Institute of Computer Engineering Technology

Coursework

Coursework	WEEK 07 - PRF Final Coursework
Name	Burger Shop
Ass. Date	30th December 2023

Case Study

The iHungry Burger shop, which was recently started in our city, has a large number of transactions every day, Therefor the Burger Shop owner requires a system to manage orders from Burger Shop. As you are a talented iCET student, they have thought to give you a chance to make a system for them.

Figure 1 – Usecase Diagram

Requirement

You are supposed to create a Java application to manage Burger Shop. In the application, you need to implement the following use cases.

When you run the application, you should come up with something similar to the following Command Line Interface (CLI), where the user can enter an option number that he wants to execute. This will be the Home Page of the application that you will be developing.

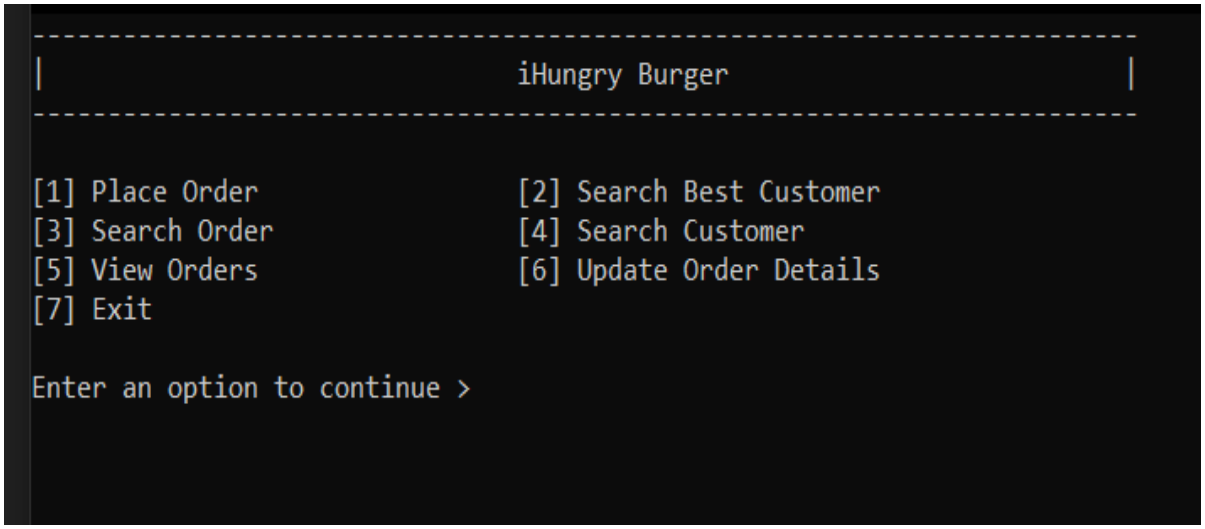

Figure 2 – Home Page of the iHungry Burger Shop Management System

01. Place Order

The system keeps five details related to the Place Order. They are Order ID, Customer ID, Customer Name, Burger Quantity, and Order Status.

- Order ID – The Order ID should be generated by the system and the Order ID should start with 'B' and should have 4 letters. When the user selects the Place Order option on the home page when the Place Order window is loaded, the order ID should be generated by the system and the user should place the order under that ID. Order ID can not be generated randomly and generate the next Order ID according to the last Order ID. Order ID can not be repeated.
- Customer ID – The user should input Customer ID(Customer phone number is the Customer ID). When entering the phone number, it should be validated. Phone numbers should start with "0" and must have 10 numbers. If the user has entered an invalid phone number (as an example, start without "0" or the phone number has more than 10 numbers), the user should be kept prompted until he enters a valid phone number.
- Customer Name – After the user inputs the Customer ID, the System should check if this customer ID is already added to the Order, search for that Customer ID, and display the name of the customer. Customer ID can repeat in Place Order. If the Customer ID is not in the list, the User should enter the Customer Name.

- Burger Quantity - The user should input Burger Quantity. Any value greater than 0 can be input as Burger Quantity.
- Order Status - There are 3 order statuses. They are PREPARING, DELIVERED, CANCEL. The order status should be declared as a static final variable. Declare that static variable as 0 for PREPARING, 1 for DELIVERED, and 2 for CANCEL. When an order is placed, the system gives PREPARING as its initial status.

After entering details system should display the Total Bill Value. The quantity entered by the user should be multiplied by the value of the Burger and the total value should be calculated and displayed by the system. (The value of the Burger is Rs.500/= and it is constant)

After the order details are entered successfully, ask the user "Are you confirming the order " , if the user enters "Y" order details are added to the system and if the user enters "N" order details are not added to the system.

After the order is placed successfully, asked from the user "Do you want to place another order "If the user enters "Y" the user can place the order again, and if the user enters "N" the user can go to the main menu.

Figure 3 – Place Order

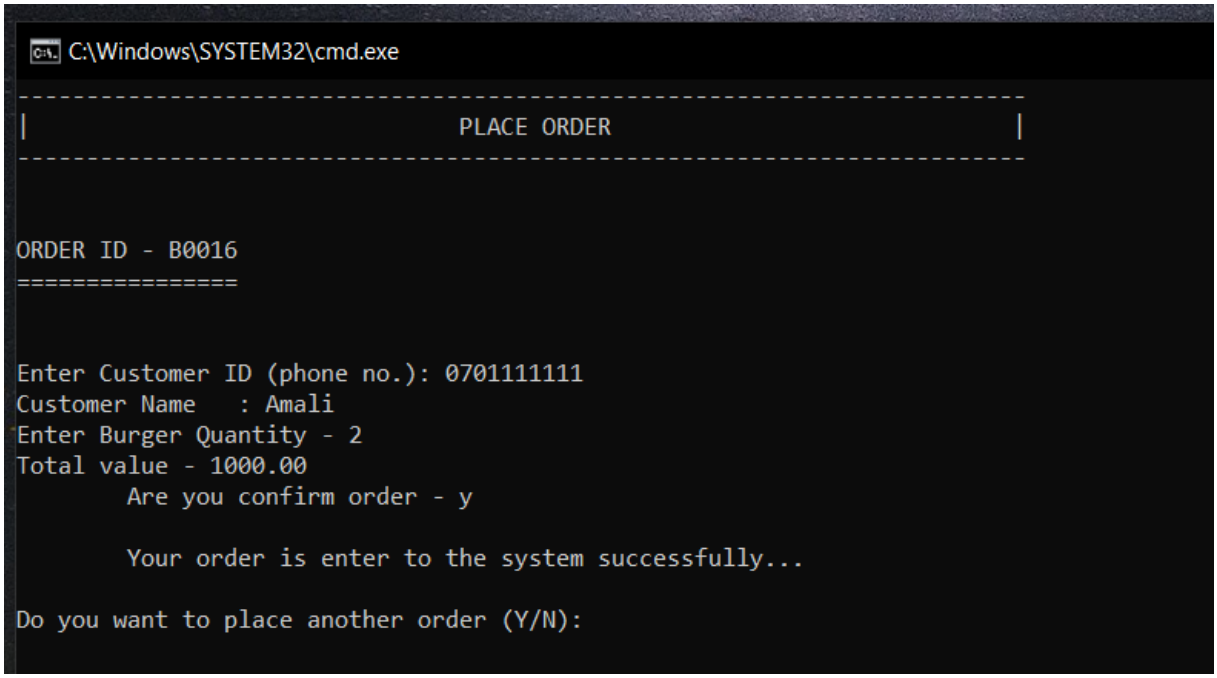

Figure 4 – Place Order (confirm order)

02. Search Best Customer (Demo video)

This displays all the Customers according to the descending order of their total of all purchases. Search the best customer table display in Figure 5. It should prompt whether the user wants to stay in here or go back to the main menu.

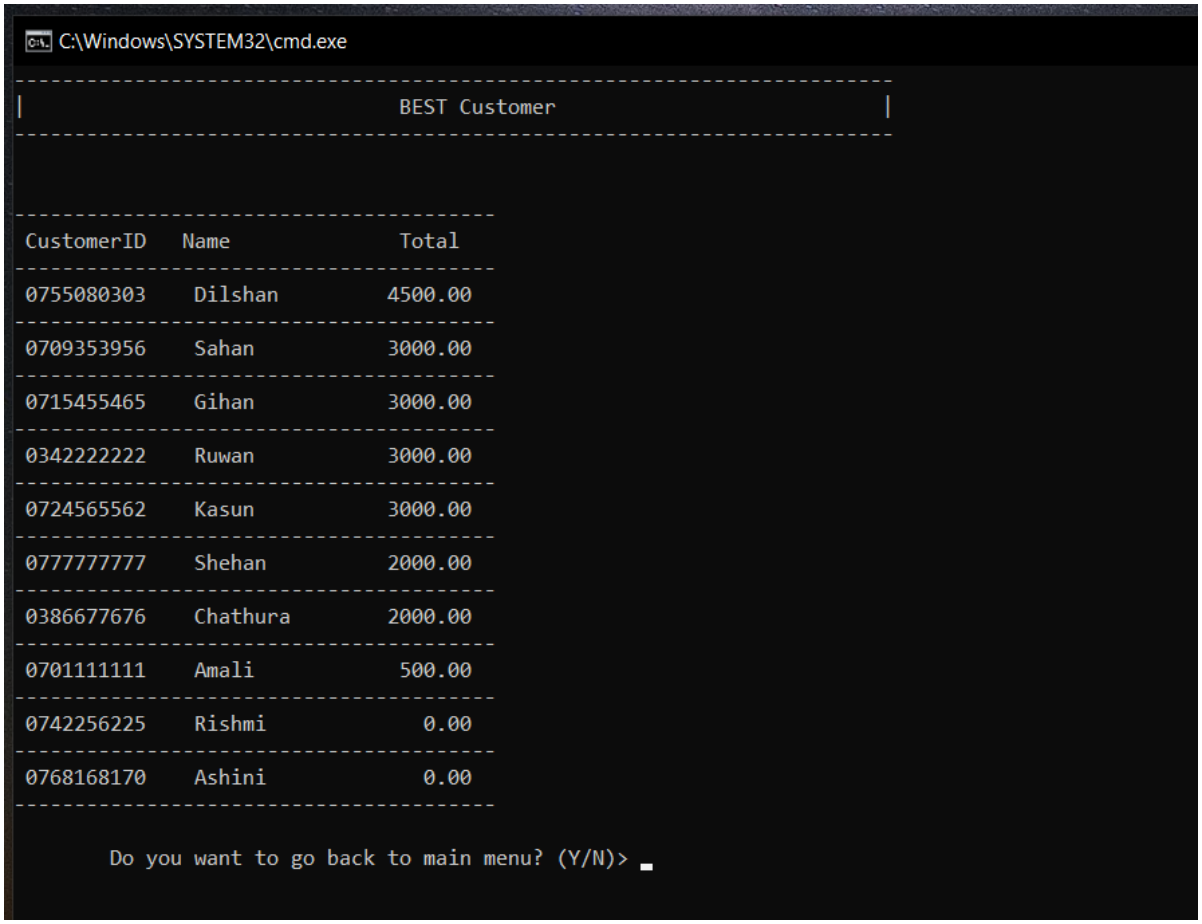

Figure 5 – Search Best Customer

03. Search Order (Demo)

With this option, the user can view Order details. First, the user needs to enter a valid Order ID. After the user enters the Order ID, the system should search if this Order ID is an existing Order ID or not. If this Order ID hasn't been added yet, then it should be displayed like Figure 6. If the Order ID is correct, display order details like Figure 7.

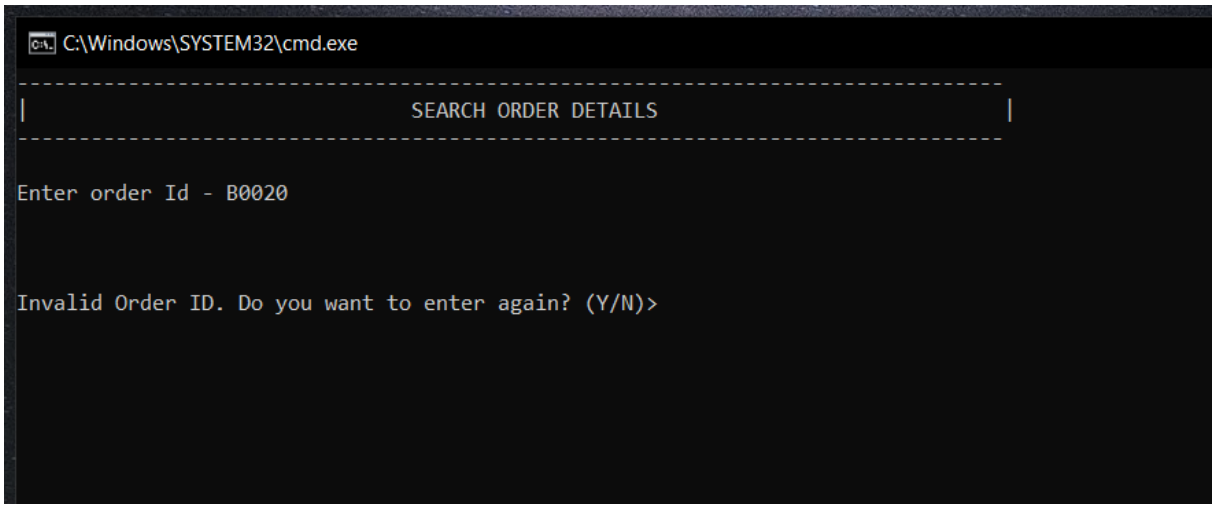

Figure 6 – Search Order (Warning – invalid order)

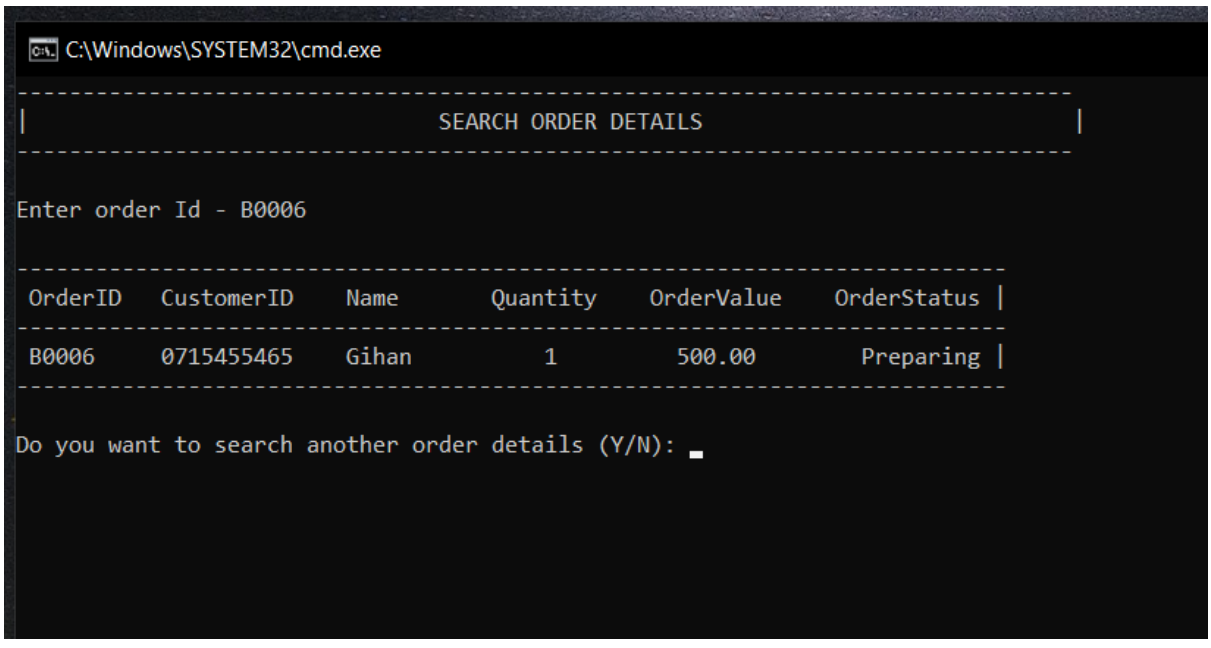

Figure 7 – Search Order (successful)

04. Search Customer (Demo)

With this option, the user can view Customer details. First, he needs to enter a valid Customer ID. otherwise, it should handle like previously. If the Customer ID hasn't been added yet, then it should be displayed like below.

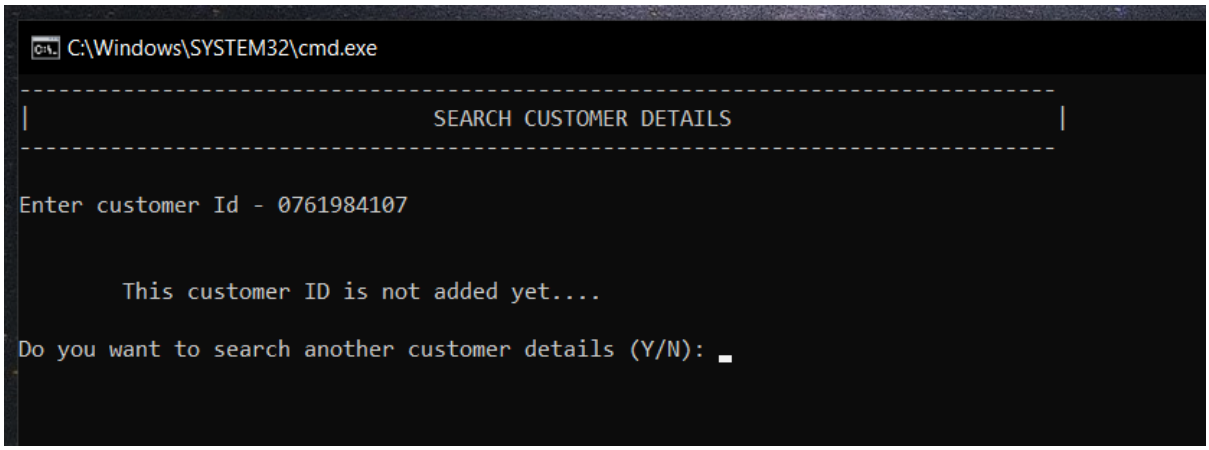

Figure 8 – Search Customer (Warning – customer not added)

If the Customer ID have been already added, then a comprehensive detail table should be displayed, which contains details of all orders placed by that Customer. Once the information has been displayed, the user should be prompted whether to continue seeking Customer details or go back to the main menu.

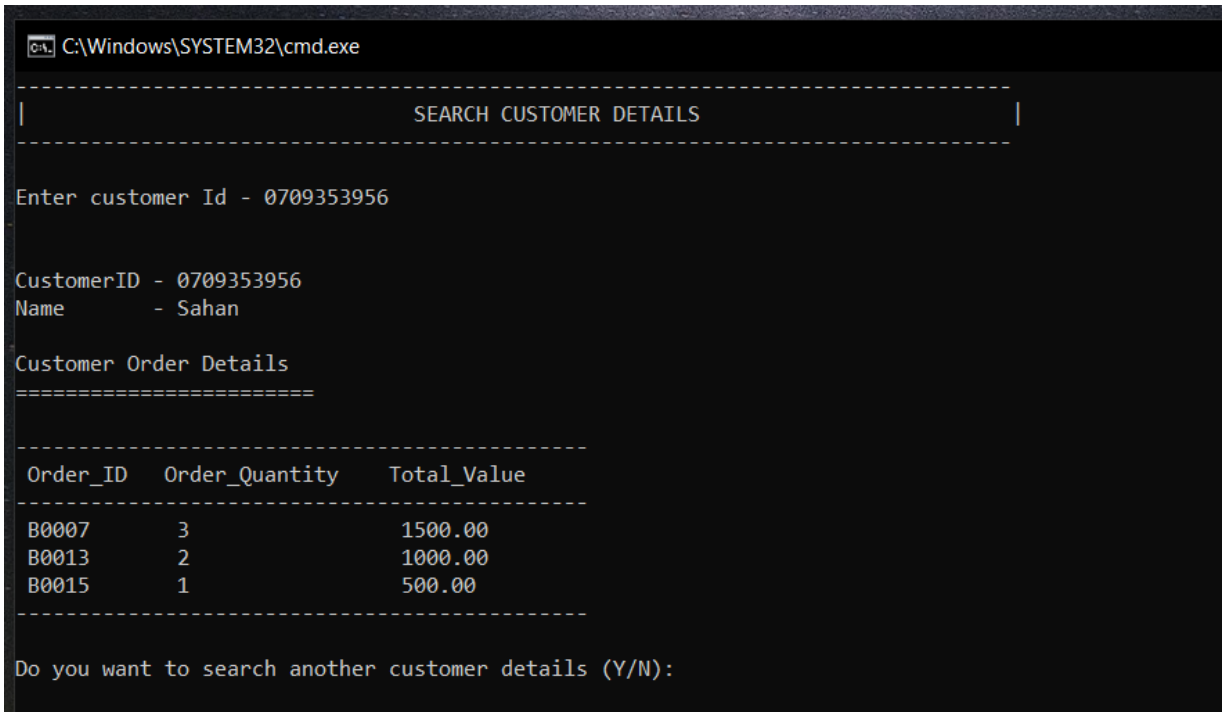

Figure 9 – Search Customer (successful)

05. View Orders (Demo)

With this option, the user can view Order details under the 3 order categories. If user enter 1, user can view Order details of Delivered Order(figure 11), If the user enters 2, the user can view the Order details of Preparing Order(figure 12) If the user enters 3, the user can view the Order details of the Canceled Order(figure 13). Just like above, in the end, it should prompt whether the user wants to stay here or go back to the main menu.

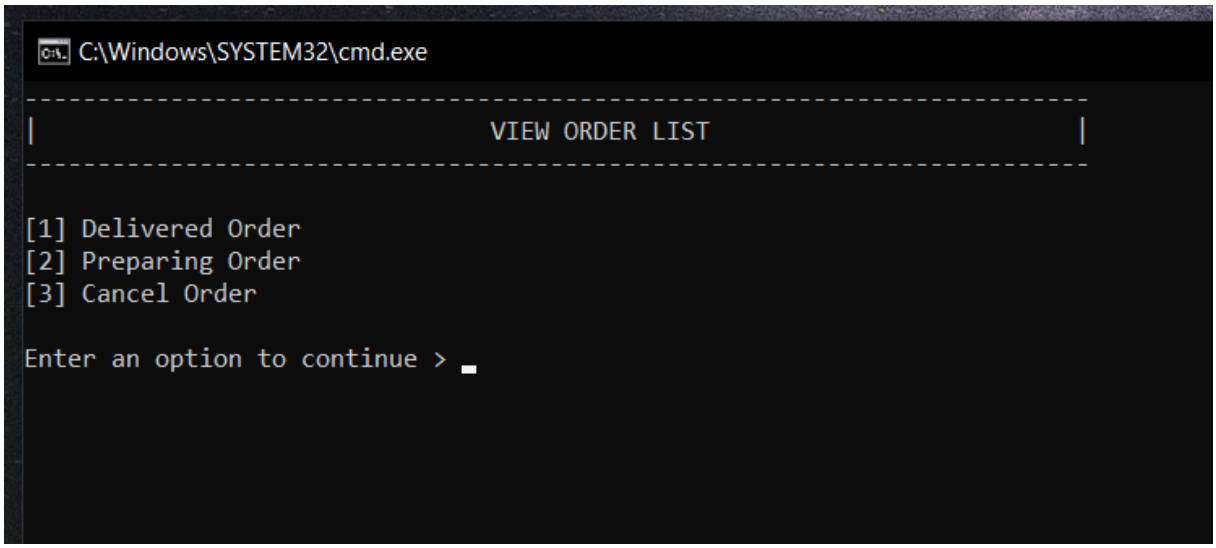

Figure 10 – Home Page of View Order List

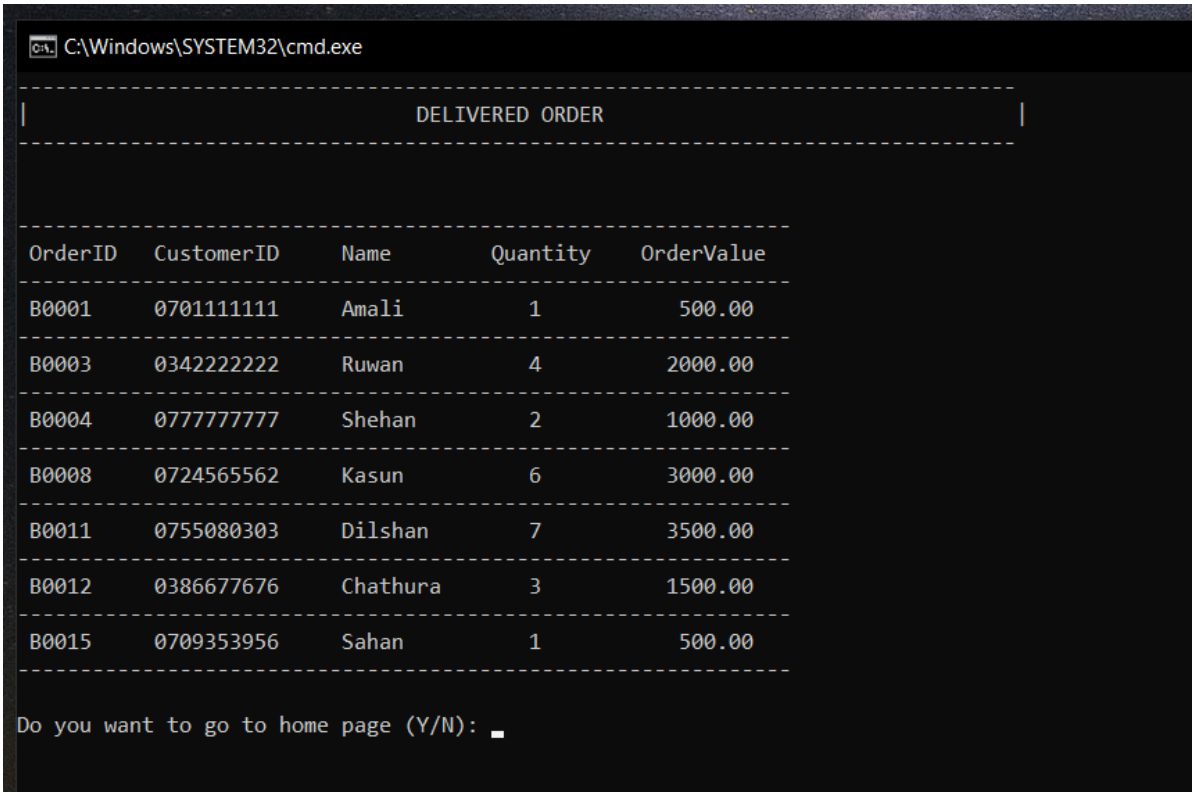

Figure 11 – display Delivered Order list

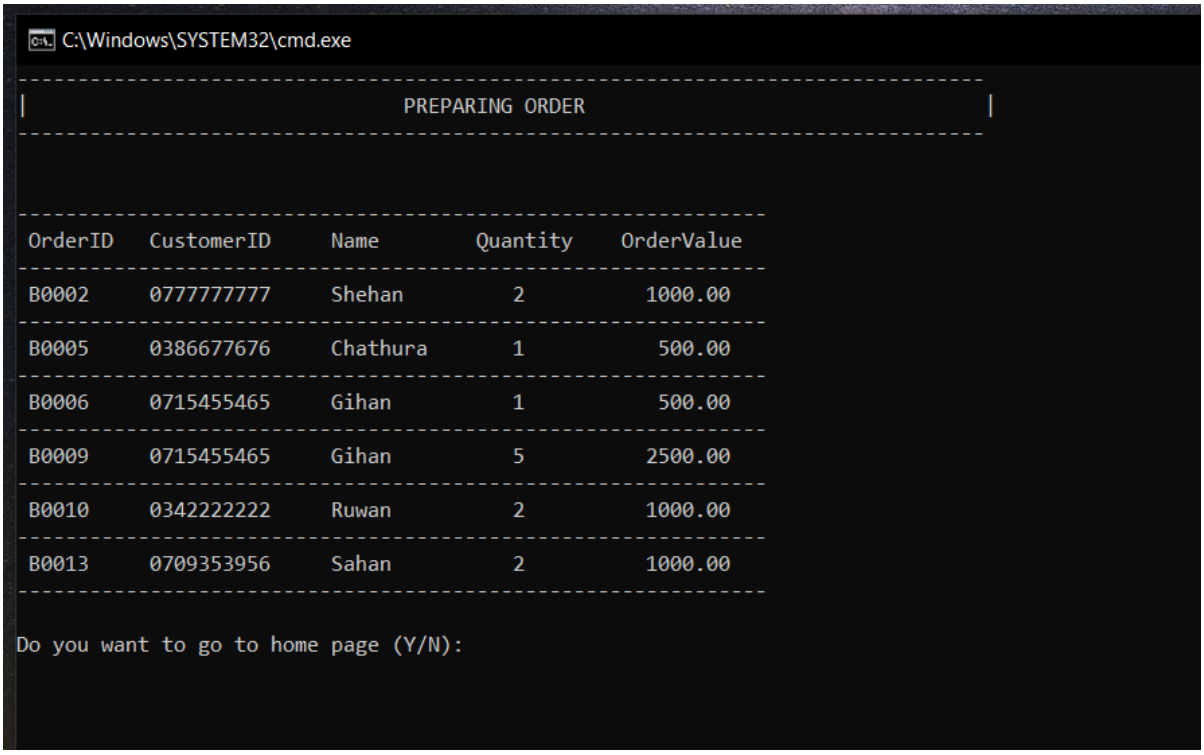

Figure 12 – display Preparing Order list

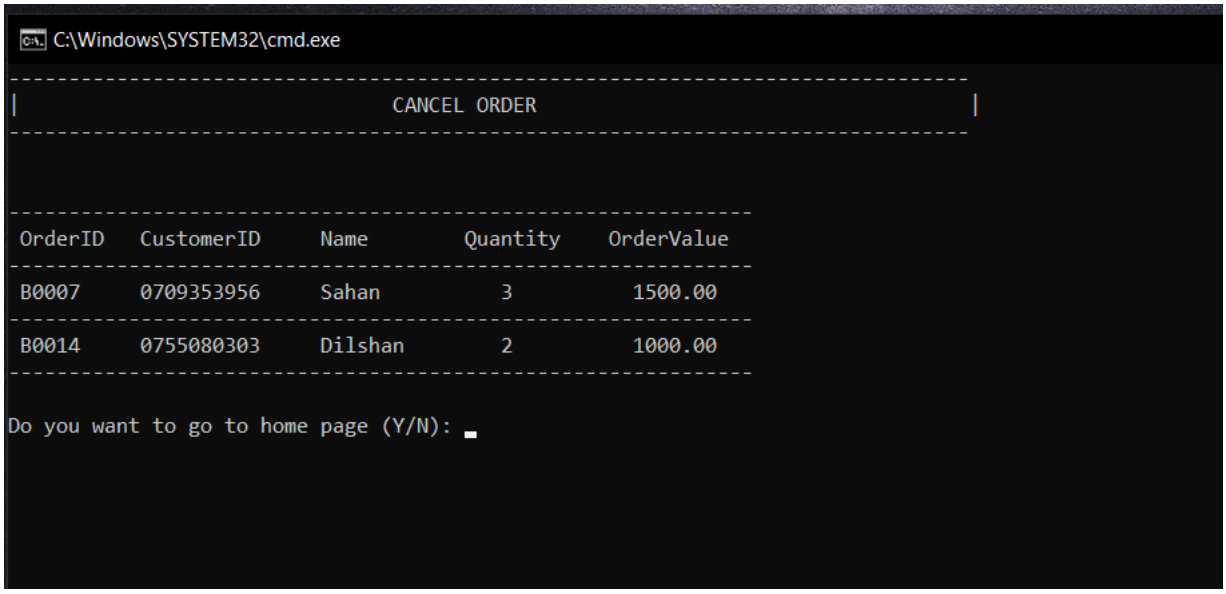

Figure 13 – display Cancel Order list

06. Update Order Details (Demo)

With this, the user can update previously added Order Details. First, the user needs to find the Order should be updated via Order ID(Invalid Order ID's should be handled like previously). Order Details can be updated only in preparing orders. If that order is already Delivered or Cancelled, it should display to the user as below.

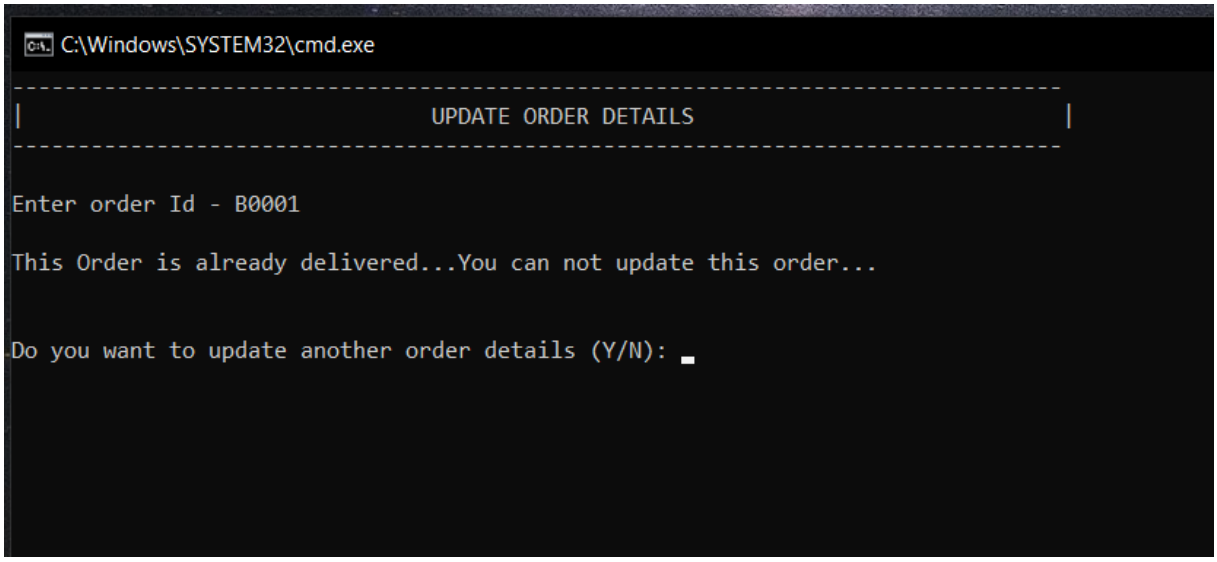

Figure 14 – Update Order (Warning – already delivered order)

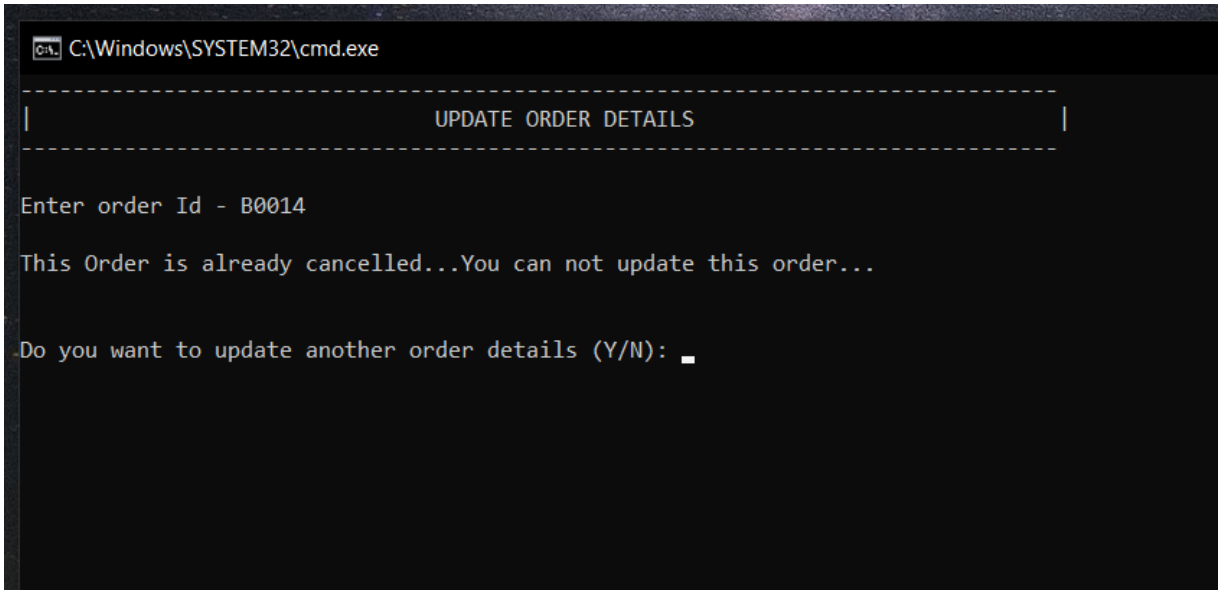

Figure 15 – Update Order (Warning – already cancel order)

Once the user has entered a valid Order ID, it is going to display the current details, which are the Customer ID, Customer name, Quantity, Order Value and Order status. After display Order Details asked from the user that user want to update Order Quantity or Order status.

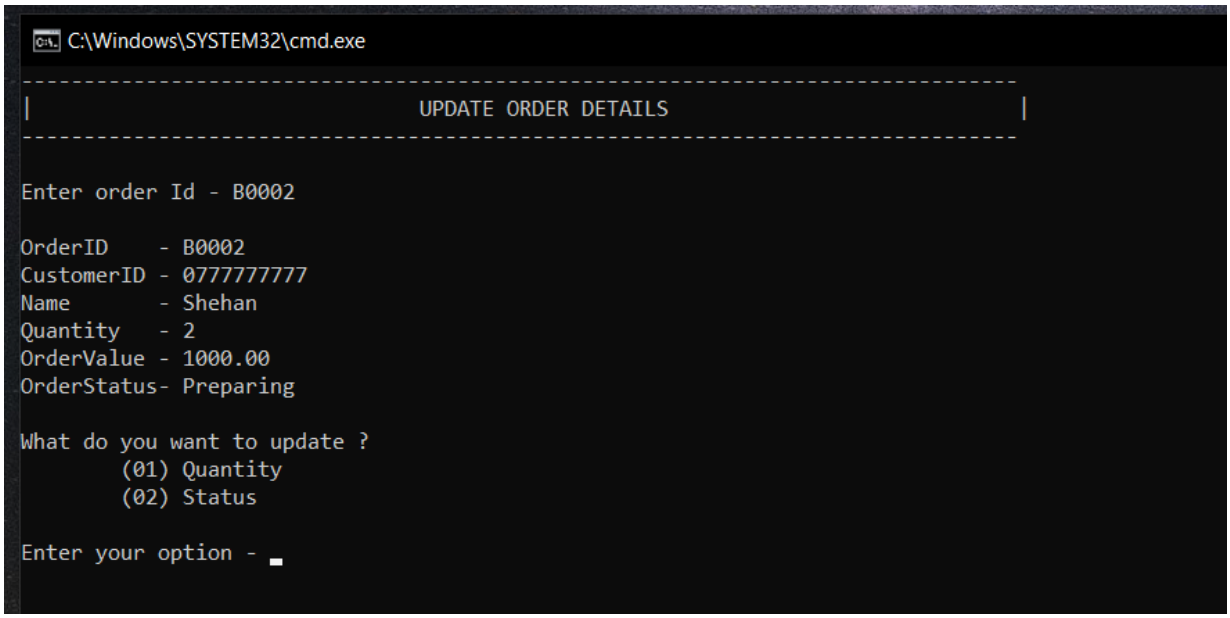

Figure 16 – update option of Update Order

If the user select Quantity(1), prompt the window related to the Quantity and user can update Burger Quantity. Then it prompts for new Order quantity. Again, quantity should be validated here. Once the update has been done successfully, display new Order Quantity and it should prompt whether to continue updating or go back to the main menu.

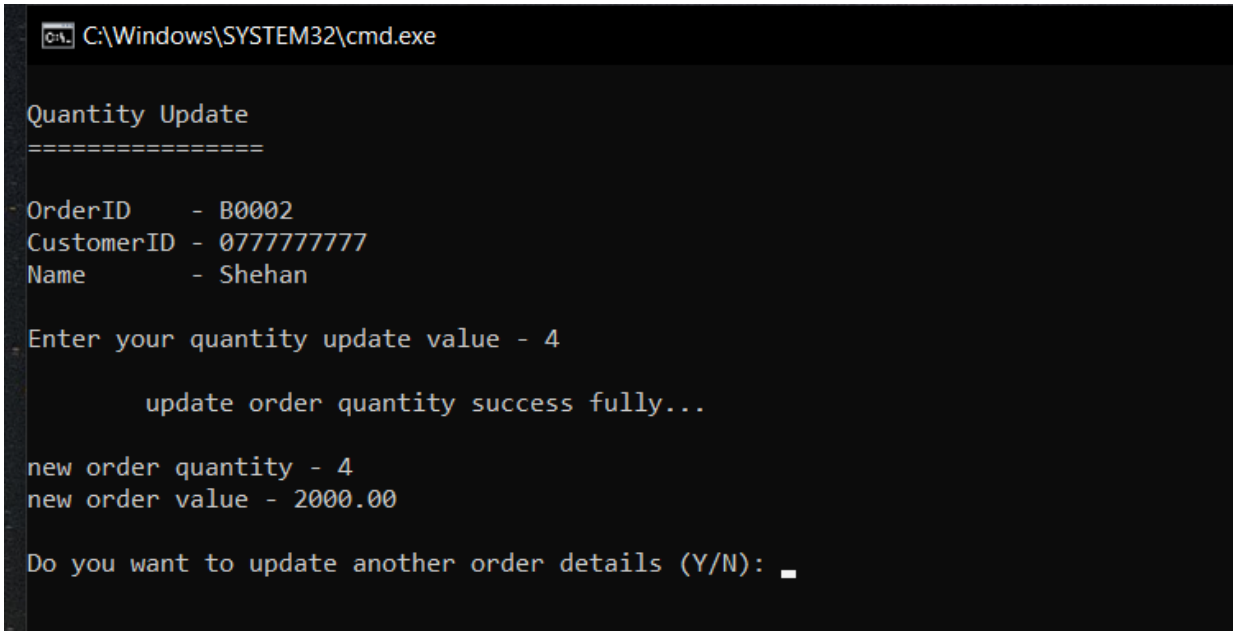

Figure 17 – Quantity Update (successful)

If user select Status(2), prompt the window related to the Order Status and user can update Order Status. Then it prompts for new Order Status. Once the update has been done successfully, display new Order Status and it should prompt whether to continue updating or go back to the main menu.

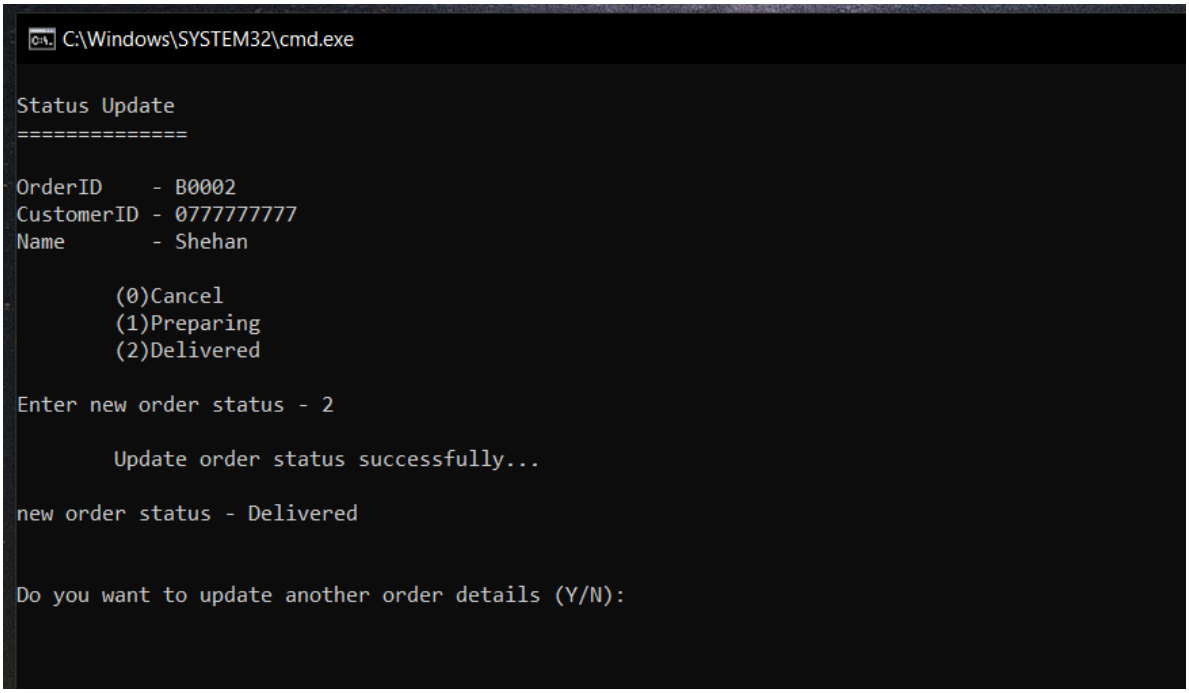

Figure 18 – Status Update (successful

Guideline

- Refer to the Coursework Guidelines at the end to understand the specific guidelines to be followed when developing the project required.
- You can't create classes except for the class that holds the main method.
- You can create as many methods and Arrays as you wish in the only class that you have.
- Use the Scanner class to get input from the command-line interface.
- All validations that have been mentioned in this document should be implemented.
- You can use the following code to declare the Burger price.

```
final static double BURGERPRICE=500;
```

- You can use the following code to declare order status.

```
//Order status
   public static final int PREPARING=0;
   public static final int DELIVERED=1;
   public static final int CANCEL=2;
```

- When you create an Exit option in the program, you can use the below code.

```
//exit
   public static void exit(){
      clearConsole();
      System.out.println("\n\t\tYou left the program...\n");
      System.exit(0);
   }
```

- It is not required to clear the command line screen while navigating between the options. But doing so highly recommend it.
- The code to clear the command line from inside a Java application is as follows. You can use this code when you need to clear the command line.

```
public final static void clearConsole() {
   try {
      final String os = System.getProperty("os.name");
      if (os.contains("Windows")) {
         new ProcessBuilder("cmd", "/c", "cls").inheritIO().start().waitFor();
      } else {
         System.out.print("\033[H\033[2J");
         System.out.flush();
        }
    } catch (final Exception e) {
       e.printStackTrace();
       // Handle any exceptions.
     }
}
```

- Demo Videos are given at relevant places for you to understand the coursework requirement better and Demo videos may help you to clarify your doubts to some extent.