Assignment - 01

Date of Sub :- 29/August/2020

* Title: Use of generics and collections

Problem Statement: Design a system with the help of advance data structure in Java and enhance the system using collections and generics build a Bus Reservation System.

* Objective: To understand and implement java generics and collections

* Outcomes: We will understand the basics of java generics and collections

* S/W AND H/W Requirements:
Notepad ++, JDK 8.0, Windows 10, Intel i5 processor, 8 GB RAM

* THEORY:

i) Collections in Java:

- collections in Java is framework that provides on architecture to store and manipulate group of objects
- All the operations that you perform on data such as searching, sorting, insertion, deletion, manipulation, etc can be performed by Java Collect

- Java ArrayList class:
It uses a dynamic array for storing the element. inherits abstract list class implements list interface

- It can contain duplicate entries.

Teacher's Sign.: ___

- It allows random access
- Slow manipulation

Methods:

1) void add(int index, object element) - insert at specific position
2) void clear () - removes all elements
3) boolean addAll (Collection c) - append at end
4) Object clone ()

eg. ArrayList < Integer> no. = new ArrayList<integer>(

Application: Arraylist is widely used datastructure since it is dynamic, resizable and randomly accessible.

- Java LinkedList class:
  - Use doubly linkedlist to store elements

Methods:

1) void add(int index, object elements)
2) void addFirst(object o)
3) void addLast (object o)
4) int size ()
5) boolean contains (object o)

Application: can be used to implement stack, queue.

- Java Hashset class:
  - Used to create a collection that uses hash tabl for storage
    - contains unique element

- storage element by hashing

Methods:

1> void clear ( )
2> boolean contains (Object o)
3> boolean add (object o)
4> boolean isEmpty ( )

example

```
Hashset < String> name = new Hashset <String> ( );
```

Application: Hashset is exclusively used since lookup time is less

Java HashMap Class:

- Stores value in key - value pairs.
- Access them by an index of another type

Methods

1> void clear ( )
2> boolean isEmpty ( )
3> void put (object key, object value)
4> void remove (object key)
5> boolean remove (object key, object value)

example:

```
HashMap <Integer, string> items = new HashMap <Integer
                                          string> ( );
```

Application: Implementation provides constant time performation for basic operation (put and get)

ii) Generic class: A class that can refer to any type is generic class.

example:
```
class MyGen <T>
  { T obj;
   void add (T obj){
        this.obj = obj;}
     T get()
      { return obj;}
  }
```

T type indicates that it can refer to any type

Features of Java:

1) Object Oriented: In java, everything is an object. Java can be easily extended since it is based on the object made.

2) Platform Independent:
Unlike many other programming language including C and C++ when Java is completed, it is not compiled into platform specific machine

3) Simple: Java is designed to be easy to learn.

4) Secure: With java's secure feature, it enables to develope virus-free, tamperless systems.

5) Portable: Being architectural-neutral and having no implementation dependent aspects of the specifications makes java portable.

Collections (interface)

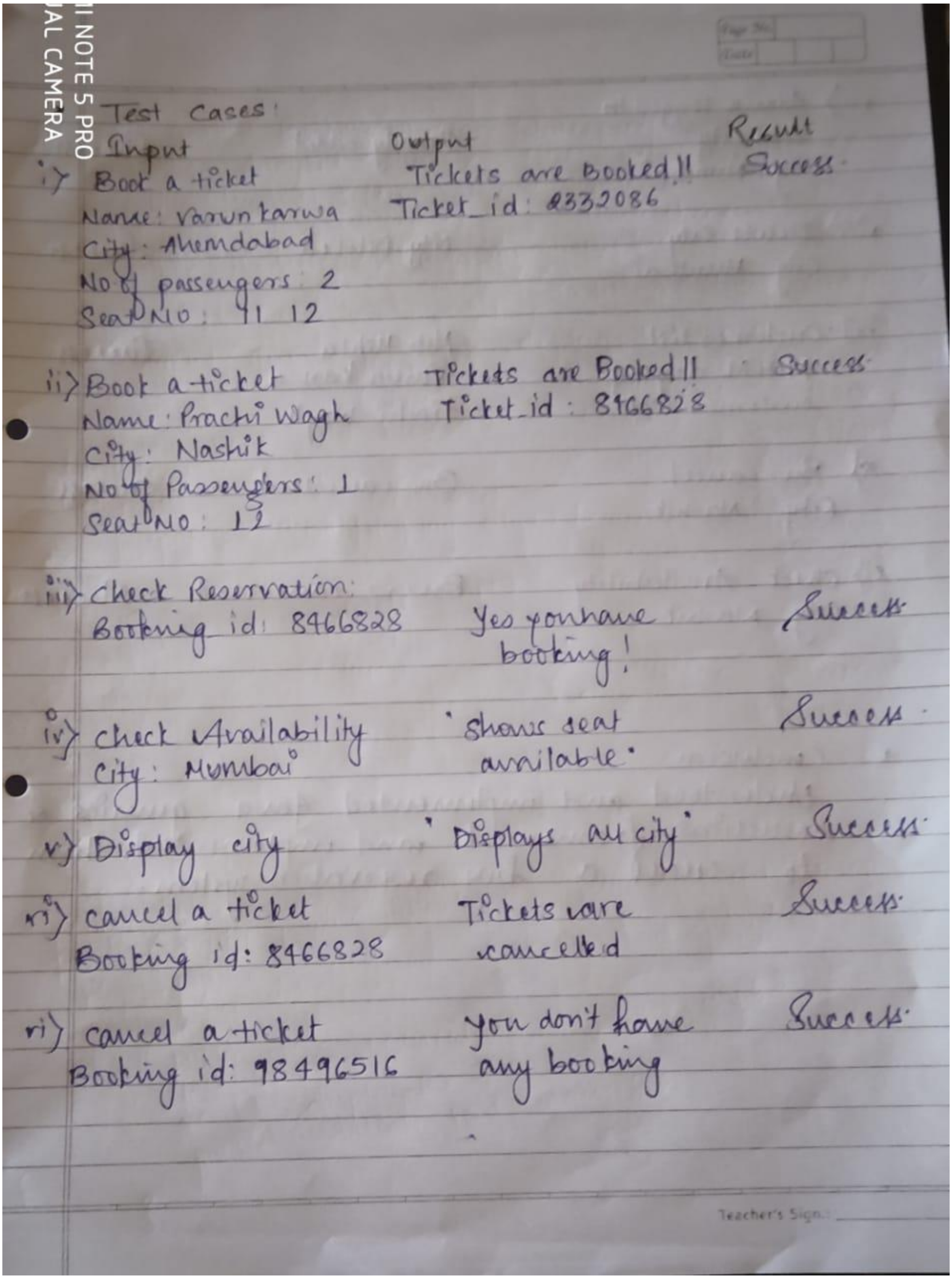
List (interface) implemented	Set (interface) implemented	Queue (interface) implemented	Map (interface) implemented
ArrayList (class)	Hashset (class)	Priority Queue (class)	HashMap (class)
LinkedList (class)	LinkedHashset (class)	Dequeue (interface)	Linked HashMap (class)
Vectors	Treeset (class)	ArrayDequeue (class)	TreeMap (class)

* class Structure:

i) class customer

```
{ public string name, destination_city, contact number
  public int number_of_passengers, booking_id;
 ArrayList <string> passengers = new ArrayList <string>
 ArrayList <Integer> seat_no = new ArrayList <Integer
   public void read_customer()
   {
   }
   public void print_details()
      {
      }
}
```

```
> public class Booking {
    public ArrayList <String>  cities = new ArrayList <String>();
    public HashSet < customer> customer_list = new
                                  HashSet < customer> ();
    public HashMap< String, ArrayList> buses = new
                                  HashMap <String , ArrayList>
                                                      ();
    public Queue <String> display_list = new
                                  PriorityQueue <String>
                                                      ();
    public void add_cities ()
      { }
    public void add_buses ()
      { }
    public void_city ()
      { }
     public void delete_city()
       { }
    public void display_city()
      { }
    public void add_bus ()
      { }
    public void cancel_bus()
      { }
    public void book ( customer c )
      { }
     public customer check (int booking_id)
       { }
    public void cancel (customer c) { }
    public void available (String city){ }
    void display () { }
    public static void main(){ }
}
```



Page No.
Date

Test Cases:

Input	Output	Result
i) Book a ticket Name: Varun karwa City: Ahemdabad No of passengers: 2 Seat No: 11 12	Tickets are Booked!! Ticket_id: 8332086	Success.
ii) Book a ticket Name: Prachi Wagh City: Nashik No of Passengers: 1 Seat No: 12	Tickets are Booked!! Ticket_id: 8466828	Success.
iii) Check Reservation: Booking id: 8466828	Yes you have booking!	Success.
iv) check Availability City: Mumbai	'shows seat available'	Success.
v) Display city	'Displays all city'	Success.
vi) cancel a ticket Booking id: 8466828	Tickets are cancelled	Success.
vi) cancel a ticket Booking id: 98496516	You don't have any booking	Success.

Teacher's Sign.:

vii) Admin Logged In Add City City : Mumbai	Already added	Success
viii) Delete City City : Mumbai	City Deleted	Success
ix] - Customes list: Miti M. display customer list	Miti Mehta Varun Karwa Prachi Wagh	Success.
x) Bus cancel City : Nashik	Bus cancelled	Success.
xi) Check Availability City : Nashik	Bus is cancelled	Success

* Conclusion:

Understood and implemented Java generics & collections framework and implemented them to create a bus reservation system.

Code:

```
import java.util.*;
class customer
{
	public String name,destination_city,contact_number;
	public int number_of_passengers,booking_id;
	ArrayList<String> passengers=new ArrayList<String>();
	ArrayList<Integer> seat_no = new ArrayList<Integer>();
	public void read_customer()
	{
		Scanner sc=new Scanner(System.in);
		System.out.println("Enter Name");
		name=sc.nextLine();
		System.out.println("Enter Destination City");
		destination_city=sc.next();
		System.out.println("Enter number of passengers");
		number_of_passengers=Integer.parseInt(sc.next());
		System.out.println("Enter Contact number");
		contact_number=sc.next();
		for(int i=0;i<number_of_passengers;i++)
		{
			System.out.println("Enter Passenger's Name");
			passengers.add(sc.next());
		}
	}
	public void print_details()
	{
		System.out.println("=====Customer Deatils=====");
		System.out.println("Booking Id:"+booking_id);
		System.out.println("Name of the Customer:" + name);
		System.out.println("Contact Number:" + contact_number);
		System.out.println("Destination City:" + destination_city);
		System.out.println("Number of Passengers Travelling:" + 
number_of_passengers);
		System.out.println("List of Passenger's:");
		for(Integer i=0;i<number_of_passengers;i++)
		{
			System.out.println(passengers.get(i));
			System.out.println("Seat Number:"+seat_no.get(i));
		}
	}
}
public class Booking
{
	public ArrayList<String> Cities=new ArrayList<String>();
	public HashSet<customer> customer_list = new HashSet<customer>();
	public HashMap<String, ArrayList> buses = new HashMap<String, 
ArrayList>();
	public Queue <String> display_list= new PriorityQueue<String>();
	public void add_cities()
	{
		Cities.add("Mumbai");
		Cities.add("Ahemdabad");
```

```
        Cities.add("Kolhapur");
        Cities.add("Nashik");
        Cities.add("Indore");
        Cities.add("Nagpur");
        Cities.add("Jalgaon");
}
public void add_buses()
{
        for(String c:Cities)
        {
                ArrayList<Integer> bus = new ArrayList<Integer>();
                for(Integer i=0;i<40;i++)
                        bus.add(0);
                buses.put(c,bus);
        }
}
public void add_city(String city)
{
        if(!Cities.contains(city))
        {
                Cities.add(city);
                add_bus(city);
                System.out.println("City added");
        }
        else
                System.out.println("City already added");
```

```
}
public void delete_city(String city)
{
        if(Cities.contains(city))
        {
                Cities.remove(city);
                buses.remove(city);
                System.out.println("City deleted");
        }
        else
                System.out.println("City not present!");
}
public void display_city()
{
        System.out.println("We have service to the following cities:");
        for(String str:Cities)
                System.out.println("\t"+str);
}
public void add_bus(String city)
{
        if(Cities.contains(city))
        {
                ArrayList<Integer> bus=new ArrayList<Integer>();
                for(Integer i=0;i<40;i++)
                        bus.add(0);
                buses.put(city,bus);
        System.out.println("Bus added");
    }
    else
        System.out.println("City not Available");
}
public void cancel_bus(String city)
{
    if(Cities.contains(city))
    {
        buses.remove(city);
    }
    else
        System.out.println("City not present");
}
public void book(customer c)
{
    Scanner sc=new Scanner(System.in);
    int seat;
    Random rand=new Random();
    c.read_customer();
    if(!Cities.contains(c.destination_city))
    {
        System.out.println("Sorry!! This city is not available.");
        return;
    }
    if(!buses.containsKey(c.destination_city))
```

```
{
    System.out.println("Bus is cancelled for this city");
    return;
}
ArrayList <Integer> bus=buses.get(c.destination_city);
available(c.destination_city);
if(!bus.contains(0))
{
    System.out.println("This Bus is Full!!");
    return;
}
System.out.println("Enter Seat Numbers:");
for(int i=0;i<c.number_of_passengers;i++)
{
    seat=Integer.parseInt(sc.next());
    if(bus.get(seat-1)==0)
    {
        bus.set(seat-1,1);
        c.seat_no.add(seat);
    }
    else
    {
        System.out.println("Seat is already booked!!");
        System.out.println("Enter another seat:");
        i--;
    }
```

```
        }
        String city=c.destination_city;
        buses.replace(city,bus);
        customer_list.add(c);
        display_list.add(c.name);
        c.booking_id=rand.nextInt(9999999);
        System.out.println("Tickets are Booked!!");
        c.print_details();
}
public customer check(int booking_id)
{
        for(customer c:customer_list)
                if(c.booking_id==booking_id)
                        return c;
        return null;
}
public void cancel(customer c)
{
        ArrayList<Integer> bus=buses.get(c.destination_city);
        for(int i=0;i<c.number_of_passengers;i++)
        {
                bus.set(c.seat_no.get(i)-1,0);
        }
        buses.replace(c.destination_city,bus);
        customer_list.remove(c);
        System.out.println("Ticket is cancelled.");
```

```
	}
	public void available(String city)
	{
		if(Cities.contains(city))
		{
			if(buses.containsKey(city))
			{
				ArrayList<Integer> bus=buses.get(city);
				System.out.println("Seats Available(0 means
available):");
				int j=0;
				while(j<bus.size())
				{
					System.out.print("\t"+bus.get(j++)+"
"+bus.get(j++));
					System.out.println("\t"+bus.get(j++)+"
"+bus.get(j++));
				}
			}
			else
				System.out.println("Bus is cancelled");
		}
		else
			System.out.println("City not available");
	}
	void display()
	{
		Iterator it=display_list.iterator();
		while(it.hasNext())
		{
			System.out.println(it.next());
		}
	}
	public static void main(String args[])
	{
		Scanner sc=new Scanner(System.in);
		System.out.println("==========BUS RESERVATION
SYSTEM==========");
		int choice1,choice2,choice3,booking_id;
		String city;
		Booking b= new Booking();
		b.add_cities();
		b.add_buses();
		do
		{
			System.out.println("Login as:");
			System.out.println("1.Admin");
			System.out.println("2.Customer");
			System.out.println("3.Exit");
			choice1=Integer.parseInt(sc.next());
			switch(choice1)
			{
				case 1:System.out.println("Logged in as Admin");
					do
				{
					System.out.println("1.Add City");
					System.out.println("2.Delete City");
					System.out.println("3.Add Bus");
					System.out.println("4.Cancel Bus");
					System.out.println("5.Customer List");
					System.out.println("6.Log Out");
					choice2=Integer.parseInt(sc.next());
					switch(choice2)
					{
						case 1:System.out.println("Enter
City to be added:");
							city=sc.next();
							b.add_city(city);
							break;
						case 2:System.out.println("Enter
City name to be deleted");
							city=sc.next();
							b.delete_city(city);
							break;
						case 3:System.out.println("Enter
Bus Destination City:");
							city=sc.next();
							b.add_bus(city);
							break;
						case 4:
```

```
                                System.out.println("Enter
bus destination city which you want to cancel");
                                city=sc.next();
                                b.cancel_bus(city);
                                break;
                            case
5:System.out.println("Customer List in Booking Order:");
                                b.display();
                                break;
                            case 6:System.out.println("Logging
Out...");
                        }
                    }while(choice2!=6);
                    break;
                case 2:System.out.println("Logged in as Customer");
                    do
                    {
                        System.out.println("1.Book a Ticket.");
                        System.out.println("2.Cancel a Ticket.");
                        System.out.println("3.Check
Reservation.");
                        System.out.println("4.Check
Availability.");
                        System.out.println("5.City List");
                        System.out.println("6.Log Out");
                        choice3=Integer.parseInt(sc.next());
                        switch(choice3)
                        {
```

```
							case 1:
								customer c=new
customer();
								b.book(c);
								break;
							case 2:
								System.out.println("Enter
your Booking Id:");

	booking_id=Integer.parseInt(sc.next());
								customer
can=b.check(booking_id);
								if(can!=null)
								{
									b.cancel(can);
								}
								else
								{

	System.out.println("You don't have any booking in bus or your booking
id is incorrect.");
								}
								break;
							case 3:
								System.out.println("Enter
your Booking Id:");

	booking_id=Integer.parseInt(sc.next());
```

```
                                        customer
ch=b.check(booking_id);
                                        if(ch!=null)
                                        {

    System.out.println("Yes! You have booking.");
                                            ch.print_details();
                                        }
                                        else
                                        {

    System.out.println("No!! You don't have booking or your booking id is
incorrect.");
                                        }
                                        break;
                                    case 4:
                                        System.out.println("Enter
destination city:");
                                        city=sc.next();
                                        b.available(city);
                                        break;
                                    case 5:
                                        b.display_city();
                                        break;
                                    case 6:System.out.println("Logging
Out...");
                                }
                            }while(choice3!=6);
```

```
                                    break;
                               case 3:System.out.println("Exiting...");
                         }
                    }while(choice1!=3);
               }
          }
```

OUTPUT:

```
Microsoft Windows [Version 10.0.18362.778]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\varun karwa>F:

F:\>cd "Bus Reservation System"

F:\Bus Reservation System>java Booking
==========BUS RESERVATION SYSTEM==========
Login as:
1.Admin
2.Customer
3.Exit
2
Logged in as Customer
1.Book a Ticket.
2.Cancel a Ticket.
3.Check Reservation.
4.Check Availability.
5.City List
6.Log Out
1
Enter Name
Varun Karwa
```

```
Enter Destination City
Ahemdabad
Enter number of passengers
2
Enter Contact number
7617209448
Enter Passenger's Name
Varun
Enter Passenger's Name
Miti
Seats Available(0 means available):
    0 0   0 0
    0 0   0 0
    0 0   0 0
    0 0   0 0
    0 0   0 0
    0 0   0 0
    0 0   0 0
    0 0   0 0
    0 0   0 0
    0 0   0 0
Enter Seat Numbers:
11 12
Tickets are Booked!!
=====Customer Deatils=====
Booking Id:3478805
```

```
Name of the Customer:Varun Karwa

Contact Number:7617209448

Destination City:Ahemdabad

Number of Passengers Travelling:2

List of Passenger's:

Varun

Seat Number:11

Miti

Seat Number:12

1.Book a Ticket.

2.Cancel a Ticket.

3.Check Reservation.

4.Check Availability.

5.City List

6.Log Out

1

Enter Name

Siddhi

Enter Destination City

Ahemdabad

Enter number of passengers

1

Enter Contact number

46849651651

Enter Passenger's Name

Siddhi
```

```
Seats Available(0 means available):
        0 0    0 0
        0 0    0 0
        0 0    1 1
        0 0    0 0
        0 0    0 0
        0 0    0 0
        0 0    0 0
        0 0    0 0
        0 0    0 0
        0 0    0 0
Enter Seat Numbers:
11
Seat is already booked!!
Enter another seat:
10
Tickets are Booked!!
=====Customer Deatils=====
Booking Id:8641143
Name of the Customer:Siddhi
Contact Number:46849651651
Destination City:Ahemdabad
Number of Passengers Travelling:1
List of Passenger's:
Siddhi
Seat Number:10
```

```
1.Book a Ticket.
2.Cancel a Ticket.
3.Check Reservation.
4.Check Availability.
5.City List
6.Log Out
3
Enter your Booking Id:
8641143
Yes! You have booking.
=====Customer Deatils=====
Booking Id:8641143
Name of the Customer:Siddhi
Contact Number:46849651651
Destination City:Ahemdabad
Number of Passengers Travelling:1
List of Passenger's:
Siddhi
Seat Number:10
1.Book a Ticket.
2.Cancel a Ticket.
3.Check Reservation.
4.Check Availability.
5.City List
6.Log Out
4
```

```
Enter destination city:
Mumbai
Seats Available(0 means available):
        0 0    0 0
        0 0    0 0
        0 0    0 0
        0 0    0 0
        0 0    0 0
        0 0    0 0
        0 0    0 0
        0 0    0 0
        0 0    0 0
        0 0    0 0
1.Book a Ticket.
2.Cancel a Ticket.
3.Check Reservation.
4.Check Availability.
5.City List
6.Log Out
2
Enter your Booking Id:
8641143
Ticket is cancelled.
1.Book a Ticket.
2.Cancel a Ticket.
3.Check Reservation.
```

```
4.Check Availability.
5.City List
6.Log Out
6
Logging Out...
Login as:
1.Admin
2.Customer
3.Exit
1
Logged in as Admin
1.Add City
2.Delete City
3.Add Bus
4.Cancel Bus
5.Customer List
6.Log Out
1
Enter City to be added:
Nashik
City already added
1.Add City
2.Delete City
3.Add Bus
4.Cancel Bus
5.Customer List
```

```
6.Log Out
1
Enter City to be added:
Bhusawal
Bus added
City added
1.Add City
2.Delete City
3.Add Bus
4.Cancel Bus
5.Customer List
6.Log Out
2
Enter City name to be deleted
Mumbai
City deleted
1.Add City
2.Delete City
3.Add Bus
4.Cancel Bus
5.Customer List
6.Log Out
4
Enter bus destination city which you want to cancel
Nashik
1.Add City
```

```
2.Delete City
3.Add Bus
4.Cancel Bus
5.Customer List
6.Log Out
5
Customer List in Booking Order:
Siddhi
Varun Karwa
1.Add City
2.Delete City
3.Add Bus
4.Cancel Bus
5.Customer List
6.Log Out
6
Logging Out...
Login as:
1.Admin
2.Customer
3.Exit
2
Logged in as Customer
1.Book a Ticket.
2.Cancel a Ticket.
3.Check Reservation.
```

```
4.Check Availability.
5.City List
6.Log Out
5
We have service to the following cities:
        Ahemdabad
        Kolhapur
        Nashik
        Indore
        Nagpur
        Jalgaon
        Bhusawal
1.Book a Ticket.
2.Cancel a Ticket.
3.Check Reservation.
4.Check Availability.
5.City List
6.Log Out
4
Enter destination city:
Nashik
Bus is cancelled
1.Book a Ticket.
2.Cancel a Ticket.
3.Check Reservation.
4.Check Availability.
```

```
5.City List
6.Log Out
6
Logging Out...
Login as:
1.Admin
2.Customer
3.Exit
3
Exiting…
```