UNIVERSITI TEKNOLOGI MARA

CSC186: OBJECT-ORIENTED PROGRAMMING

GROUP PROJECT REPORT

Group	:	JCS1102A
Lecturer's Name	:	PUAN ZURIATI BINTI ISMAIL @ KHORI

GROUP MEMBER	STUDENT ID
ALIFF WAFIUDDIN BIN MOHD AZIZUL KYUSYAIRI	2021611452
MUHAMMAD 'IRFAN BIN RAHMAT	2021877566
NUR AIN BINTI MOHD SOFIAN	2021631594

TABLE OF CONTENT

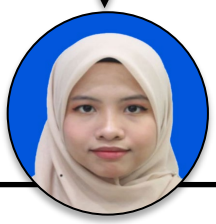
ORGANIZATIONAL STRUCTURE

ALIFF WAFIUDDIN BIN MOHD AZIZUL KYUSYAIRI

- Team project leader
- Creating UML Diagram
- Code for Customer, Local class
- Do sort numbers of ticket in ascending order (list of processing)
- Write a project report

MUHAMMAD 'IRFAN BIN RAHMAT

- Team project member
- Creating UML diagram
- Do coding for Ticket and Island class
- Do count number of tickets (list of processing)
- Do search customer's name in the system
- Write a project report



NUR AIN BINTI MOHD SOFIAN

- Team project member
- Creating UML (subclasses) and use case diagram
- Do coding for Main program class
- Do total and average, minimum and maximum of ticket (list of processing)
- Write a project report

1. **Project Title:**
   Bus Ticket Reservation

2. **Introduction:**

Ticket Nation is one of the new projects that had been assigned to provide the reservation bus ticket platform for people or traveler who want to travel all over Malaysia. The platform enables traveler to reserve and order the ticket to make sure traveler will have the best travel experience in their life. This system will provide user to choose the destination that they want to go as they can arrive at the right place at the exact time.

To make sure the system will provide what the customer wants, they will need to provide their name, age, phone number and the quantity of ticket. By fulfill all this requirement, this system will give customer the ticket to make sure they will have permission to have a bus service to make sure they can go travel to the destination that they want and arrive safely without any problem or issues. As stated in the UML Diagram on page 6 in this proposal as the information needed in the development of Ticket Nation bus ticket reservation system. The following class Customer defines all the details required for the user information while class Ticket will define all the details for the ticket info for customer. The system will be able to count number of tickets for each destination, determine the maximum and minimum number of ticket that customer have bought, calculate average ticket price, and calculate the total price of tickets. Besides that, discount also provided for customer which are count by their age which is children will enjoy 20% discount while elder will entitled 25% discount.

In the end of the system, it will display all the user information that user had enter. So, user will know that the transaction that they have made was successful. The final receipt will show the final price, customer’s detail, and the bus route.

3. Lists of Processing/Objectives:

i. Count the number of tickets for different destinations that the customers decided to go.

ii. Count the maximum and minimum number of tickets the customers bought.

iii. Calculate the average of ticket's price for all customers.

iv. Calculate the price of the tickets for different destinations with discount and without discount:

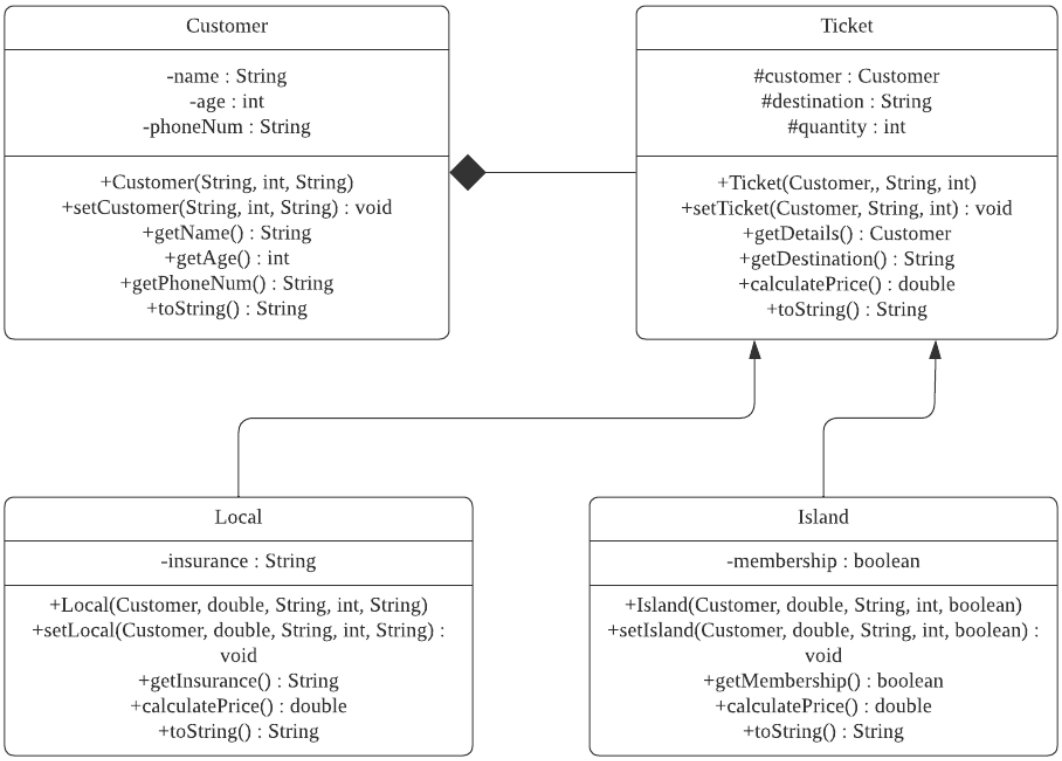
- Children: 20%
- Elder: 25%

v. Calculate total price of tickets for all customers.

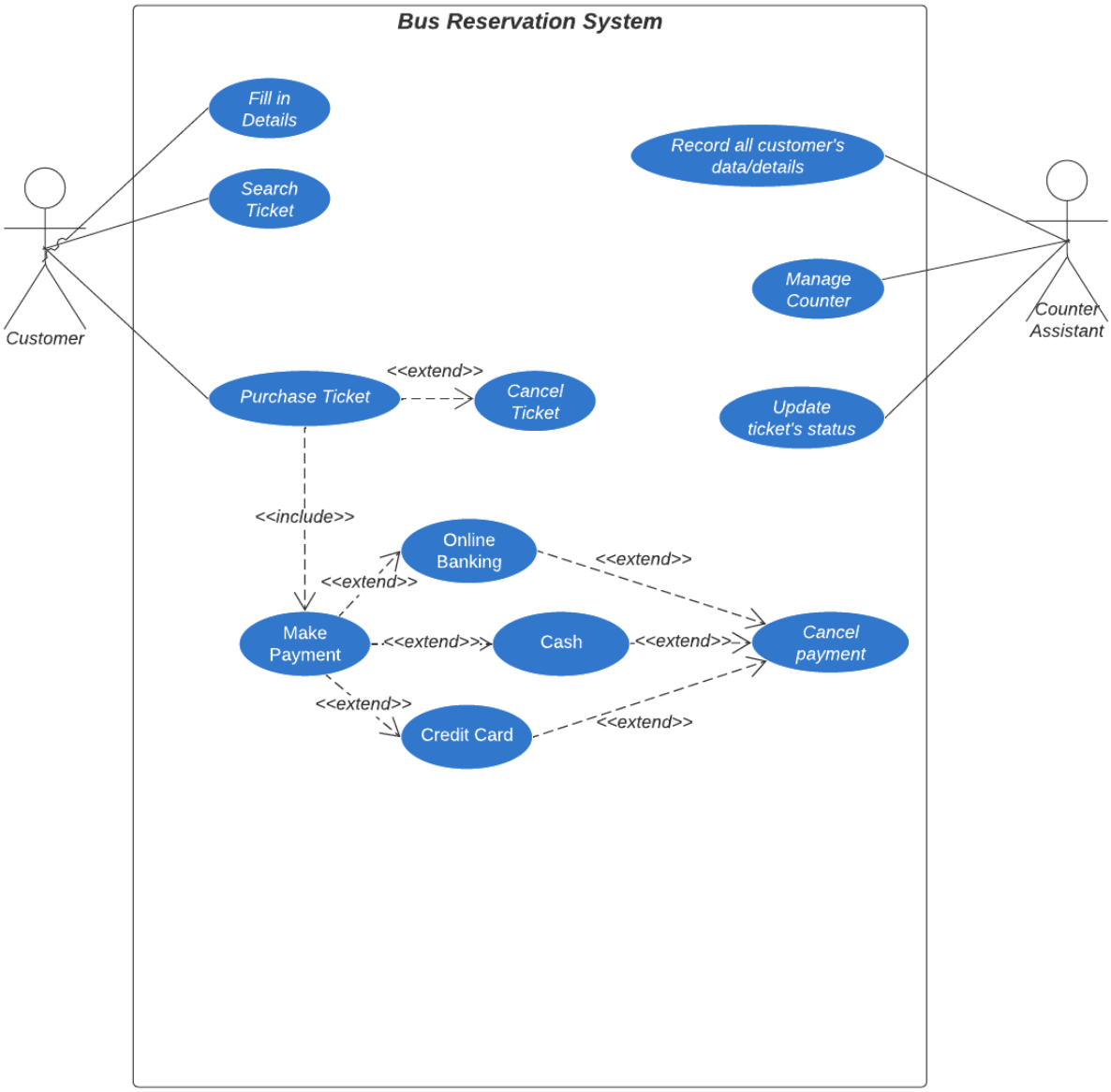
vi. Search customer's name and update customer's information.

vii. Sort the customers based on the quantity of tickets in ascending order.

UML DIAGRAM


Customer
-name : String
-age : int
-phoneNum : String
+Customer(String, int, String)
+setCustomer(String, int, String) : void
+getName() : String
+getAge() : int
+getPhoneNum() : String
+toString() : String
Ticket
#customer : Customer
#destination : String
#quantity : int
+Ticket(Customer,, String, int)
+setTicket(Customer, String, int) : void
+getDetails() : Customer
+getDestination() : String
+calculatePrice() : double
+toString() : String
Local
-insurance : String
+Local(Customer, double, String, int, String)
+setLocal(Customer, double, String, int, String) : void
+getInsurance() : String
+calculatePrice() : double
+toString() : String
Island
-membership : boolean
+Island(Customer, double, String, int, boolean)
+setIsland(Customer, double, String, int, boolean) : void
+getMembership() : boolean
+calculatePrice() : double
+toString() : String

USE CASE DIAGRAM



INPUT DATA

For input, we use txt file as there are many customer's input to be insert in our system.

We use bufferedReader and StringTokenizer for reading txt file.

reservation.txt

```
L,Lim Xin Yo,19,018-7773264,TBS,5,Yes
L,Joanne Arumugam,67,017-83741102,Stesen Kuala Terengganu,3,No
I,Siti Sarah,26,011-2536648,Pulau Kapas,3,Yes
I,Mohd Kidam,12,013-2849921,Pulau Tioman,1,No
L,Mohd Syahmi,30,012-7589654,Terminal Melaka Sentral,6,No
I,Nur Dahlia,25,011-2546397,Pulau Perhentian,2,No
I,Christine,32,013-2469785,Pulau Kapas,2,No
L,Alex Wang,16,014-7832467,Terminal Melaka Sentral,1,Yes
L,Sudirman,65,013-7321632,TBS,2,No
I,Benjamin,22,014-9872315,Pulau Tioman,5,Yes
L,Lala,19,019-2463759,Kuala Perlis bus Terminal,3,Yes
L,Muhammad Yusri Hasan,22,017-6532647,Terminal Melaka Sentral,4,No
I,Musa,14,011-974235,Pulau Kapas,6,No
I,Syafiqah,33,013-9324578,Pulau Perhentian,7,Yes
I,Mei Ling,24,015-7862145,Pulau Kapas,4,No
```

We also use a JOptionPane for user to search customer's name to make sure they are in our system.

```
THE NAME YOU SEARCHED FOR WAS FOUND!

                                    LIST OF ALL CUSTOMERS [SORTING PURPOSE]
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------

Customer Details >>
Customer Name : Mohd Kidam
Customer age : 12
Phone Number : 013-2849921
Destination : Pulau Tioman
MemberShip: false
Total Price: RM 200.00
```

CLASS DEFINITION OF INHERITANCE AND POLYMORPHISM

CLASS CUSTOMER

```
public class Customer
{
    private String name, phoneNum;
    private int age;

    public Customer(String name, String phoneNum, int age)
//parameterized constructor
    {
        this.name = name;
        this.phoneNum = phoneNum;
        this.age = age;
    }
    public void setCustomer(String a, String b, int c) //mutator
    {
        this.name = a;
        this.phoneNum = b;
        this.age = c;
    }
    //accessor
    public String getName()
    {
        return name;
    }
    public int getAge()
    {
        return age;
    }
    public String getPhoneNum()
    {
        return phoneNum;
    }
    public String toString() //toString for Customer
    {
        return "\nCustomer Name : " + name +
                "\nCustomer age : " + age +
                "\nPhone Number : " + phoneNum;
    }
}
```

CLASS TICKET

```
public abstract class Ticket
{
   protected int quantity;
   protected Customer customer;
   protected String destination;

   public Ticket(int qty, Customer cust, String desti)
   {
      quantity = qty;
      customer = cust;
      destination = desti;

   }

   public void setTicket(int a, Customer b, String c)
   {
      quantity = a;
      customer = b;
      destination = c;
   }

   public int getQuantity()
   {
      return quantity;
   }

   public Customer getDetails()
   {
      return customer;
   }
   public String getDestination()
   {
      return destination;
   }

   public abstract double calculatePrice();

   public String toString() //toString for Ticket
   {
      return "\nCustomer Details >> " +
               customer.toString() +
               "\nDestination : " + destination;
   }
}
```

CLASS LOCAL

```
import java.text.DecimalFormat;
public class Local extends Ticket
{
    private String insurance;
    DecimalFormat df = new DecimalFormat("0.00");

    public Local(int qty, Customer cust, String desti, String ins)
    {
        super(qty, cust, desti);
        insurance = ins;
    }

    public void setLocal(int qty, Customer cust, String desti, String
newIns)
    {
        super.setTicket(qty, cust, desti);
        insurance = newIns;
    }

    public String getInsurance()
    {
        return insurance;
    }

    public double calculatePrice()
    {
        double charge = 0.0, total = 0.0, insuredCharge = 0;

        //base ticket price
        if(super.getDestination().equals("TBS"))
          charge = 20 * super.getQuantity();
        else if(super.getDestination().equals("Terminal Melaka
Sentral"))
          charge = 15 * super.getQuantity();
        else if(super.getDestination().equals("Kuala Perlis bus
Terminal"))
          charge = 70 * super.getQuantity();
        else //Kuala Terengganu Bus Station
          charge = 30 * super.getQuantity();

        //ticket with insurance
        if(getInsurance().equalsIgnoreCase("Yes"))
          insuredCharge = charge + (1.5 * super.getQuantity());
```

```
        else
          insuredCharge = charge;


        //ticket with discount + insurance
        if(customer.getAge() <= 7)
          total = insuredCharge - (insuredCharge * 0.20);
        else if(customer.getAge() >= 60)
          total = insuredCharge - (insuredCharge * 0.25);
        else
          total = insuredCharge;

        return total;
    }

    public String toString() //toString for Local
    {
        return super.toString() +
              "\nInsurance:" + getInsurance() +
              "\nTotal Price: RM " + df.format(calculatePrice());
    }

}
```

CLASS ISLAND

```
import java.text.DecimalFormat;
public class Island extends Ticket
{
    private boolean membership;
    DecimalFormat df = new DecimalFormat("0.00");

    public Island(int a,Customer b, String c, boolean d)
//parameterized constructor
    {
        super(a, b, c);
        membership = d;
    }

    public void setIsland(int a,Customer b, String c, boolean d)
//mutator
    {
        super.setTicket(a, b, c);
        membership = d;
    }

    //accessor
    public boolean getMembership()
    {
        return membership;
    }

    public double calculatePrice() //calculation
    {
        double charge = 0.0, member = 0.0, total = 0.0;

        //base ticket price
        if(super.getDestination().equals("Pulau Kapas"))
        charge = 100 * super.getQuantity();
        else if(super.getDestination().equals("Pulau Perhentian"))
        charge = 150 * super.getQuantity();
        else if(super.getDestination().equals("Pulau Tioman"))
        charge = 200 * super.getQuantity();
        else //PULAU LANGKAWI
        charge = 300 * super.getQuantity();

        //ticket with discount + insurance
        if(customer.getAge() <= 7)
          total = charge - (charge * 0.20);
```

```
        else if(customer.getAge() >= 60)
          total = charge - (charge * 0.25);
        else
          total = charge;

        //ticket with membership discount
        if(membership)
        member = total - (total * 0.15);
        else
        member = total;

        return member;
    }

    public String boolString()
    {
        if(membership)
        return "Yes";
        else
        return "No";
    }

    public String toString() //toString for Island - Printer Method
    {
        return super.toString() +
            "\nMemberShip: " + membership +
            "\nTotal Price: RM " + df.format(calculatePrice());
    }
}
```

CLASS BUS RESERVATION (MAIN PROGRAM)

```
import java.io.*;
import java.util.*;
import javax.swing.JOptionPane;
import java.text.DecimalFormat;
public class BusReservation
{
    public static void main(String[] args) throws IOException
    {
        try
        {
            DecimalFormat df = new DecimalFormat("0.00");
            DecimalFormat dt = new DecimalFormat("0.00");

            FileReader fr = new FileReader("reservation.txt");
            BufferedReader br = new BufferedReader(fr);

            String line;
            Ticket [] t = new Ticket[100];
            int [] array = new int[100];
            int i = 0;

            while((line = br.readLine())!=null)
            {
                StringTokenizer st = new StringTokenizer(line,",");
                String choice = st.nextToken();
                String cName = st.nextToken();
                int cAge = Integer.parseInt(st.nextToken());
                String cPhone = st.nextToken();

                Customer c = new Customer(cName, cPhone, cAge);
                if(choice.equalsIgnoreCase("L"))
                {
                    String tDestination = st.nextToken();
                    int tQty = Integer.parseInt(st.nextToken());
                    String tInsurance = st.nextToken();
                    array[i] = tQty;
                    t[i] = new Local(tQty, c, tDestination,
tInsurance);
                }
                else
                {
                    String tDestination = st.nextToken();
                    int tQty = Integer.parseInt(st.nextToken());
```

```
                    String tMem = st.nextToken();
                    boolean tMember = false;
                    if(tMem.equalsIgnoreCase("Yes"))
                    tMember = true;
                    array[i] = tQty;
                    t[i] = new Island(tQty, c, tDestination, tMember);
                }
                i++;
            }

            PrintWriter pw = new PrintWriter("bus.txt");
            pw.println("                                                LIST
OF ALL CUSTOMERS [MANAGEMENT PURPOSE]
");
            pw.println("-----------------------------------------------
-----------------------------------------------------------------
----------------------|---------------|");
            pw.println(String.format("%-25s%-12s%-20s%-30s%-20s%-20s%-
20s","NAME","AGE","PHONE
NO","DESTINATION","QUANTITY","INSURANCE","MEMBERSHIP"));
            pw.println("-----------------------------------------------
-----------------------------------------------------------------
----------------------|---------------|");
            for(int a=0;a<i;a++)
            {
                if(t[a] instanceof Local)
                {
                    Local l = (Local)t[a];
                    pw.println(String.format("%-25s%-12s%-20s%-30s%-
20s%-20s%-
20s",l.customer.getName(),l.customer.getAge(),l.customer.getPhoneNum()
,l.getDestination(),l.getQuantity(),l.getInsurance(),"NA"));
                }
                else if(t[a] instanceof Island)
                {
                    Island is = (Island)t[a];
                    pw.println(String.format("%-25s%-12s%-20s%-30s%-
20s%-20s%-
20s",is.customer.getName(),is.customer.getAge(),is.customer.getPhoneNu
m(),is.getDestination(),is.getQuantity(),"NA",is.boolString()));
                }
            }

            //total & average of ticket price (all customer)
```

```
            double total = 0.0, totalLo = 0.0, totalIs = 0.0, average
= 0.0;
            for(int a=0;a<i;a++)
            {
                if(t[a] instanceof Local)//calculate total
                {
                    Local l = (Local)t[a];
                    totalLo += l.calculatePrice();
                }
                else if(t[a] instanceof Island)
                {
                    Island is = (Island)t[a];
                    totalIs += is.calculatePrice();
                }
                total = totalLo + totalIs;
                average = total/i;
            }

            //max quantity of ticket
            int max = -9999;
            for(int a=0;a<i;a++)
            {
                if(t[a] instanceof Local)
                {
                    Local l = (Local)t[a];
                    if(l.getQuantity() > max)
                    max = l.getQuantity();
                }
                else if(t[a] instanceof Island)
                {
                    Island is = (Island)t[a];
                    if(is.getQuantity() > max)
                    max = is.getQuantity();
                }
                if(t[a] instanceof Local && t[a] instanceof Island)
                {
                    Local l = (Local)t[a];
                    Island is = (Island)t[a];
                    if(l.getQuantity()>is.getQuantity())
                    max = l.getQuantity();
                    else
                    max = is.getQuantity();
                }
            }
```

```
            //min quantity of ticket
            int min = 9999;
            for(int a=0;a<i;a++)
            {
                if(t[a] instanceof Local)
                {
                    Local l = (Local)t[a];
                    if(l.getQuantity() < min)
                    min = l.getQuantity();
                }
                else if(t[a] instanceof Island)
                {
                    Island is = (Island)t[a];
                    if(is.getQuantity() < min)
                    min = is.getQuantity();
                }
                if(t[a] instanceof Local && t[a] instanceof Island)
                {
                    Local l = (Local)t[a];
                    Island is = (Island)t[a];
                    if(l.getQuantity()<is.getQuantity())
                    min = l.getQuantity();
                    else
                    min = is.getQuantity();
                }
            }
            pw.println("-----------------------------------------------
-------------------------------------------------------------------
----------------------|--------------|");
            pw.println(String.format("%-25s%-12s%-20s%-30s%-20s%-36s%-
20s","TOTAL(RM)","","","","","",df.format(total)));
            pw.println(String.format("%-25s%-12s%-20s%-30s%-20s%-36s%-
20s","Average(RM)","","","","","",dt.format(average)));
            pw.println("-----------------------------------------------
-------------------------------------------------------------------
----------------------|--------------|");
            pw.println(String.format("%-25s%-12s%-20s%-30s%-20s%-39s%-
20s","Maximum number of Ticket","","","","","",max));
            pw.println(String.format("%-25s%-12s%-20s%-30s%-20s%-39s%-
20s","Minimum number of Ticket","","","","","",min));
            pw.println("-----------------------------------------------
-------------------------------------------------------------------
----------------------|--------------|");

            //search customer
```

```
            String search = JOptionPane.showInputDialog("Search
customer's name : ");
            boolean found = false;
            int indexFound = 0;
            for(int a=0;a<i;a++)
            {
                if(t[a] instanceof Local)
                {
                    Local l = (Local)t[a];
                    if(l.customer.getName().equals(search))
                    {
                        found = true;
                        indexFound = a;
                    }
                }
                else if(t[a] instanceof Island)
                {
                    Island is = (Island)t[a];
                    if(is.customer.getName().equals(search))
                    {
                        found = true;
                        indexFound = a;
                    }
                }
            }
            if(found)//updating
            {
                t[indexFound].customer.setCustomer("Aria Jenne","011-
2345567",4);
                System.out.println("THE NAME YOU SEARCHED FOR WAS
FOUND AND RECORD UPDATED!");
            }
            else
            System.out.println("THE NAME YOU SEARCHED FOR IS NOT EXIST
AND RECORD NOT FOUND!");

            //sort customers based on quantity of ticket
            int swap = 0, n = i;
            for(int a=0;a<(n-1);a++)
            {
                for(int x=0;x<(n-a-1);x++) //ascending order
                {
                    if(array[x]>array[x+1])
                    {
                       swap = array[x];
```

```
                        array[x] = array[x+1];
                        array[x+1] = swap;
                    }
                }
            }

            //count for ticket quantity
            int count = 0;
            for(int x = 0; x < t.length; x++)
            {
                if(array[x] > 0)
                {
                    count += t[x].getQuantity();
                }
            }



            System.out.println("\n
LIST OF ALL CUSTOMERS [AFTER SORTING]
");
            System.out.println("-------------------------------------
-------------------------------------------------------------------
-----------");
            for(int c=0;c<n;c++)
            {
                if(c==0)
                {
                    for(int j=0;j<n;j++)
                    {
                        if(array[c] == t[j].getQuantity())
                        System.out.println(t[j].toString());
                    }
                }
                else if(array[c] != array[c-1])
                {
                    for(int j=0;j<n;j++)
                    {
                        if(array[c] == t[j].getQuantity())
                        System.out.println(t[j].toString());
                    }
                }
            }
```

```
            System.out.println("\nThe number of tickets bought by all
customers are : " + count + " in total.");

            System.out.println("-----------------------------------
-------------------------------------------------------------------
-----------");
            pw.close();
            br.close();
        }
            catch(FileNotFoundException e){System.out.println("File
cannot be found!");}


         catch(Exception e){System.out.println("Cannot read the
Data!");}
    }
}
```

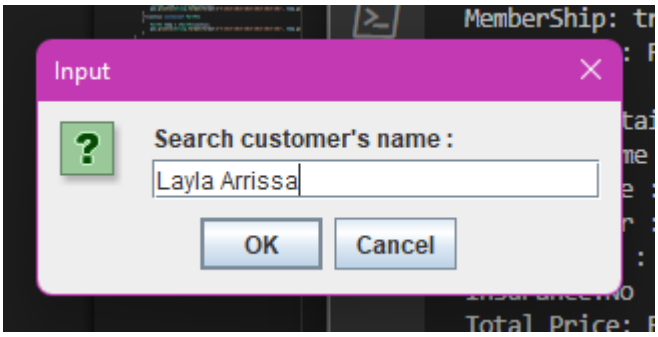
DISPLAY INFORMATION AND SAMPLE INTERFACES

Diagram below shows an output sample that using FileWriter and BufferedWriter method.

bus.txt

```
                                        LIST OF ALL CUSTOMERS [MANAGEMENT PURPOSE]
-------------------------------------------------------------------------------------------------------------------------------------------|--------------|
NAME                    AGE         PHONE NO          DESTINATION                QUANTITY          INSURANCE          MEMBERSHIP
-------------------------------------------------------------------------------------------------------------------------------------------|--------------|
Lim Xin Yo              19          018-7773264       TBS                        5                 Yes                NA
Joanne Arumugam         67          017-83741102      Stesen Kuala Terengganu    3                 No                 NA
Siti Sarah              26          011-2536648       Pulau Kapas                3                 NA                 Yes
Mohd Kidam              12          013-2849921       Pulau Tioman               1                 NA                 No
Mohd Syahmi             30          012-7589654       Terminal Melaka Sentral    6                 No                 NA
Nur Dahlia              25          011-2546397       Pulau Perhentian           2                 NA                 No
Christine               32          013-2469785       Pulau Kapas                2                 NA                 No
Alex Wang               16          014-7832467       Terminal Melaka Sentral    1                 Yes                NA
Sudirman                65          013-7321632       TBS                        2                 No                 NA
Benjamin                22          014-9872315       Pulau Tioman               5                 NA                 Yes
Lala                    19          019-2463759       Kuala Perlis bus Terminal  3                 Yes                NA
Muhammad Yusri Hasan    22          017-6532647       Terminal Melaka Sentral    4                 No                 NA
Musa                    14          011-974235        Pulau Kapas                6                 NA                 No
Syafiqah                33          013-9324578       Pulau Perhentian           7                 NA                 Yes
Mei Ling                24          015-7862145       Pulau Kapas                4                 NA                 No
-------------------------------------------------------------------------------------------------------------------------------------------|--------------|
TOTAL(RM)                                                                                                                                  2638.50
Average(RM)                                                                                                                                175.90
-------------------------------------------------------------------------------------------------------------------------------------------|--------------|
Maximum number of Ticket                                                                                                                   7
Minimum number of Ticket                                                                                                                   1
-------------------------------------------------------------------------------------------------------------------------------------------|--------------|
```

Diagram below shows a pop out that using JOptionPane for management to fill in the customer's name to search customer's name if their name is in the system or not.



This is the output if customer's name was found and it will updated the customer's name and phone number.

```
THE NAME YOU SEARCHED FOR WAS FOUND AND RECORD UPDATED!
```

```
Customer Details >>
Customer Name : Aria Jenne
Customer age : 4
Phone Number : 011-2345567
Destination : Kuala Perlis bus Terminal
Insurance:Yes
Total Price: RM 171.60
```

While this is the output if customer's name does not exist in the system

```
THE NAME YOU SEARCHED FOR IS NOT EXIST AND RECORD NOT FOUND!
```

This is the output sample that had been sorting in ascending order with a total count of ticket reservation

```
                                    LIST OF ALL CUSTOMERS [AFTER SORTING]
-------------------------------------------------------------------------------------------------------------------------

Customer Details >>
Customer Name : Mohd Kidam
Customer age : 12
Phone Number : 013-2849921
Destination : Pulau Tioman
MemberShip: false
Total Price: RM 200.00

Customer Details >>
Customer Name : Alex Wang Jay
Customer age : 16
Phone Number : 014-7832467
Destination : Terminal Melaka Sentral
Insurance:Yes
Total Price: RM 16.50

Customer Details >>
Customer Name : Nur Dahlia
Customer age : 25
Phone Number : 011-2546397
Destination : Pulau Perhentian
MemberShip: false
Total Price: RM 300.00

Customer Details >>
Customer Name : Christine Lee
Customer age : 32
Phone Number : 013-2469785
Destination : Pulau Kapas
MemberShip: false
Total Price: RM 200.00

Customer Details >>
Customer Name : Sudirman Awang
Customer age : 65
Phone Number : 013-7321632
Destination : TBS
Insurance:No
Total Price: RM 30.00

Customer Details >>
Customer Name : Joanne Arumugam
Customer age : 67
Phone Number : 017-8374110
Destination : Stesen Kuala Terengganu
Insurance:No
Total Price: RM 67.50
```

```
Customer Details >>
Customer Name : Siti Sarah
Customer age : 26
Phone Number : 011-2536648
Destination : Pulau Kapas
MemberShip: true
Total Price: RM 255.00

Customer Details >>
Customer Name : Aria Jenne
Customer age : 4
Phone Number : 011-2345567
Destination : Kuala Perlis bus Terminal
Insurance:Yes
Total Price: RM 171.60

Customer Details >>
Customer Name : Muhammad Yusri
Customer age : 22
Phone Number : 017-6532647
Destination : Terminal Melaka Sentral
Insurance:No
Total Price: RM 60.00

Customer Details >>
Customer Name : Ye Mei Ling
Customer age : 24
Phone Number : 015-7862145
Destination : Pulau Kapas
MemberShip: false
Total Price: RM 400.00

Customer Details >>
Customer Name : Lim Xin Yo
Customer age : 19
Phone Number : 018-7773264
Destination : TBS
Insurance:Yes
Total Price: RM 107.50

Customer Details >>
Customer Name : Benjamin Mason
Customer age : 22
Phone Number : 014-9872315
Destination : Pulau Tioman
MemberShip: true
Total Price: RM 850.00

Customer Details >>
Customer Name : Mohd Syahmi
Customer age : 30
Phone Number : 012-7589654
Destination : Terminal Melaka Sentral
Insurance:No
Total Price: RM 90.00
```

```
Customer Details >>
Customer Name : Musa Muhammad
Customer age : 14
Phone Number : 011-9742359
Destination : Pulau Kapas
MemberShip: false
Total Price: RM 600.00

Customer Details >>
Customer Name : Syafiqah Razali
Customer age : 33
Phone Number : 013-9324578
Destination : Pulau Perhentian
MemberShip: true
Total Price: RM 892.50

The number of tickets bought by all customers are : 54 in total.
```

In the end of program, it is show that the total ticket bought by all customers which are 54 tickets. All the sequence in the diagram is sorted by the quantity of tickets which is sort in ascending order

REFERENCES

1) Tech Raj, Jan 3, 2017, Sorting Arrays in Java - Tutorial | Selection Sort and Bubble Sort, Youtube. https://youtu.be/ZQ8AaxHnAb0

2) Lucid Software, Feb 8, 2018, UML Use Case Diagram Tutorial, Youtube. https://youtu.be/zid-MVo7M-E

3) UML Class Diagram Tutorial – Visual Paradigm. https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/