_____DAL.CS_____

```
using System;
using System.Configuration;
using System.Data;
using System.Data.SqlClient;
using System.Text;

namespace DataAccessLayer
{
   public class ConnectionDb
   {
      /PLEASE DO NOT REMOVE/MODIFY THE VARIABLES and METHODS - InsertData and GetAllData LISTED BELOW/
      /Internally used method starts/

      public static SqlConnection sqlConnection = null;
      public static SqlCommand sqlCommand = null;
      public static SqlDataAdapter sqlDataAdapter = new SqlDataAdapter();
      public static DataSet dataset = new DataSet();
      public static DataTable dataTable;
      static readonly string connection = ConfigurationManager.ConnectionStrings["HotelManagement"].ConnectionString;

      public static int InsertData()
      {
```

```
        using (sqlConnection = new SqlConnection(connection))
        {
            sqlConnection.Open();
            StringBuilder commandBuilder = new StringBuilder();
            commandBuilder.Append("if not exists(select 1 from reservation where iRoomNo=101
and dCheckInDate='2010-03-02' and dCheckoutDate='2017-03-04') ");
            commandBuilder.Append("begin insert into reservation
values(134,'03/02/2010','03/04/2017',101,'Booked',6000) end");

            SqlCommand cmd = new SqlCommand(commandBuilder.ToString())
            {
                Connection = sqlConnection
            };
            return cmd.ExecuteNonQuery();
        }
    }

    public static DataTable GetAllData()
    {
        using (sqlConnection = new SqlConnection(connection))
        {
            sqlConnection.Open();
            sqlCommand = new SqlCommand("select * from Reservation", sqlConnection);
            sqlDataAdapter = new SqlDataAdapter(sqlCommand);
            dataset = new DataSet();
            sqlDataAdapter.Fill(dataset);
```

```
        dataTable = dataset.Tables[0];
        return dataTable;
    }
}
/Internally used method end/




public DataTable GetRoomTypes()
{

    try
    {
        sqlConnection=new SqlConnection(connection);
        sqlConnection.Open();
        sqlCommand=new SqlCommand("select * from Roomtype",sqlConnection);
        sqlDataAdapter=new SqlDataAdapter(sqlCommand);
        dataTable=new DataTable();
        sqlDataAdapter.Fill(dataTable);
    }
    catch(Exception ex)
    {
```

```
            Console.WriteLine(ex.Message);


        }
        finally {sqlConnection.Close();}
        return dataTable;
    }
    public int AddBooking(int cust_id,DateTime check_in,DateTime check_out,int
room_no,string reservation_Status,float total_Charge)
    {
        int reservationID=0;
        int res;
        try{
            if(ValidateBooking(check_in,room_no)==false)
            {
                sqlConnection=new SqlConnection(connection);
                sqlConnection.Open();
                sqlCommand=new SqlCommand("insert into Reservation
values(@p1,@p2,@p3,@p4,@p5,@p6)",sqlConnection);
                sqlCommand.Parameters.AddWithValue("@p1",cust_id);
                sqlCommand.Parameters.AddWithValue("@p2",check_in);
                sqlCommand.Parameters.AddWithValue("@p3",check_out);
                sqlCommand.Parameters.AddWithValue("@p4",room_no);
                sqlCommand.Parameters.AddWithValue("@p5",reservation_Status);
                sqlCommand.Parameters.AddWithValue("@p6",total_Charge);


                res=sqlCommand.ExecuteNonQuery();
```

```
                reservationID++;
            }
        }
        catch(Exception ex)
        {
            Console.WriteLine(ex.Message);
        }
        finally
        {
            sqlConnection.Close();
        }
        return reservationID;
    }
    public static bool ValidateBooking(DateTime checkInDate,int roomNo)
    {
        using(sqlConnection=new SqlConnection(connection))
        {
            sqlConnection.Open();
            sqlCommand = new SqlCommand("select * from Reservation where iRoomNo=@p1
and @p2 between dCheckInDate and dCheckoutDate",sqlConnection);
            sqlCommand.Parameters.AddWithValue("@p1",roomNo);
            sqlCommand.Parameters.AddWithValue("@p2",checkInDate);
            sqlDataAdapter=new SqlDataAdapter(sqlCommand);
            dataset=new DataSet();
            sqlDataAdapter.Fill(dataset);
```

```
            if(dataset.Tables[0].Rows.Count>0)
            {
               return true;
            }
            return false;
         }
      }
      public DataTable GetRoomDetails(string roomType)
      {
         try{
            using (sqlConnection=new SqlConnection(connection))
            {
               sqlConnection.Open();

            sqlCommand=new SqlCommand("select * from Roomtype where
Roomtype=@p1",sqlConnection);
             sqlCommand.Parameters.AddWithValue("@p1",roomType);

            sqlDataAdapter=new SqlDataAdapter(sqlCommand);
            dataTable=new DataTable();
            sqlDataAdapter.Fill(dataTable);
            }

         }
         catch(Exception ex)
```

```
        {
            Console.WriteLine(ex.Message);
        }
        return dataTable;
    }
    public static DataTable SearchRoomByDate(DateTime searchDate)
    {
        DataTable dataTable=null;
        try
        {
            using(sqlConnection=new SqlConnection(connection))
            {
                sqlConnection.Open();
                sqlCommand=new SqlCommand("select * from Reservation where @p1 between
dCheckInDate and dCheckoutDate",sqlConnection);
                sqlCommand.Parameters.AddWithValue("@p1",searchDate);
                sqlDataAdapter=new SqlDataAdapter(sqlCommand);
                dataTable=new DataTable();
                sqlDataAdapter.Fill(dataTable);
            }

        }
        catch(Exception ex)
        {
            Console.WriteLine(ex.Message);
        }
```

```
            return dataTable;
        }


    }
}
```

_____Program.cs_____

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using DataAccessLayer;

namespace HotelMgmt
{
    internal class Program
    {

        public static void Main(string[] args)
        {
            Reservation reservation = new Reservation();
            while (true)
```

```
{
    try
    {
        Console.WriteLine("---------Welcome to the  hotel --------");
        Console.WriteLine("Enter your choice");
        Console.WriteLine("1. Make a New booking");
        Console.WriteLine("2.Booking Details based on a date");
        Console.WriteLine("3.Exit");
        int choice = Convert.ToInt32(Console.ReadLine());
        switch (choice)
        {
            case 1:
                NewBooking(reservation);
                break;
            case 2:
                SearchByDate(reservation);
                break;
            default:
                Environment.Exit(0);
                break;

        }
    }
    catch(Exception ex)
    {
```

```
                Console.WriteLine(ex.Message);
            }

        }

    }
    private static void NewBooking(Reservation reservation)
    {
        Console.WriteLine("_Make a New Booking_");
        Console.WriteLine("Please Enter the following Details");
        Console.WriteLine("Enter the Customer id");
        int cus_id = Convert.ToInt32(Console.ReadLine());
        Console.WriteLine("Enter the Date on which you want to check in (mm/dd/yyyy format)");
        DateTime check_in = Convert.ToDateTime(Console.ReadLine());
        Console.WriteLine("Enter the Date on which you want to check out (mm/dd/yyyy
format)");
        DateTime check_out = Convert.ToDateTime(Console.ReadLine());
        var reservationDetail = Reservation.CheckRoomAvailablity(cus_id, check_in, check_out);
        if (reservationDetail != null)
        {
            if (reservation.DisplayandAddBookingDetails(reservationDetail))
            {
                Console.WriteLine("Your Booking Details are:");
                Console.WriteLine("Room Number:"+reservationDetail.RoomNumber);
                Console.WriteLine("Check In
Date:"+Convert.ToDateTime(reservationDetail.CheckinDate).Date.ToString("MM/dd/yyyy"));
```

```
                Console.WriteLine("Check Out Date:"+
Convert.ToDateTime(reservationDetail.CheckoutDate).Date.ToString("MM/dd/yyyy"));

                Console.WriteLine("The number of days for your saty
 :"+reservationDetail.StayDuration);

                Console.WriteLine("Total Charges:"+reservationDetail.TotalCharge);

                Console.WriteLine("Room Booking Successful");

            }

            else

                Console.WriteLine("No rooms available for the selected dates. Please try with
another set of dates");



        }

    }

    private static void SearchByDate(Reservation reservation)

    {

        Console.WriteLine("Enter the Date to View the room bookings (mm/dd/yyyy):");

        DateTime searchDate = Convert.ToDateTime(Console.ReadLine());

       Reservation.SearchByDate(searchDate);

    }









  }

}
```

_____RESERVATION.CS_____

```
using System;
```

```
using DataAccessLayer;

namespace HotelMgmt
{
    public class ReservationDetail
    {
        public int CustomerId { get; set; }
        public DateTime CheckinDate { get; set; }
        public DateTime CheckoutDate { get; set; }
        public int StayDuration { get; set; }
        public float TotalCharge { get; set; }
        public string RoomType { get; set; }
        public int RoomNumber { get; set; }
        public string ReservationStatus { get; set; }
    }
    public class Reservation
    {
        public static void SearchByDate(DateTime searchDate)
        {
            ConnectionDb dal = new ConnectionDb();
            var reservationDetail = ConnectionDb.SearchRoomByDate(searchDate);
            if (reservationDetail.Rows.Count != 0)
            {
                for (int i = 0; i < reservationDetail.Rows.Count; i++)
                {
```

```
            Console.WriteLine("Room Id:" + reservationDetail.Rows[i][0].ToString());
            Console.WriteLine("CheckInDate:" +
Convert.ToDateTime(reservationDetail.Rows[i][1]).Date.ToString("MM/dd/yyyy"));
            Console.WriteLine("CheckOutDate:" +
Convert.ToDateTime(reservationDetail.Rows[i][2]).Date.ToString("MM/dd/yyyy"));
            Console.WriteLine("Total Charge:" + reservationDetail.Rows[i][3].ToString());
          }
        }
        else
        {
          Console.WriteLine("No booking available for the selected date.");
        }
      }
      public static  ReservationDetail CheckRoomAvailablity(int customerId, DateTime
checkInDate, DateTime checkOutDate)
      {
        ReservationDetail reservationDetail = null;
        int stayDuration=0;
        if (checkInDate > checkOutDate)
          throw new Exception("Sorry the Checkout date should be greater than Checkin date");


        ConnectionDb dAL = new ConnectionDb();
         var roomTypes = dAL.GetRoomTypes();
        Console.WriteLine("The Room Types avaliable are :");
        for (int i = 0; i < roomTypes.Rows.Count; i++)
```

```
        {
            if (roomTypes.Rows[i][0].ToString() == "DX")
                Console.WriteLine("Room Type -" + roomTypes.Rows[i][0].ToString() + " Charges:--->
4500");
            if (roomTypes.Rows[i][0].ToString() == "GR")
                Console.WriteLine("Room Type -" + roomTypes.Rows[i][0].ToString() + " Charges:--->
3000");


        }
        stayDuration =Convert.ToInt32((checkOutDate-checkInDate).TotalDays);



        Console.WriteLine("Please choose the Room Type");
        string room_Type = Console.ReadLine();
        Console.WriteLine("The number of days for your Stay :{0}",stayDuration);
        int total_charge = 0;
        bool room_type_Status = false;
        int room_Number;
        if (room_Type == "DX")
        {
            total_charge = 4500 * stayDuration;
            room_type_Status = true;
        }
        else if (room_Type == "GR")
        {
```

```
            total_charge = 3000 * stayDuration;
            room_type_Status = true;
        }
        else
        {
            Console.WriteLine("Sorry, please choose DX or GR");
            room_type_Status = false;
        }
        if (room_type_Status)
        {
            var room_Details = dAL.GetRoomDetails(room_Type);
            for (int i = 0; i < room_Details.Rows.Count; i++)
            {
                Console.WriteLine(room_Details.Rows[i][1].ToString() + " " +
room_Details.Rows[i][0].ToString());
            }
            Console.WriteLine("Please Choose the Room of the Type:{0}", room_Type);
            room_Number = Convert.ToInt32(Console.ReadLine());
            reservationDetail = new ReservationDetail()
            {
                CustomerId = customerId,
                CheckinDate = checkInDate,
                CheckoutDate = checkOutDate,
                StayDuration = stayDuration,
                RoomNumber = room_Number,
                TotalCharge = total_charge,
```

```
                RoomType = room_Type,
                ReservationStatus = room_type_Status ? "Booked" : "Available",
            };
        }

        return reservationDetail;
    }
    public  bool DisplayandAddBookingDetails(ReservationDetail reservationDetail)
    {
        ConnectionDb dal = new ConnectionDb();
        int reservationId = dal.AddBooking(reservationDetail.CustomerId,
reservationDetail.CheckinDate, reservationDetail.CheckoutDate,
reservationDetail.RoomNumber,reservationDetail.ReservationStatus,
reservationDetail.TotalCharge);
        if (reservationId > 0)
        {
            Console.WriteLine("Booking Successfull");
            return true;
        }
        else
        {
            Console.WriteLine(" Room booking can't be done. Please try again later");
            return false;
        }


    }
```

```
    }
}
```