UNIVERSITI TEKNOLOGI MARA

COLLEGE OF COMPUTING, INFORMATICS AND MATHEMATICS

UITM CAWANGAN JOHOR,

KAMPUS SEGAMAT

FINAL REPORT FOR

D ‘ GANU RESTAURANT FOOD ORDERING SYSTEMS

GROUP PROJECT 248

GROUP MEMBER	STUDENT ID
AHMAD DANIAL BIN HISHAM	2023492972
MUHAMMAD DANIEL BIN M ZULKIFLI	2023882308
MUHAMMAD AMIRUN HAMIZAN BIN KHAMISAN	2023601668

GROUP : JCDCS1103C

LECTURER NAME : MADAM MAZLYDA ABD RAHMAN

DATE SUBMISSION :

TABLE OF CONTENT

NO	CONTENT	PAGE
1.	Introduction of project and group members.	
2.	Distribution of works between team members	
3.	Complete coding of all classes.	
4.	Sample input and output.	
5.	Conclusions of your finding: which of the data structure best applied in your case study	

INTODUCTION OF PROJECT AND GROUP MEMBERS

Our restaurant's Food Ordering System simplifies managing customer orders and restaurant operations with several key features. The system includes comprehensive customer data management and order tracking capabilities.

First, all customer information is displayed in a customer list, allowing easy access to essential details. Customers can also view specific information such as those whose food price is less than $10. The system enables staff to identify the highest and lowest food prices for each customer efficiently.

Additionally, it counts and displays the number of customers who purchase a specific food item, providing valuable insights into customer preferences. The system also allows for the reorganization of data by transferring customer details into various linked lists based on payment methods, such as custLL, creditLL, onlineLL, and cashLL.

The number of customers using each payment method is counted and displayed, offering a clear view of payment trends. Staff can search the creditCardLL linked list to update order statuses for customers who paid via credit card. Similarly, the average food price can be calculated and searched within the cashLL linked list.

Moreover, the system allows staff to search by customer name to update food items in their order, ensuring flexibility and accuracy in managing customer preferences. Lastly, in the end of output our coding will generate a report that count each of food are being buy from customer

This enhanced system with real-time updates and an intuitive design ensures a convenient and efficient experience for both customers and restaurant staff.

The application will be able to:

-Display all the data in customer list.

-Display information of customer where the foodprice < 10.

-Find highest and lowest food price of a customer.

-Count and display number of customer that buy any food name.

-Remove all the data in customer list into custLL,creditLL,onlineLL,cashLL.

-count and display number of payment method of a customer.

-search customer id in creditCardLL to update orderStatus.

-search average food price in cashLL.

-search customer name to update food name.

-calculate total price for each payment method.

-generate a report that count each of food are being buy from customer

LIST OF PROCESSING

- INSERT
  - To insert the customer's personal information such as name, phone number and email
  - To insert the customer's order for food and drinks
  - To insert the customer's delivery address

- CALCULATE
- Find the highest and lowest food price of a customer.
- Calculate the total price for each payment method.
- Search the average food price in cashLL

- TRAVERSING & COUNT
- Display all the data in the customer list.
- Display information of customers where the food price < 10.
- Count and display the number of customers that buy any specific food item.
- Count and display the number of payment methods of a customer.

- REMOVE
- Remove all the data in the customer list into custLL, creditLL, onlineLL, cashLL.

- SEARCH
- Search customer ID in creditCardLL to update orderStatus.
- Search customer name to update food name.

- UPDATE
- Update orderStatus by searching the customer ID in creditCardLL.

  Update food name by searching the customer name.

2. Distribution of works between team members

GROUP MEMEBRS

	**MUHAMMAD AMIRUN HAMIZAN BIN KHAMISAN** **2023601668**
	**AHMAD DANIAL BIN ISHAM** **2023601668**
	**MUHAMMAD DANIEL BIN M ZULKIFLI** **2023882308**

DISTRIBUTION OF WORKS BETWEEN TEAM MEMBERS

AMIRUN HAMIZAN BIN KHAMISAN	1. Propose the project idea 2. Assignment of responsibilities among group members 3. Design and implementation of all class definitions. 4. Provide guidance and assist in development of the linked list and queue main classes.
AHMAD DANIAL BIN ISHAM	1. Design and implement the queue main class. 2. Helping with the making of the project's proposal. 3. Helping with the making of the final report.
MUHAMMAD DANIEL BIN M ZULKIFLI	1. Design and implement the linked list main class. 2. Helping with the making of the project's proposal. 3. Helping with the making of the coding segment

3.Complete Coding of All cases

INPUT FILE .TXT

C001,Ahmad Zulkifli,0123456789,ahmad.zulkifli@gmail.com,O001,2025-01-12 12:00,complete,Nasi Lemak,4.00,cash

C002,Siti Nurhaliza,0987654321,siti.nurhaliza@gmail.com,O002,2025-01-12 12:30,pending,Char Kway Teow,6.00,credit card

C003,Muhammad Hafiz,0112233445,muhammad.hafiz@yahoo.com,O003,2025-01-12 13:00,complete,Nasi Lemak,4.00,online transfer

C004,Zainab Abdullah,0223344556,zainab.abdullah@outlook.com,O004,2025-01-12 13:15,complete,Nasi Kerabu,7.00,credit card

C005,Lim Beng Hock,0334455667,lim.benghock@gmail.com,O005,2025-01-12 14:00,cancel,Char Kway Teow,6.00,cash

C006,Noor Fatimah,0445566778,noor.fatimah@hotmail.com,O006,2025-01-12 14:30,pending,Roti Canai,2.00,online transfer

C007,Ravi Kumar,0556677889,ravi.kumar@gmail.com,O007,2025-01-12 15:00,complete,Nasi Dagang,6.50,cash

C008,Tan Mei Ling,0667788990,tan.meiling@yahoo.com,O008,2025-01-12 15:15,complete,Nasi Dagang,6.50,credit card

C009,Chong Wei,0778899001,chong.wei@outlook.com,O009,2025-01-12 16:00,complete,Char Kway Teow,6.00,online transfer

C010,Nurul Ain,0889900112,nurul.ain@gmail.com,O010,2025-01-12 16:30,cancel,Nasi Lemak,4.00,credit card

C011,Azman Latif,0990011223,azman.latif@gmail.com,O011,2025-01-12 17:00,pending,Roti Canai,2.00,cash

C012,Aisyah Rahman,0101122334,aisyah.rahman@yahoo.com,O012,2025-01-12 17:15,complete,Nasi Kerabu,7.00,online transfer

C013,Nagarajan Raj,0111233445,nagarajan.raj@gmail.com,O013,2025-01-12 18:00,pending,Nasi Dagang,6.50,credit card

C014,Liew Hong,0122233445,liew.hong@gmail.com,O014,2025-01-12 18:30,complete,Char Kway Teow,6.00,cash

C015,Farid Ismail,0133344556,farid.ismail@hotmail.com,O015,2025-01-12 19:00,cancel,Roti Canai,2.00,online transfer

C016,Sarah Zahira,0144455667,sarah.zahira@gmail.com,O016,2025-01-12 19:15,complete,Nasi Kerabu,7.00,credit card

C017,Teo Kian Seng,0155566778,teo.kianseng@yahoo.com,O017,2025-01-12 20:00,complete,Nasi Dagang,6.50,online transfer

C018,Rosnah Daud,0166677889,rosnah.daud@gmail.com,O018,2025-01-12 20:30,pending,Nasi Dagang,6.50,cash

C019,Hisham Mokhtar,0177788990,hisham.mokhtar@outlook.com,O019,2025-01-12 21:00,complete,Char Kway Teow,6.00,credit card

C020,Lee Choon Wah,0188899001,lee.choonwah@gmail.com,O020,2025-01-12 21:30,cancel,Roti Canai,2.00,online transfer

C021,Ahmad Bakri,0199900112,ahmad.bakri@gmail.com,O021,2025-01-12 22:00,complete,Nasi Lemak,4.00,cash

C022,Nur Syafiqah,0200011223,nur.syafiqah@gmail.com,O022,2025-01-12 22:30,pending,Nasi Lemak,4.00,credit card

C023,Roslan Hamid,0211122334,roslan.hamid@yahoo.com,O023,2025-01-12 23:00,complete,Char Kway Teow,6.00,online transfer

C024,Amira Hassan,0222233445,amira.hassan@gmail.com,O024,2025-01-12 23:15,complete,Nasi Dagang,6.50,cash

C025,Wan Fariz,0233344556,wan.fariz@outlook.com,O025,2025-01-12 23:30,complete,Nasi Kerabu,7.00,credit card

C026,Chee Yin,0244455667,chee.yin@gmail.com,O026,2025-01-12 23:45,cancel,Nasi Dagang,6.50,online transfer

C027,Fatimah Yaakob,0255566778,fatimah.yaakob@yahoo.com,O027,2025-01-13 00:00,complete,Char Kway Teow,6.00,cash

C028,Singh Amarjit,0266677889,singh.amarjit@gmail.com,O028,2025-01-13 00:15,pending,Nasi Dagang,6.50,credit card

C029,Noor Hafizah,0277788990,noor.hafizah@gmail.com,O029,2025-01-13 00:30,complete,Roti Canai,2.00,online transfer

C030,Zamri Nasir,0288899001,zamri.nasir@gmail.com,O030,2025-01-13 01:00,complete,Nasi Kerabu,7.00,cash

CLASS NODE

```
/***
Coder: Roslan S, UiTM Pahang, roslancs@uitm.edu.my
Year: 2012
***/

public class Node<E>{
   E data;
   Node next;

   public Node(E data) {
```

```
        this.data = data;
    }

}
```

CLASS DEFINITION LINKED LIST

```
public class LinkedList<E> {

    private Node<E> head, current, tail;

    public LinkedList() {
        head = current = tail = null;
    }

    public boolean isEmpty() {
        return head == null;
    }

    public void addFirst(E data) {
        Node newNode = new Node(data);
        newNode.next = this.head;
        this.head = newNode;
        if(this.tail == null) {
            this.tail = this.head;
        }
    }

    public void addLast(E data) {
        Node newNode = new Node(data);

        if(this.tail == null) {
```

```
        this.head = this.tail = newNode;
    }
    else {
        this.tail.next = newNode;
        this.tail = this.tail.next;
    }
}

public E getFirst() {
    if (this.isEmpty()) {
        return null;
    }
    else {
        this.current = this.head;
        return this.current.data;
    }
}

public E getLast() {
    if (this.isEmpty()) {
        return null;
    }
    else {
        return this.tail.data;
    }
}

public E getNext() {
    if (this.current == this.tail) {
        return null;
    }
    else {
        this.current = this.current.next;
```

```
        return this.current.data;
    }
}

public void clear() {
    this.head = this.current = this.tail = null;

}

public boolean contains(E data) {
    boolean isContain = false;
    this.current = this.head;

    while (this.current != null) {
        if (data.equals(this.current.data)) {
            isContain = true;
            break;
        }
    }

    return isContain;
}

public E removeFirst() {
    if (this.isEmpty()) {
        return null;
    }
    else {
        this.current = this.head;
        this.head = this.head.next;
        if (this.head == null)
            this.tail = null;
```

```
        return current.data;
    }
}

public E removeLast() {
    if (this.isEmpty())
        return null;
    else if (this.head == this.tail) {
        this.current = this.head;
        this.head = this.tail = null;
        return current.data;
    }
    else {
        this.current = this.head;
        while (this.current.next != tail) {
            this.current = this.current.next;
        }
        Node<E> temp = this.tail;
        this.tail = this.current;
        this.tail.next = null;
        return temp.data;
    }
}

public E removeAfter(E data) {
    if (this.isEmpty()) {
        return null;
    }
    else if (this.head == this.tail) {
        this.current = this.head;
        this.head = this.tail = null;
        return current.data;
```

```
    }
    else {
        Node<E> previous = this.head;
        while (previous.next != null) {
            if (data.equals(previous.data))
            {
                break;
            }
            previous = previous.next;
        }
        current = previous.next;
        previous.next = current.next;
        return current.data;
    }
}

public String toString() {
    StringBuilder result = new StringBuilder("[");
    if (this.isEmpty()) {
        result.append("The list is empty]");
    }
    else {
        this.current = this.head;
        while (this.current != null) {
            result.append(this.current.data);
            this.current = this.current.next;
            if (this.current != null)
                result.append(", ");
            else
                result.append("]");
        }
    }
    return result.toString();
```

```
    }
}
```

CLASS DEFINITION QUEUE

```
public class Queue <E>
{
    private LinkedList <E> list;
    public Queue() {list = new LinkedList<E>();}

    public void enqueue(E data) {
        list.addLast(data);
    }

    public E dequeue() {
        return list.removeFirst();
    }

    public E getFront() {
        return list.getFirst();
    }

    public boolean isEmpty() {
        return list.isEmpty();
    }
}
```

CLASS DEFINITION RESTAURANT

```
import java.text.*;
public class Restaurant{
```

```
    private String orderID;
    private String orderTime;
    private String orderStatus;
    private String foodName;
    private double foodPrice;
    private String paymentMethod;

    public Restaurant(String orderID,String orderTime,String orderStatus,String
foodName,double foodPrice,String paymentMethod){
        this.orderID = orderID;
        this.orderTime = orderTime;
        this.orderStatus = orderStatus;
        this.foodName = foodName;
        this.foodPrice = foodPrice;
        this.paymentMethod = paymentMethod;
    }

    public String getOrderID(){return orderID;}
    public String getOrderTime(){return orderTime;}
    public String getOrderStatus(){return orderStatus;}
    public String getFoodName(){return foodName;}
    public double getFoodPrice(){return foodPrice;}
    public String getPaymentMethod(){return paymentMethod;}

    public void setStatusOrder(String newOrderStatus){
        orderStatus = newOrderStatus;
    }

    public void setPaymentMethod(String newPaymentMethod){
        paymentMethod = newPaymentMethod;
    }
```

```
    public String toString(){
        return String.format(
            "| %-14s | %-16.2f | %-14s | $%-9.2f | %-17s | %-17s |\n" ,
           orderID, orderTime, orderStatus,foodName, foodPrice,  paymentMethod);
    }
}
```

CLASS DEFINITION CUSTOMER

```
import java.text.*;

public class Customer{
    private String custID;
    private String custName;
    private String custPhoneNum;
    private String custEmail;
    private Restaurant restaurant;

     DecimalFormat df = new DecimalFormat("0.00");

    public Customer(String custID,String custName,String custPhoneNum, String
custEmail,Restaurant restaurant)
    {
       this.custID = custID;
       this.custName = custName ;
       this.custPhoneNum = custPhoneNum;
       this.custEmail = custEmail;
       this.restaurant = restaurant;
    }

    public String getCustID(){return custID;}
    public String getCustName(){return custName;}
```

```
   public String getCustPhoneNum(){return custPhoneNum;}
   public String getCustEmail(){return custEmail;}
   public Restaurant getRestaurant(){return restaurant;}

   public String toString() {

       return String.format(
          "| %-13s | %-18s | %-16s | %-30s | %-14s | %-18s | %-13s | %-10.2f | %-17s | %-
17s |",
          custID, custName, custPhoneNum, custEmail,
          restaurant.getOrderID(), restaurant.getOrderTime(),
          restaurant.getOrderStatus(), restaurant.getFoodPrice(),
          restaurant.getFoodName(), restaurant.getPaymentMethod()
       );
   }




}

TEST CLASS TestRestaurantLL
import java.util.*;
import java.io.*;
import java.text.*;
import java.text.DecimalFormat;


public class TestRestaurantLL{
   public static void main(String[]args) throws IOException
   {

       FileReader fr = new FileReader("Cust.txt");
```

```
BufferedReader br = new BufferedReader(fr);
DecimalFormat df = new DecimalFormat("0.00");
Scanner scanner = new Scanner(System.in);
scanner.useDelimiter("\n");

LinkedList <Customer> CustomerList = new LinkedList();

//5 Create LinkedLists for custom
 LinkedList<Customer> custLL = new LinkedList();
 LinkedList<Customer> creditLL = new LinkedList();
 LinkedList<Customer> onlineLL = new LinkedList();
 LinkedList<Customer> cashLL = new LinkedList();

String custID;
String custName;
String custPhoneNum;
String custEmail;

 String orderID;
 String orderTime;
 String orderStatus;
 double foodPrice;
 String foodName;
 String paymentMethod;

String inData = null;
while((inData = br.readLine()) != null)
{
  StringTokenizer st = new StringTokenizer(inData, ",");

  custID = st.nextToken();
  custName = st.nextToken();
```

```
            custPhoneNum = st.nextToken();
            custEmail = st.nextToken();

            orderID = st.nextToken();
            orderTime = st.nextToken();
            orderStatus = st.nextToken();
            foodName = st.nextToken();
            foodPrice = Double.parseDouble(st.nextToken());
            paymentMethod = st.nextToken();

            Restaurant restaurant = new Restaurant(orderID, orderTime,
orderStatus,foodName, foodPrice, paymentMethod);
            Customer customer = new Customer(custID, custName, custPhoneNum,
custEmail,restaurant);



            CustomerList.addFirst(customer);

        }
        br.close();
        fr.close();

         System.out.printf("%-20s %-10s\n", "Food Item", "Price (RM)");
         System.out.println("---------------------------------");
         System.out.printf("%-20s %-10.2f\n", "Nasi Lemak", 4.00);
         System.out.printf("%-20s %-10.2f\n", "Char Kway Teow", 6.00);
         System.out.printf("%-20s %-10.2f\n", "Roti Canai", 2.00);
         System.out.printf("%-20s %-10.2f\n", "Nasi Dagang", 6.50);
         System.out.printf("%-20s %-10.2f\n", "Nasi Kerabu", 7.00);

        //1. Display customer order
```

```
        System.out.println("\n+---------------+--------------------+-----------------+------------
Display Customer Order Information-------------------+---------------+--------------------+-----------
----+------------+------");

        System.out.println("\n+---------------+--------------------+-----------------+--------------------
----------+---------------+--------------------+--------------+------------+------------------+-----------------
+");

        System.out.println("| Customer ID   | Customer Name      | Phone Number     |
Email                        | Order ID       | Order Time         | Order Status  | Food Price | Food
Name        | Payment Method    |");

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


        Customer obj;

        obj  = CustomerList.getFirst();

        while( obj != null){

           System.out.println(obj.toString());

           obj = CustomerList.getNext();

        }

        System.out.println("\n+---------------+--------------------+-----------------+--------------------
----------+---------------+--------------------+--------------+------------+------------------+-----------------
+");


        // 2. Count and display number of customers that buy a specific payment
method

         System.out.println("\nCount and Display number of customer payment
method");

         System.out.println("\n+---------------+--------------------+-----------------+--------------------
-----------+---------------+--------------------+--------------+------------+------------------+-----------------
+");


         System.out.print("\nEnter the payment method to search for(cash | online |
credit ): ");

         String searchPaymentMethod = scanner.nextLine().trim();

         int cashCount = 0;

         int onlineCount = 0;

         int creditCount = 0;
```

```
        obj = CustomerList.getFirst();
        while( obj != null){
           if(obj.getRestaurant().getPaymentMethod().equalsIgnoreCase("Cash")){
              cashCount++;
           }
           else if(obj.getRestaurant().getPaymentMethod().equalsIgnoreCase("Online
Transfer")){
              onlineCount++;
           }
           else if(obj.getRestaurant().getPaymentMethod().equalsIgnoreCase("Credit
Card")){
              creditCount++;
           }
           obj = CustomerList.getNext();
           }

        if(searchPaymentMethod.equalsIgnoreCase("Cash")){
              System.out.printf("Number of customers who pay with '%s': %d%n",
searchPaymentMethod, cashCount);
           }
           else if(searchPaymentMethod.equalsIgnoreCase("Online")){
               System.out.printf("Number of customers who pay with '%s': %d%n",
searchPaymentMethod, onlineCount);
           }
           else if(searchPaymentMethod.equalsIgnoreCase("Credit")){
               System.out.printf("Number of customers who pay with '%s': %d%n",
searchPaymentMethod, creditCount);
           }

        System.out.println("\n+---------------+--------------------+-----------------+--------------------
----------+---------------+-------------------+--------------+-----------+------------------+-----------------
+");

        // 3. Search customer ID in customer list to update orderStatus
        System.out.println("\nSearch customer id and Update Status");
```

```
        System.out.println("\n\n+---------------+--------------------+-----------------+-------------------
-------------+---------------+--------------------+--------------+------------+------------------+----------------
--+");


        System.out.print("\nEnter the Customer ID to search for and update order
status: ");
        String searchCustomerID = scanner.nextLine();
        boolean customerFound = false;


        obj = CustomerList.getFirst(); // Start from the first customer in the linked list
        while (obj != null) {
           if (obj.getCustID().equalsIgnoreCase(searchCustomerID)) {
              customerFound = true;
              System.out.println("\nCustomer found:");
              System.out.println(obj.toString()); // Display current customer details
              System.out.println();
              System.out.println("\n+---------------+--------------------+-----------------+--------------
-----------------+---------------+--------------------+--------------+------------+------------------+-------------
------+");
              System.out.println("\nUpdate Status");
              System.out.println("\n+---------------+--------------------+-----------------+--------------
-----------------+---------------+--------------------+--------------+------------+------------------+-------------
------+");
              System.out.print("Enter the new order status: ");
              String newOrderStatus = scanner.nextLine().trim();
              obj.getRestaurant().setStatusOrder(newOrderStatus); // Update order
status


              System.out.println("\nOrder status updated successfully!");
              System.out.println();
              System.out.println(obj.toString()); // Display updated customer details
              System.out.println("\n+---------------+--------------------+-----------------+--------------
-----------------+---------------+--------------------+--------------+------------+------------------+-------------
------+");


           }
```

```
            obj = CustomerList.getNext(); // Move to the next customer
        }




        System.out.println("\n+---------------+-------------------+-----------------+---------------
Update Customer Information--------------------------+--------------+-----------+-----------------
+------------------+");


        obj  = CustomerList.getFirst();
        while( obj != null){
          System.out.println(obj.toString());
          obj = CustomerList.getNext();
        }


        System.out.println("\n+---------------+-------------------+-----------------+-------------------
-----------+---------------+-------------------+--------------+-----------+------------------+-----------------
+");


        //4
        System.out.println("\nSearch Customer Name and Update Payment Method");
        System.out.println("\n+---------------+-------------------+-----------------+-------------------
-----------+---------------+-------------------+--------------+-----------+------------------+-----------------
+");




        System.out.print("\nEnter the Customer Name to search for and update
payment method: ");
        String searchCustomerName = scanner.nextLine();
        boolean nameFound = false;


        obj = CustomerList.getFirst(); // Start from the first customer in the linked list
        while (obj != null) {
          if (obj.getCustName().equalsIgnoreCase(searchCustomerName)) {
            nameFound = true;
```

```
                System.out.println("\nCustomer found:");
                System.out.println(obj.toString()); // Display current customer details
                System.out.println();
                System.out.println("+---------------+-------------------+-----------------+-----------------
---------------+---------------+-------------------+--------------+------------+------------------+--------------
----+");


                System.out.println("\n\n+---------------+-------------------+-----------------+------------
-------------------+---------------+-------------------+--------------+------------+------------------+----------
---------+");


                System.out.print("Enter the new payment method: ");
                String newPaymentMethod = scanner.nextLine().trim();
                obj.getRestaurant().setPaymentMethod(newPaymentMethod); // Update
order status


                System.out.println("\nCustomer payment method updated successfully!");
                System.out.println();
                System.out.println(obj.toString()); // Display updated customer details
                System.out.println("\n+---------------+-------------------+-----------------+--------------
-----------------+---------------+-------------------+--------------+------------+------------------+------------
------+");
                }
            obj = CustomerList.getNext(); // Move to the next customer
          }


          System.out.println("\n+---------------+-------------------+-----------------+--------------------
-----------+---------------+-------------------+--------------+------------+------------------+------------------
+");




          obj  = CustomerList.getFirst();


          while( obj != null){
            System.out.println(obj.toString());
```

```
            obj = CustomerList.getNext();
        }

        System.out.println("\n+---------------+--------------------+-----------------+--------------------
-----------+---------------+--------------------+--------------+-----------+------------------+-----------------
+");

        System.out.print("\nEnter the payment method to search for(cash | online |
credit): ");
        String searchPayment = scanner.nextLine().trim();
        double totalPriceC = 0.00;
        double totalPriceOn9 = 0.00 ;
        double totalPriceCard = 0.00;

        obj = CustomerList.getFirst();
        while( obj != null){
            if(obj.getRestaurant().getPaymentMethod().equalsIgnoreCase("Cash")){
                totalPriceC += obj.getRestaurant().getFoodPrice();
            }
            else if(obj.getRestaurant().getPaymentMethod().equalsIgnoreCase("Online
Transfer")){
                totalPriceOn9 += obj.getRestaurant().getFoodPrice();
            }
            else if(obj.getRestaurant().getPaymentMethod().equalsIgnoreCase("Credit
Card")){
                totalPriceCard += obj.getRestaurant().getFoodPrice();
            }
            obj = CustomerList.getNext();
            }

        if(searchPayment.equalsIgnoreCase("Cash")){
                System.out.println("Total price cutomer who pay with cash : "
+df.format(totalPriceC));
            }
```

```
            else if(searchPayment.equalsIgnoreCase("Online")){
                System.out.printf("Total price cutomer who pay with online payment : "
+df.format(totalPriceOn9));
            }
            else if(searchPayment.equalsIgnoreCase("Credit")){
                System.out.printf("Total price cutomer who pay with : "
+df.format(totalPriceCard));
            }

        System.out.println("\n+---------------+-------------------+-----------------+--------------------
-----------+---------------+-------------------+--------------+-----------+------------------+------------------
+");

         //5 remove all the data in cust list into custLL,creditLL,onlineLL
        // Transfer all data from CustomerList
        System.out.println("\nSort Customer Payment Method");
        System.out.println("\n+---------------+-------------------+-----------------+--------------------
-----------+---------------+-------------------+--------------+-----------+------------------+------------------
+");
        while (!CustomerList.isEmpty()) {
            Customer customer = CustomerList.removeFirst(); // Remove from
CustomerList
            custLL.addFirst(customer);  // Add to custLL

            // Check payment method and add to appropriate lists
            if (customer.getRestaurant().getPaymentMethod().equalsIgnoreCase("Credit
Card")) {
                creditLL.addFirst(customer);
            } else if
(customer.getRestaurant().getPaymentMethod().equalsIgnoreCase("Online Transfer"))
{
                onlineLL.addFirst(customer);
            } else if
(customer.getRestaurant().getPaymentMethod().equalsIgnoreCase("Cash")){
                cashLL.addFirst(customer);
            }
        }
```

```
        System.out.println("\nAll customers (custLL):");
        Customer current = custLL.getFirst();
        while (current != null) {
           System.out.println(current.toString());
           current = custLL.getNext();
        }

        System.out.println("\n+---------------+-------------------+-----------------+--------------------
-----------+---------------+-------------------+--------------+------------+------------------+------------------
+");

        System.out.println("\nCustomers who paid with Credit Card (creditLL):");
        current = creditLL.getFirst();
        while (current != null) {
           System.out.println(current.toString());
           current = creditLL.getNext();
        }

        System.out.println("\n+---------------+-------------------+-----------------+--------------------
-----------+---------------+-------------------+--------------+------------+------------------+------------------
+");



        System.out.println("\nCustomers who paid with Online Transfer (onlineLL):");
        current = onlineLL.getFirst();
        while (current != null) {
           System.out.println(current.toString());
           current = onlineLL.getNext();
        }

        System.out.println("\n+---------------+-------------------+-----------------+--------------------
-----------+---------------+-------------------+--------------+------------+------------------+------------------
+");
```

```
        System.out.println("\nCustomers who paid with Cash Payment (cashLL):");
        current = cashLL.getFirst();
        while (current != null) {
            System.out.println(current.toString());
            current = cashLL.getNext();
        }

        System.out.println("\n+---------------+--------------------+-----------------+--------------------
-----------+---------------+--------------------+--------------+------------+------------------+------------------
+");



         //6. display information of customer where the foodprice < 10.00
         System.out.println("\n+---------------+--------------------+-----------------+---------------
Display Customer Food Price less than RM10-------------+---------------+--------------------+----
-----------+------------+-");
        System.out.println("\n+---------------+--------------------+-----------------+--------------------
----------+---------------+--------------------+--------------+------------+------------------+------------------
+");
        System.out.println("| Customer ID   | Customer Name      | Phone Number     |
Email                      | Order ID       | Order Time         | Order Status  | Food Price | Food
Name        | Payment Method    |");
        System.out.println("+---------------+--------------------+-----------------+----------------------
-------+---------------+--------------------+--------------+------------+------------------+------------------
+");
        obj = custLL.getFirst();
        while( obj != null){
            if(obj.getRestaurant().getFoodPrice() < 10.00){
                System.out.println(obj.toString());
            }
            obj = custLL.getNext();
        }

        System.out.println("\n+---------------+--------------------+-----------------+--------------------
----------+---------------+--------------------+--------------+------------+------------------+------------------
+");
```

```
        //7. find highest and lowest price of a customer



        Customer highestObj = custLL.getFirst();
        Customer lowestObj = custLL.getFirst();
        obj = custLL.getFirst();


        while(obj != null){


         if(obj.getRestaurant().getFoodPrice() >
highestObj.getRestaurant().getFoodPrice()){
            highestObj = obj ;
        }
        else if(obj.getRestaurant().getFoodPrice() <
lowestObj.getRestaurant().getFoodPrice()){
            lowestObj = obj;
        }
        obj = custLL.getNext();




        }


        System.out.println("\n+---------------+--------------------+------------------+---------------
Display Customer Highset and Lowest Price-------------+----------------+--------------------+------
---------+-----------+-");


        System.out.println("\nCustomer with the Highest Food Price:");
        System.out.println(highestObj);


        System.out.println("\nCustomer with the Lowest Food Price:");
        System.out.println(lowestObj);
```

```
        System.out.println("\n+---------------+--------------------+-----------------+--------------------
-----------+---------------+--------------------+--------------+-----------+------------------+-----------------
+");



        // 8. Calculate and display the average food price in customer list
        double totalFoodPrice = 0;
        int customerCount = 0;

        System.out.println("\n+---------------+--------------------+-----------------+---------------
Display average food price among all customer---------+--------------+-----------+--------------
-----+------------------+");

        obj = custLL.getFirst(); // Start from the first customer in the linked list
        while (obj != null) {
            totalFoodPrice += obj.getRestaurant().getFoodPrice(); // Accumulate food
prices
            customerCount++; // Count each customer
            obj = custLL.getNext(); // Move to the next customer
        }

        if (customerCount > 0) {
            double averageFoodPrice = totalFoodPrice / customerCount;
            System.out.printf("\nThe average food price among all customers is:
RM%.2f%n", averageFoodPrice);
        } else {
            System.out.println("\nNo customers available to calculate the average food
price.");
        }

        System.out.println("\n+---------------+--------------------+-----------------+--------------------
-----------+---------------+--------------------+--------------+-----------+------------------+-----------------
+");



       //9
```

```
        System.out.print("\nEnter the payment method to search for(cash | online | credit): ");
        String searchPaymentLL = scanner.nextLine().trim();
        double totalPriceCLL = 0.00;
        double totalPriceOn9LL = 0.00 ;
        double totalPriceCardLL = 0.00;



        obj = custLL.getFirst();
        while( obj != null){
           if(obj.getRestaurant().getPaymentMethod().equalsIgnoreCase("Cash")){
              totalPriceCLL += obj.getRestaurant().getFoodPrice();
           }
           else if(obj.getRestaurant().getPaymentMethod().equalsIgnoreCase("Online Transfer")){
              totalPriceOn9LL += obj.getRestaurant().getFoodPrice();
           }
           else if(obj.getRestaurant().getPaymentMethod().equalsIgnoreCase("Credit Card")){
              totalPriceCardLL += obj.getRestaurant().getFoodPrice();
           }
           obj = custLL.getNext();
           }


        if(searchPayment.equalsIgnoreCase("Cash")){
              System.out.println("Total price cutomer who pay with cash : " +df.format(totalPriceCLL));
           }
           else if(searchPayment.equalsIgnoreCase("Online")){
               System.out.printf("Total price cutomer who pay with online payment : " +df.format(totalPriceOn9LL));
           }
           else if(searchPayment.equalsIgnoreCase("Credit")){
               System.out.printf("Total price cutomer who pay with : " +df.format(totalPriceCardLL));
```

```
            }

        System.out.println("\n+---------------+--------------------+-----------------+--------------------
-----------+---------------+--------------------+--------------+-----------+-----------------+-----------------
+");
        //10 - Generate Report

        System.out.println("\n+---------------+--------------------+-----------------+--------------------
-----------+---------------+--------------------+--------------+-----------+-----------------+-----------------
+");
        System.out.println("\nReport for food names and the number of the
customers");
        System.out.println("\nFood Report:");
        System.out.println("+--------------------+-----------------+");
        System.out.println("| Food Name         | Total Customers |");
        System.out.println("+--------------------+-----------------+");

        int nlemakC = 0;
        int CKTc = 0;
        int rotiCanaiC=0;
        int nDagangC = 0;
        int nKerabuC = 0;

        obj = custLL.getFirst();

        while(obj != null)
        {
           if(obj.getRestaurant().getFoodName().equalsIgnoreCase("Nasi Lemak"))
           {
              nlemakC++;
            }
           else if ( obj.getRestaurant().getFoodName().equalsIgnoreCase("Char Kway
Teow"))
           {
              CKTc++;
```

```
            }
            else if(obj.getRestaurant().getFoodName().equalsIgnoreCase("Roti Canai"))
            {
               rotiCanaiC++;
            }
            else if(obj.getRestaurant().getFoodName().equalsIgnoreCase("Nasi
Dagang"))
            {
               nDagangC++;
            }
            else if(obj.getRestaurant().getFoodName().equalsIgnoreCase("Nasi
Kerabu")){
               nKerabuC++;
            }

            obj = custLL.getNext();

         }
         System.out.printf("| %-17s | %-16d |\n", "Nasi Lemak", nlemakC);
         System.out.printf("| %-17s | %-16d |\n", "Char Kway Tiaw", CKTc);
         System.out.printf("| %-17s | %-16d |\n", "Roti Canai", rotiCanaiC);
         System.out.printf("| %-17s | %-16d |\n", "Nasi Dagang",nDagangC);
         System.out.printf("| %-17s | %-16d |\n", "Nasi Kerabu", nKerabuC);

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


   }
}
```

TEST CLASS TestRestaurantQ

```
import java.util.*;
```

```
import java.io.*;
import java.text.*;
import java.text.DecimalFormat;

public class TestRestaurantQ
{
   public static void main(String[]args) throws IOException
   {
      FileReader fr = new FileReader("Cust.txt");
      BufferedReader br = new BufferedReader(fr);
      DecimalFormat df = new DecimalFormat("#.##");
      Scanner scanner = new Scanner(System.in);
      scanner.useDelimiter("\n");

      Queue <Customer> CustomerQ = new Queue();
      Queue<Customer> tempQ = new Queue();

      // Create Queue For Custom
      Queue <Customer> custQ = new Queue();
      Queue <Customer> creditQ = new Queue();
      Queue <Customer> onlineQ = new Queue();
      Queue <Customer> cashQ = new Queue();

     String custID, custName, custPhoneNum, custEmail;

      String orderID;
      String orderTime;
      String orderStatus;
      double foodPrice;
      String foodName;
      String paymentMethod;
      int NameFood;
```

```
String inData = null;
while((inData = br.readLine()) != null)
  {
    StringTokenizer st = new StringTokenizer(inData, ",");

    custID = st.nextToken();
    custName = st.nextToken();
    custPhoneNum = st.nextToken();
    custEmail = st.nextToken();

    orderID = st.nextToken();
    orderTime = st.nextToken();
    orderStatus = st.nextToken();
    foodName = st.nextToken();
    foodPrice = Double.parseDouble(st.nextToken());
    paymentMethod = st.nextToken();

    Restaurant restaurant = new Restaurant (orderID, orderTime, orderStatus,
foodName, foodPrice, paymentMethod);

    Customer customer = new Customer(custID, custName, custPhoneNum,
custEmail,restaurant);

    CustomerQ.enqueue(customer);

  }
br.close();
fr.close();
   System.out.printf("%-20s %-10s\n", "Food Item", "Price (RM)");
   System.out.println("-----------------------------------");
   System.out.printf("%-20s %-10.2f\n", "Nasi Lemak", 4.00);
```

```
        System.out.printf("%-20s %-10.2f\n", "Char Kway Teow", 6.00);
        System.out.printf("%-20s %-10.2f\n", "Roti Canai", 2.00);
        System.out.printf("%-20s %-10.2f\n", "Nasi Dagang", 6.50);
        System.out.printf("%-20s %-10.2f\n", "Nasi Kerabu", 7.00);


    //1. Display customer order

    System.out.println("\nDisplay Customer Order Information");
    System.out.println("\n+---------------+--------------------+-----------------+-------------------------
-------+---------------+-------------------+--------------+------------+------------------+-----------------
+");
    System.out.println("| Customer ID   | Customer Name      | Phone Number     |
Email                       | Order ID       | Order Time         | Order Status  | Food Price | Food
Name        | Payment Method    |");
    System.out.println("+---------------+--------------------+-----------------+-------------------------
-----+---------------+-------------------+--------------+------------+------------------+-----------------+");


    Customer obj;
    while(!CustomerQ.isEmpty())
    {
        obj = CustomerQ.dequeue();
        System.out.println(obj.toString());
        tempQ.enqueue(obj);
    }

    while(!tempQ.isEmpty())
    {
         CustomerQ.enqueue(tempQ.dequeue());//yang buang tadi akan masuk balik
    }

    System.out.println("\n+---------------+--------------------+-----------------+-------------------------
-------+---------------+-------------------+--------------+------------+------------------+-----------------
+");
```

```
        // 2. count and display number of customer payment method
        System.out.println("\nCount and Display number of customer payment
method");
        System.out.println("\n+---------------+-------------------+-----------------+-----------------------
-------+---------------+-------------------+--------------+-----------+------------------+------------------
+");


        System.out.print("\nEnter the payment method to search for(cash | online |
credit): ");
        String searchPaymentMethod = scanner.nextLine().trim();


        int cashCount = 0;
        int onlineCount = 0;
        int creditCount = 0;



        while(!CustomerQ.isEmpty()){


            obj = CustomerQ.dequeue();
            if(obj.getRestaurant().getPaymentMethod().equalsIgnoreCase("Cash")){
                cashCount++;
            }
            else if(obj.getRestaurant().getPaymentMethod().equalsIgnoreCase("Online
Transfer")){
                onlineCount++;
            }
            else if(obj.getRestaurant().getPaymentMethod().equalsIgnoreCase("Credit
Card")){
                creditCount++;
            }
            tempQ.enqueue(obj);
        }
```

```
        if(searchPaymentMethod.equalsIgnoreCase("Cash")){

                System.out.printf("Number of customers who pay with '%s': %d%n",
searchPaymentMethod, cashCount);

            }

               else if(searchPaymentMethod.equalsIgnoreCase("Online Transfer")){

                   System.out.printf("Number of customers who pay with '%s': %d%n",
searchPaymentMethod, onlineCount);

            }

               else if(searchPaymentMethod.equalsIgnoreCase("Credit Card")){

                   System.out.printf("Number of customers who pay with '%s': %d%n",
searchPaymentMethod, creditCount);

            }


        System.out.println("\n+---------------+--------------------+-----------------+------------------------
--------+---------------+--------------------+--------------+------------+------------------+------------------
+");




         // Restore custQ from tempQ

        while (!tempQ.isEmpty()) {

           CustomerQ.enqueue(tempQ.dequeue());

        }


        // 3. search customer ID and update order status

        System.out.println("\nSearch customer id and Update Status");

        System.out.println("\n\n+---------------+--------------------+-----------------+----------------------
----------+---------------+--------------------+--------------+------------+------------------+------------------
+");


        System.out.print("\nEnter the Customer ID to search for and update order status:
");

        String searchCustomerID = scanner.nextLine();

        boolean customerFound = false;
```

```
        while (!CustomerQ.isEmpty())
        {
            obj = CustomerQ.dequeue();

            if (obj.getCustID().equalsIgnoreCase(searchCustomerID))
            {



                customerFound = true;
                System.out.println("\nCustomer found:");
                System.out.println(obj.toString()); // Display current customer details
                System.out.println();
                System.out.println("\n+---------------+--------------------+-----------------+---------------
----------------+---------------+--------------------+--------------+------------+------------------+-------------
------+");
                System.out.println("\nUpdate Status");
                System.out.println("\n+---------------+--------------------+-----------------+---------------
----------------+---------------+--------------------+--------------+------------+------------------+-------------
------+");


                System.out.print("\nEnter the new order status: ");
                String newOrderStatus = scanner.nextLine().trim();
                obj.getRestaurant().setStatusOrder(newOrderStatus); // Update order
status
                System.out.println("\nOrder status updated successfully!");
                System.out.println();
                System.out.println(obj.toString()); // Display updated customer details
                System.out.println("\n+---------------+--------------------+-----------------+---------------
----------------+---------------+--------------------+--------------+------------+------------------+-------------
------+");


            }
            tempQ.enqueue(obj); // Move to the next customer
        }
```

```
        System.out.println("\n+---------------+-------------------+-----------------+---------------
Update Customer Information-------------------------+--------------+------------+-----------------
+-----------------+");


        // Restore custQ from tempQ
        while (!tempQ.isEmpty())
        {
           CustomerQ.enqueue(tempQ.dequeue());
        }









        while( !CustomerQ.isEmpty())
        {
              obj = CustomerQ.dequeue();
             System.out.println(obj.toString());
             tempQ.enqueue(obj);
        }


        System.out.println("\n+---------------+-------------------+-----------------+-----------------------
-------+---------------+-------------------+--------------+------------+-----------------+-----------------
+");


        // Restore custQ from tempQ
        while (!tempQ.isEmpty())
        {
           CustomerQ.enqueue(tempQ.dequeue());
        }
```

```
        // 4 - Search Customer Name and Update Payment Method
        System.out.println("\nSearch Customer Name and Update Payment Method");
        System.out.println("\n+---------------+--------------------+-----------------+-----------------------
-------+---------------+--------------------+--------------+------------+------------------+-----------------
+");

        System.out.print("\nEnter the Customer Name to search for and update payment
method: ");
        String searchCustomerName = scanner.nextLine();
        boolean nameFound = false;

        while ( !CustomerQ.isEmpty())
        {
            obj = CustomerQ.dequeue();

            if (obj.getCustName().equalsIgnoreCase(searchCustomerName)) {

                nameFound = true;
                System.out.println("\nCustomer found:");
                System.out.println(obj.toString()); // Display current customer details
                System.out.println();
                System.out.println("+---------------+--------------------+-----------------+----------------
---------------+---------------+--------------------+--------------+------------+------------------+-------------
----+");

                System.out.println("\n\n+---------------+--------------------+-----------------+-----------
--------------------+---------------+--------------------+--------------+------------+------------------+----------
---------+");

                System.out.print("Enter the new payment method: ");
                String newPaymentMethod = scanner.nextLine().trim();
                obj.getRestaurant().setPaymentMethod(newPaymentMethod); // Update
order status
```

```
                System.out.println("\nCustomer payment method updated successfully!");
                System.out.println();

                System.out.println(obj.toString()); // Display updated customer details
                System.out.println("\n+---------------+--------------------+-----------------+---------------
----------------+---------------+--------------------+--------------+------------+------------------+------------
------+");
                }
            tempQ.enqueue(obj);
        }

        System.out.println("\n+---------------+--------------------+-----------------+-------------------------
--------+---------------+--------------------+--------------+------------+------------------+------------------
+");

         // Restore custQ from tempQ
        while (!tempQ.isEmpty())
        {
           CustomerQ.enqueue(tempQ.dequeue());
        }



        while( !CustomerQ.isEmpty())
           {
              obj = CustomerQ.dequeue();
              System.out.println(obj.toString());
               tempQ.enqueue(obj);
           }

        System.out.println("\n+---------------+--------------------+-----------------+-------------------------
--------+---------------+--------------------+--------------+------------+------------------+------------------
+");

         // Restore custQ from tempQ
        while (!tempQ.isEmpty())
```

```
        {
            CustomerQ.enqueue(tempQ.dequeue());
        }

        //5 -remove all the data in cust list into custQ,creditQ,onlineQ

        // Transfer all data from CustomerList
        System.out.println("\nSort Customer Payment Method");
        System.out.println("\n+---------------+-------------------+-----------------+-----------------------
-------+---------------+-------------------+---------------+------------+------------------+------------------
+");

        while (!CustomerQ.isEmpty())
        {
            obj = CustomerQ.dequeue(); // Remove from CustomerList



            // Check payment method and add to appropriate lists
            if (obj.getRestaurant().getPaymentMethod().equalsIgnoreCase("Credit
Card")) {
                creditQ.enqueue(obj);
            } else if (obj.getRestaurant().getPaymentMethod().equalsIgnoreCase("Online
Transfer")) {
                onlineQ.enqueue(obj);
            } else if
(obj.getRestaurant().getPaymentMethod().equalsIgnoreCase("Cash")){
                cashQ.enqueue(obj);
            }
            tempQ.enqueue(obj);
        }

        // Restore custQ from tempQ
        while (!tempQ.isEmpty())
        {
```

```
            CustomerQ.enqueue(tempQ.dequeue());
        }

        System.out.println("\nAll customers (custQ):");



        while (!CustomerQ.isEmpty())
        {
            obj = CustomerQ.dequeue();
            System.out.println(obj.toString());
            tempQ.enqueue(obj);
        }

        System.out.println("\nCustomers who paid with Credit Card (creditLL):");

        while (!creditQ.isEmpty())
        {
            obj = creditQ.dequeue();
            System.out.println(obj.toString());
        }

        System.out.println("\nCustomers who paid with Online Transfer (onlineLL):");
        while (!onlineQ.isEmpty())
        {
            obj = onlineQ.dequeue();
            System.out.println(obj.toString());
        }

        System.out.println("\n+---------------+-------------------+-----------------+-----------------------
-------+---------------+-------------------+--------------+------------+------------------+------------------
+");

        System.out.println("\nCustomers who paid with Cash Payment (cashLL):");
```

```
        while (!cashQ.isEmpty())
        {
            obj = cashQ.dequeue();
            System.out.println(obj.toString());
        }

        System.out.println("\n+---------------+-------------------+-----------------+-----------------------
-------+---------------+-------------------+--------------+------------+------------------+------------------
+");
        // Restore custQ from tempQ
        while (!tempQ.isEmpty())
        {
            CustomerQ.enqueue(tempQ.dequeue());
        }

        //6. Display information of customer where the foodprice < 10.00
        System.out.println("\nDisplay Customer Information where the food price below
than RM 10.00");
        System.out.println("\n+---------------+-------------------+-----------------+---------------
Display Customer Food Price less than RM10-------------+----------------+--------------------+----
-----------+------------+-");
        System.out.println("\n+---------------+-------------------+-----------------+-----------------------
-------+---------------+-------------------+--------------+------------+------------------+------------------
+");
        System.out.println("| Customer ID   | Customer Name      | Phone Number     |
Email                    | Order ID       | Order Time         | Order Status  | Food Price | Food
Name        | Payment Method    |");
        System.out.println("+---------------+-------------------+-----------------+-----------------------
-----+---------------+-------------------+--------------+------------+------------------+------------------+");


        while(!CustomerQ.isEmpty())
        {
            obj = CustomerQ.dequeue();
           if(obj.getRestaurant().getFoodPrice() < 10.00)
           {
```

```
                System.out.println(obj.toString());
            }
            tempQ.enqueue(obj);
        }

        System.out.println("\n+---------------+--------------------+-----------------+------------------------
-------+---------------+--------------------+--------------+------------+------------------+-----------------
+");



         while(!tempQ.isEmpty())
        {
            CustomerQ.enqueue(tempQ.dequeue());//yang buang tadi akan masuk balik
        }

        //7. Find Highest And Lowest price of customer order

        Customer highestObj = null;
        Customer lowestObj = null;




        // Initialize highestObj and lowestObj with the first customer
        if (!CustomerQ.isEmpty())
        {
            highestObj = CustomerQ.dequeue();
            lowestObj = highestObj; // Both start as the same first customer
            tempQ.enqueue(highestObj); // Enqueue back into tempQ for further
processing
        }

        // Compare remaining customers to find the highest and lowest food prices
        while (!CustomerQ.isEmpty())
```

```
        {
            obj = CustomerQ.dequeue();
            if (obj.getRestaurant().getFoodPrice() >
highestObj.getRestaurant().getFoodPrice())
            {
            highestObj = obj;
            }
            if (obj.getRestaurant().getFoodPrice() <
lowestObj.getRestaurant().getFoodPrice())
            {
            lowestObj = obj;
            }
            tempQ.enqueue(obj);
        }

        // Restore the original queue
        while (!tempQ.isEmpty())
        {
            CustomerQ.enqueue(tempQ.dequeue());
        }


        System.out.println("\n+---------------+--------------------+-----------------+---------------
Display Customer Highest and Lowest Price-------------+----------------+--------------------+------
---------+-----------+-");

        System.out.println("\nCustomer with the Highest Food Price:");
        System.out.println(highestObj);
        System.out.println("\nCustomer with the Lowest Food Price:");
        System.out.println(lowestObj);

        System.out.println("\n+---------------+--------------------+-----------------+-----------------------
-------+----------------+--------------------+---------------+------------+------------------+-----------------
+");
```

```
// 8. Calculate And Display the Average food Price in Customer List


double totalFoodPrice = 0;
int customerCount = 0;


System.out.println("\n+---------------+-------------------+-----------------+---------------
Display average food price among all customer---------+---------------+------------+-------------
-----+------------------+");




while (!CustomerQ.isEmpty())
{
    obj = CustomerQ.dequeue();


    totalFoodPrice += obj.getRestaurant().getFoodPrice(); // Accumulate food
prices
    customerCount++; // Count each customer


    tempQ.enqueue(obj); // Move to the next customer
}


if (customerCount > 0)
{
    double averageFoodPrice = totalFoodPrice / customerCount;
    System.out.printf("\nThe average food price among all customers is:
RM%.2f%n", averageFoodPrice);
} else
{
    System.out.println("\nNo customers available to calculate the average food
price.");
}
```

```
        System.out.println("\n+---------------+-------------------+-----------------+-----------------------
--------+---------------+-------------------+--------------+------------+------------------+-----------------
+");



        // Restore custQ from tempQ
        while (!tempQ.isEmpty())
        {
           CustomerQ.enqueue(tempQ.dequeue());
        }


         // 9 - Calculate total Price of Payment method
        System.out.print("\nEnter the payment method to calculate total price for(cash |
online | credit): ");
        String searchPayment = scanner.nextLine().trim();
        double totalPriceC = 0.00;
        double totalPriceOn9 = 0.00 ;
        double totalPriceCard = 0.00;




        while(!CustomerQ.isEmpty())
        {
           obj = CustomerQ.dequeue();
              if(obj.getRestaurant().getPaymentMethod().equalsIgnoreCase("Cash"))
              {
                 totalPriceC += obj.getRestaurant().getFoodPrice();
              }
              else if(obj.getRestaurant().getPaymentMethod().equalsIgnoreCase("Online
Transfer"))
              {
                 totalPriceOn9 += obj.getRestaurant().getFoodPrice();
              }
```

```
            else if(obj.getRestaurant().getPaymentMethod().equalsIgnoreCase("Credit
Card"))
            {
               totalPriceCard += obj.getRestaurant().getFoodPrice();
            }
            tempQ.enqueue(obj);
        }

        // Restore custQ from tempQ
        while (!tempQ.isEmpty())
        {
           CustomerQ.enqueue(tempQ.dequeue());
        }

        if(searchPayment.equalsIgnoreCase("Cash")){
                System.out.println("Total price customer who pay with cash : " +
df.format(totalPriceC));
            }
            else if(searchPayment.equalsIgnoreCase("Online Transfer")){
                System.out.printf("Total price customer who pay with online payment : " +
df.format(totalPriceOn9));
            }
            else if(searchPayment.equalsIgnoreCase("Credit")){
                System.out.printf("Total price customer who pay with : " +
df.format(totalPriceCard));


            }




        System.out.println("\n+---------------+--------------------+-----------------+-----------------------
--------+----------------+--------------------+---------------+------------+------------------+------------------
+");
        // 10. Generate a report for food names and the number of customers
        System.out.println("\nReport for food names and the number of the customers");
```

```
        System.out.println("\nFood Report:");
        System.out.println("+------------------+-----------------+");
        System.out.println("| Food Name        | Total Customers |");
        System.out.println("+------------------+-----------------+");

        // Outer loop to traverse through CustomerQ
        while (!CustomerQ.isEmpty()) {
            obj = CustomerQ.dequeue(); // Dequeue a customer
            String currentFoodName = obj.getRestaurant().getFoodName(); // Get the food
name
            boolean alreadyCounted = false;

            // Check if this food name has already been processed
            Queue<Customer> checkTempQ = new Queue<>(); // Temporary queue for
iteration
            while (!tempQ.isEmpty()) {
                Customer tempCustomer = tempQ.dequeue();
                if (tempCustomer.getRestaurant().getFoodName().equals(currentFoodName))
{
                    alreadyCounted = true;
                }
                checkTempQ.enqueue(tempCustomer); // Restore tempQ
            }

            // Restore tempQ after checking
            while (!checkTempQ.isEmpty()) {
                tempQ.enqueue(checkTempQ.dequeue());
            }

            if (!alreadyCounted) {
                int foodCount = 1; // Start with the current customer

                //Inner loop to count occurrences of the current food name
                Queue<Customer> innerTempQ = new Queue<>();
```

```
            while (!CustomerQ.isEmpty())
            {
               Customer tempObj = CustomerQ.dequeue();
               if (tempObj.getRestaurant().getFoodName().equals(currentFoodName)) {
                  foodCount++;
               }
               innerTempQ.enqueue(tempObj); // Restore the original queue
            }

            // Restore CustomerQ from innerTempQ
            while (!innerTempQ.isEmpty()) {
               CustomerQ.enqueue(innerTempQ.dequeue());
            }

            // Display the result for the current food name
            System.out.printf("| %-17s | %-16d |\n", currentFoodName, foodCount);

         }

         tempQ.enqueue(obj); // Add the current customer to tempQ
      }

      // Restore CustomerQ from tempQ
      while (!tempQ.isEmpty()) {
      CustomerQ.enqueue(tempQ.dequeue());
      }

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

   }
}
```

4. SAMPLE INPUT AND OUTPUT

Nasi Lemak	4.00
Char Kway Teow	6.00
Roti Canai	2.00
Nasi Dagang	6.50
Nasi Kerabu	7.00

Display Customer Order Information

Customer ID	Customer Name	Phone Number	Email	Order ID	Order Time	Order Status	Food Price	Food Name	Payment Method
C001	Ahmad Zulkifli	0123456789	ahmad.zulkifli@gmail.com	O001	2025-01-12 12:00	complete	4.00	Nasi Lemak	cash
C002	Siti Nurhaliza	0987654321	siti.nurhaliza@gmail.com	O002	2025-01-12 12:30	pending	6.00	Char Kway Teow	credit card
C003	Muhammad Hafiz	0112233445	muhammad.hafiz@yahoo.com	O003	2025-01-12 13:00	complete	4.00	Nasi Lemak	online transfer
C004	Zainab Abdullah	0223344556	zainab.abdullah@outlook.com	O004	2025-01-12 13:15	complete	7.00	Nasi Kerabu	credit card
C005	Lim Beng Hock	0334455667	lim.benghock@gmail.com	O005	2025-01-12 14:00	cancel	6.00	Char Kway Teow	cash
C006	Noor Fatimah	0445566778	noor.fatimah@hotmail.com	O006	2025-01-12 14:30	pending	2.00	Roti Canai	online transfer
C007	Ravi Kumar	0556677889	ravi.kumar@gmail.com	O007	2025-01-12 15:00	complete	6.50	Nasi Dagang	cash
C008	Tan Mei Ling	0667788990	tan.meiling@yahoo.com	O008	2025-01-12 15:15	complete	6.50	Nasi Dagang	credit card
C009	Chong Wei	0778899001	chong.wei@outlook.com	O009	2025-01-12 16:00	complete	6.00	Char Kway Teow	online transfer
C010	Nurul Ain	0889900112	nurul.ain@gmail.com	O010	2025-01-12 16:30	cancel	4.00	Nasi Lemak	credit card
C011	Azman Latif	0990011223	azman.latif@gmail.com	O011	2025-01-12 17:00	pending	2.00	Roti Canai	cash
C012	Aisyah Rahman	0101122334	aisyah.rahman@yahoo.com	O012	2025-01-12 17:15	complete	7.00	Nasi Kerabu	online transfer
C013	Nagarajan Raj	0111233445	nagarajan.raj@gmail.com	O013	2025-01-12 18:00	pending	6.50	Nasi Dagang	credit card
C014	Liew Hong	0122233445	liew.hong@gmail.com	O014	2025-01-12 18:30	complete	6.00	Char Kway Teow	cash
C015	Farid Ismail	0133344556	farid.ismail@hotmail.com	O015	2025-01-12 19:00	cancel	2.00	Roti Canai	online transfer
C016	Sarah Zahira	0144455667	sarah.zahira@gmail.com	O016	2025-01-12 19:15	complete	7.00	Nasi Kerabu	credit card
C017	Teo Kian Seng	0155566778	teo.kianseng@yahoo.com	O017	2025-01-12 20:00	complete	6.50	Nasi Dagang	online transfer
C018	Rosnah Daud	0166677889	rosnah.daud@gmail.com	O018	2025-01-12 20:30	pending	6.50	Nasi Dagang	cash
C019	Hisham Mokhtar	0177788990	hisham.mokhtar@outlook.com	O019	2025-01-12 21:00	complete	6.00	Char Kway Teow	credit card
C020	Lee Choon Wah	0188899001	lee.choonwah@gmail.com	O020	2025-01-12 21:30	cancel	2.00	Roti Canai	online transfer
C021	Ahmad Bakri	0199900112	ahmad.bakri@gmail.com	O021	2025-01-12 22:00	complete	4.00	Nasi Lemak	cash
C022	Nur Syafiqah	0200011223	nur.syafiqah@gmail.com	O022	2025-01-12 22:30	pending	4.00	Nasi Lemak	credit card
C023	Roslan Hamid	0211122334	roslan.hamid@yahoo.com	O023	2025-01-12 23:00	complete	6.00	Char Kway Teow	online transfer
C024	Amira Hassan	0222233445	amira.hassan@gmail.com	O024	2025-01-12 23:15	complete	6.50	Nasi Dagang	cash
C025	Wan Fariz	0233344556	wan.fariz@outlook.com	O025	2025-01-12 23:30	complete	7.00	Nasi Kerabu	credit card
C026	Chee Yin	0244455667	chee.yin@gmail.com	O026	2025-01-12 23:45	cancel	6.50	Nasi Dagang	online transfer
C027	Fatimah Yaakob	0255566778	fatimah.yaakob@yahoo.com	O027	2025-01-13 00:00	complete	6.00	Char Kway Teow	cash
C028	Singh Amarjit	0266677889	singh.amarjit@gmail.com	O028	2025-01-13 00:15	pending	6.50	Nasi Dagang	credit card
C029	Noor Hafizah	0277788990	noor.hafizah@gmail.com	O029	2025-01-13 00:30	complete	2.00	Roti Canai	online transfer
C030	Zamri Nasir	0288899001	zamri.nasir@gmail.com	O030	2025-01-13 01:00	complete	7.00	Nasi Kerabu	cash

Count and Display number of customer payment method

Enter the payment method to search for(cash | online | credit): cash
Number of customers who pay with 'cash': 10

Search customer id and Update Status

Enter the Customer ID to search for and update order status: C001

Customer found:

C001	Ahmad Zulkifli	0123456789	ahmad.zulkifli@gmail.com	O001	2025-01-12 12:00	complete	4.00	Nasi Lemak	cash

Update Status

Enter the new order status: pending

Order status updated successfully!

C001	Ahmad Zulkifli	0123456789	ahmad.zulkifli@gmail.com	O001	2025-01-12 12:00	pending	4.00	Nasi Lemak	cash

Update Customer Information

C001	Ahmad Zulkifli	0123456789	ahmad.zulkifli@gmail.com	O001	2025-01-12 12:00	pending	4.00	Nasi Lemak	cash
C002	Siti Nurhaliza	0987654321	siti.nurhaliza@gmail.com	O002	2025-01-12 12:30	pending	6.00	Char Kway Teow	credit card
C003	Muhammad Hafiz	0112233445	muhammad.hafiz@yahoo.com	O003	2025-01-12 13:00	complete	4.00	Nasi Lemak	online transfer
C004	Zainab Abdullah	0223344556	zainab.abdullah@outlook.com	O004	2025-01-12 13:15	complete	7.00	Nasi Kerabu	credit card
C005	Lim Beng Hock	0334455667	lim.benghock@gmail.com	O005	2025-01-12 14:00	cancel	6.00	Char Kway Teow	cash
C006	Noor Fatimah	0445566778	noor.fatimah@hotmail.com	O006	2025-01-12 14:30	pending	2.00	Roti Canai	online transfer
C007	Ravi Kumar	0556677889	ravi.kumar@gmail.com	O007	2025-01-12 15:00	complete	6.50	Nasi Dagang	cash
C008	Tan Mei Ling	0667788990	tan.meiling@yahoo.com	O008	2025-01-12 15:15	complete	6.50	Nasi Dagang	credit card
C009	Chong Wei	0778899001	chong.wei@outlook.com	O009	2025-01-12 16:00	complete	6.00	Char Kway Teow	online transfer
C010	Nurul Ain	0889900112	nurul.ain@gmail.com	O010	2025-01-12 16:30	cancel	4.00	Nasi Lemak	credit card
C011	Azman Latif	0990011223	azman.latif@gmail.com	O011	2025-01-12 17:00	pending	2.00	Roti Canai	cash
C012	Aisyah Rahman	0101122334	aisyah.rahman@yahoo.com	O012	2025-01-12 17:15	complete	7.00	Nasi Kerabu	online transfer
C013	Nagarajan Raj	0111233445	nagarajan.raj@gmail.com	O013	2025-01-12 18:00	pending	6.50	Nasi Dagang	credit card
C014	Liew Hong	0122233445	liew.hong@gmail.com	O014	2025-01-12 18:30	complete	6.00	Char Kway Teow	cash
C015	Farid Ismail	0133344556	farid.ismail@hotmail.com	O015	2025-01-12 19:00	cancel	2.00	Roti Canai	online transfer
C016	Sarah Zahira	0144455667	sarah.zahira@gmail.com	O016	2025-01-12 19:15	complete	7.00	Nasi Kerabu	credit card
C017	Teo Kian Seng	0155566778	teo.kianseng@yahoo.com	O017	2025-01-12 20:00	complete	6.50	Nasi Dagang	online transfer
C018	Rosnah Daud	0166677889	rosnah.daud@gmail.com	O018	2025-01-12 20:30	pending	6.50	Nasi Dagang	cash
C019	Hisham Mokhtar	0177788990	hisham.mokhtar@outlook.com	O019	2025-01-12 21:00	complete	6.00	Char Kway Teow	credit card
C020	Lee Choon Wah	0188899001	lee.choonwah@gmail.com	O020	2025-01-12 21:30	cancel	2.00	Roti Canai	online transfer
C021	Ahmad Bakri	0199900112	ahmad.bakri@gmail.com	O021	2025-01-12 22:00	complete	4.00	Nasi Lemak	cash
C022	Nur Syafiqah	0200011223	nur.syafiqah@gmail.com	O022	2025-01-12 22:30	pending	4.00	Nasi Lemak	credit card
C023	Roslan Hamid	0211122334	roslan.hamid@yahoo.com	O023	2025-01-12 23:00	complete	6.00	Char Kway Teow	online transfer
C024	Amira Hassan	0222233445	amira.hassan@gmail.com	O024	2025-01-12 23:15	complete	6.50	Nasi Dagang	cash
C025	Wan Fariz	0233344556	wan.fariz@outlook.com	O025	2025-01-12 23:30	complete	7.00	Nasi Kerabu	credit card
C026	Chee Yin	0244455667	chee.yin@gmail.com	O026	2025-01-12 23:45	cancel	6.50	Nasi Dagang	online transfer
C027	Fatimah Yaakob	0255566778	fatimah.yaakob@yahoo.com	O027	2025-01-13 00:00	complete	6.00	Char Kway Teow	cash
C028	Singh Amarjit	0266677889	singh.amarjit@gmail.com	O028	2025-01-13 00:15	pending	6.50	Nasi Dagang	credit card
C029	Noor Hafizah	0277788990	noor.hafizah@gmail.com	O029	2025-01-13 00:30	complete	2.00	Roti Canai	online transfer
C030	Zamri Nasir	0288899001	zamri.nasir@gmail.com	O030	2025-01-13 01:00	complete	7.00	Nasi Kerabu	cash

Search Customer Name and Update Payment Method

```
+--------------+--------------------+----------------+----------------------------+---------------+------------------+--------------+-------------------+------------------+

Enter the Customer Name to search for and update payment method: Zamri Nasir

Customer found:
| C030         | Zamri Nasir        | 0288899001     | zamri.nasir@gmail.com      | O030          | 2025-01-13 01:00 | complete     | 7.00         | Nasi Kerabu       | cash             |

+--------------+--------------------+----------------+----------------------------+---------------+------------------+--------------+-------------------+------------------+


+--------------+--------------------+----------------+----------------------------+---------------+------------------+--------------+-------------------+------------------+
Enter the new payment method: credit

Customer payment method updated successfully!

| C030         | Zamri Nasir        | 0288899001     | zamri.nasir@gmail.com      | O030          | 2025-01-13 01:00 | complete     | 7.00         | Nasi Kerabu       | credit           |

+--------------+--------------------+----------------+----------------------------+---------------+------------------+--------------+-------------------+------------------+
+--------------+--------------------+----------------+----------------------------+---------------+------------------+--------------+-------------------+------------------+
| C001         | Ahmad Zulkifli     | 0123456789     | ahmad.zulkifli@gmail.com   | O001          | 2025-01-12 12:00 | pending      | 4.00         | Nasi Lemak        | cash             |
| C002         | Siti Nurhaliza     | 0987654321     | siti.nurhaliza@gmail.com   | O002          | 2025-01-12 12:30 | pending      | 6.00         | Char Kway Teow    | credit card      |
| C003         | Muhammad Hafiz     | 0112233445     | muhammad.hafiz@yahoo.com   | O003          | 2025-01-12 13:00 | complete     | 4.00         | Nasi Lemak        | online transfer  |
| C004         | Zainab Abdullah    | 0223344556     | zainab.abdullah@outlook.com| O004          | 2025-01-12 13:15 | complete     | 7.00         | Nasi Kerabu       | credit card      |
| C005         | Lim Beng Hock      | 0334455667     | lim.benghock@gmail.com     | O005          | 2025-01-12 14:00 | cancel       | 6.00         | Char Kway Teow    | cash             |
| C006         | Noor Fatimah       | 0445566778     | noor.fatimah@hotmail.com   | O006          | 2025-01-12 14:30 | pending      | 2.00         | Roti Canai        | online transfer  |
| C007         | Ravi Kumar         | 0556677889     | ravi.kumar@gmail.com       | O007          | 2025-01-12 15:00 | complete     | 6.50         | Nasi Dagang       | cash             |
| C008         | Tan Mei Ling       | 0667788990     | tan.meiling@yahoo.com      | O008          | 2025-01-12 15:15 | complete     | 6.50         | Nasi Dagang       | credit card      |
| C009         | Chong Wei          | 0778899001     | chong.wei@outlook.com      | O009          | 2025-01-12 16:00 | complete     | 6.00         | Char Kway Teow    | online transfer  |
| C010         | Nurul Ain          | 0889900112     | nurul.ain@gmail.com        | O010          | 2025-01-12 16:30 | cancel       | 4.00         | Nasi Lemak        | credit card      |
| C011         | Azman Latif        | 0990011223     | azman.latif@gmail.com      | O011          | 2025-01-12 17:00 | pending      | 2.00         | Roti Canai        | cash             |
| C012         | Aisyah Rahman      | 0101122334     | aisyah.rahman@yahoo.com    | O012          | 2025-01-12 17:15 | complete     | 7.00         | Nasi Kerabu       | online transfer  |
| C013         | Nagarajan Raj      | 0111233445     | nagarajan.raj@gmail.com    | O013          | 2025-01-12 18:00 | pending      | 6.50         | Nasi Dagang       | credit card      |
| C014         | Liew Hong          | 0122233445     | liew.hong@gmail.com        | O014          | 2025-01-12 18:30 | complete     | 6.00         | Char Kway Teow    | cash             |
| C015         | Farid Ismail       | 0133344556     | farid.ismail@hotmail.com   | O015          | 2025-01-12 19:00 | cancel       | 2.00         | Roti Canai        | online transfer  |
| C016         | Sarah Zahira       | 0144455667     | sarah.zahira@gmail.com     | O016          | 2025-01-12 19:15 | complete     | 7.00         | Nasi Kerabu       | credit card      |
| C017         | Teo Kian Seng      | 0155566778     | teo.kianseng@yahoo.com     | O017          | 2025-01-12 20:00 | complete     | 6.50         | Nasi Dagang       | online transfer  |
| C018         | Rosnah Daud        | 0166677889     | rosnah.daud@gmail.com      | O018          | 2025-01-12 20:30 | pending      | 6.50         | Nasi Dagang       | cash             |
| C019         | Hisham Mokhtar     | 0177788990     | hisham.mokhtar@outlook.com | O019          | 2025-01-12 21:00 | complete     | 6.00         | Char Kway Teow    | credit card      |
| C020         | Lee Choon Wah      | 0188899001     | lee.choonwah@gmail.com     | O020          | 2025-01-12 21:30 | cancel       | 2.00         | Roti Canai        | online transfer  |
| C021         | Ahmad Bakri        | 0199900112     | ahmad.bakri@gmail.com      | O021          | 2025-01-12 22:00 | complete     | 4.00         | Nasi Lemak        | cash             |
| C022         | Nur Syafiqah       | 0200011223     | nur.syafiqah@gmail.com     | O022          | 2025-01-12 22:30 | pending      | 4.00         | Nasi Lemak        | credit card      |
| C023         | Roslan Hamid       | 0211122334     | roslan.hamid@yahoo.com     | O023          | 2025-01-12 23:00 | complete     | 6.00         | Char Kway Teow    | online transfer  |
| C024         | Amira Hassan       | 0222233445     | amira.hassan@gmail.com     | O024          | 2025-01-12 23:15 | complete     | 6.50         | Nasi Dagang       | cash             |
| C025         | Wan Fariz          | 0233344556     | wan.fariz@outlook.com      | O025          | 2025-01-12 23:30 | complete     | 7.00         | Nasi Kerabu       | credit card      |
| C026         | Chee Yin           | 0244455667     | chee.yin@gmail.com         | O026          | 2025-01-12 23:45 | cancel       | 6.50         | Nasi Dagang       | online transfer  |
| C027         | Fatimah Yaakob     | 0255566778     | fatimah.yaakob@yahoo.com   | O027          | 2025-01-13 00:00 | complete     | 6.00         | Char Kway Teow    | cash             |
| C028         | Singh Amarjit      | 0266677889     | singh.amarjit@gmail.com    | O028          | 2025-01-13 00:15 | pending      | 6.50         | Nasi Dagang       | credit card      |
| C029         | Noor Hafizah       | 0277788990     | noor.hafizah@gmail.com     | O029          | 2025-01-13 00:30 | complete     | 2.00         | Roti Canai        | online transfer  |
| C030         | Zamri Nasir        | 0288899001     | zamri.nasir@gmail.com      | O030          | 2025-01-13 01:00 | complete     | 7.00         | Nasi Kerabu       | credit           |

+--------------+--------------------+----------------+----------------------------+---------------+------------------+--------------+-------------------+------------------+
```

Sort Customer Payment Method

```
+--------------+--------------------+----------------+----------------------------+---------------+------------------+--------------+-------------------+------------------+

All customers (custQ):
| C001         | Ahmad Zulkifli     | 0123456789     | ahmad.zulkifli@gmail.com   | O001          | 2025-01-12 12:00 | pending      | 4.00         | Nasi Lemak        | cash             |
| C002         | Siti Nurhaliza     | 0987654321     | siti.nurhaliza@gmail.com   | O002          | 2025-01-12 12:30 | pending      | 6.00         | Char Kway Teow    | credit card      |
| C003         | Muhammad Hafiz     | 0112233445     | muhammad.hafiz@yahoo.com   | O003          | 2025-01-12 13:00 | complete     | 4.00         | Nasi Lemak        | online transfer  |
| C004         | Zainab Abdullah    | 0223344556     | zainab.abdullah@outlook.com| O004          | 2025-01-12 13:15 | complete     | 7.00         | Nasi Kerabu       | credit card      |
| C005         | Lim Beng Hock      | 0334455667     | lim.benghock@gmail.com     | O005          | 2025-01-12 14:00 | cancel       | 6.00         | Char Kway Teow    | cash             |
| C006         | Noor Fatimah       | 0445566778     | noor.fatimah@hotmail.com   | O006          | 2025-01-12 14:30 | pending      | 2.00         | Roti Canai        | online transfer  |
| C007         | Ravi Kumar         | 0556677889     | ravi.kumar@gmail.com       | O007          | 2025-01-12 15:00 | complete     | 6.50         | Nasi Dagang       | cash             |
| C008         | Tan Mei Ling       | 0667788990     | tan.meiling@yahoo.com      | O008          | 2025-01-12 15:15 | complete     | 6.50         | Nasi Dagang       | credit card      |
| C009         | Chong Wei          | 0778899001     | chong.wei@outlook.com      | O009          | 2025-01-12 16:00 | complete     | 6.00         | Char Kway Teow    | online transfer  |
| C010         | Nurul Ain          | 0889900112     | nurul.ain@gmail.com        | O010          | 2025-01-12 16:30 | cancel       | 4.00         | Nasi Lemak        | credit card      |
| C011         | Azman Latif        | 0990011223     | azman.latif@gmail.com      | O011          | 2025-01-12 17:00 | pending      | 2.00         | Roti Canai        | cash             |
| C012         | Aisyah Rahman      | 0101122334     | aisyah.rahman@yahoo.com    | O012          | 2025-01-12 17:15 | complete     | 7.00         | Nasi Kerabu       | online transfer  |
| C013         | Nagarajan Raj      | 0111233445     | nagarajan.raj@gmail.com    | O013          | 2025-01-12 18:00 | pending      | 6.50         | Nasi Dagang       | credit card      |
| C014         | Liew Hong          | 0122233445     | liew.hong@gmail.com        | O014          | 2025-01-12 18:30 | complete     | 6.00         | Char Kway Teow    | cash             |
| C015         | Farid Ismail       | 0133344556     | farid.ismail@hotmail.com   | O015          | 2025-01-12 19:00 | cancel       | 2.00         | Roti Canai        | online transfer  |
| C016         | Sarah Zahira       | 0144455667     | sarah.zahira@gmail.com     | O016          | 2025-01-12 19:15 | complete     | 7.00         | Nasi Kerabu       | credit card      |
| C017         | Teo Kian Seng      | 0155566778     | teo.kianseng@yahoo.com     | O017          | 2025-01-12 20:00 | complete     | 6.50         | Nasi Dagang       | online transfer  |
| C018         | Rosnah Daud        | 0166677889     | rosnah.daud@gmail.com      | O018          | 2025-01-12 20:30 | pending      | 6.50         | Nasi Dagang       | cash             |
| C019         | Hisham Mokhtar     | 0177788990     | hisham.mokhtar@outlook.com | O019          | 2025-01-12 21:00 | complete     | 6.00         | Char Kway Teow    | credit card      |
| C020         | Lee Choon Wah      | 0188899001     | lee.choonwah@gmail.com     | O020          | 2025-01-12 21:30 | cancel       | 2.00         | Roti Canai        | online transfer  |
| C021         | Ahmad Bakri        | 0199900112     | ahmad.bakri@gmail.com      | O021          | 2025-01-12 22:00 | complete     | 4.00         | Nasi Lemak        | cash             |
| C022         | Nur Syafiqah       | 0200011223     | nur.syafiqah@gmail.com     | O022          | 2025-01-12 22:30 | pending      | 4.00         | Nasi Lemak        | credit card      |
| C023         | Roslan Hamid       | 0211122334     | roslan.hamid@yahoo.com     | O023          | 2025-01-12 23:00 | complete     | 6.00         | Char Kway Teow    | online transfer  |
| C024         | Amira Hassan       | 0222233445     | amira.hassan@gmail.com     | O024          | 2025-01-12 23:15 | complete     | 6.50         | Nasi Dagang       | cash             |
| C025         | Wan Fariz          | 0233344556     | wan.fariz@outlook.com      | O025          | 2025-01-12 23:30 | complete     | 7.00         | Nasi Kerabu       | credit card      |
| C026         | Chee Yin           | 0244455667     | chee.yin@gmail.com         | O026          | 2025-01-12 23:45 | cancel       | 6.50         | Nasi Dagang       | online transfer  |
| C027         | Fatimah Yaakob     | 0255566778     | fatimah.yaakob@yahoo.com   | O027          | 2025-01-13 00:00 | complete     | 6.00         | Char Kway Teow    | cash             |
| C028         | Singh Amarjit      | 0266677889     | singh.amarjit@gmail.com    | O028          | 2025-01-13 00:15 | pending      | 6.50         | Nasi Dagang       | credit card      |
| C029         | Noor Hafizah       | 0277788990     | noor.hafizah@gmail.com     | O029          | 2025-01-13 00:30 | complete     | 2.00         | Roti Canai        | online transfer  |
| C030         | Zamri Nasir        | 0288899001     | zamri.nasir@gmail.com      | O030          | 2025-01-13 01:00 | complete     | 7.00         | Nasi Kerabu       | credit           |
```

```
Customers who paid with Credit Card (creditLL):
| C002          | Siti Nurhaliza     | 0987654321      | siti.nurhaliza@gmail.com       | O002          | 2025-01-12 12:30   | pending       | 6.00       | Char Kway Teow    | credit card       |
| C004          | Zainab Abdullah    | 0223344556      | zainab.abdullah@outlook.com    | O004          | 2025-01-12 13:15   | complete      | 7.00       | Nasi Kerabu       | credit card       |
| C008          | Tan Mei Ling       | 0667788990      | tan.meiling@yahoo.com          | O008          | 2025-01-12 15:15   | complete      | 6.50       | Nasi Dagang       | credit card       |
| C010          | Nurul Ain          | 0889900112      | nurul.ain@gmail.com            | O010          | 2025-01-12 16:30   | cancel        | 4.00       | Nasi Lemak        | credit card       |
| C013          | Nagarajan Raj      | 0111233445      | nagarajan.raj@gmail.com        | O013          | 2025-01-12 18:00   | pending       | 6.50       | Nasi Dagang       | credit card       |
| C016          | Sarah Zahira       | 0144455667      | sarah.zahira@gmail.com         | O016          | 2025-01-12 19:15   | complete      | 7.00       | Nasi Kerabu       | credit card       |
| C019          | Hisham Mokhtar     | 0177788990      | hisham.mokhtar@outlook.com     | O019          | 2025-01-12 21:00   | complete      | 6.00       | Char Kway Teow    | credit card       |
| C022          | Nur Syafiqah       | 0200011223      | nur.syafiqah@gmail.com         | O022          | 2025-01-12 22:30   | pending       | 4.00       | Nasi Lemak        | credit card       |
| C025          | Wan Fariz          | 0233344556      | wan.fariz@outlook.com          | O025          | 2025-01-12 23:30   | complete      | 7.00       | Nasi Kerabu       | credit card       |
| C028          | Singh Amarjit      | 0266677889      | singh.amarjit@gmail.com        | O028          | 2025-01-13 00:15   | pending       | 6.50       | Nasi Dagang       | credit card       |

Customers who paid with Online Transfer (onlineLL):
| C003          | Muhammad Hafiz     | 0112233445      | muhammad.hafiz@yahoo.com       | O003          | 2025-01-12 13:00   | complete      | 4.00       | Nasi Lemak        | online transfer   |
| C006          | Noor Fatimah       | 0445566778      | noor.fatimah@hotmail.com       | O006          | 2025-01-12 14:30   | pending       | 2.00       | Roti Canai        | online transfer   |
| C009          | Chong Wei          | 0778899001      | chong.wei@outlook.com          | O009          | 2025-01-12 16:00   | complete      | 6.00       | Char Kway Teow    | online transfer   |
| C012          | Aisyah Rahman      | 0101122334      | aisyah.rahman@yahoo.com        | O012          | 2025-01-12 17:15   | complete      | 7.00       | Nasi Kerabu       | online transfer   |
| C015          | Farid Ismail       | 0133344556      | farid.ismail@hotmail.com       | O015          | 2025-01-12 19:00   | cancel        | 2.00       | Roti Canai        | online transfer   |
| C017          | Teo Kian Seng      | 0155566778      | teo.kianseng@yahoo.com         | O017          | 2025-01-12 20:00   | complete      | 6.50       | Nasi Dagang       | online transfer   |
| C020          | Lee Choon Wah      | 0188899001      | lee.choonwah@gmail.com         | O020          | 2025-01-12 21:30   | cancel        | 2.00       | Roti Canai        | online transfer   |
| C023          | Roslan Hamid       | 0211122334      | roslan.hamid@yahoo.com         | O023          | 2025-01-12 23:00   | complete      | 6.00       | Char Kway Teow    | online transfer   |
| C026          | Chee Yin           | 0244455667      | chee.yin@gmail.com             | O026          | 2025-01-12 23:45   | cancel        | 6.50       | Nasi Dagang       | online transfer   |
| C029          | Noor Hafizah       | 0277788990      | noor.hafizah@gmail.com         | O029          | 2025-01-13 00:30   | complete      | 2.00       | Roti Canai        | online transfer   |

Customers who paid with Cash Payment (cashLL):
| C001          | Ahmad Zulkifli     | 0123456789      | ahmad.zulkifli@gmail.com       | O001          | 2025-01-12 12:00   | pending       | 4.00       | Nasi Lemak        | cash              |
| C005          | Lim Beng Hock      | 0334455667      | lim.benghock@gmail.com         | O005          | 2025-01-12 14:00   | cancel        | 6.00       | Char Kway Teow    | cash              |
| C007          | Ravi Kumar         | 0556677889      | ravi.kumar@gmail.com           | O007          | 2025-01-12 15:00   | complete      | 6.50       | Nasi Dagang       | cash              |
| C011          | Azman Latif        | 0990011223      | azman.latif@gmail.com          | O011          | 2025-01-12 17:00   | pending       | 2.00       | Roti Canai        | cash              |
| C014          | Liew Hong          | 0122233445      | liew.hong@gmail.com            | O014          | 2025-01-12 18:30   | complete      | 6.00       | Char Kway Teow    | cash              |
| C018          | Rosnah Daud        | 0166677889      | rosnah.daud@gmail.com          | O018          | 2025-01-12 20:30   | pending       | 6.50       | Nasi Dagang       | cash              |
| C021          | Ahmad Bakri        | 0199900112      | ahmad.bakri@gmail.com          | O021          | 2025-01-12 22:00   | complete      | 4.00       | Nasi Lemak        | cash              |
| C024          | Amira Hassan       | 0222233445      | amira.hassan@gmail.com         | O024          | 2025-01-12 23:15   | complete      | 6.50       | Nasi Dagang       | cash              |
| C027          | Fatimah Yaakob     | 0255566778      | fatimah.yaakob@yahoo.com       | O027          | 2025-01-13 00:00   | complete      | 6.00       | Char Kway Teow    | cash              |

Display Customer Information where the food price below than RM 10.00

----------Display Customer Food Price less than RM10----------

| Customer ID   | Customer Name      | Phone Number    | Email                          | Order ID      | Order Time         | Order Status  | Food Price | Food Name         | Payment Method    |
| C001          | Ahmad Zulkifli     | 0123456789      | ahmad.zulkifli@gmail.com       | O001          | 2025-01-12 12:00   | pending       | 4.00       | Nasi Lemak        | cash              |
| C002          | Siti Nurhaliza     | 0987654321      | siti.nurhaliza@gmail.com       | O002          | 2025-01-12 12:30   | pending       | 6.00       | Char Kway Teow    | credit card       |
| C003          | Muhammad Hafiz     | 0112233445      | muhammad.hafiz@yahoo.com       | O003          | 2025-01-12 13:00   | complete      | 4.00       | Nasi Lemak        | online transfer   |
| C004          | Zainab Abdullah    | 0223344556      | zainab.abdullah@outlook.com    | O004          | 2025-01-12 13:15   | complete      | 7.00       | Nasi Kerabu       | credit card       |
| C005          | Lim Beng Hock      | 0334455667      | lim.benghock@gmail.com         | O005          | 2025-01-12 14:00   | cancel        | 6.00       | Char Kway Teow    | cash              |
| C006          | Noor Fatimah       | 0445566778      | noor.fatimah@hotmail.com       | O006          | 2025-01-12 14:30   | pending       | 2.00       | Roti Canai        | online transfer   |
| C007          | Ravi Kumar         | 0556677889      | ravi.kumar@gmail.com           | O007          | 2025-01-12 15:00   | complete      | 6.50       | Nasi Dagang       | cash              |
| C008          | Tan Mei Ling       | 0667788990      | tan.meiling@yahoo.com          | O008          | 2025-01-12 15:15   | complete      | 6.50       | Nasi Dagang       | credit card       |
| C009          | Chong Wei          | 0778899001      | chong.wei@outlook.com          | O009          | 2025-01-12 16:00   | complete      | 6.00       | Char Kway Teow    | online transfer   |
| C010          | Nurul Ain          | 0889900112      | nurul.ain@gmail.com            | O010          | 2025-01-12 16:30   | cancel        | 4.00       | Nasi Lemak        | credit card       |
| C011          | Azman Latif        | 0990011223      | azman.latif@gmail.com          | O011          | 2025-01-12 17:00   | pending       | 2.00       | Roti Canai        | cash              |
| C012          | Aisyah Rahman      | 0101122334      | aisyah.rahman@yahoo.com        | O012          | 2025-01-12 17:15   | complete      | 7.00       | Nasi Kerabu       | online transfer   |
| C013          | Nagarajan Raj      | 0111233445      | nagarajan.raj@gmail.com        | O013          | 2025-01-12 18:00   | pending       | 6.50       | Nasi Dagang       | credit card       |
| C014          | Liew Hong          | 0122233445      | liew.hong@gmail.com            | O014          | 2025-01-12 18:30   | complete      | 6.00       | Char Kway Teow    | cash              |
| C015          | Farid Ismail       | 0133344556      | farid.ismail@hotmail.com       | O015          | 2025-01-12 19:00   | cancel        | 2.00       | Roti Canai        | online transfer   |
| C016          | Sarah Zahira       | 0144455667      | sarah.zahira@gmail.com         | O016          | 2025-01-12 19:15   | complete      | 7.00       | Nasi Kerabu       | credit card       |
| C017          | Teo Kian Seng      | 0155566778      | teo.kianseng@yahoo.com         | O017          | 2025-01-12 20:00   | complete      | 6.50       | Nasi Dagang       | online transfer   |
| C018          | Rosnah Daud        | 0166677889      | rosnah.daud@gmail.com          | O018          | 2025-01-12 20:30   | pending       | 6.50       | Nasi Dagang       | cash              |
| C019          | Hisham Mokhtar     | 0177788990      | hisham.mokhtar@outlook.com     | O019          | 2025-01-12 21:00   | complete      | 6.00       | Char Kway Teow    | credit card       |
| C020          | Lee Choon Wah      | 0188899001      | lee.choonwah@gmail.com         | O020          | 2025-01-12 21:30   | cancel        | 2.00       | Roti Canai        | online transfer   |
| C021          | Ahmad Bakri        | 0199900112      | ahmad.bakri@gmail.com          | O021          | 2025-01-12 22:00   | complete      | 4.00       | Nasi Lemak        | cash              |
| C022          | Nur Syafiqah       | 0200011223      | nur.syafiqah@gmail.com         | O022          | 2025-01-12 22:30   | pending       | 4.00       | Nasi Lemak        | credit card       |
| C023          | Roslan Hamid       | 0211122334      | roslan.hamid@yahoo.com         | O023          | 2025-01-12 23:00   | complete      | 6.00       | Char Kway Teow    | online transfer   |
| C024          | Amira Hassan       | 0222233445      | amira.hassan@gmail.com         | O024          | 2025-01-12 23:15   | complete      | 6.50       | Nasi Dagang       | cash              |
| C025          | Wan Fariz          | 0233344556      | wan.fariz@outlook.com          | O025          | 2025-01-12 23:30   | complete      | 7.00       | Nasi Kerabu       | credit card       |
| C026          | Chee Yin           | 0244455667      | chee.yin@gmail.com             | O026          | 2025-01-12 23:45   | cancel        | 6.50       | Nasi Dagang       | online transfer   |
| C027          | Fatimah Yaakob     | 0255566778      | fatimah.yaakob@yahoo.com       | O027          | 2025-01-13 00:00   | complete      | 6.00       | Char Kway Teow    | cash              |
| C028          | Singh Amarjit      | 0266677889      | singh.amarjit@gmail.com        | O028          | 2025-01-13 00:15   | pending       | 6.50       | Nasi Dagang       | credit card       |
| C029          | Noor Hafizah       | 0277788990      | noor.hafizah@gmail.com         | O029          | 2025-01-13 00:30   | complete      | 2.00       | Roti Canai        | online transfer   |
| C030          | Zamri Nasir        | 0288899001      | zamri.nasir@gmail.com          | O030          | 2025-01-13 01:00   | complete      | 7.00       | Nasi Kerabu       | credit            |

----------Display Customer Highest and Lowest Price----------

Customer with the Highest Food Price:
| C004          | Zainab Abdullah    | 0223344556      | zainab.abdullah@outlook.com    | O004          | 2025-01-12 13:15   | complete      | 7.00       | Nasi Kerabu       | credit card       |

Customer with the Lowest Food Price:
| C006          | Noor Fatimah       | 0445566778      | noor.fatimah@hotmail.com       | O006          | 2025-01-12 14:30   | pending       | 2.00       | Roti Canai        | online transfer   |

---------Display average food price among all customer---------

The average food price among all customers is: RM5.30

Enter the payment method to calculate total price for(cash | online | credit): online

Report for food names and the number of the customers

Food Report:
| Food Name         | Total Customers |
| Nasi Lemak        | 5               |
| Char Kway Teow    | 7               |
| Nasi Kerabu       | 5               |
| Roti Canai        | 5               |
| Nasi Dagang       | 8               |
```

5. CONCLUSION

The Food Ordering System is designed to simplify the management of customer orders and restaurant operations through an intuitive and efficient approach. Using Linked List data structures offers several benefits for this application.Linked Lists are well-suited for this case because they allow easy traversal and navigation of data. They enable efficient management of customer information, order tracking, and payment methods without the risk of data loss. In contrast, Queues presented challenges such as difficulties in navigation and the need to restore data after every operation, which often resulted in data loss.By implementing Linked List data structures, the system ensures smooth operations such as transferring customer data, classifying payment methods, and updating orders. This choice provides a reliable, effective, and user-friendly solution that enhances the overall functionality of the Food Ordering System.