ASSESSMENT ITEM 2—JAVA PROGRAM USING AN ARRAY OF OBJECTS

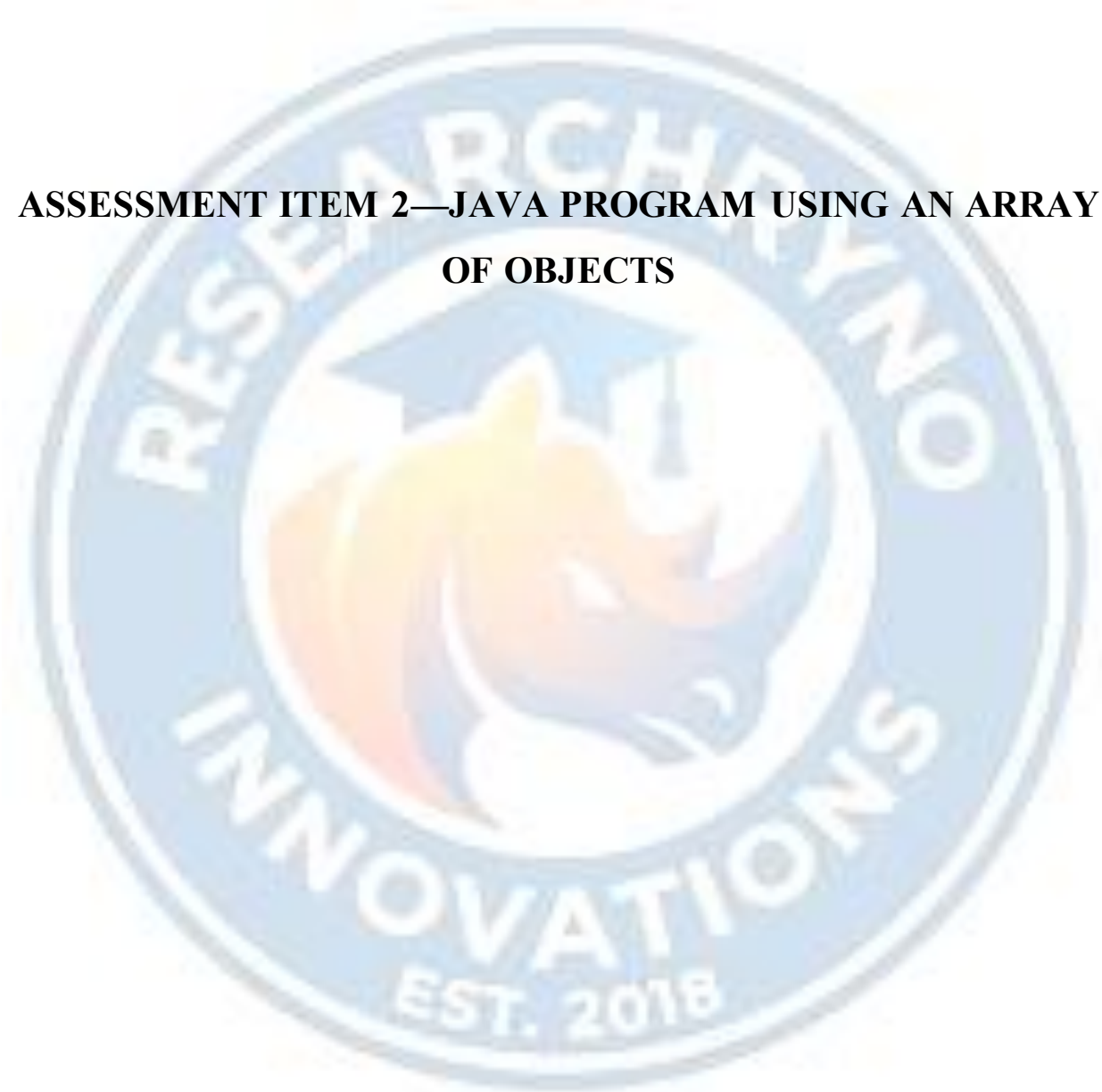

Table of Contents

Introduction

Java is a relatively basic programming language that is class based and object-oriented. It is designed for programmers to create code that can run wherever that code is compiled and run all platforms that support java. Java basically makes for writing, debugging and compiling programming to make it easy.

In this assignment the developer will implement a Java programming application which should be a console application for handling reservations for Yeppoon Cabins. The programming includes the room booking system where the names of booking and number of nights can be stored in an array of objects.

Design and Implementation

Design

The design section contains an UML class diagram of the Booking class to elaborate the entire customer booking process where the customer can choose from a booking menu (Htay and Khaing, 2018). These diagrams will provide a better understanding to the reader.

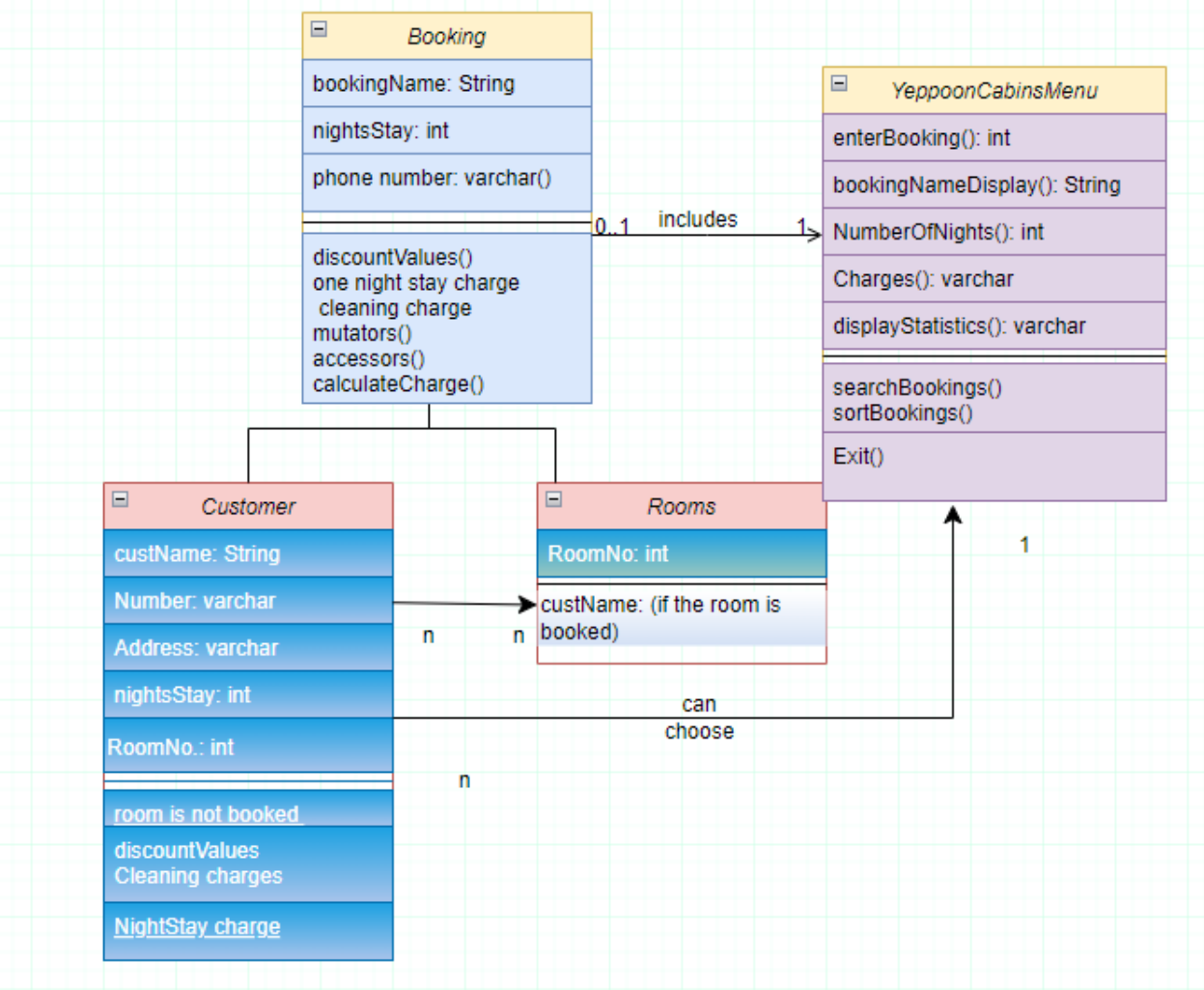

Figure 1: UML Class Diagram

(Source: Self-Created)

The above image represents a class diagram with ***Booking***, ***YepponCabinsMenu***, Customer and Rooms classes. The class diagram represents the entire database design of the application where the classes are the tables of the database (Assunção *et al.*2020).

In the ***Booking*** class the attributes are ***bookingName*** (datatype: string), ***nightsStay*** (datatype: integer), ***phoneNumber*** (datatype: varchar). The class also includes constructors and methods such as ***discountValues(), one night stay charge***, ***cleaning charge***, ***calculateCharge()***.

The ***YepponCabinsMenu*** class contains the booking menu from which the customers will be able to choose the options. The options are ***enterBooking*** (datatype: integer), ***bookingNameDisplay*** (datatype: string), ***NumberOfNights*** (datatype: integer), ***charges*** (datatype: varchar). In this class it also contains constructors and methods like ***searchBookings()***, ***sortBookings()*** and ***exit()***.

In this UML diagram the relationship between the ***Booking*** class and the ***YeppoonCabinsMenu*** class is ***one-to-one relationship***.

The Booking class includes two individual classes such as ***Customer*** class and ***Rooms*** class.

In Customer class the columns are ***custName*** (datatype: string), ***Number*** (datatype: varchar), ***Address*** (datatype: varchar), ***nightStay*** (datatype: integer), ***RoomNo*** (datatype: integer).

In the ***Rooms*** class the attributes are ***RoomNo*** (datatype: integer).

In the above diagram the relationship between ***Customer*** class and ***Booking*** class is ***many-to-many relationship*** and relationship between ***Customer*** class and ***YeppoonCabinsMenu*** is ***many-to-one relationship***.

Implementation

The implementation section includes the screenshots of console program outputs where the developer has used Java classes, methods, Scanner class, if conditions and so on. The Programming will help in booking rooms in Yeppoon cabins hotel.

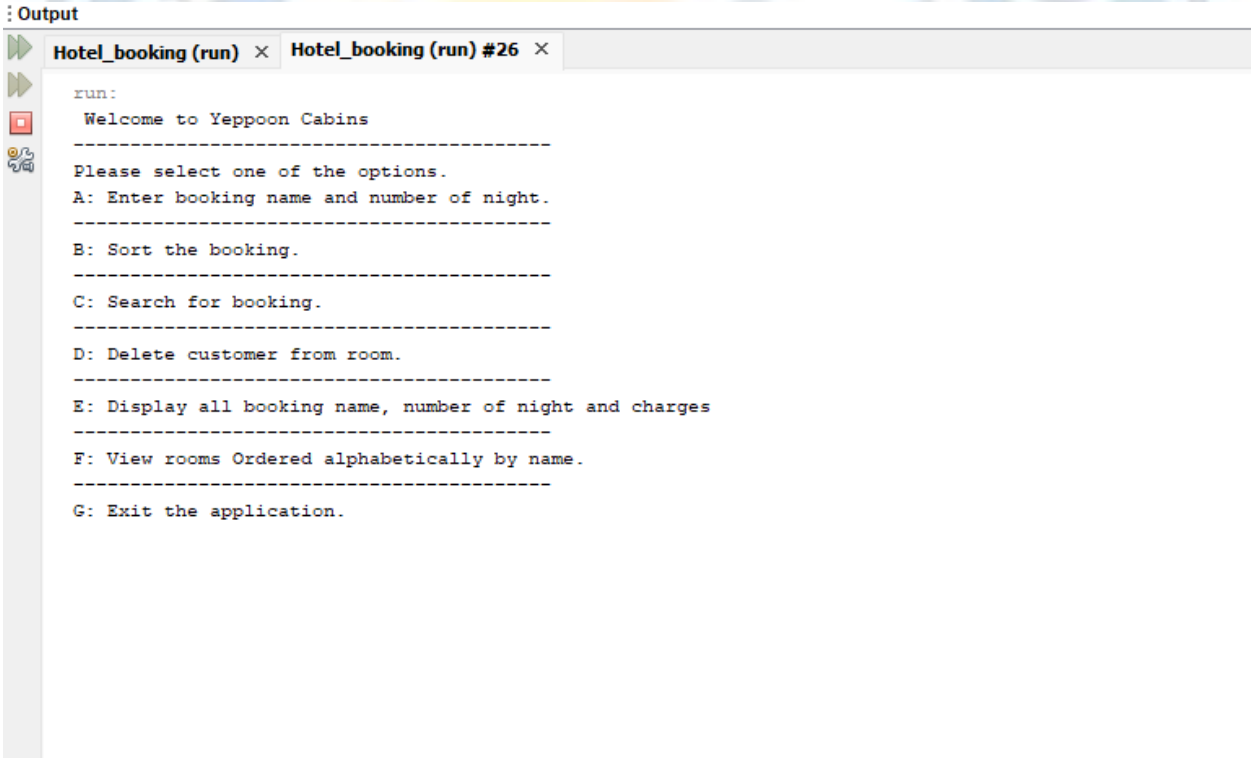

Figure 2: Yeppoon Cabins Menu Console

(Source: Apache Netbeans)

The above image shows the console output of Yeppoon Cabins Menu from where the customers have to select the options.

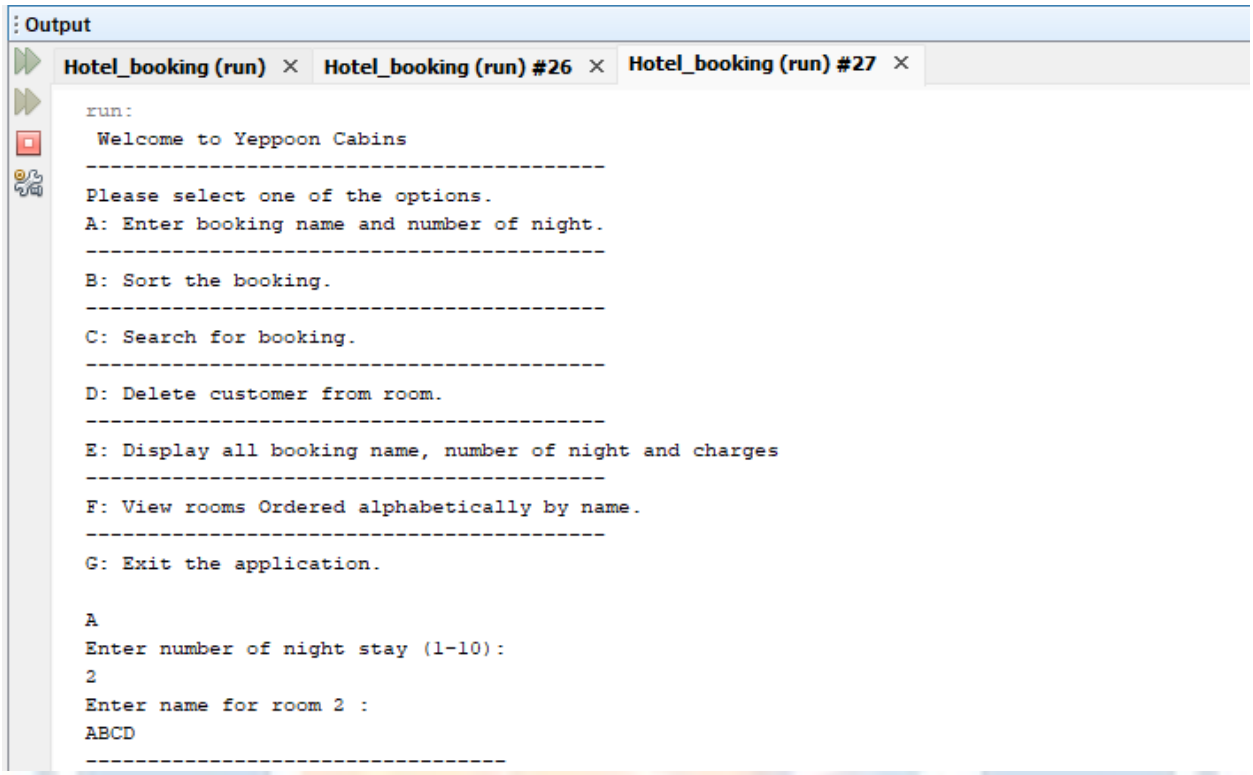

Figure 3: Booking room process

(Source: Apache Netbeans)

The above output shows that the customer has selected a menu option of booking name and number of night stay.

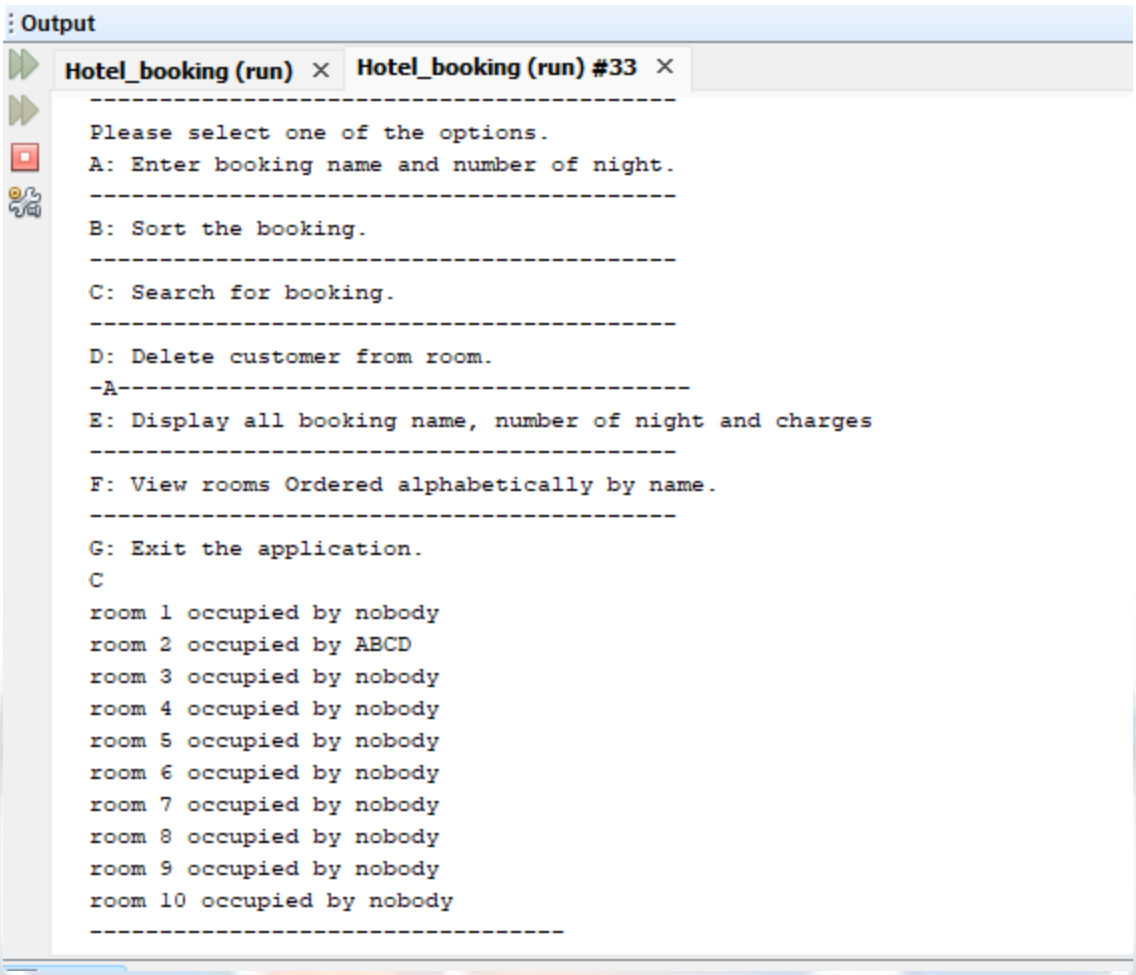

Figure 4: Sorting rooms by booking status

(Source: Apache Netbeans)

The above output shows that the admin can display and check the rooms by its booking name in ascending order.

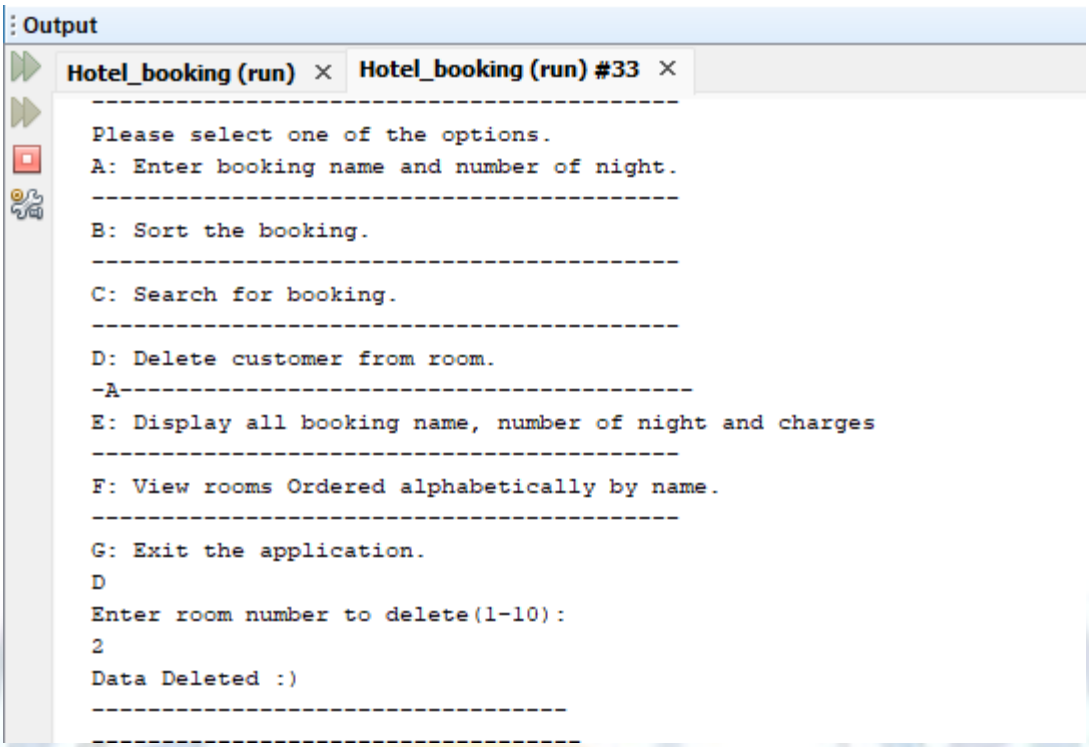

Figure 5: Deleting data

(Source: Apache Netbeans)

The above image shows that the admin can delete the customer data, when customers leave the room.

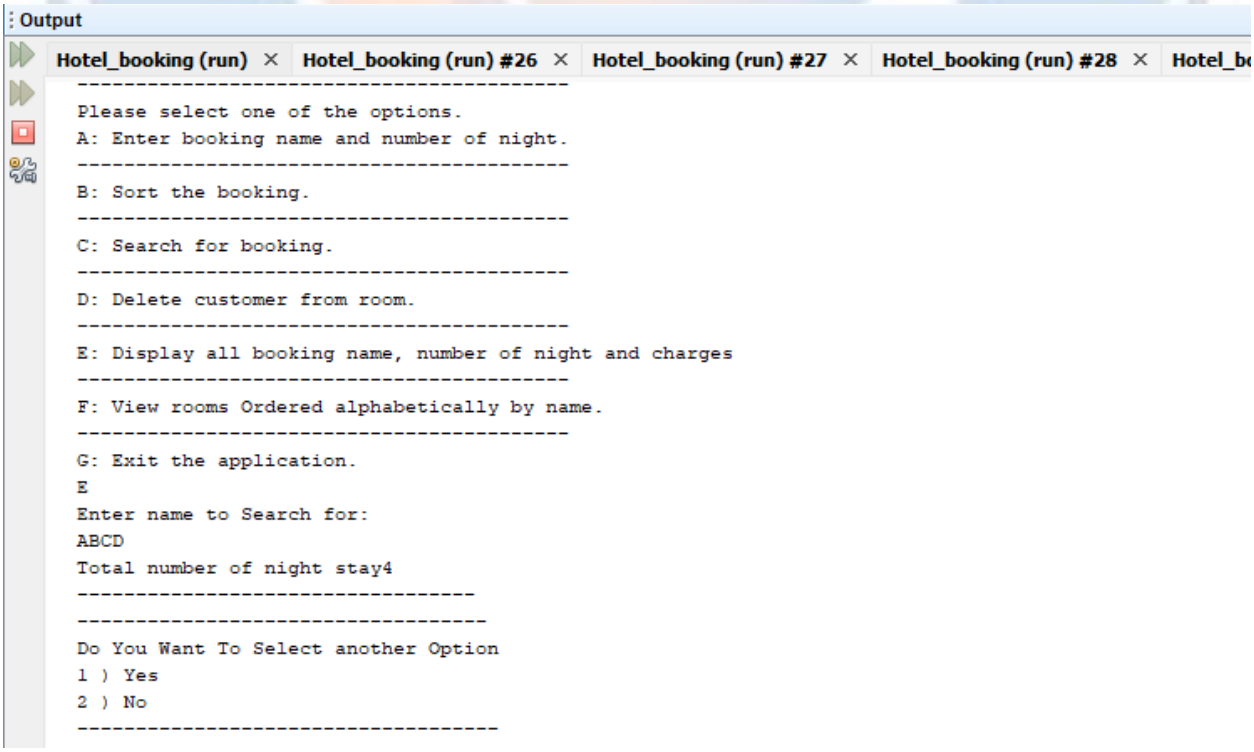

Figure 6: Selecting Menu Options

(Source: Apache Netbeans)

The above output shows admin can search customer details with their name and total number of night stay.

Test Plan

The testing plan shows that the program is working properly. Here each test cases are provided with screenshots.

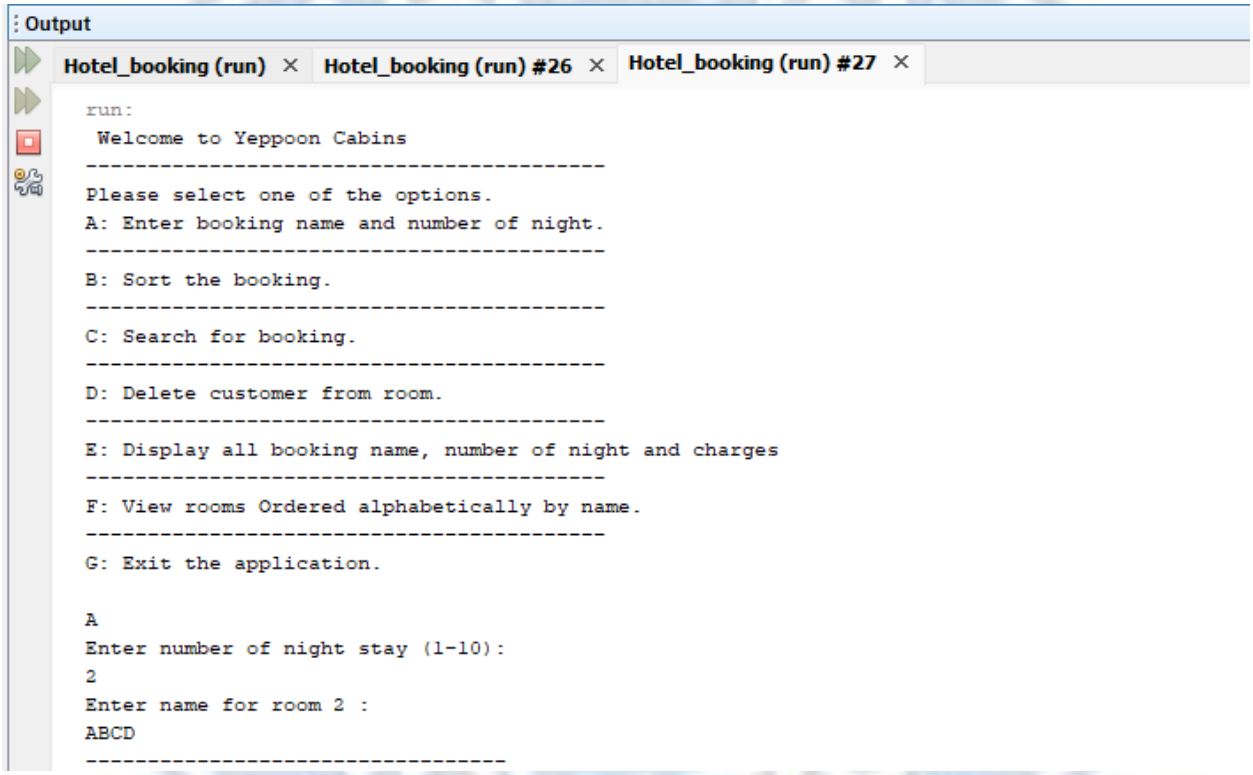

Figure 7: Selecting Menu Options testing

(Source: Apache Netbeans)

In the above image admin can search the customer data with their night stay number.

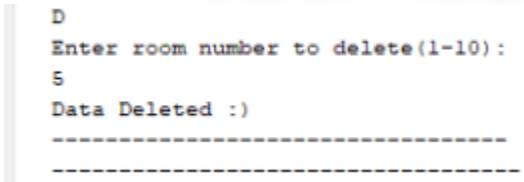

Figure 8: Data Deleted Successfully

(Source: Apache Netbeans)

```
F: View rooms Ordered alphabetically by name.
----------------------------------------
G: Exit the application.
C
room 1 occupied by nobody
room 2 occupied by ABCD
room 3 occupied by nobody
room 4 occupied by nobody
room 5 occupied by nobody
room 6 occupied by nobody
room 7 occupied by nobody
room 8 occupied by nobody
room 9 occupied by nobody
room 10 occupied by nobody
----------------------------------
```

Figure 9: Sorting rooms by booking status testing

(Source: Apache Netbeans)

```
F: View rooms Ordered alphabetically by name.
----------------------------------------
G: Exit the application.
E
Enter name to Search for:
ABCD
Total number of night stay4
----------------------------------
----------------------------------
Do You Want To Select another Option
1 ) Yes
2 ) No
```

Figure 10: Total Number of Night Stay testing

(Source: Apache Netbeans)

Conclusion

The developer has successfully completed the assignment by implementing the Java application of the customer booking system in Yeppoon Cabins. Each requirement of the console program has been fulfilled accordingly. An UML class diagram has been created elaborating the entire booking process console program using Java programming has been executed where there are two main classes Booking and YeppoonCabinsMenu.

It is a simple Java program which will help customers to book rooms in Yeppoons Cabins and also this assignment will clear the basics of java programming.

Reference

Assunção, W.K., Vergilio, S.R. and Lopez-Herrejon, R.E., 2020. Automatic extraction of product line architecture and feature models from UML class diagram variants. *Information and Software Technology*, *117*, p.106198.

Cai, Z., Liu, Y., Gan, Y., Li, J. and Feng, Y., 2019. Design and implementation of online mall system based on java web. *International Journal of Performability Engineering*, *15*(12), p.3237.

Htay, S.N. and Khaing, M. 2018, *Designing Object Oriented Models through UML diagrams for Hotel Front Office System* (Doctoral dissertation, MERAL Portal).

Lucas, W., Bonifácio, R., Canedo, E.D., Marcílio, D. and Lima, F., 2019, September. Does the introduction of lambda expressions improve the comprehension of java programs?. In *Proceedings of the XXXIII Brazilian Symposium on Software Engineering* (pp. 187-196).

Appendices

Appendix 1: Booking.java

```
package Booking_application;

import java.io.BufferedReader;
import java.io.FileInputStream;
import java.io.FileWriter;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import static java.time.Clock.system;
import java.util.Arrays;
import java.util.Scanner;

public class YeppoonCabinsMenu {

    private static boolean MainMenu = true;
    private static boolean SubMenu = true;


    public static void main(String[] args) throws IOException {
        Scanner input = new Scanner(System.in);
        Booking[] myHotel = new Booking[10];
        myHotel[0] = new Booking();
        myHotel[1] = new Booking();
        myHotel[2] = new Booking();
        myHotel[3] = new Booking();
        myHotel[4] = new Booking();
        myHotel[5] = new Booking();
        myHotel[6] = new Booking();
        myHotel[7] = new Booking();
```

```
    myHotel[8] = new Booking();
    myHotel[9] = new Booking();
    int roomNum = 0;

initialise(myHotel);

    while (MainMenu) {
        while (SubMenu) {

            System.out.println(" Welcome to Yeppoon Cabins ");
            System.out.println("----------------------------------------");
            System.out.println("Please select one of the options.");
            System.out.println("A: Enter booking name and number of night.");
            System.out.println("----------------------------------------");
            System.out.println("B: Sort the booking.");
            System.out.println("----------------------------------------");
            System.out.println("C: Search for booking.");
            System.out.println("----------------------------------------");
            System.out.println("D: Delete customer from room.");
            System.out.println("-A--------------------------------------");
            System.out.println("E: Display all booking name, number of night and charges");
            System.out.println("----------------------------------------");
            System.out.println("F: View rooms Ordered alphabetically by name.");
            System.out.println("----------------------------------------");
            System.out.println("G: Exit the application.");



            String Selection = input.next();
            Selection = Selection.toUpperCase();
```

```
switch (Selection) {
    case "A":
        BookAOrder(myHotel, roomNum);
        break;
    case "B":
        CheckIfEmpty(myHotel);
        break;
    case "C":
        ViewAllRooms(myHotel);
        break;
    case "D":
        DeleteCustomerFromRoom(myHotel, roomNum);
        break;
    case "E":
        FindRoomFromCustomerName(myHotel);
        break;
    case "F":
        ViewRoomsOrderedAlphabeticallyByName(myHotel);
        break;



    default:
        System.out.println("Invalid Selection");
        break;
}
System.out.println("---------------------------------");
System.out.println("---------------------------------");
```

```
            System.out.println("Do You Want To Select another Option\n1 ) Yes\n2 ) No");
            System.out.println("-----------------------------------");
            System.out.println("-----------------------------------");
            if (input.nextInt() == 1) {
               SubMenu = true;
            } else {
               SubMenu = false;
            }
         }
         SubMenu = true;
         System.out.println("-----------------------------------");
         System.out.println("-----------------------------------");
         System.out.println("Would You Like To Continue With The Program\n1 ) Yes\n2 ) No");
         System.out.println("-----------------------------------");
         System.out.println("-----------------------------------");
         if (input.nextInt() == 1) {
            MainMenu = true;
         } else {
            System.out.println("");
            System.exit(0);

         }
      }

}

private static void initialise(Booking[] myHotel) {
   for (int x = 0; x < myHotel.length; x++) {
      myHotel[x].setName("nobody");
   }
}
```

```
private static void CheckIfEmpty(Booking[] myHotel) {
    for (int x = 0; x < myHotel.length; x++) {
        if (myHotel[x].getName().equals("nobody"))  {
            System.out.println("room " + (x + 1) + " is empty");
        }
    }
}

private static void BookAOrder(Booking[] myHotel, int roomNum) {
    String roomName;
    Scanner input = new Scanner(System.in);
    System.out.println("Enter number of night stay (1-10):");
    roomNum = input.nextInt() - 1;
    System.out.println("Enter name for room " + (roomNum + 1) + " :");
    roomName = input.next();
    myHotel[roomNum].setName(roomName);
}

private static void ViewAllRooms(Booking[] myHotel) {
    for (int x = 0; x < myHotel.length; x++) {
        System.out.println("room " + (x + 1) + " occupied by " + myHotel[x].getName());
    }
}

private static void DeleteCustomerFromRoom(Booking[] myHotel, int roomNum) {
    Scanner input = new Scanner(System.in);
    System.out.println("Enter room number to delete(1-10):");
    roomNum = input.nextInt() - 1;
    myHotel[roomNum].setName("nobody");
    System.out.println("Data Deleted :)");
```

```
    }

    private static void FindRoomFromCustomerName(Booking[] myHotel) {
        Scanner input = new Scanner(System.in);
        String orderName;
        System.out.println("Enter name to Search for:");
        orderName = input.next();
        int x;
        boolean Checker = true;
        for (x = 0; x < myHotel.length; x++) {
            if (orderName.equals(myHotel[x].getName())) {

                System.out.println("Total number of night stay" + x );
                Checker = true;

            }
        }
        if (Checker == false) {
            System.out.println("There are no Rooms Booked with that name\n(make sure you've used
the correct names)");
        }
    }

    private static void ViewRoomsOrderedAlphabeticallyByName(Booking[] myHotel) {
        String[] myStrArray = new String[myHotel.length];
        for (int i = 0; i < myHotel.length; i++) {
            myStrArray[i] = myHotel[1].getName();
        }


        Arrays.sort(myStrArray);
```

```
        for (int a = 0; a < myStrArray.length; a++) {
            System.out.println(myStrArray[a]);
        }

    }


}
```

Appendix 1: Booking.java

```
package Booking_application;

public class Booking {

    private String  BookingName;
    int NightStay;

    public Booking() {
        BookingName = "A";
        NightStay = 1;
    }

    public void setName(String aName) {

        BookingName = aName;
    }
     public void setNumber(int aStay) {

        NightStay = aStay;
    }
```

```
    public String getName() {
        return BookingName;
    }


    public int getNumber() {
        return NightStay;
    }
}
```