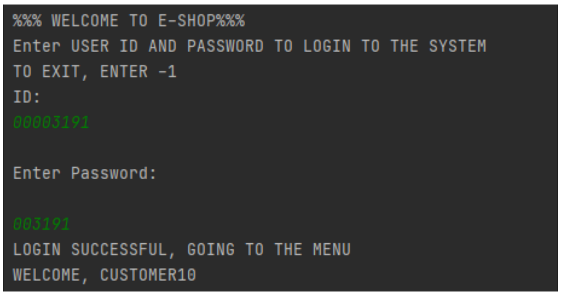


```
%%% WELCOME TO E-SHOP%%%
Enter USER ID AND PASSWORD TO LOGIN TO THE SYSTEM
TO EXIT, ENTER -1
ID:
00003191

Enter Password:

123
WRONG PASSWORD, TRY AGAIN
```

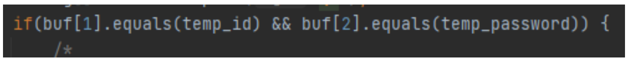


```
if(temp_password.length()!=password_length) {
    System.out.println("WRONG PASSWORD, TRY AGAIN\n\n");
}
else {
```

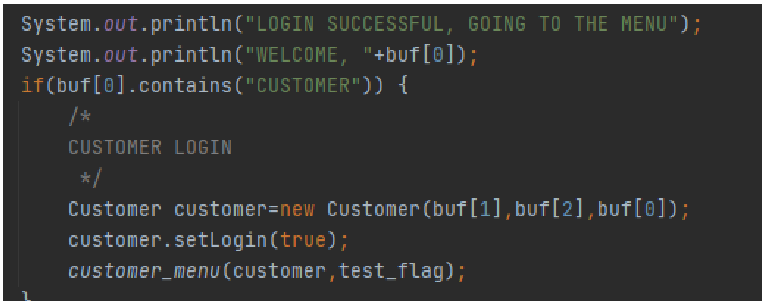
Login successful (Customer)



```
%%% WELCOME TO E-SHOP%%%
Enter USER ID AND PASSWORD TO LOGIN TO THE SYSTEM
TO EXIT, ENTER -1
ID:
00003191

Enter Password:

003191
LOGIN SUCCESSFUL, GOING TO THE MENU
WELCOME, CUSTOMER10
```

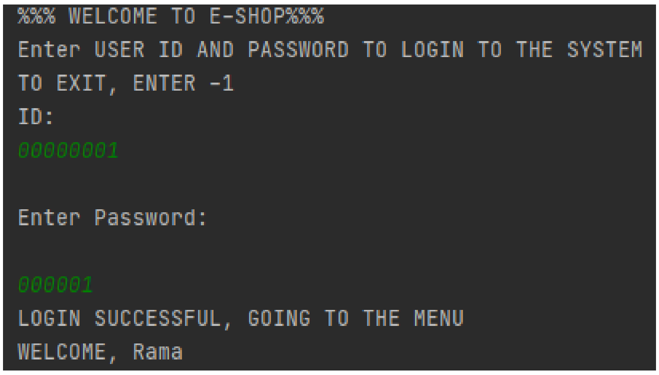


```
if(buf[1].equals(temp_id) && buf[2].equals(temp_password)) {
    /*
```

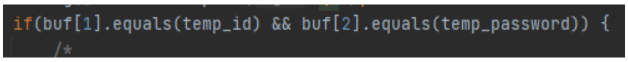


```
System.out.println("LOGIN SUCCESSFUL, GOING TO THE MENU");
System.out.println("WELCOME, "+buf[0]);
if(buf[0].contains("CUSTOMER")) {
    /*
    CUSTOMER LOGIN
     */
    Customer customer=new Customer(buf[1],buf[2],buf[0]);
    customer.setLogin(true);
    customer_menu(customer,test_flag);
}
```

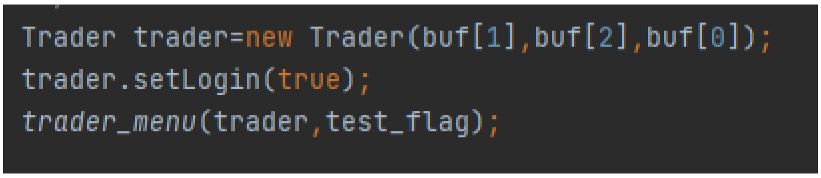
Login successful (Trader)



```
%%% WELCOME TO E-SHOP%%%
Enter USER ID AND PASSWORD TO LOGIN TO THE SYSTEM
TO EXIT, ENTER -1
ID:
00000001

Enter Password:

000001
LOGIN SUCCESSFUL, GOING TO THE MENU
WELCOME, Rama
```

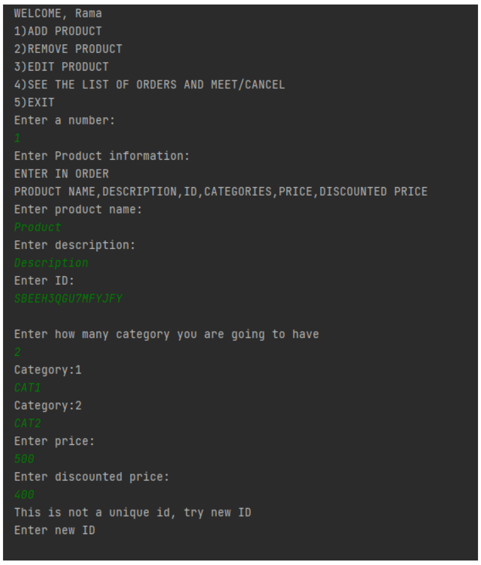


```
if(buf[1].equals(temp_id) && buf[2].equals(temp_password)) {
    /*
```

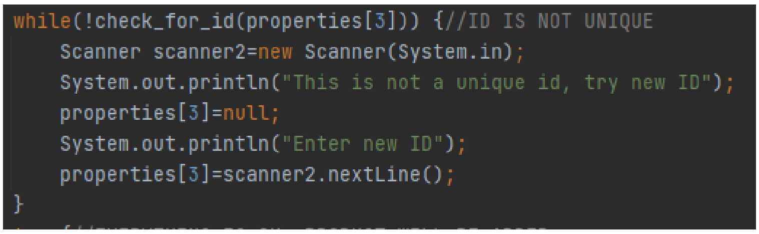


```
Trader trader=new Trader(buf[1],buf[2],buf[0]);
trader.setLogin(true);
trader_menu(trader,test_flag);
```

When traders adds a product, they did not give an unique ID

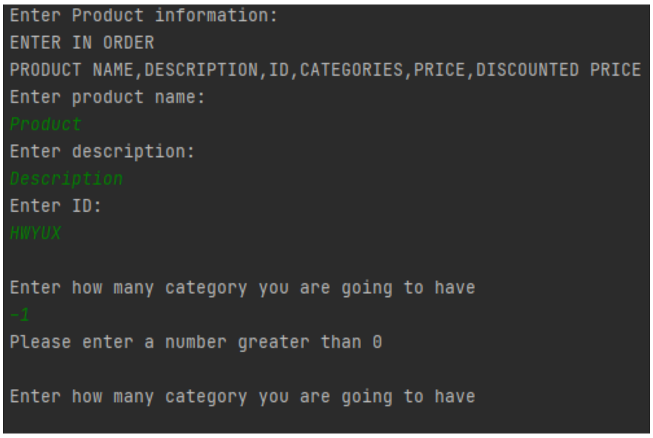


```
WELCOME, Rama
1)ADD PRODUCT
2)REMOVE PRODUCT
3)EDIT PRODUCT
4)SEE THE LIST OF ORDERS AND MEET/CANCEL
5)EXIT
Enter a number:
1
Enter Product information:
ENTER IN ORDER
PRODUCT NAME,DESCRIPTION,ID,CATEGORIES,PRICE,DISCOUNTED PRICE
Enter product name:
Product
Enter description:
Description
Enter ID:
SBEEH3QGU7MFYJFY

Enter how many category you are going to have
2
Category:1
CAT1
Category:2
CAT2
Enter price:
500
Enter discounted price:
400
This is not a unique id, try new ID
Enter new ID
```

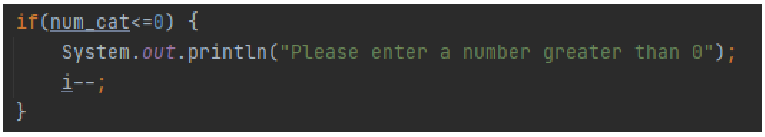


```
while(!check_for_id(properties[3])) {//ID IS NOT UNIQUE
    Scanner scanner2=new Scanner(System.in);
    System.out.println("This is not a unique id, try new ID");
    properties[3]=null;
    System.out.println("Enter new ID");
    properties[3]=scanner2.nextLine();
}
```

When Trader adds a product, they should enter number of category greater than 0.

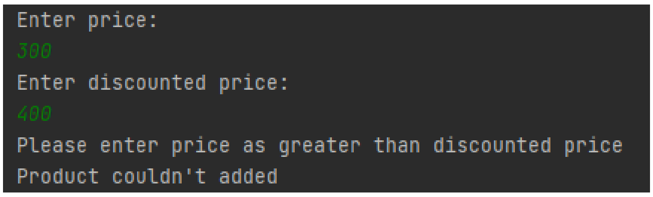


```
Enter Product information:
ENTER IN ORDER
PRODUCT NAME,DESCRIPTION,ID,CATEGORIES,PRICE,DISCOUNTED PRICE
Enter product name:
Product
Enter description:
Description
Enter ID:
HWYUX

Enter how many category you are going to have
-1
Please enter a number greater than 0

Enter how many category you are going to have
```

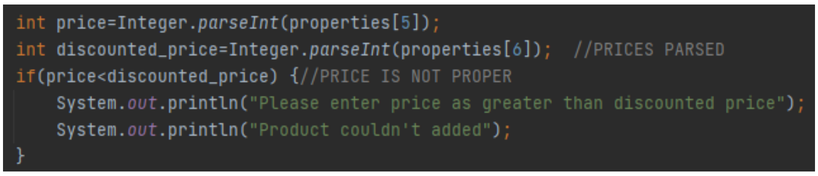


```
if(num_cat<=0) {
    System.out.println("Please enter a number greater than 0");
    i--;
}
```

When Trader adds a product, they should enter price greater than discounted price.

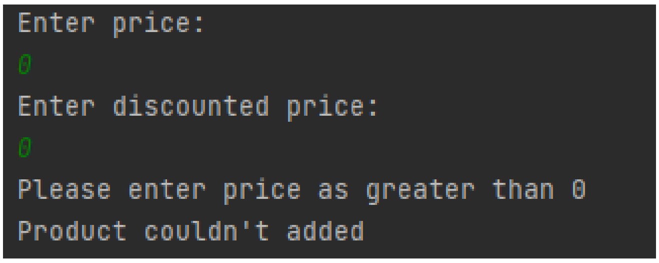


```
Enter price:
300
Enter discounted price:
400
Please enter price as greater than discounted price
Product couldn't added
```

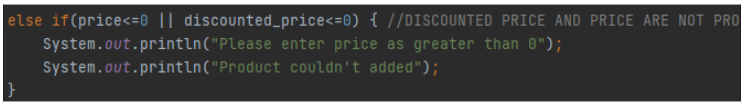


```
int price=Integer.parseInt(properties[5]);
int discounted_price=Integer.parseInt(properties[6]);  //PRICES PARSED
if(price<discounted_price) {//PRICE IS NOT PROPER
    System.out.println("Please enter price as greater than discounted price");
    System.out.println("Product couldn't added");
}
```

When Trader adds a product, they should enter price or discounted price greater than 0

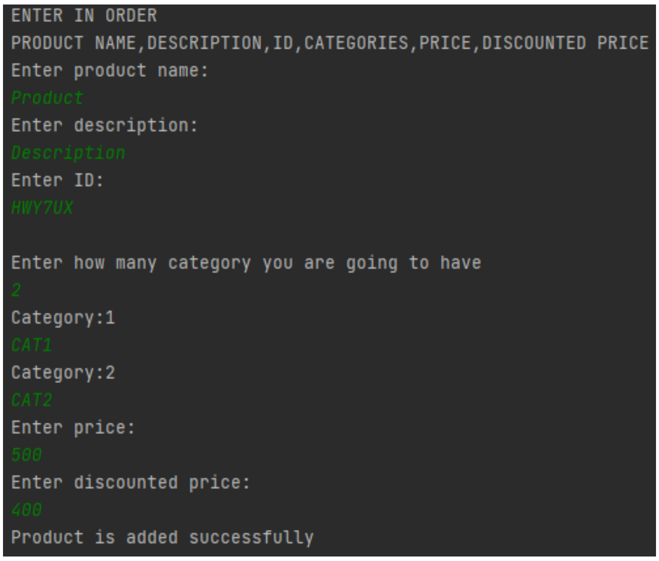


```
Enter price:
0
Enter discounted price:
0
Please enter price as greater than 0
Product couldn't added
```

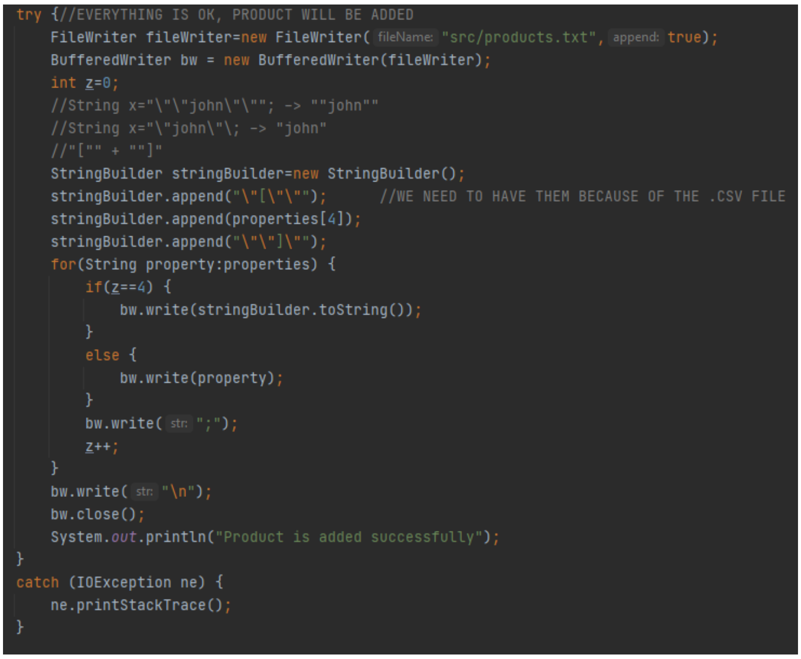


```
else if(price<=0 || discounted_price<=0) { //DISCOUNTED PRICE AND PRICE ARE NOT PRO
    System.out.println("Please enter price as greater than 0");
    System.out.println("Product couldn't added");
}
```

Trader added product successfully

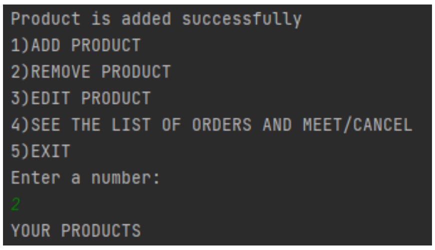


```
ENTER IN ORDER
PRODUCT NAME,DESCRIPTION,ID,CATEGORIES,PRICE,DISCOUNTED PRICE
Enter product name:
Product
Enter description:
Description
Enter ID:
HWY7UX

Enter how many category you are going to have
2
Category:1
CAT1
Category:2
CAT2
Enter price:
500
Enter discounted price:
400
Product is added successfully
```

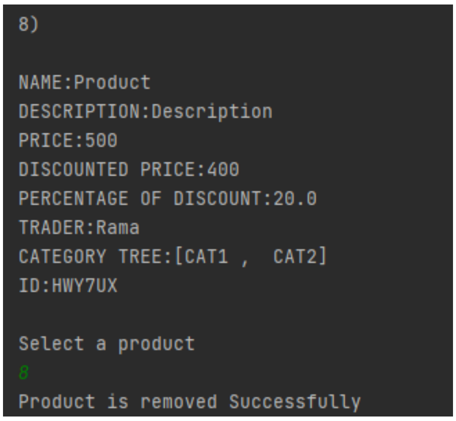


```
try {//EVERYTHING IS OK, PRODUCT WILL BE ADDED
    FileWriter fileWriter=new FileWriter( fileName: "src/products.txt", append: true);
    BufferedWriter bw = new BufferedWriter(fileWriter);
    int z=0;
    //String x="\"\"john\"\""; -> ""john""
    //String x="\"john\"\; -> "john"
    //"["" + ""]"
    StringBuilder stringBuilder=new StringBuilder();
    stringBuilder.append("\"[\"\"");      //WE NEED TO HAVE THEM BECAUSE OF THE .CSV FILE
    stringBuilder.append(properties[4]);
    stringBuilder.append("\"\"]\"");
    for(String property:properties) {
        if(z==4) {
            bw.write(stringBuilder.toString());
        }
        else {
            bw.write(property);
        }
        bw.write( str: ";");
        z++;
    }
    bw.write( str: "\n");
    bw.close();
    System.out.println("Product is added successfully");
}
catch (IOException ne) {
    ne.printStackTrace();
}
```

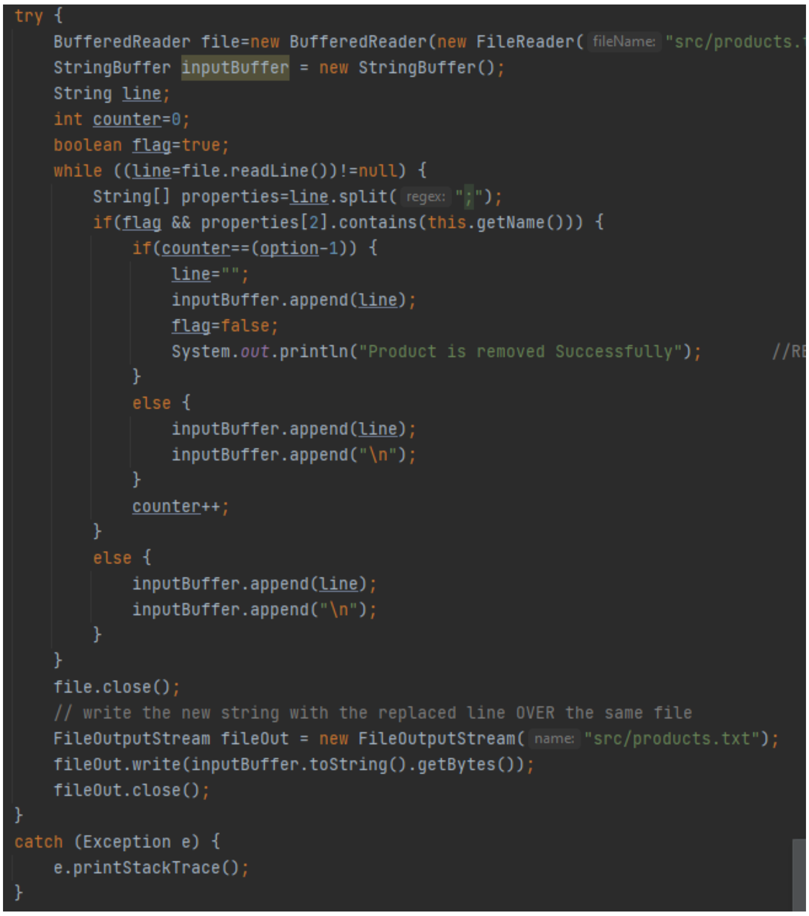
When trader removes a product, product removed successfully



```
Product is added successfully
1)ADD PRODUCT
2)REMOVE PRODUCT
3)EDIT PRODUCT
4)SEE THE LIST OF ORDERS AND MEET/CANCEL
5)EXIT
Enter a number:
2
YOUR PRODUCTS
```

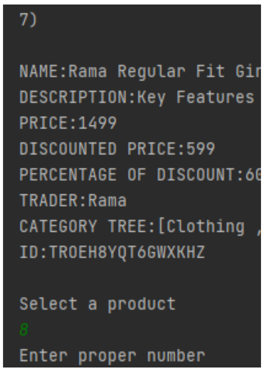


```
8)

NAME:Product
DESCRIPTION:Description
PRICE:500
DISCOUNTED PRICE:400
PERCENTAGE OF DISCOUNT:20.0
TRADER:Rama
CATEGORY TREE:[CAT1 ,  CAT2]
ID:HWY7UX

Select a product
8
Product is removed Successfully
```



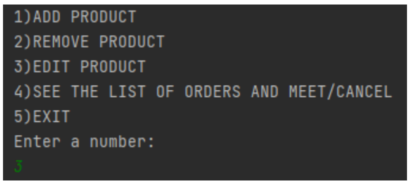
```
try {
    BufferedReader file=new BufferedReader(new FileReader( fileName: "src/products.
    StringBuffer inputBuffer = new StringBuffer();
    String line;
    int counter=0;
    boolean flag=true;
    while ((line=file.readLine())!=null) {
        String[] properties=line.split( regex: ";");
        if(flag && properties[2].contains(this.getName())) {
            if(counter==(option-1)) {
                line="";
                inputBuffer.append(line);
                flag=false;
                System.out.println("Product is removed Successfully");       //RE
            }
            else {
                inputBuffer.append(line);
                inputBuffer.append("\n");
            }
            counter++;
        }
        else {
            inputBuffer.append(line);
            inputBuffer.append("\n");
        }
    }
    file.close();
    // write the new string with the replaced line OVER the same file
    FileOutputStream fileOut = new FileOutputStream( name: "src/products.txt");
    fileOut.write(inputBuffer.toString().getBytes());
    fileOut.close();
}
catch (Exception e) {
    e.printStackTrace();
}
```

When trader removes a product, they entered wrong index.

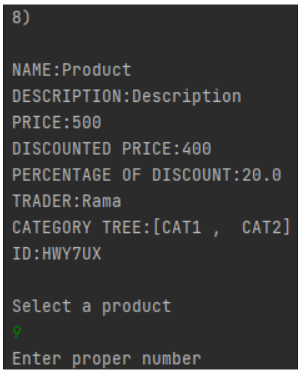


```
7)

NAME:Rama Regular Fit Gir
DESCRIPTION:Key Features
PRICE:1499
DISCOUNTED PRICE:599
PERCENTAGE OF DISCOUNT:60
TRADER:Rama
CATEGORY TREE:[Clothing ,
ID:TROEH8YQT6GWXKHZ

Select a product
8
Enter proper number
```

When trader edits products, entered wrong product index.

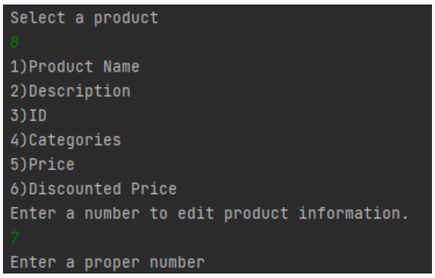


```
1)ADD PRODUCT
2)REMOVE PRODUCT
3)EDIT PRODUCT
4)SEE THE LIST OF ORDERS AND MEET/CANCEL
5)EXIT
Enter a number:
3
```



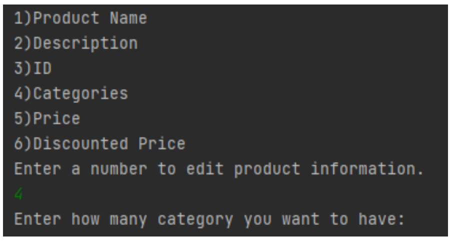
```
8)

NAME:Product
DESCRIPTION:Description
PRICE:500
DISCOUNTED PRICE:400
PERCENTAGE OF DISCOUNT:20.0
TRADER:Rama
CATEGORY TREE:[CAT1 ,  CAT2]
ID:HWY7UX

Select a product
9
Enter proper number
```

When trader edits products, entered wrong information index.

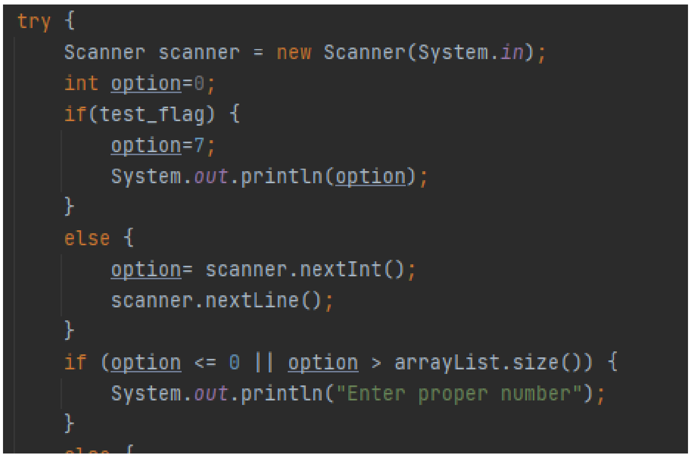


```
Select a product
8
1)Product Name
2)Description
3)ID
4)Categories
5)Price
6)Discounted Price
Enter a number to edit product information.
7
Enter a proper number
```

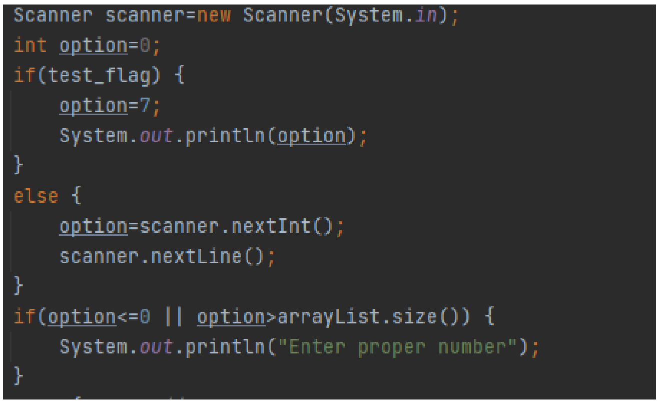
When trader edits products category, entered wrong category index.



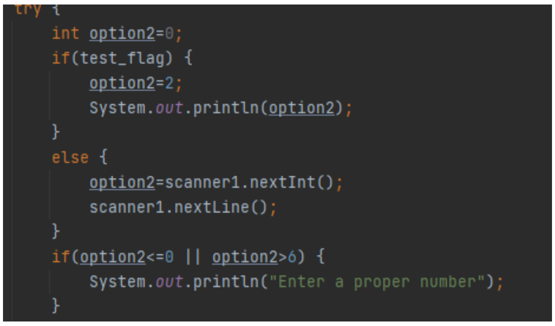
```
1)Product Name
2)Description
3)ID
4)Categories
5)Price
6)Discounted Price
Enter a number to edit product information.
4
Enter how many category you want to have:
```



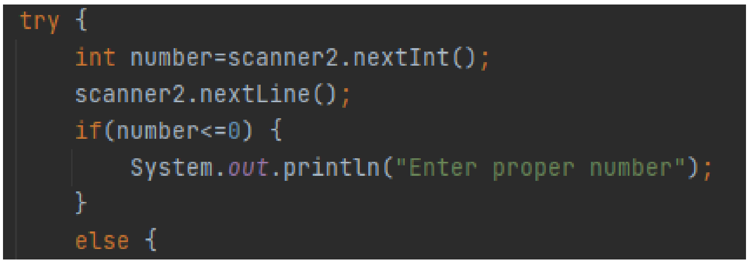
```
try {
    Scanner scanner = new Scanner(System.in);
    int option=0;
    if(test_flag) {
        option=7;
        System.out.println(option);
    }
    else {
        option= scanner.nextInt();
        scanner.nextLine();
    }
    if (option <= 0 || option > arrayList.size()) {
        System.out.println("Enter proper number");
    }
    else {
```



```
Scanner scanner=new Scanner(System.in);
int option=0;
if(test_flag) {
    option=7;
    System.out.println(option);
}
else {
    option=scanner.nextInt();
    scanner.nextLine();
}
if(option<=0 || option>arrayList.size()) {
    System.out.println("Enter proper number");
}
```

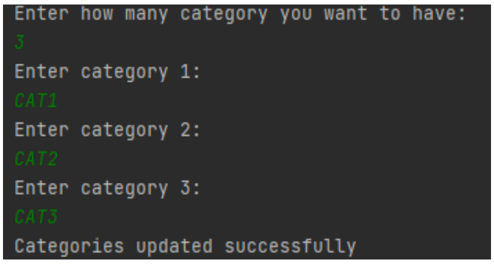


```
try {
    int option2=0;
    if(test_flag) {
        option2=2;
        System.out.println(option2);
    }
    else {
        option2=scanner1.nextInt();
        scanner1.nextLine();
    }
    if(option2<=0 || option2>6) {
        System.out.println("Enter a proper number");
    }
```



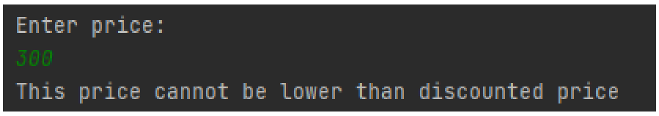
```
try {
    int number=scanner2.nextInt();
    scanner2.nextLine();
    if(number<=0) {
        System.out.println("Enter proper number");
    }
    else {
```

When trader edits products category, category updated successfully.



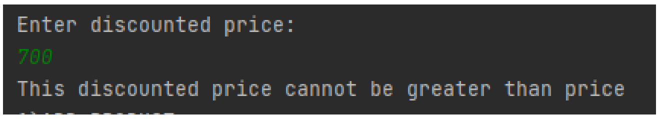
```
Enter how many category you want to have:
3
Enter category 1:
CAT1
Enter category 2:
CAT2
Enter category 3:
CAT3
Categories updated successfully
```

When trader edits products price, price is not greater than discounted price.



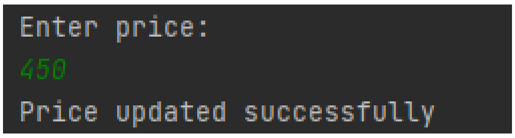
```
Enter price:
300
This price cannot be lower than discounted price
```

When trader edits products discounted price, discounted price is not less than price.



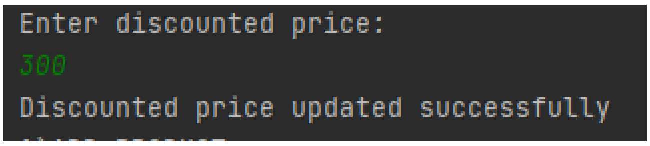
```
Enter discounted price:
700
This discounted price cannot be greater than price
```

When trader edits products price, edit is successful



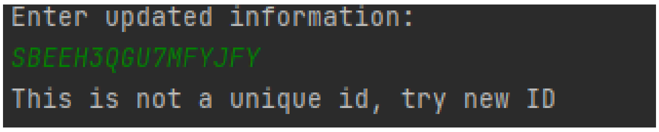
```
Enter price:
450
Price updated successfully
```

When trader edits products discounted price, edit is successful



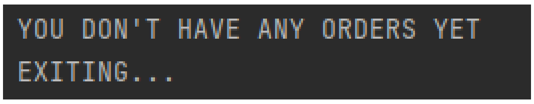
```
Enter discounted price:
300
Discounted price updated successfully
```

When trader edits products ID, they entered not a unique id.



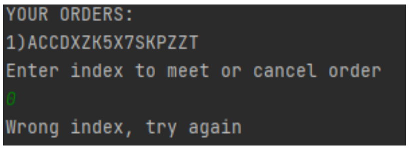
```
Enter updated information:
SBEEH3QGU7MFYJFY
This is not a unique id, try new ID
```

When trader tries to see orders, they don't have any orders yet.

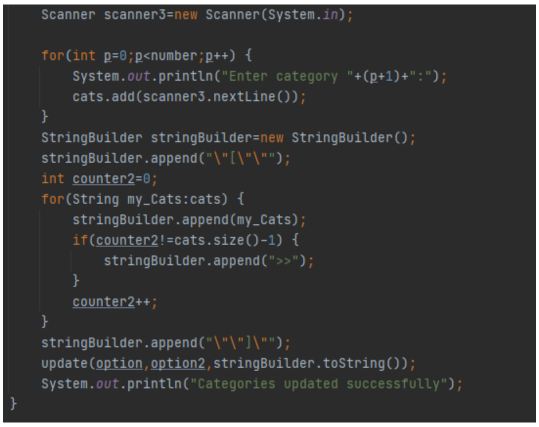


```
YOU DON'T HAVE ANY ORDERS YET
EXITING...
```

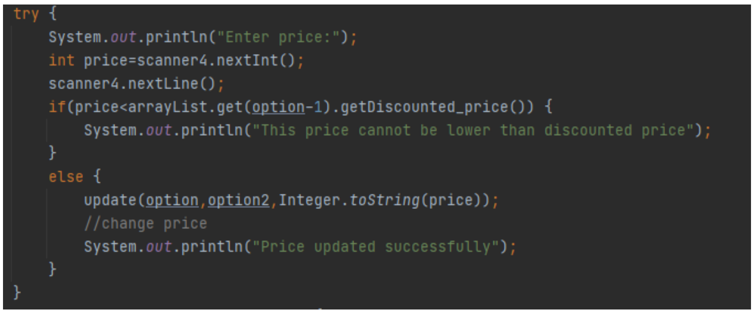
When trader tries to see orders, they entered wrong index.



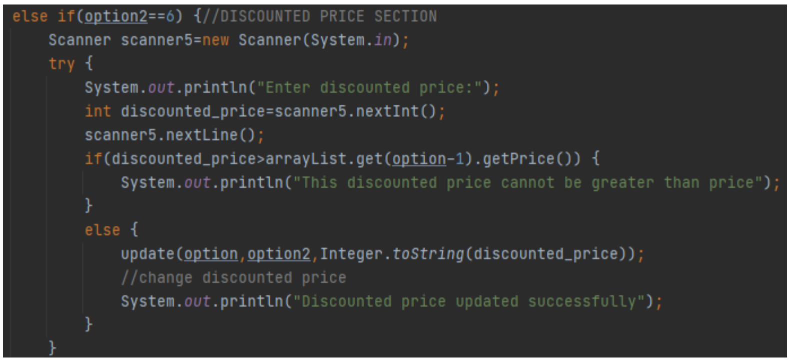
```
YOUR ORDERS:
1)ACCDXZK5X7SKPZZT
Enter index to meet or cancel order
0
Wrong index, try again
```



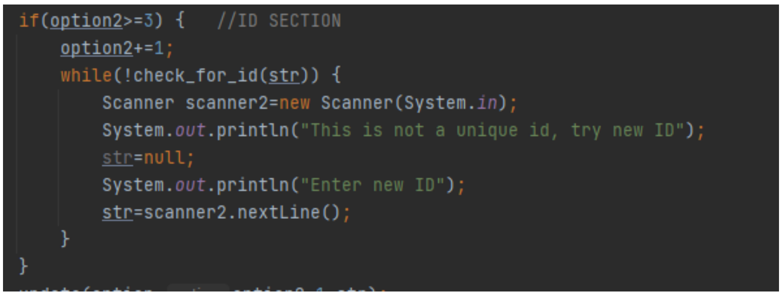
```
    Scanner scanner3=new Scanner(System.in);

    for(int p=0;p<number;p++) {
        System.out.println("Enter category "+(p+1)+":");
        cats.add(scanner3.nextLine());
    }
    StringBuilder stringBuilder=new StringBuilder();
    stringBuilder.append("\"[\"\"");
    int counter2=0;
    for(String my_Cats:cats) {
        stringBuilder.append(my_Cats);
        if(counter2!=cats.size()-1) {
            stringBuilder.append(">>");
        }
        counter2++;
    }
    stringBuilder.append("\"\"]\"");
    update(option,option2,stringBuilder.toString());
    System.out.println("Categories updated successfully");
}
```



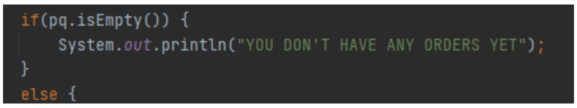
```
try {
    System.out.println("Enter price:");
    int price=scanner4.nextInt();
    scanner4.nextLine();
    if(price<arrayList.get(option-1).getDiscounted_price()) {
        System.out.println("This price cannot be lower than discounted price");
    }
    else {
        update(option,option2,Integer.toString(price));
        //change price
        System.out.println("Price updated successfully");
    }
}
```



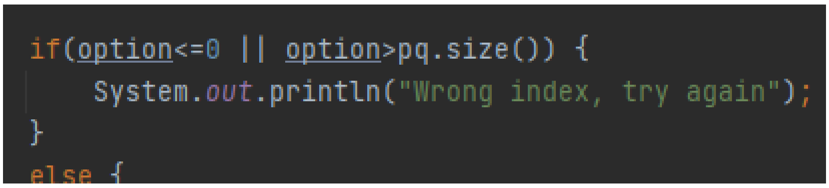
```
else if(option2==6) {//DISCOUNTED PRICE SECTION
    Scanner scanner5=new Scanner(System.in);
    try {
        System.out.println("Enter discounted price:");
        int discounted_price=scanner5.nextInt();
        scanner5.nextLine();
        if(discounted_price>arrayList.get(option-1).getPrice()) {
            System.out.println("This discounted price cannot be greater than price");
        }
        else {
            update(option,option2,Integer.toString(discounted_price));
            //change discounted price
            System.out.println("Discounted price updated successfully");
        }
    }
```



```
if(option2>=3) {   //ID SECTION
    option2+=1;
    while(!check_for_id(str)) {
        Scanner scanner2=new Scanner(System.in);
        System.out.println("This is not a unique id, try new ID");
        str=null;
        System.out.println("Enter new ID");
        str=scanner2.nextLine();
    }
}
```

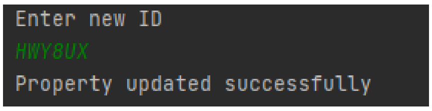


```
if(pq.isEmpty()) {
    System.out.println("YOU DON'T HAVE ANY ORDERS YET");
}
else {
```

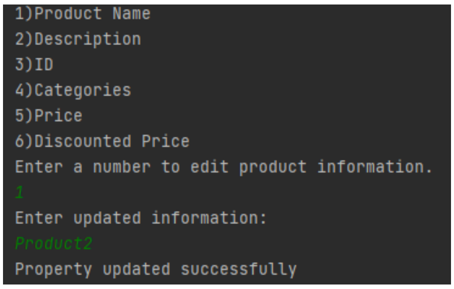


```
if(option<=0 || option>pq.size()) {
    System.out.println("Wrong index, try again");
}
else {
```

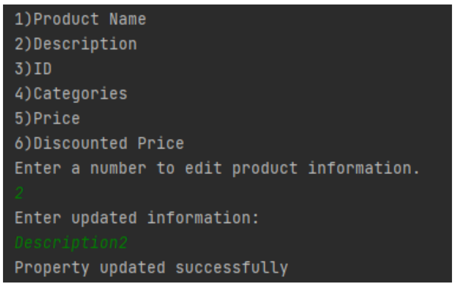
When trader edits products ID,name or Description ; edit is successful



```
Enter new ID
HWY8UX
Property updated successfully
```

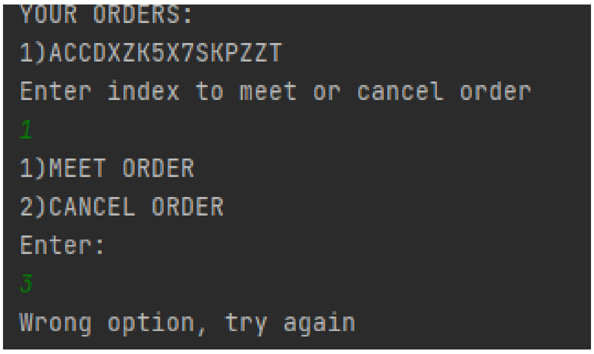


```
1)Product Name
2)Description
3)ID
4)Categories
5)Price
6)Discounted Price
Enter a number to edit product information.
1
Enter updated information:
Product2
Property updated successfully
```



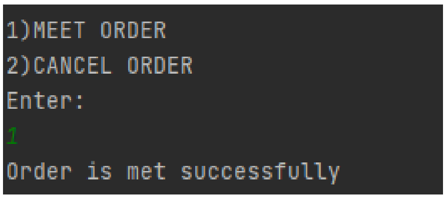
```
1)Product Name
2)Description
3)ID
4)Categories
5)Price
6)Discounted Price
Enter a number to edit product information.
2
Enter updated information:
Description2
Property updated successfully
```

When trader tries to meet/cancel order, entered wrong menu index.



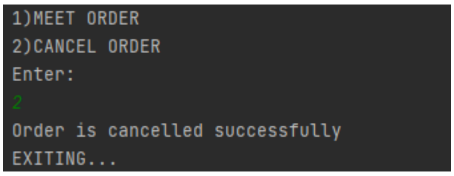
```
YOUR ORDERS:
1)ACCDXZK5X7SKPZZT
Enter index to meet or cancel order
1
1)MEET ORDER
2)CANCEL ORDER
Enter:
3
Wrong option, try again
```

When trader tries to meet order, order is met successfully



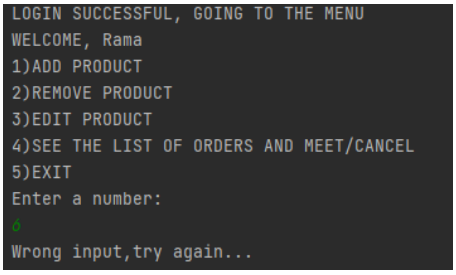
```
1)MEET ORDER
2)CANCEL ORDER
Enter:
1
Order is met successfully
```

When trader tries to cancel order, order is cancelled successfully

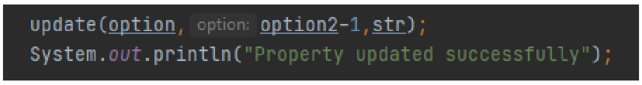


```
1)MEET ORDER
2)CANCEL ORDER
Enter:
2
Order is cancelled successfully
EXITING...
```

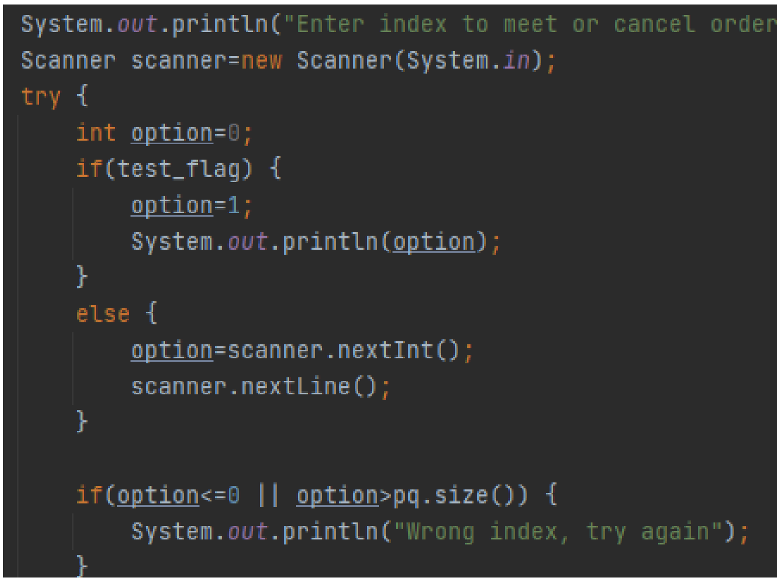
Trader entered wrong menu input



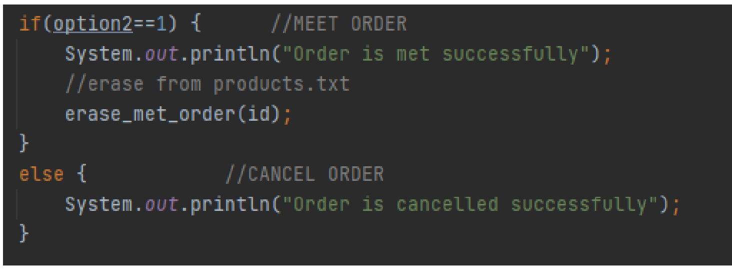
```
LOGIN SUCCESSFUL, GOING TO THE MENU
WELCOME, Rama
1)ADD PRODUCT
2)REMOVE PRODUCT
3)EDIT PRODUCT
4)SEE THE LIST OF ORDERS AND MEET/CANCEL
5)EXIT
Enter a number:
6
Wrong input,try again...
```



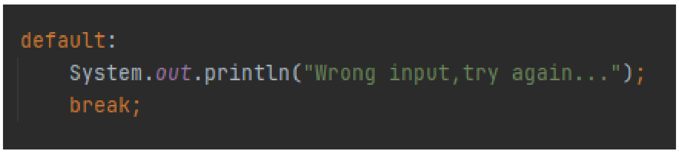
```java
update(option, option: option2-1,str);
System.out.println("Property updated successfully");
```



```java
System.out.println("Enter index to meet or cancel order
Scanner scanner=new Scanner(System.in);
try {
    int option=0;
    if(test_flag) {
        option=1;
        System.out.println(option);
    }
    else {
        option=scanner.nextInt();
        scanner.nextLine();
    }

    if(option<=0 || option>pq.size()) {
        System.out.println("Wrong index, try again");
    }
```

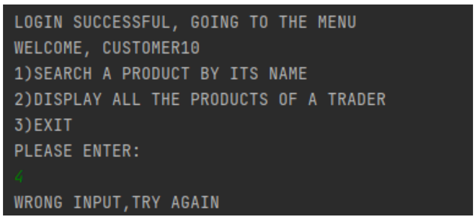


```java
if(option2==1) {      //MEET ORDER
    System.out.println("Order is met successfully");
    //erase from products.txt
    erase_met_order(id);
}
else {            //CANCEL ORDER
    System.out.println("Order is cancelled successfully");
}
```



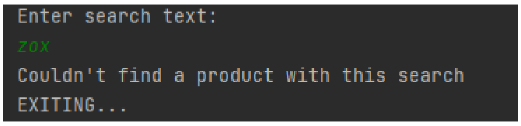
```java
default:
    System.out.println("Wrong input,try again...");
    break;
```

Customer entered wrong input menu



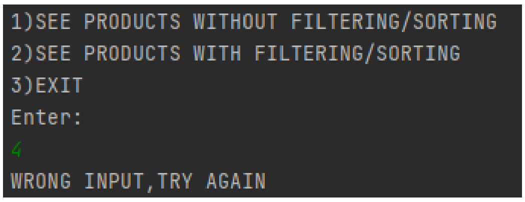
```
LOGIN SUCCESSFUL, GOING TO THE MENU
WELCOME, CUSTOMER10
1)SEARCH A PRODUCT BY ITS NAME
2)DISPLAY ALL THE PRODUCTS OF A TRADER
3)EXIT
PLEASE ENTER:
4
WRONG INPUT,TRY AGAIN
```

Customer entered a text and there is no product with this text



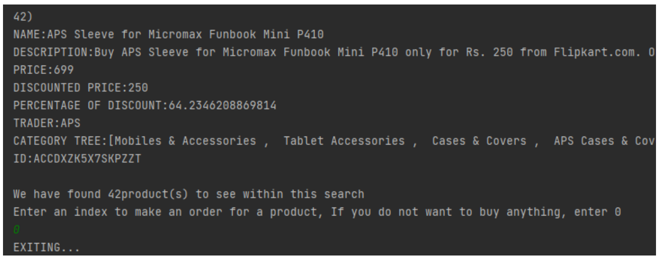
```
Enter search text:
zox
Couldn't find a product with this search
EXITING...
```

Customer entered wrong search menu input



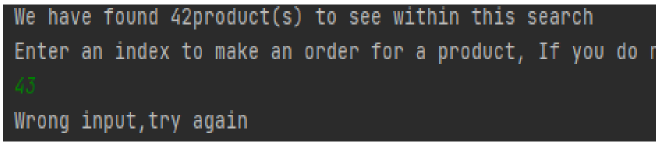
```
1)SEE PRODUCTS WITHOUT FILTERING/SORTING
2)SEE PRODUCTS WITH FILTERING/SORTING
3)EXIT
Enter:
4
WRONG INPUT,TRY AGAIN
```

Customer's search is successful



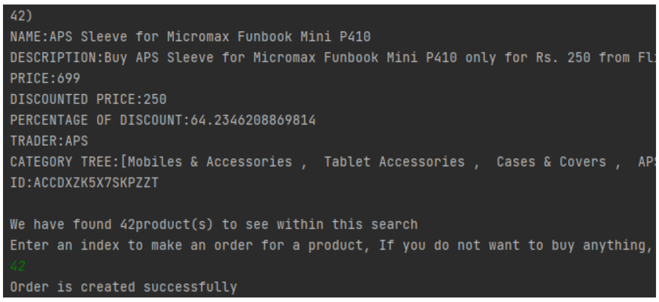
```
42)
NAME:APS Sleeve for Micromax Funbook Mini P410
DESCRIPTION:Buy APS Sleeve for Micromax Funbook Mini P410 only for Rs. 250 from Flipkart.com. O
PRICE:699
DISCOUNTED PRICE:250
PERCENTAGE OF DISCOUNT:64.2346208869814
TRADER:APS
CATEGORY TREE:[Mobiles & Accessories ,  Tablet Accessories ,  Cases & Covers ,  APS Cases & Cov
ID:ACCDXZK5X7SKPZZT

We have found 42product(s) to see within this search
Enter an index to make an order for a product, If you do not want to buy anything, enter 0
0
EXITING...
```

Customer entered wrong product index after search



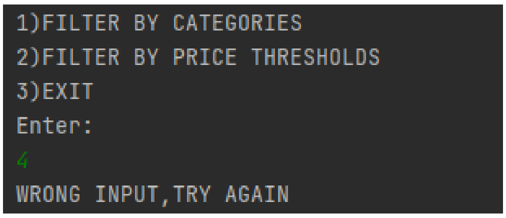
```
We have found 42product(s) to see within this search
Enter an index to make an order for a product, If you do n
43
Wrong input,try again
```

Customer created order successfully

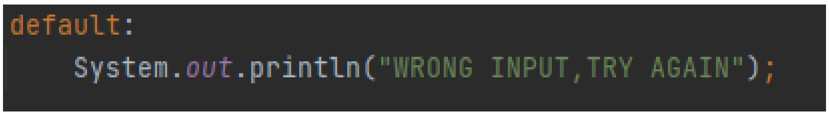


```
42)
NAME:APS Sleeve for Micromax Funbook Mini P410
DESCRIPTION:Buy APS Sleeve for Micromax Funbook Mini P410 only for Rs. 250 from Fl
PRICE:699
DISCOUNTED PRICE:250
PERCENTAGE OF DISCOUNT:64.2346208869814
TRADER:APS
CATEGORY TREE:[Mobiles & Accessories ,  Tablet Accessories ,  Cases & Covers ,  AP
ID:ACCDXZK5X7SKPZZT

We have found 42product(s) to see within this search
Enter an index to make an order for a product, If you do not want to buy anything,
42
Order is created successfully
```

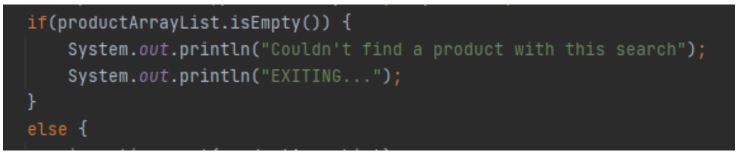
Customer entered wrong input menu when tries to filter products



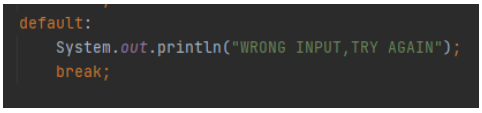
```
1)FILTER BY CATEGORIES
2)FILTER BY PRICE THRESHOLDS
3)EXIT
Enter:
4
WRONG INPUT,TRY AGAIN
```



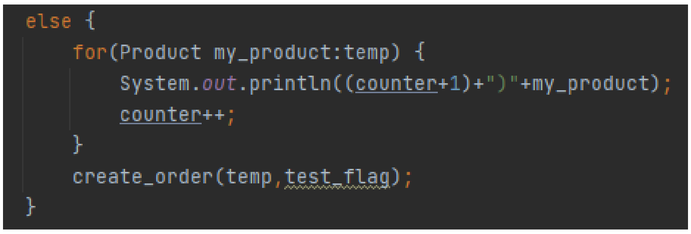
```java
default:
    System.out.println("WRONG INPUT,TRY AGAIN");
```



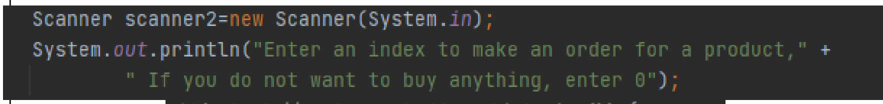
```java
if(productArrayList.isEmpty()) {
    System.out.println("Couldn't find a product with this search");
    System.out.println("EXITING...");
}
else {
```



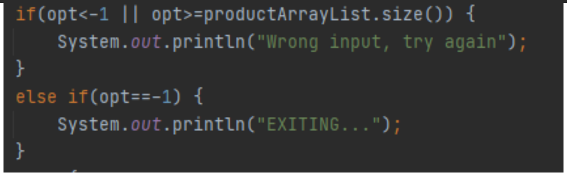
```java
default:
    System.out.println("WRONG INPUT,TRY AGAIN");
    break;
```



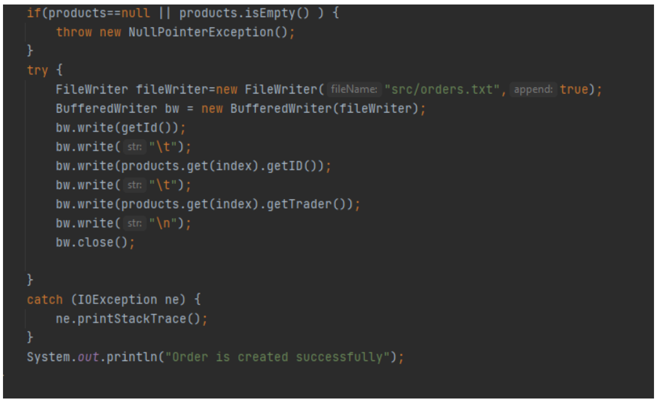
```java
else {
    for(Product my_product:temp) {
        System.out.println((counter+1)+")"+my_product);
        counter++;
    }
    create_order(temp,test_flag);
}
```



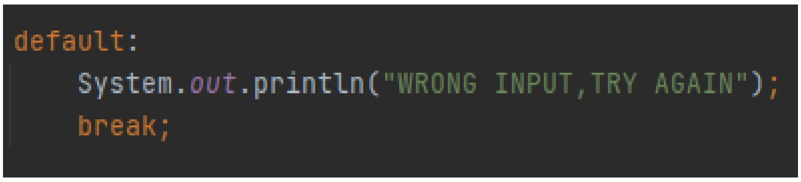
```java
Scanner scanner2=new Scanner(System.in);
System.out.println("Enter an index to make an order for a product," +
        " If you do not want to buy anything, enter 0");
```



```java
if(opt<-1 || opt>=productArrayList.size()) {
    System.out.println("Wrong input, try again");
}
else if(opt==-1) {
    System.out.println("EXITING...");
}
```

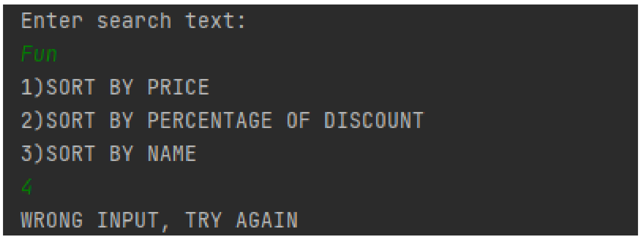


```java
if(products==null || products.isEmpty() ) {
    throw new NullPointerException();
}
try {
    FileWriter fileWriter=new FileWriter( fileName: "src/orders.txt", append: true);
    BufferedWriter bw = new BufferedWriter(fileWriter);
    bw.write(getId());
    bw.write( str: "\t");
    bw.write(products.get(index).getID());
    bw.write( str: "\t");
    bw.write(products.get(index).getTrader());
    bw.write( str: "\n");
    bw.close();

}
catch (IOException ne) {
    ne.printStackTrace();
}
System.out.println("Order is created successfully");
```

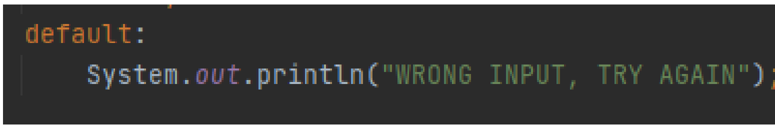


```java
default:
    System.out.println("WRONG INPUT,TRY AGAIN");
    break;
```

Customer entered wrong input menu when tries to sort products

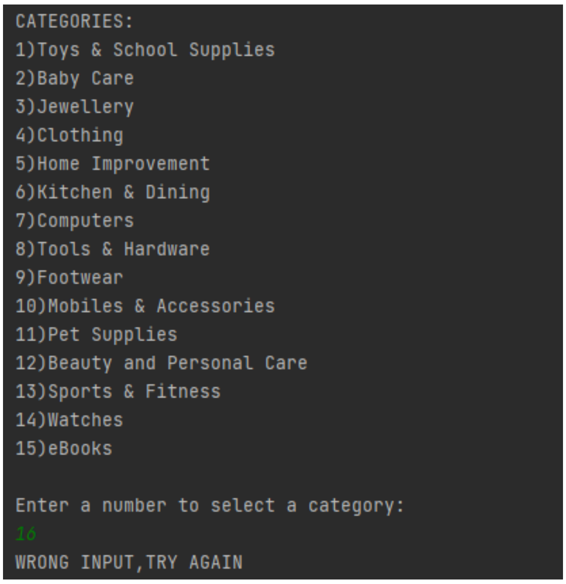


```
Enter search text:
Fun
1)SORT BY PRICE
2)SORT BY PERCENTAGE OF DISCOUNT
3)SORT BY NAME
4
WRONG INPUT, TRY AGAIN
```

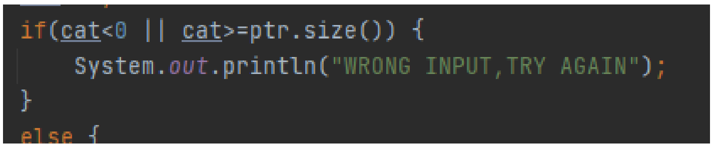


```
default:
    System.out.println("WRONG INPUT, TRY AGAIN");
```

Customer entered wrong input menu when tries to select a category

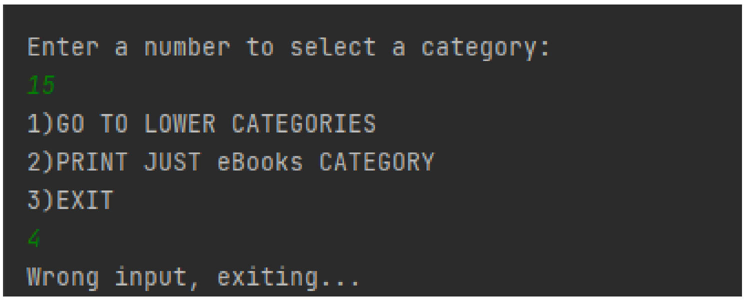


```
CATEGORIES:
1)Toys & School Supplies
2)Baby Care
3)Jewellery
4)Clothing
5)Home Improvement
6)Kitchen & Dining
7)Computers
8)Tools & Hardware
9)Footwear
10)Mobiles & Accessories
11)Pet Supplies
12)Beauty and Personal Care
13)Sports & Fitness
14)Watches
15)eBooks

Enter a number to select a category:
16
WRONG INPUT,TRY AGAIN
```

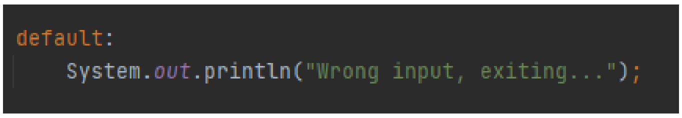


```
if(cat<0 || cat>=ptr.size()) {
    System.out.println("WRONG INPUT,TRY AGAIN");
}
else {
```

Customer entered wrong input menu when selecting lower categories or print products

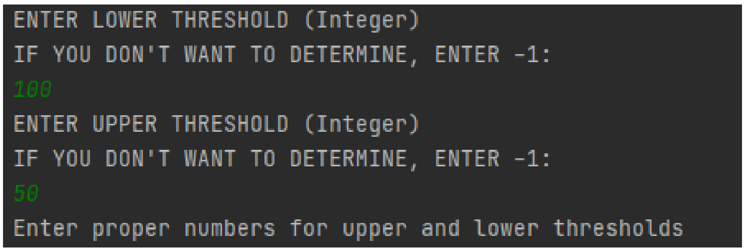


```
Enter a number to select a category:
15
1)GO TO LOWER CATEGORIES
2)PRINT JUST eBooks CATEGORY
3)EXIT
4
Wrong input, exiting...
```

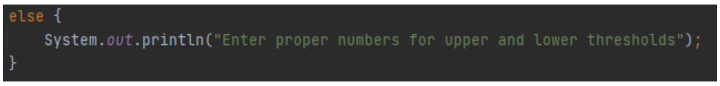


```
default:
    System.out.println("Wrong input, exiting...");
```

Customer entered wrong values for upper and lower thresholds

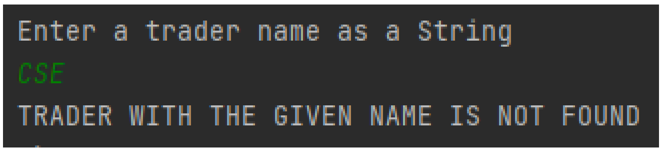


```
ENTER LOWER THRESHOLD (Integer)
IF YOU DON'T WANT TO DETERMINE, ENTER -1:
100
ENTER UPPER THRESHOLD (Integer)
IF YOU DON'T WANT TO DETERMINE, ENTER -1:
50
Enter proper numbers for upper and lower thresholds
```

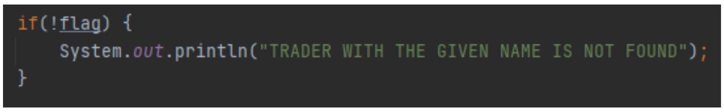


```
else {
    System.out.println("Enter proper numbers for upper and lower thresholds");
}
```

Customer entered wrong trader name when tries to display products of a trader



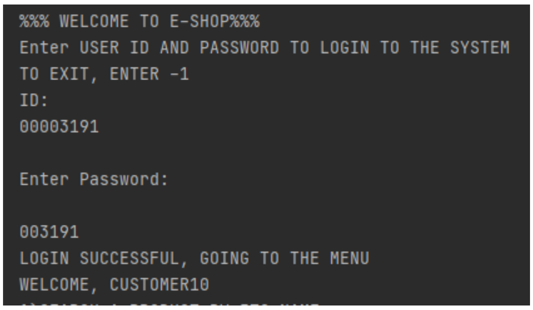
```
Enter a trader name as a String
CSE
TRADER WITH THE GIVEN NAME IS NOT FOUND
```



```
if(!flag) {
    System.out.println("TRADER WITH THE GIVEN NAME IS NOT FOUND");
}
```

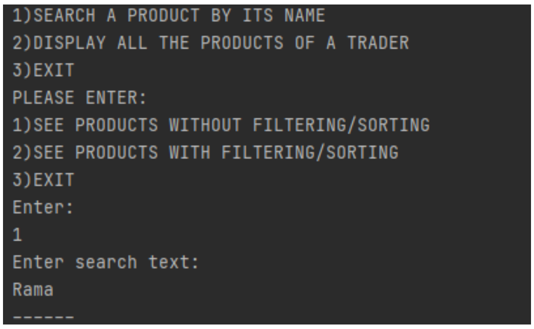
4. RUNNING COMMANDS AND RESULTS

Customer logins to the system



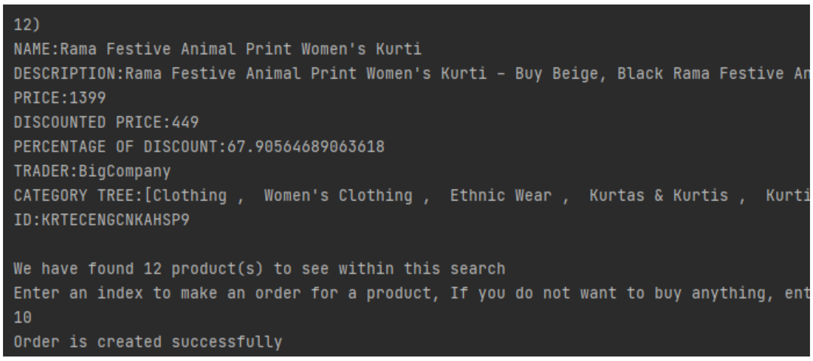
```
%%% WELCOME TO E-SHOP%%%
Enter USER ID AND PASSWORD TO LOGIN TO THE SYSTEM
TO EXIT, ENTER -1
ID:
00003191

Enter Password:

003191
LOGIN SUCCESSFUL, GOING TO THE MENU
WELCOME, CUSTOMER10
```

Customer searches a product by giving name



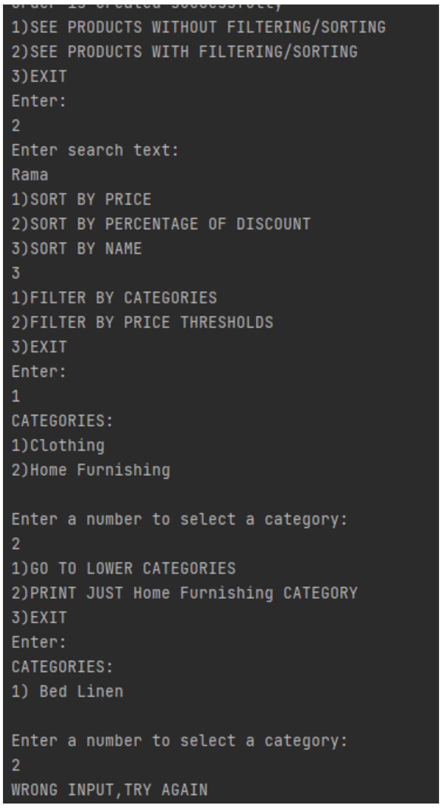
```
1)SEARCH A PRODUCT BY ITS NAME
2)DISPLAY ALL THE PRODUCTS OF A TRADER
3)EXIT
PLEASE ENTER:
1)SEE PRODUCTS WITHOUT FILTERING/SORTING
2)SEE PRODUCTS WITH FILTERING/SORTING
3)EXIT
Enter:
1
Enter search text:
Rama
------
```

Customer creates an order



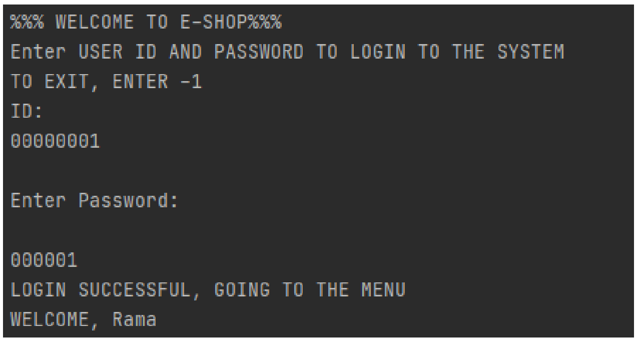
```
12)
NAME:Rama Festive Animal Print Women's Kurti
DESCRIPTION:Rama Festive Animal Print Women's Kurti - Buy Beige, Black Rama Festive An
PRICE:1399
DISCOUNTED PRICE:449
PERCENTAGE OF DISCOUNT:67.90564689063618
TRADER:BigCompany
CATEGORY TREE:[Clothing ,  Women's Clothing ,  Ethnic Wear ,  Kurtas & Kurtis ,  Kurti
ID:KRTECENGCNKAHSP9

We have found 12 product(s) to see within this search
Enter an index to make an order for a product, If you do not want to buy anything, ent
10
Order is created successfully
```

Customer wants to filter-sort



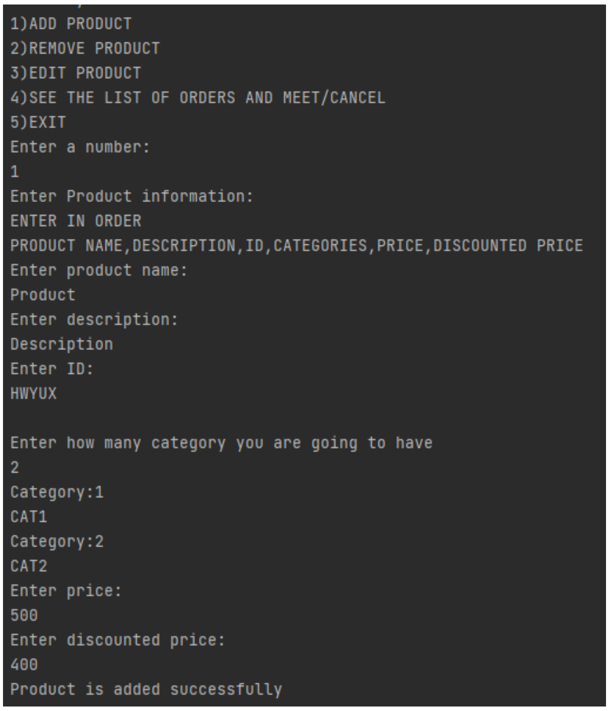
```
1)SEE PRODUCTS WITHOUT FILTERING/SORTING
2)SEE PRODUCTS WITH FILTERING/SORTING
3)EXIT
Enter:
2
Enter search text:
Rama
1)SORT BY PRICE
2)SORT BY PERCENTAGE OF DISCOUNT
3)SORT BY NAME
3
1)FILTER BY CATEGORIES
2)FILTER BY PRICE THRESHOLDS
3)EXIT
Enter:
1
CATEGORIES:
1)Clothing
2)Home Furnishing

Enter a number to select a category:
2
1)GO TO LOWER CATEGORIES
2)PRINT JUST Home Furnishing CATEGORY
3)EXIT
Enter:
CATEGORIES:
1) Bed Linen

Enter a number to select a category:
2
WRONG INPUT,TRY AGAIN
```

Trader logins to the system



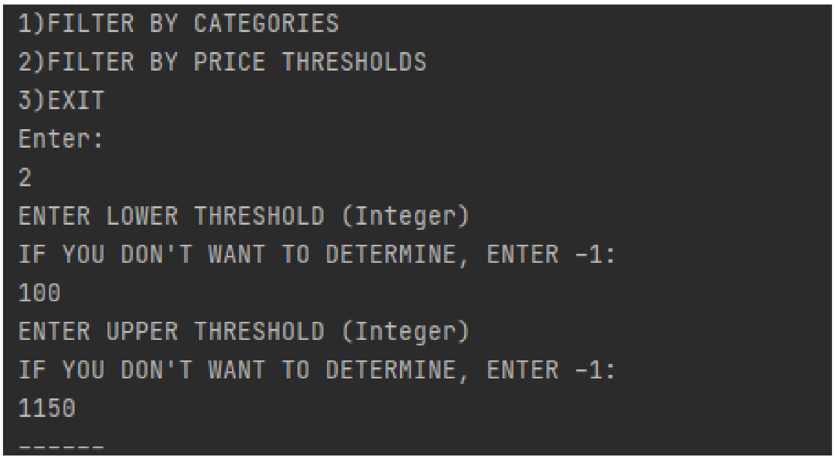
```
%%% WELCOME TO E-SHOP%%%
Enter USER ID AND PASSWORD TO LOGIN TO THE SYSTEM
TO EXIT, ENTER -1
ID:
00000001

Enter Password:

000001
LOGIN SUCCESSFUL, GOING TO THE MENU
WELCOME, Rama
```

Trader adds a product



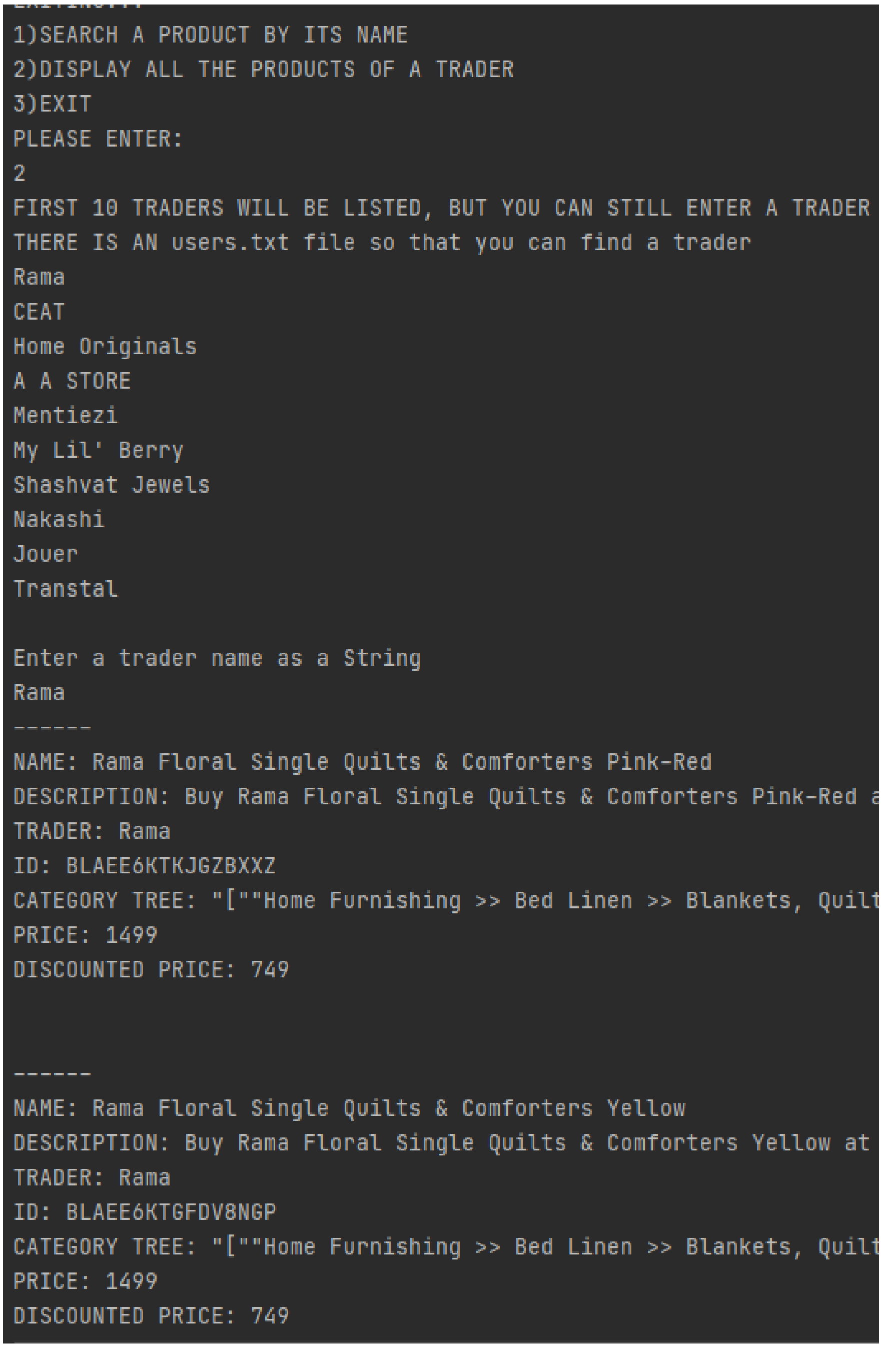
```
1)ADD PRODUCT
2)REMOVE PRODUCT
3)EDIT PRODUCT
4)SEE THE LIST OF ORDERS AND MEET/CANCEL
5)EXIT
Enter a number:
1
Enter Product information:
ENTER IN ORDER
PRODUCT NAME,DESCRIPTION,ID,CATEGORIES,PRICE,DISCOUNTED PRICE
Enter product name:
Product
Enter description:
Description
Enter ID:
HWYUX

Enter how many category you are going to have
2
Category:1
CAT1
Category:2
CAT2
Enter price:
500
Enter discounted price:
400
Product is added successfully
```

Customer wants to filter by price threshold



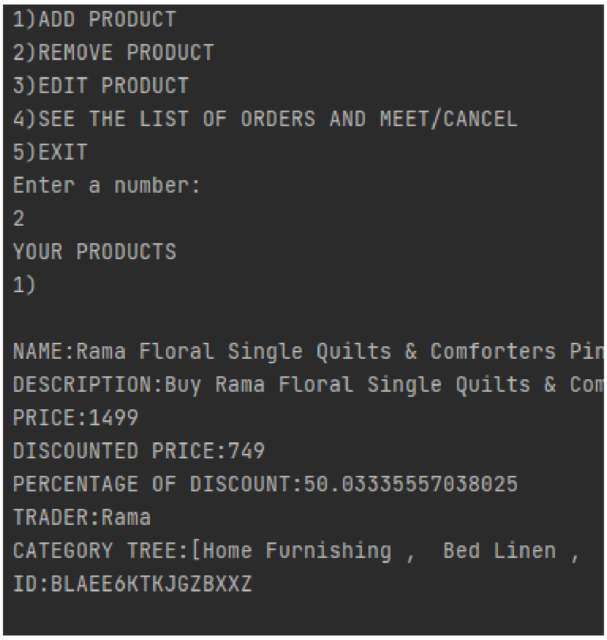
```
1)FILTER BY CATEGORIES
2)FILTER BY PRICE THRESHOLDS
3)EXIT
Enter:
2
ENTER LOWER THRESHOLD (Integer)
IF YOU DON'T WANT TO DETERMINE, ENTER -1:
100
ENTER UPPER THRESHOLD (Integer)
IF YOU DON'T WANT TO DETERMINE, ENTER -1:
1150
------
```

Customer wants to display products of a trader



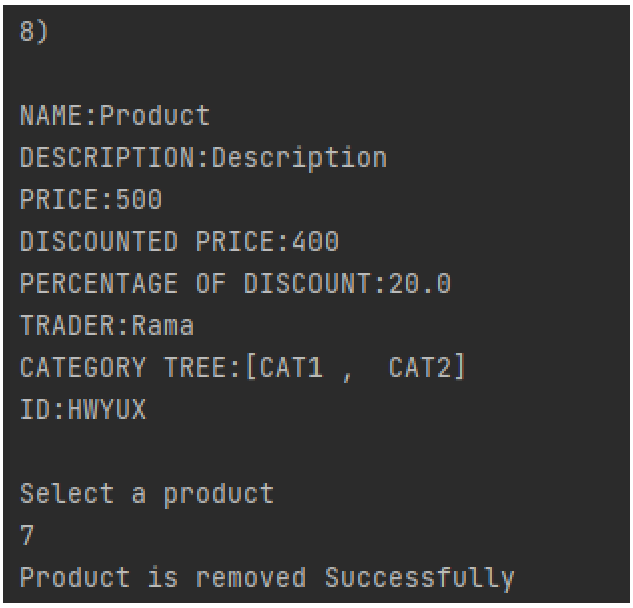
```
1)SEARCH A PRODUCT BY ITS NAME
2)DISPLAY ALL THE PRODUCTS OF A TRADER
3)EXIT
PLEASE ENTER:
2
FIRST 10 TRADERS WILL BE LISTED, BUT YOU CAN STILL ENTER A TRADER
THERE IS AN users.txt file so that you can find a trader
Rama
CEAT
Home Originals
A A STORE
Mentiezi
My Lil' Berry
Shashvat Jewels
Nakashi
Jouer
Transtal

Enter a trader name as a String
Rama
------
NAME: Rama Floral Single Quilts & Comforters Pink-Red
DESCRIPTION: Buy Rama Floral Single Quilts & Comforters Pink-Red a
TRADER: Rama
ID: BLAEE6KTKJGZBXXZ
CATEGORY TREE: "[""Home Furnishing >> Bed Linen >> Blankets, Quilt
PRICE: 1499
DISCOUNTED PRICE: 749


------
NAME: Rama Floral Single Quilts & Comforters Yellow
DESCRIPTION: Buy Rama Floral Single Quilts & Comforters Yellow at
TRADER: Rama
ID: BLAEE6KTGFDV8NGP
CATEGORY TREE: "[""Home Furnishing >> Bed Linen >> Blankets, Quilt
PRICE: 1499
DISCOUNTED PRICE: 749
```

Trader removes a product



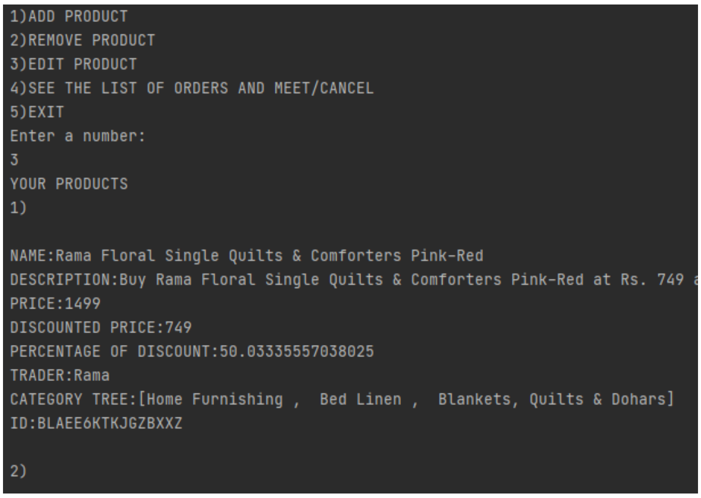
```
1)ADD PRODUCT
2)REMOVE PRODUCT
3)EDIT PRODUCT
4)SEE THE LIST OF ORDERS AND MEET/CANCEL
5)EXIT
Enter a number:
2
YOUR PRODUCTS
1)

NAME:Rama Floral Single Quilts & Comforters Pin
DESCRIPTION:Buy Rama Floral Single Quilts & Com
PRICE:1499
DISCOUNTED PRICE:749
PERCENTAGE OF DISCOUNT:50.03335557038025
TRADER:Rama
CATEGORY TREE:[Home Furnishing ,  Bed Linen ,
ID:BLAEE6KTKJGZBXXZ
```

Trader removes a product (continue)



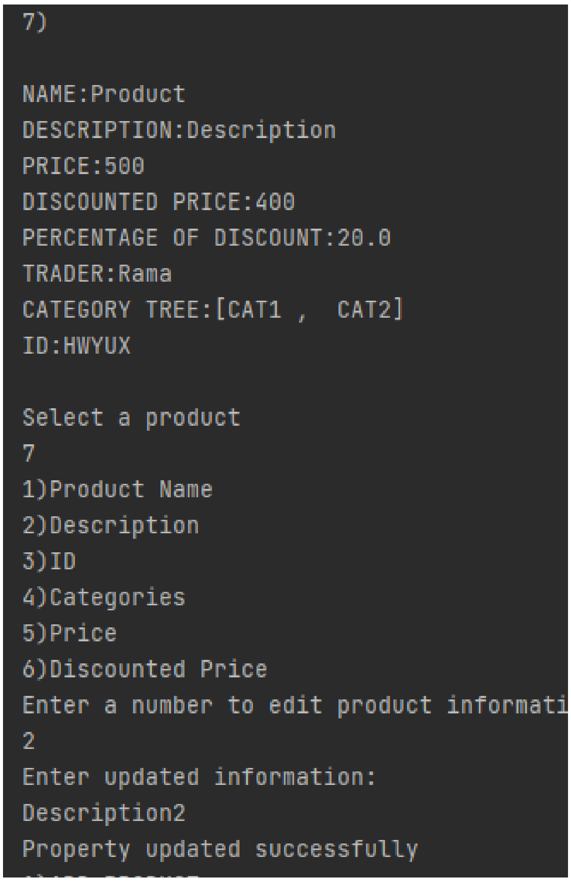
```
8)

NAME:Product
DESCRIPTION:Description
PRICE:500
DISCOUNTED PRICE:400
PERCENTAGE OF DISCOUNT:20.0
TRADER:Rama
CATEGORY TREE:[CAT1 ,  CAT2]
ID:HWYUX

Select a product
7
Product is removed Successfully
```

Trader edits a product



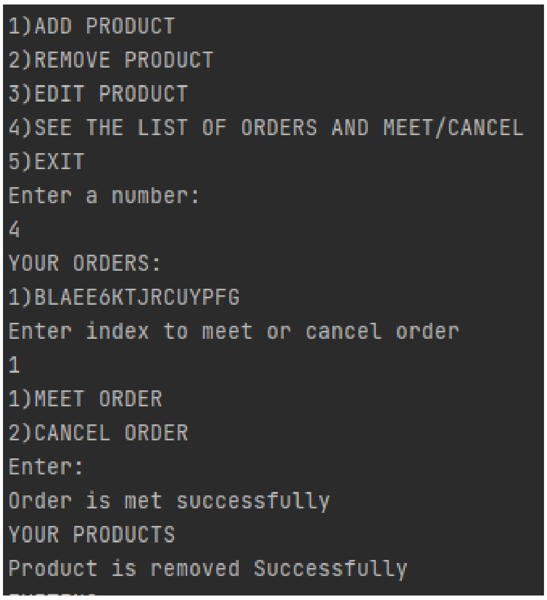
```
1)ADD PRODUCT
2)REMOVE PRODUCT
3)EDIT PRODUCT
4)SEE THE LIST OF ORDERS AND MEET/CANCEL
5)EXIT
Enter a number:
3
YOUR PRODUCTS
1)

NAME:Rama Floral Single Quilts & Comforters Pink-Red
DESCRIPTION:Buy Rama Floral Single Quilts & Comforters Pink-Red at Rs. 749
PRICE:1499
DISCOUNTED PRICE:749
PERCENTAGE OF DISCOUNT:50.03335557038025
TRADER:Rama
CATEGORY TREE:[Home Furnishing ,  Bed Linen ,  Blankets, Quilts & Dohars]
ID:BLAEE6KTKJGZBXXZ

2)
```

Trader edits a product (continued)



```
7)

NAME:Product
DESCRIPTION:Description
PRICE:500
DISCOUNTED PRICE:400
PERCENTAGE OF DISCOUNT:20.0
TRADER:Rama
CATEGORY TREE:[CAT1 ,  CAT2]
ID:HWYUX

Select a product
7
1)Product Name
2)Description
3)ID
4)Categories
5)Price
6)Discounted Price
Enter a number to edit product informati
2
Enter updated information:
Description2
Property updated successfully
```

Trader meets/cancels an order



```
1)ADD PRODUCT
2)REMOVE PRODUCT
3)EDIT PRODUCT
4)SEE THE LIST OF ORDERS AND MEET/CANCEL
5)EXIT
Enter a number:
4
YOUR ORDERS:
1)BLAEE6KTJRCUYPFG
Enter index to meet or cancel order
1
1)MEET ORDER
2)CANCEL ORDER
Enter:
Order is met successfully
YOUR PRODUCTS
Product is removed Successfully
```

- **Some helper informations**

Some ID-passwords of the traders and customers

	ID	PASSWORD
Rama	00000001	000001
CEAT	00000002	000002
Home Originals	00000003	000003
A A STORE	00000004	000004
Mentiezi	00000005	000005
My Lil' Berry	00000006	000006
Shashvat Jewels	00000007	000007
Nakashi	00000008	000008
Jouer	00000009	000009
Transtal	00000010	000010
CUSTOMER1	00003182	003182
CUSTOMER2	00003183	003183
CUSTOMER3	00003184	003184
CUSTOMER4	00003185	003185
CUSTOMER5	00003186	003186
CUSTOMER6	00003187	003187
CUSTOMER7	00003188	003188
CUSTOMER8	00003189	003189
CUSTOMER9	00003190	003190
CUSTOMER10	00003191	003191