Hotel Management System

Project Final Report

Computer Science, Software engineering Project

CS – 254

Group: 06 (2015)

Submitted by:

1. PS-1641 - P.Wijith Bandara
2. PS-1666 –Y.Yutharshan
3. PS-1749 – P.A.Harsha Madusanka
4. PS-1715 – K.A.B.M Kaluaarachchi
5. PS-1755 – P.Saranya
6. PS-1680 - Sarith Lakmina Mulholland

Abstract

This is the final report document for developed hotel management system for Lotus Hotel .It consists of the milestones in development of finalized hotel management system. As previously mentioned current manual system used by hotel, caused for decrement in growth of success and efficiency of the hotel.

Coding was handled through an Object-oriented approach. Used methodologies made project work load light and provided the ease of developing. The system was evaluated by several people regarding user levels of the developed system. Results of the evaluation helped for further maintenance of the product. Fully functional Lotus Hotel Management System will fulfil the main objectives and all the events of the hotel.

Acknowledgement

We would like to express our very great appreciation to our supervisor Mrs.S.Patheepan for her valuable and constructive suggestions during the planning and development of this project. Her willingness to give her time so generously has been very much appreciated. Advice given by other academic lecturers has been a great help in building the software solution.

Also we obliged to staff members of Lotus Hotel, for the valuable information provided by them in their respective fields. We are grateful for their cooperation during the period of developing the system

Table of Contents

List of Figures

List of Acronyms and Abbreviations

H.M.S. – Hotel Management System

1. Introduction

1.1 Problem Statement

Currently Lotus Hotel is using a manual system to handle hotel processes. When a guest make a reservation, all the reservation details (including guest details) are recorded in a files and those files are stored in a special cabinet. Calculations of bills and inventory items are done by manually too.

As the current system is a file based one, management of the hotel has to put much effort on securing those files. They can be easily get damaged by a fire, insects or even by a natural disaster like tsunami. Keeping files takes much time and wastes much precious man hours. Although we can't trust the accuracy of calculations done by manually, it's not a surprise of encountering problems. If we want to check for a previous room record or a reservation detail, management will be in a great problem. It's a tough and time taking process to search for a record in a file

If we summarize all the drawbacks of manual hotel management system we can mention as follows

- It is much time consuming.
- Often the data are lost and the employee or manager is not aware of this.
- Lots of Manual labor is required for manual record keeping.

- Data is not always reliable as it is hand written and some human errors might have occurred.
  - Example - wrong telephone number among other.
- Difficulty of searching and retrieving records.
- Slow process of reservation.
- Low in security.
- Files are prone to theft unauthorized modification due to low data security levels and standards.
- Retrieval of guest records is extremely difficult.

Above mentioned teasers caused for the decline of efficiency of the hotel and caused for many inconveniences in performing daily tasks at hotel.

The manual handling of hotel management system has always been a difficult task.

Administering the smallest of the requirements of the guests to the basic needs the Hotel Management System have to put a lot of energy in controlling and executing the day-today work along with the miscellaneous items? However, the introduction of the hotel management software has diminished the possibilities of error and limitation of the staff to provide a specific service on any given time

1.2 Product Scope

The introducing software, Hotel Management System which is implemented for Lotus Hotel will automate the major operations of the hotel. The Reservation System is to keep track in room and hall reservation and check availability. The Room Management System is for manage all room types rooms. The Hall booking System will keep track in all booking of the halls in the hotel and catering management, take away and bar also can be manage by this software. Administration department (manager) will monitor the all .There is three End Users for HMS. The End Users are Manager, Receptionist and Barman. Manager can access to all system functionalities without any restrictions. Receptionist can access to all system functionalities with limited restrictions. Receptionist can only access to the Take away section and room booking system and Barman can access to the Daily Bar and Stock Bar, Bite. To keep restrictions for each End User levels HMS can create different Login functions

2. Methodology

2.1 Requirements and Analysis

Requirement gathering in the particular project was mainly done through conducting interviews with different kinds of people of management hierarchy of the company. Though most of the information was gathered from the interviews from the owner, the developing team conducted few interviews with the hotel staff members of different levels to get a wide idea of the company. In this situation, they were enthusiastic and supported us by giving relevant information. Referring to previous hotel records was another way to gather requirements of hotel. Formal system is well defined by the forms, reports, policy manuals and organization charts. Above mentioned tactic helped a lot in providing a clear view of the hotel management system. Majorly we found the users of the system after interviews. It promotes the user roles of the system. Intention of building the system was clearly understood after this phase of software life cycle. Further analysis revealed where and when the system will be used.

After gathering and analyzing the client requirement it was expressed by a use case and the rest of the development process was built on it.

2.2 Product Functions

- Make Reservations
- Booking Rooms
- Manage Bar
- Manage Guest (Add, Update Guest)
- Manage Room Details (Add, Update, Delete)
- Manage Inventory (Add, Edit, Delete)
- Set Rates
- Retrieve Reports
- Manage Users (Add, Update, Delete)
- Add Payment
- Issue Bills

3. User Classes and Characteristics

3.1 User Classes

There are three user levels in Hotel Management System of Hotel Lotus.

I. Manager
II. Receptionist
III. Barman

3.2 Characteristics of User Classes

Manager:-

Hotel manager has the privilege of Monitoring and authorization of all the tasks handle by the system. He can access every function performed by the system. Owner of the company as well as the system can access to the administration panel which is consider the core of the system. As the main authorized person of the company manager gets the ability to manage the other users including their user levels and privileges. Meanwhile he will be able to take all the kinds of reports available in the system. As the manager of the system and the company he has the power to set room rates as well. Hotel manager has the sole right of deleting a staff member from the system database.

Barman:-

Barman is responsible for managing resources available in Bar. The reason for using a Barman is to reduce the work load done by the Manager that cannot be assigned to the receptionist, as those tasks seem much responsible. Barman has the authority to take all the reports related to the bar . Adding new inventory to the Bar, Modifying them and removing them can be done by barman.

Receptionist:-

As a hotel receptionist, he or her role will be to attain the goals of bookings and to ensure that all guests are treated with a high standard of customer service. Hierarchically receptionist role has the least accessibility to the system functions. Receptionist plays the boundary role of the system .He or she can access limited functions such as registering new guest for rooms, make reservations. Management of hotel will prefer to hire receptionist who have a good standard of general education and possibly in subjects such as English, math and IT.

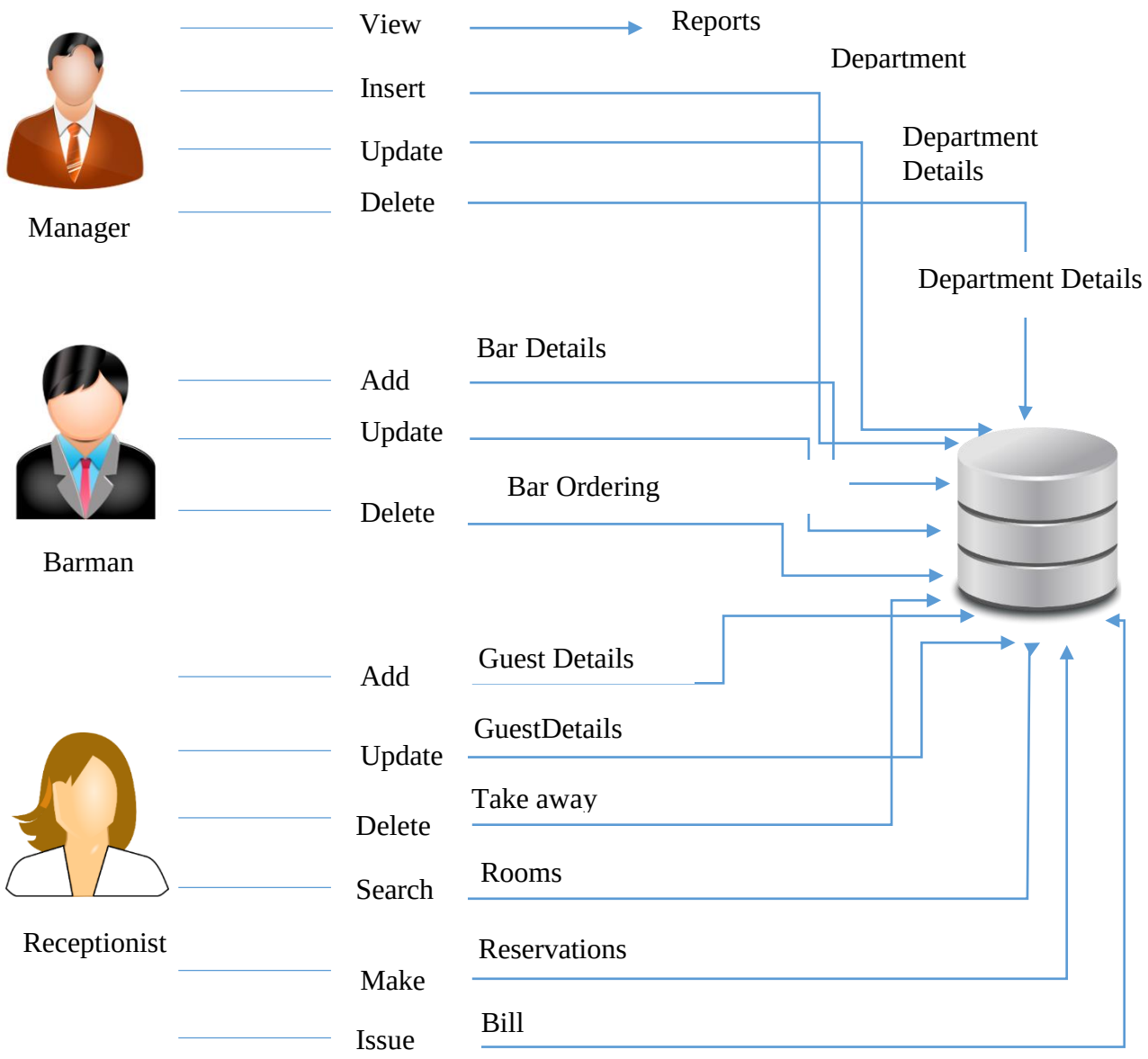
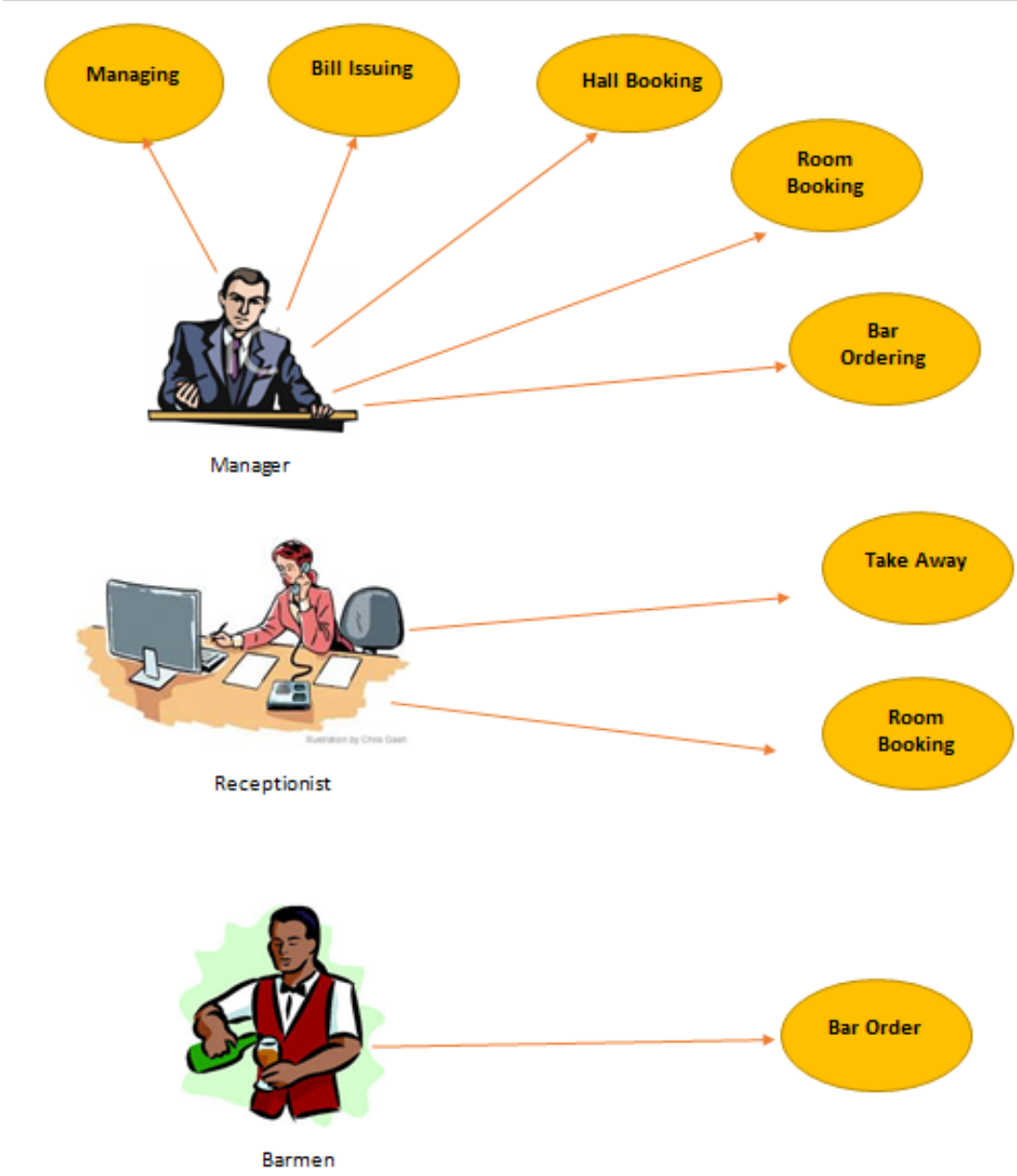


Figure 2.1.1 High Level Architecture

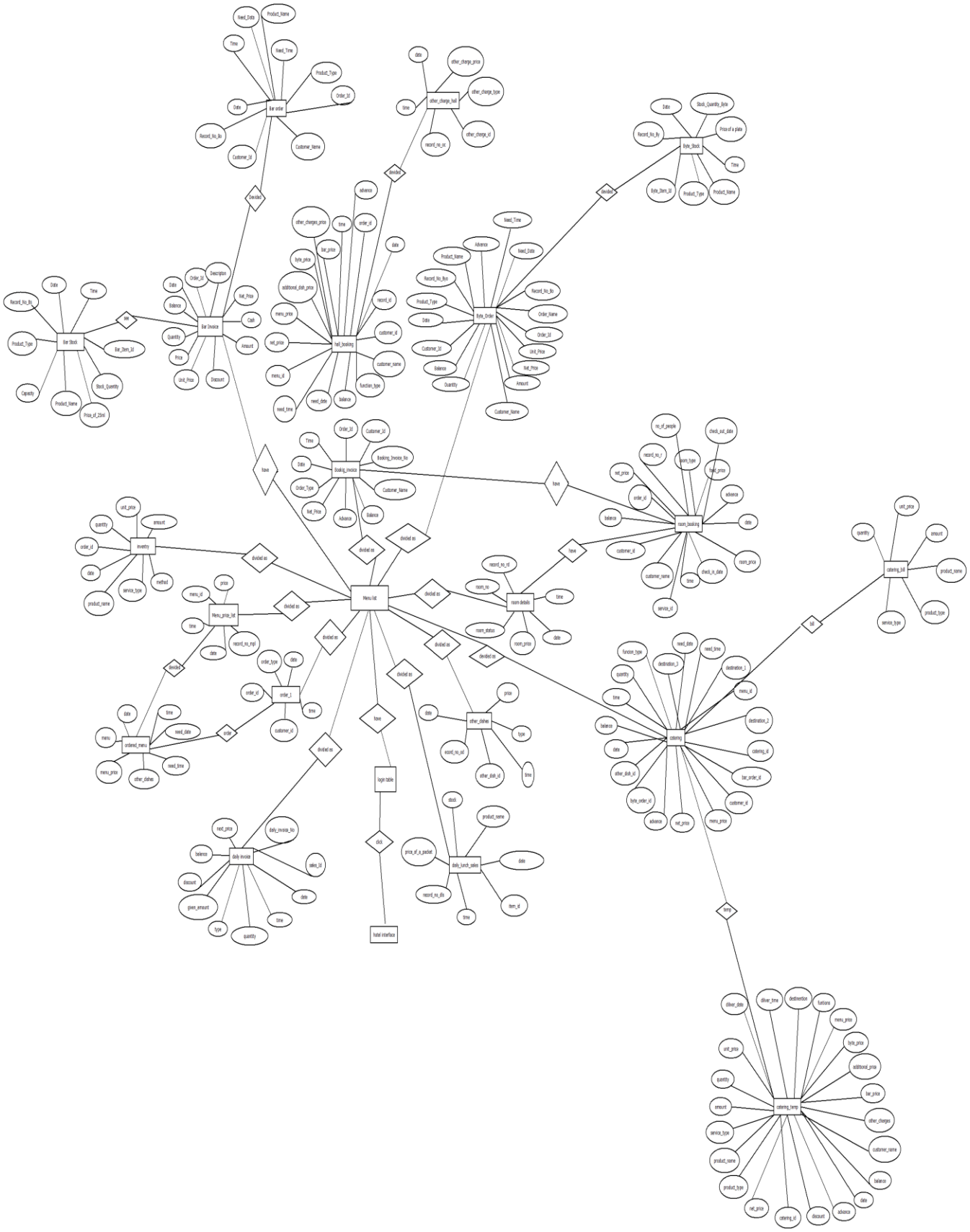


1.1 Use Case Diagram

4. Design

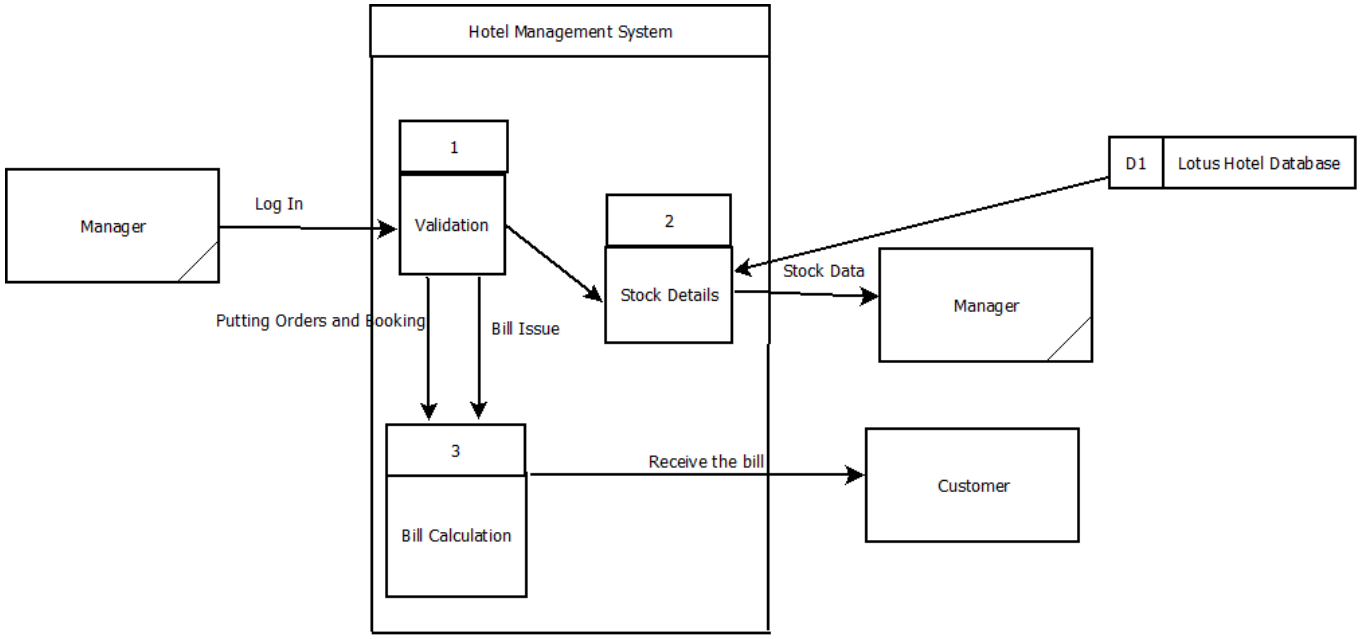
Project was designed using an object oriented approach. While developing the system we had to come across several diagrams in order to achieve the quality of the software product. After creating the ER diagram it was mapped to a relational schema and normalized. Finally the database was created.

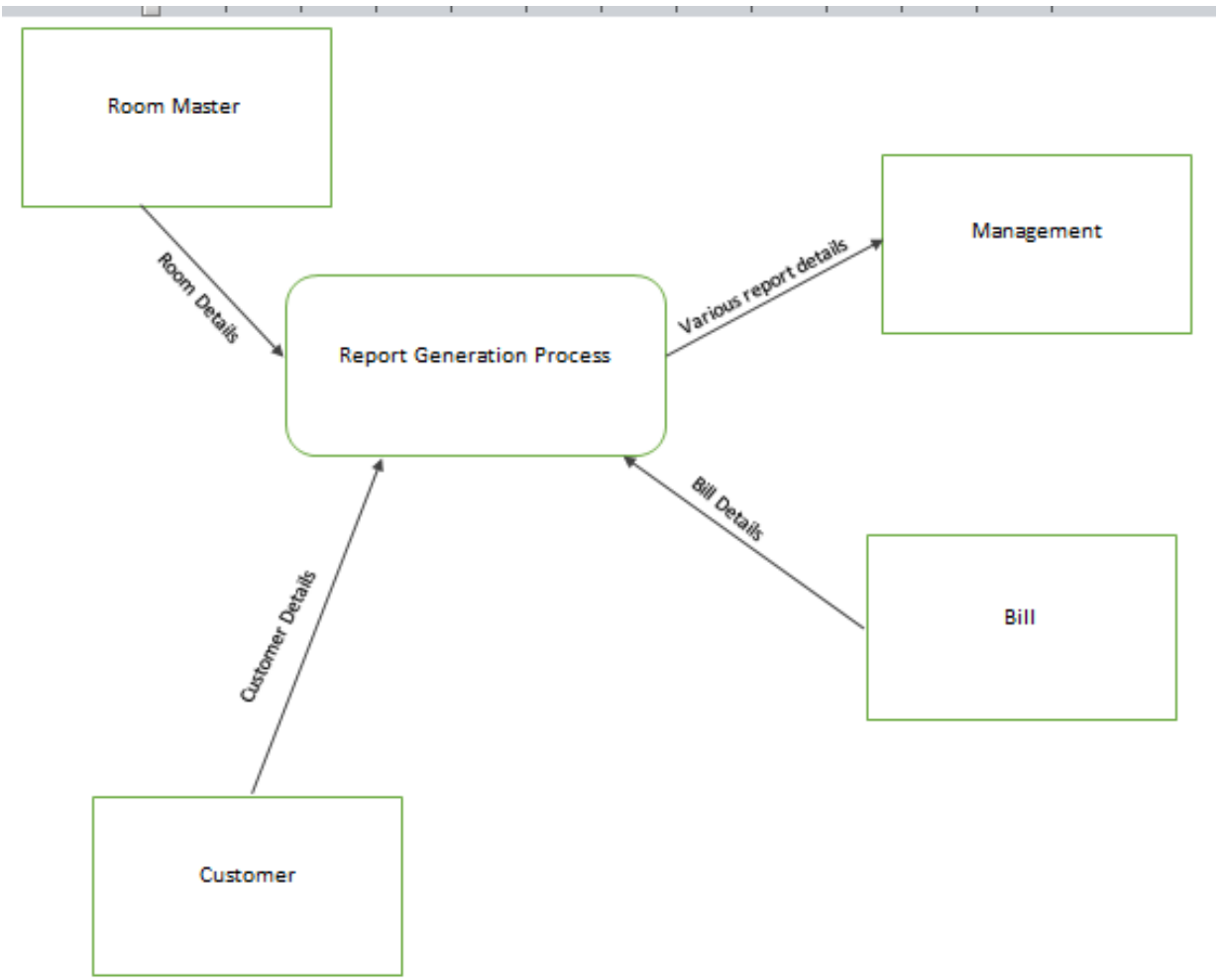
5.Diagrams



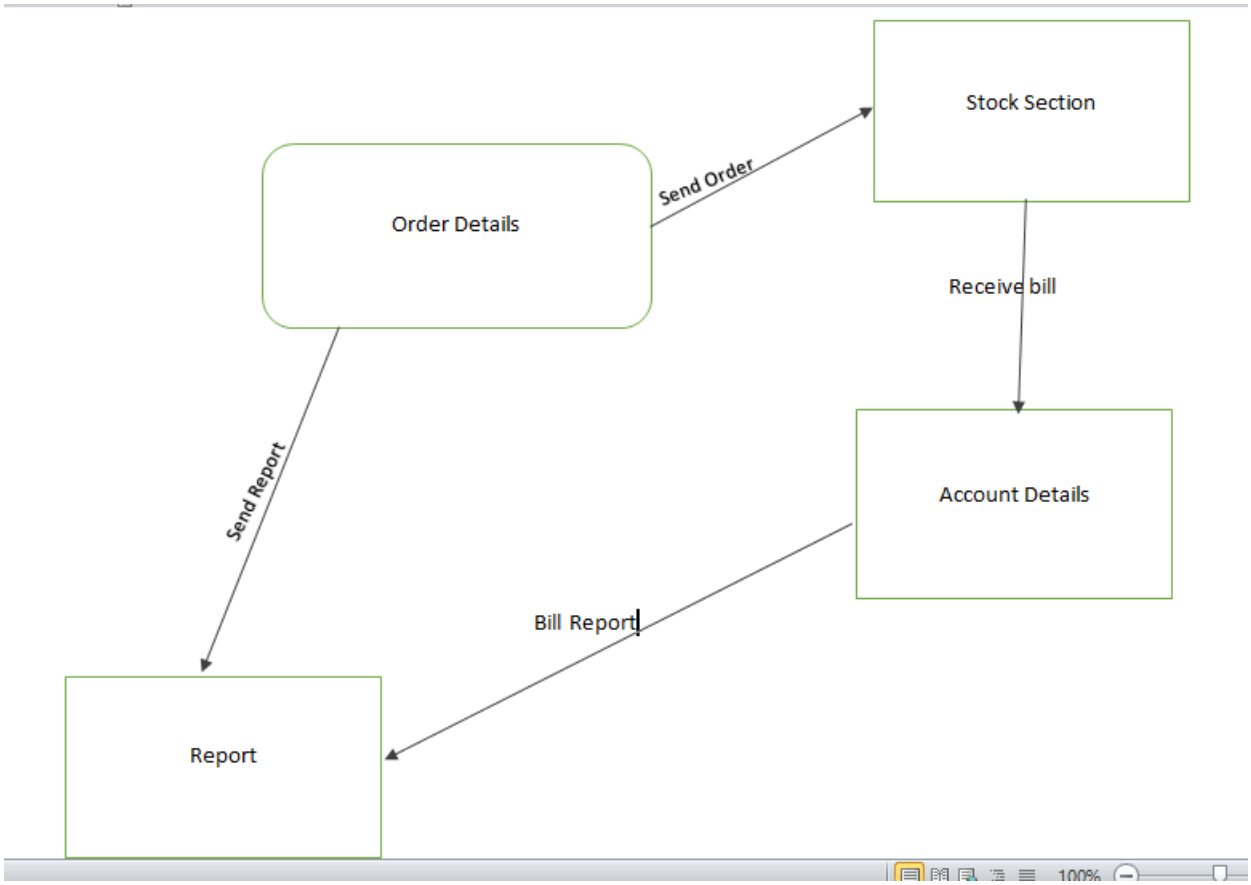
1.2 ER diagram

1.3 Level 0 diagram

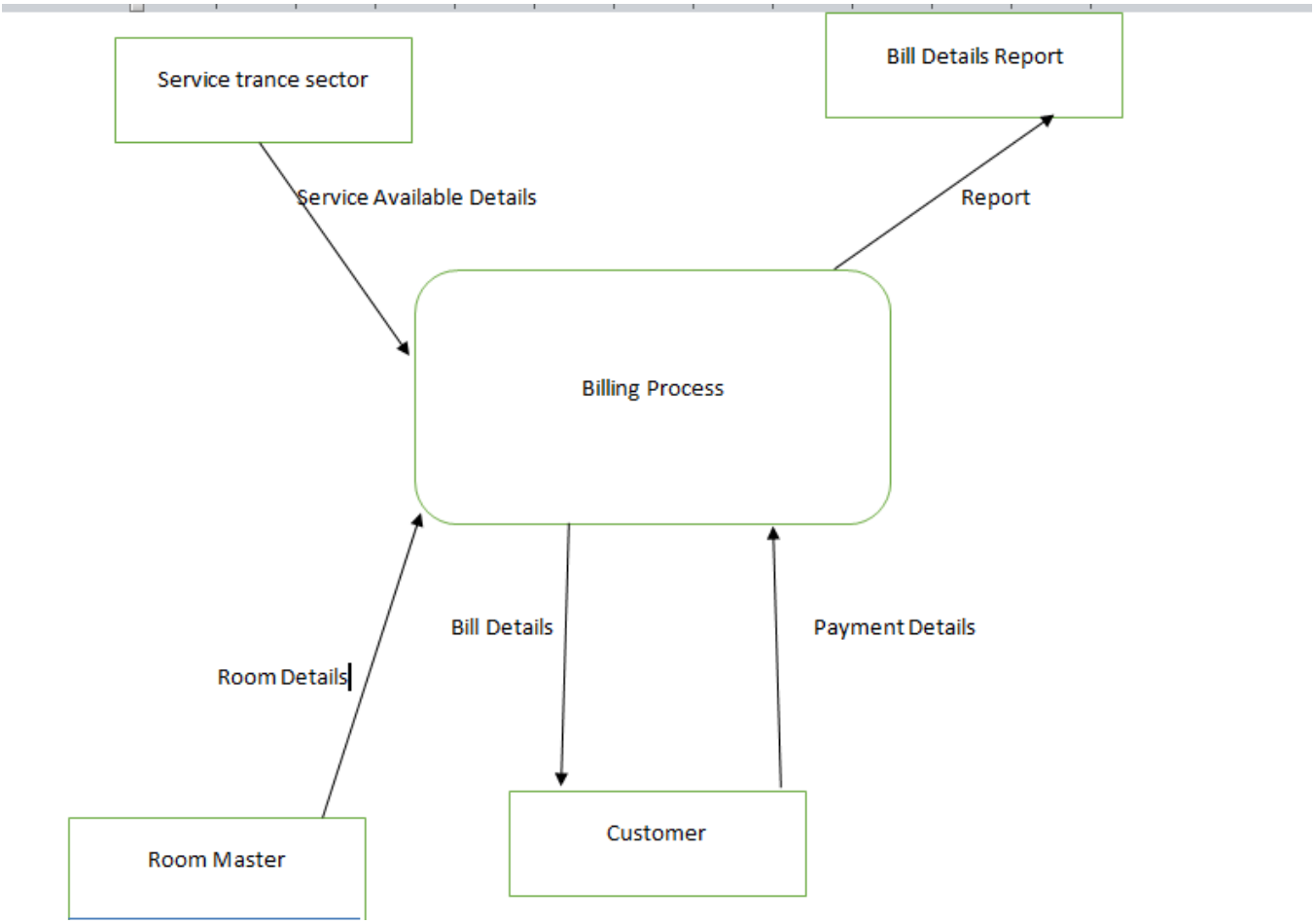




1.4 Level 1 diagram



1.5 Level 1 diagram



1.6 Level 1 diagram

6. Implementation

Database was created using MS sql server studio express 2008 which consist of the capability of producing results table efficient than the older versions. When we consider about c# coding of the project DbAccess class was created in gathering objects from the database. Redundancy of code Part belongs to making connection with the database, was reduced by creating data returning methods inside DBAccess class. It provided the gateway of windows form to access system database.

```
 {
  class ConnectionManager
  {
     public static SqlConnection connection()
     {
       string connectionString =@"Data
Source=.\SQLEXPRESS;AttachDbFilename= (connectionString);
SqlConnection con = new SqlConnection(connectionString);
         con.Open();
         return con;
     }
  }
}
```

After examining the ER diagram system database developed

7. List of Tables

7.1 Bar Invoice

Order_Id	Date	Description	Quantity	Unit_Price	Amount	Net_Price	Discount	Price	Chash	Balance
1	07-Oct-13 2:06:...	Byte Abc Bcd Pl...	1.00	150.00	150.00	150.00	600.00	600.00	500.00	100.00
1	07-Oct-13 2:06:...	Bar White Bran...	1.00	60.00	450.00	60.00	600.00	600.00	500.00	100.00
1	07-Oct-13 10:06...	Bar Arrack V.S....	1.00	45.00	337.50	45.00	397.50	397.50	200.00	197.50
1	07-Oct-13 10:06...	Bar White Bran...	1.00	60.00	60.00	60.00	397.50	397.50	200.00	197.50
2	07-Oct-13 10:12...	Byte Abc Bcd Pl...	1.00	150.00	150.00	150.00	150.00	150.00	120.00	30.00
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

7.2 Bar Order

Record_No_BO	Date	Time	Customer_Id	Customer_Na...	Order_Id	Need_Date	Need_Time	Product_Type	Product_Name	Quantity
1	8/16/2013		123456345v	qadff	NULL	8/16/2013	9:24:40 AM	Vodka(Local)	White Russian	3
2	8/16/2013		934567236v	Nimal	NULL	8/16/2013	9:26:57 AM	Vodka(Local)	White Russian	12
3	8/16/2013		123456789v	gff	NULL	8/16/2013	9:34:37 AM	Vodka(Local)	White Russian	3
4	8/27/2013		1012	Thushani Athth...	NULL	8/21/2013	12:50:51 PM	Arrack		1
5	8/27/2013		1012	56	NULL	8/21/2013	1:23:18 PM	Arrack		2
6					NULL					2
7	8/27/2013		1012	sd	NULL	8/21/2013	1:28:05 PM	Arrack		3
8					NULL					2
9	8/27/2013		1012		NULL	8/21/2013	7:14:49 PM	Arrack		2
10	8/27/2013		123456789v	gff	NULL	8/27/2013	10:09:01 PM	Arrack	Old Arrak	1

7.3 Bar Stock

Record_No_BS	Date	Time	Bar_Item_Id	Product_Type	Product_Name	Stock_Quantity	Capasity	Price_Of_25ml
129	10/6/2013	10:24:08 AM	ArrB	Arrack	Berar	1	750	60
130	10/6/2013	10:24:20 AM	ArrB1	Arrack	Viski	1	750	60
131	10/6/2013	10:24:47 AM	ArrB12	White	Viski	1	750	60
133	10/6/2013	10:25:08 AM	ArriB12	White	Brandi	2	750	60
124	10/6/2013	7:07:11 PM	AVSO	Arrack	V.S.O.A	1	750	45
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

7.4 Booking Invoice

Booking_Invoi...	Date	Time	Order_Id	Customer_Id	Customer_Na...	Order_Type	Net_Price	Advance	Balance
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

7.5 Bite Order

Record_No_BYO	Date	Time	Customer_Id	Customer_Na...	Order_Id	Record_No_BO	Need_Date	Need_Time	Product_Type	Product_Nam
1	8/16/2013	10:58:29 AM	123456789v	gff	NULL	NULL	8/16/2013	10:57:58 AM	Cba	Dcb
2	8/27/2013	12:59:49 PM	123456789v	gff	NULL	NULL	8/27/2013	12:59:22 PM	Abc	Bcd
3	8/27/2013	11:38:05 PM	123456789v	gff	NULL	NULL	8/27/2013	11:37:35 PM	Abc	Bcd
4	8/27/2013	11:38:37 PM	123456789v	gff	NULL	NULL	8/27/2013	11:37:35 PM	ch	Bcd
5	8/27/2013	11:38:39 PM	123456789v	gff	NULL	NULL	8/27/2013	11:37:35 PM	ch	Dcb
6	8/27/2013	11:38:39 PM	123456789v	gff	NULL	NULL	8/27/2013	11:37:35 PM	ch	ch1
7	8/27/2013	11:51:26 PM	123456789v	gff	NULL	NULL	NULL	11:50:47 PM	Abc	Bcd

7.6 Bite Stock

Record_No_BY	Date	Time	Byte_Item_Id	Product_Type	Product_Name	Stock_Quantit...	Price_Of_A_Pla...
1	8/14/2013	3:02:24 PM	CD	Cba	Dcb	1	120
2	8/16/2013		123	Abc	Bcd	1	150
3	8/27/2013		AB	ch	ch1	3	100
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

7.7 Catering_temp

Catering_Id	Service_Type	Product_Type	Product_Name	Quantity	Unit_Price	Amount	Net_Price	Discount	Advance	Balance
1000	Food	Menu 1	Food	33.00	675.00	22275.00	36417.00	1000.00	10000.00	26417.00
1000	Food	Other Dishes	cashewnut and ...	33.00	75.00	2475.00	36417.00	1000.00	10000.00	26417.00
1000	Food	Other Dishes	cashewnut and ...	33.00	45.00	1485.00	36417.00	1000.00	10000.00	26417.00
1000	Bar	White	Viski	1.00	1800.00	1800.00	36417.00	1000.00	10000.00	26417.00
1001	Food	Menu 2	Food	11.00	750.00	8250.00	9600.00	1.00	9600.00	0.00
1001	Bar	Arrack	V.S.O.A	1.00	1350.00	1350.00	9600.00	1.00	9600.00	0.00
1001	Bar	Arrack	V.S.O.A	1.00	1350.00	1350.00	9600.00	1.00	9600.00	0.00

7.8 Catering

Catering_Id	Customer_Id	Bar_Order_Id	Byte_Order_Id	Other_Dish_Id	Date	Time	Function_Type	Need_Date	Need_Time	Destination_1
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

7.9 Catering bill

Service_Type	Product_Type	Product_Name	Quantity	Unit_Price	Amount
Food	Menu 1	Food	2.00	675.00	1350.00
Food	Other Dishes	cashewnut and ...	2.00	100.00	200.00
Food	Other Dishes	byte	2.00	90.00	180.00
Bar	White	Viski	1.00	1800.00	1800.00
Byte	Cba	Dcb	1.00	120.00	120.00
NULL	NULL	NULL	NULL	NULL	NULL

7.10 Customer

Record_No	Date	Time	Customer_Id	Customer_Na...	Customer_Ad...	Customer_Ad...	Customer_Ad...	Customer_Tp
16	8/13/2013	11:42:14 AM	098765432v	suba	fgh	hjjjjh	gyy	761389451
15	8/13/2013	10:17:37 AM	123432156v	nimal	fggh	bnm	bnm	1234567890
14	Date	Time	123456789v	gff	jmn	jhj	yuiu	1456781235
1	8/22/2013	3:24 PM	654382356v	h	h	e		345673222
13	8/13/2013	11:42:14 AM	765432189v	suba	fgh	hjjjjh	gyy	76138945

7.11 Daily invoice

Daily_Invoice_...	Sales_Id	Date	Time	Type	Quantity	Net_Price	Discount	Given_Amount	Balance
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

7.12 Daily Lunch Sales

Record_No_DLS	Date	Time	Item_Id	Pruduct_Name	Stock	Price_Of_A_Pa...
1	8/10/2013	9:03:46 PM	E2	Egg	16	120
11	8/11/2013	12:54:36 PM	F1	Fish	20	140
14	8/15/2013	1:45:56 PM	C1	Chicken	2	240
15	8/11/2013	12:54:36 PM	V1	Vegetable	25	100
NULL	NULL	NULL	NULL	NULL	NULL	NULL

7.13 Hall booking

Record_Id	Date	Time	Order_Id	Customer_Id	Customer_Na...	Function_Type	Need_Date	Need_Time	Menu_Id	Menu_Price
9			10000	926543008v	Thushani Athth...	Wedding	8/22/2013	11:24:50 AM	menu2	75000
10			10001	926543008v	Thushani Athth...	Anniversary	8/22/2013	11:27:08 AM	menu3	190000
11			10002	926543008v	Thushani Athth...	Party	8/22/2013	11:30:57 AM	menu1	15525
12			10003	926543008v	Thushani Athth...	Party	8/22/2013	11:44:06 AM	menu3	95000
13			10004	926543008v	Thushani Athth...	Conference	8/22/2013	11:48:10 AM	menu3	95000
14			10005	926543008v	Thushani Athth...	Seminar	8/22/2013	12:13:04 PM	menu2	37500
15			10006	925151580v	Nilupuli Ranasi...	Anniversary	8/22/2013	2:15:51 PM	menu2	75000
16			10007	925151580v	nilu	Party	8/22/2013	2:18:53 PM	menu3	47500
17			10008	925151580v	Nilupuli Chandi...	Wedding	8/31/2013	2:01:08 PM	menu1	15525
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

7.14 Inventory

Order_Id	Date	Service_Type	Product_Type	Product_Name	Quantity	Unit_Price	Amount	Methord
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

7.15 Login Table

Department	UserName	Password
Barman	Amesh	123
Manager	lakmina	123
Receptionist	Nilupuli	789
s	s	s
NULL	*NULL*	*NULL*

7.16 Menu List

Record_No_ML	Date	Time	Menu_Id	Menu_Type	Menu_Items
	10/6/2013	6:55 PM			
1	8/15/2013	1:26 AM	menu1	welcome drink	fresh fruits
10	8/15/2013	1:26 AM	menu2	Rice	Plain Rice, Vege...
11	8/15/2013	1:29 AM	menu1	eee	ccc
12	8/15/2013	1:29 AM	menu2	Main Dishes	Potato Temper...
13	8/15/2013	1:29 AM	menu2	subsidiaries	Fish Cutlet,Pap...
14	8/15/2013	1:29 AM	menu2	Dessert	Watalappan,Fru...
15	8/15/2013	1:29 AM	menu2	ddd	fhfhfh
16	8/15/2013	1:29 AM	meu3	Salad	Mix Vegetable,...

7.17 Order1

Record_No_MPL	Date	Time	Menu_Id	Price
1	8/15/2013	1:00 AM	menu1	675
2	8/15/2013	1:01 AM	menu2	750
3	8/15/2013	1:01 AM	menu3	950
4	8/15/2013	1:01 AM	menu4	1200
5	8/15/2013	1:01 AM	menu5	1350
NULL	*NULL*	*NULL*	*NULL*	*NULL*

7.18 Ordered Menu

Order_Id	Customer_Id	Date	Time	Order_Type
1000	123456789v	10/8/2013	6:19:51 PM	Catering
1011	926543008v	8/26/2013	6:42:38 AM	Catering
1012		10/6/2013	8:59:13 AM	Catering
10007	925151580v	8/22/2013	2:18:53 PM	Hall Booking
10008	925151580v	8/28/2013	1:57:11 PM	Room Booking
NULL	*NULL*	*NULL*	*NULL*	*NULL*

7.19 Other Charges Hall

Record_No_OC	Other_Charge_...	Other_Charge_...	Other_Charge_...	Date	Time
1	1	hall charge	5000	8/15/2013	2:30 AM
2	2	music full band	2000	8/15/2013	2:30 AM
3	3	music 3 pcs	1000	8/15/2013	2:30 AM
4	4	DJ	500	8/15/2013	2:30 AM
5	5	chair covers & ...	75	8/15/2013	2:30 AM
6	6	light decorations	5000	8/15/2013	2:30 AM
NULL	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*

7.20 Other Dishes

Menu_Id	Date	Time	Need_Date	Need_Time	Other_Dishes	Menu_Price
NULL	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*

7.21 Room Booking

Record_No_RD	Room_No	Room_Status	Room_Price	Date	Time
1	20	a/c, normal	3500	8/28/2013	8:49 AM
2	22	a/c, normal	3500	8/28/2013	8:49 AM
3	24	a/c, normal	3500	8/28/2013	8:49 AM
4	26	a/c, mini luxuary	6000	8/28/2013	8:49 AM
5	28	A/C, normal	3500	8/28/2013	8:49 AM
NULL	NULL	NULL	NULL	NULL	NULL

7.22 Room Details

Record_No_R	Date	Time	Customer_Id	Customer_Na...	Order_Id	Room_Type	No_Of_People	Check_In_Date	Check_Out_Date	Service_ID
1	8/28/2013	1:47:32 PM	925151580v	Nilupuli Chandi...	10008	20	2	8/28/2013	8/28/2013	1
2	8/28/2013	1:49:16 PM	925151580v	Nilupuli Chandi...	10008	20	2	8/28/2013	8/28/2013	xs
3	8/28/2013	1:52:27 PM	925151580v	Nilupuli Chandi...	10008	20	2	8/28/2013	8/28/2013	jjf
4	8/28/2013	1:54:57 PM	925151580v	Nilupuli Chandi...	10008	20	2	8/28/2013	8/28/2013	22
5	8/28/2013	1:57:11 PM	925151580v	Nilupuli Chandi...	10008	20	2	8/28/2013	8/28/2013	hfh
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

8.Testing

Testing was done in several steps of the software development process. The very 1st testing scenario done was done by the developer. The designing of H.M.S. was divided into several departments depending on the work covered on them. And in a single department 1 or two developers worked to achieve the individual goals in them which supports to the goals of the overall system. These departments were divided as much as independent from each other and that made a huge impact on the testing process to make it easy. Several testing scenarios were conducted within each department by the working developers in them.

In the testing process of the system we followed many types of testing methods that are followed when a system is implementing.

With the completion of the final system we did the System Testing, the testing that estimated the ultimate state of the product as a whole. Here also we did the modifications as necessary to reduce the bugs and errors.

As the final Testing of the System we implemented, we have to do the Acceptance Testing after handing over the system to the client. In this system also we will have to be able to do the modifications as required by the client accordingly. The maintenance also will require all these test results in future to make the project a complete one

9. Evaluation

9.1Strength

- The system provide login authorization feature to prevent unauthorized user login into the Hotel Management System
- All the date that inserted to database are fully validated to ensure the accuracy of information H.M.S. is user-friendly and easy to use.
- Database backup and security.

9.2 Weaknesses

- Requirement of a web application

10. Lessons Learned

There were lots of lessons that we learned from throughout the time we spent for developing this project. Most importantly we learned to work according to a time schedule and how to manage the time to take the maximum benefit of the time we had. We faced lot of inconveniences while engaging with the project development and those incidents gave us the experiences of facing the problems and finding the appropriate solutions as necessary. This group project also let us to get the experience of working as a team and how to do it with a team spirit. We were molded with the advices given by the lecturers with every comment they gave.

There were lot of experiments that we did to get on to a better out come and those things impacted us to collect new things in to our knowledge as well as to our lives.

11. Future Work

The work in the future that we have to engage should be done with lessons we learned from the project work that we have done so far. The steps that we have to be taken should be included with a higher standard than the tasks we did in the past. We should be able to reduce the errors that we have identified all through the time we spent for the project and that should be included with the experiences we gathered but with a high standard.

Developing a web application will be scheduled for future. A web application could provide the ease of making reservations for the guest at any time. We would hope to begin it as soon as possible. The next step of the project is, handing over the system to the client. After that we have to consider about the maintenance part of the project accordingly. Here we should still be able to do the modifications as necessary and as well as according to the requirement of the client. All these steps also should be included with all the experiences and requirements gathered from the time we spent for the development of the system.

12. Conclusion

The project we developed is The Management System for the Lotus Hotel. The System gives solutions for most of the problems that we have identified in the Hotel currently. All most all the tasks were handled manually by the management and the employees work at the Hotel. The

System that we are going to introduce will address those problems accordingly

The ability to search about specific information or detail before and after doing a hotel reservation or to hotel's room in easy way with customizing guest detail and checking the validity of details are some purpose to build system which makes requirements possible with easy and fast way. Hotel Management system is built to find suitable solution for reservation and customization of rooms, guests, payments, administration and services. This system deals with the database as an end back which is based on SQL server and its interface based on visual studio c#. The interface aims to make reservation and using other tools easy to every one without needing to learn how to use.

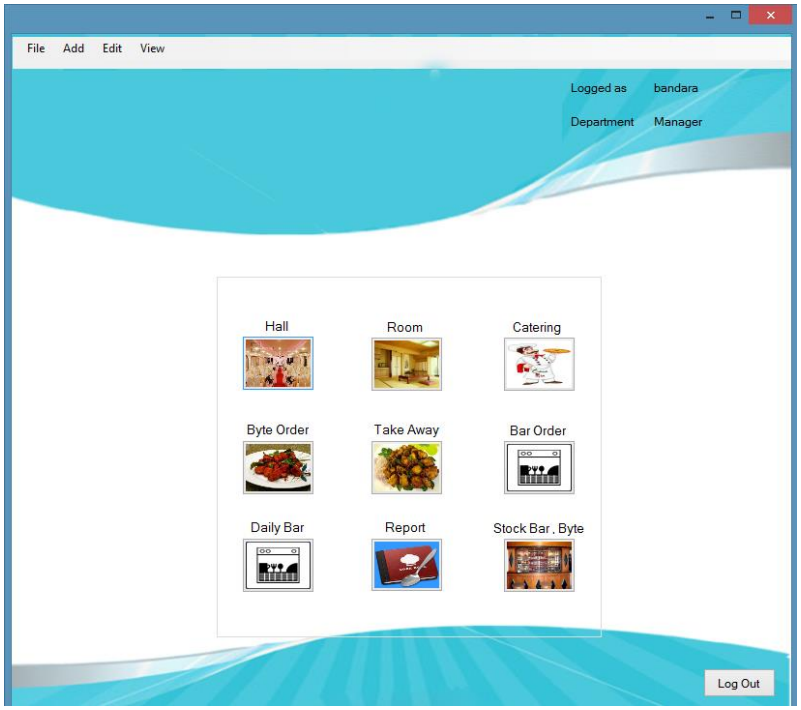
The System that is going to be handed over to the client will address most of the problems in the Hotel currently. The tasks that now are carried out manually will be able to do with the system in more easy way. The data that are now kept in large physical files will be stored in the centralized database of the system. That will reduce the damages that can be happened to the data unexpectedly. The calculations that are done manually will able to be done through the system to get more accurate results. Presently the information regarding the employees are

handled manually. The system that is introducing give the ability to handle information in a more convenient and accurate way. The details of the daily income, sales and purchasing goods can be inserted to the system easily and retrieve them whenever the user requires. Records and details of the inventory are also handled through the proposed system with giving the solutions for the problems that arise with the current manual procedure of the Hotel.

Those features of the introducing system will call upon the problems that we have encountered from the current system that is prevailing in the Hotel now to make the tasks done at the Hotel comfortably and much more efficiently

13.External Interfaces and coding's

13.1 Main Interface



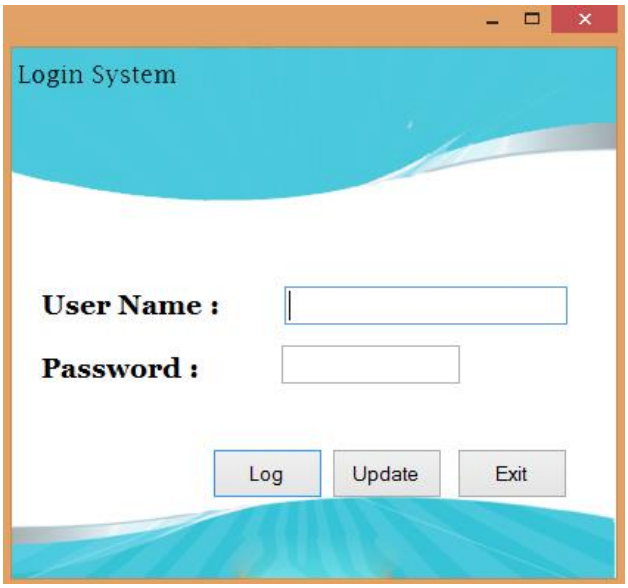

```
public partial class MainInterface : Form
   {

      string a;
      public MainInterface()
      {


         InitializeComponent();
      }

      DBOperationLoging mr = new DBOperationLoging();

      public MainInterface(string lg)
      {
         InitializeComponent();
         a = lg;
         label1.Text = a;
      }


      string l;
```

```
string dep;


private void MainInterface_Load(object sender, EventArgs e)
{
//    buttonBR.Visible = false;
//    buttonBY.Visible = false;
//    buttonCAT.Visible = false;
//    button1UP1.Visible = false;
//    buttonHB.Visible = false;
//    buttonREP.Visible = false;
//    buttonRM.Visible = false;
//    buttonTA.Visible = false;
//    buttonDB.Visible = false;


    label2nm.Text = a;

    l = label2nm.Text.ToString();


    SqlDataReader ri = mr.Log(l);
    if (ri.Read())
    {
        string dp = ri[0].ToString();
        dep = dp;

        labelDep.Text = dep;

    }

     if (labelDep.Text == "Barman")
    {

        buttonBR.Enabled = false;
        buttonBY.Enabled = false;
        buttonCAT.Enabled = false;
        buttonHB.Enabled = false;
        buttonRM.Enabled = false;
        buttonTA.Enabled = false;
    }


     if (labelDep.Text == "Receptionist")
     {
         buttonBR.Enabled = false;
         buttonBY.Enabled = false;
         buttonCAT.Enabled = false;
         button1UP1.Enabled = false;
         buttonHB.Enabled = false;
```

```
        buttonRM.Enabled = false;
        buttonDB.Enabled = false;
    }



}

private void toolStrip1_ItemClicked(object sender, ToolStripItemClickedEventArgs e)
{

}

private void buttonHB_Click(object sender, EventArgs e)
{
   Hall_Booking tr=new Hall_Booking();
   tr.Show();

}

private void groupBox1_Enter(object sender, EventArgs e)
{

}

private void buttonRM_Click(object sender, EventArgs e)
{
   Room_Booking rd = new Room_Booking();
   rd.Show();
}

private void buttonCAT_Click(object sender, EventArgs e)
{
   Catering1 ct = new Catering1();
   ct.Show();
}

private void buttonBR_Click(object sender, EventArgs e)
{
   Bar_Order bo = new Bar_Order();
   bo.Show();
}

private void buttonBY_Click(object sender, EventArgs e)
{
   Bite_Order byo = new Bite_Order();
   byo.Show();
}

private void button6_Click(object sender, EventArgs e)
```

```
        {
            TakeAwey2 tw = new TakeAwey2();
            tw.Show();
        }

        private void buttonREP_Click(object sender, EventArgs e)
        {


        }

        private void exitToolStripMenuItem_Click(object sender, EventArgs e)
        {
            this.Close();
        }
        bool logut;
        private void buttonExit1_Click(object sender, EventArgs e)
        {
            logut = true;
            Loging loo = new Loging();
            loo.ShowDialog();
            this.Close();
        }

        private void buttonUp_Click(object sender, EventArgs e)
        {
            Daily_Bar_1 db = new Daily_Bar_1();
            db.ShowDialog();
        }

        private void label2nm_Click(object sender, EventArgs e)
        {

        }

        private void button1UP1_Click(object sender, EventArgs e)
        {
            Update up = new Update();
            up.Show();
        }
    private void MainInterface_FormClosed(object sender, FormClosedEventArgs e)
        {
            if(logut==true)
            { Application.Exit(); }
        }

        private void panel1_Paint(object sender, PaintEventArgs e)
          }
      }
    }
```

13.2 Login interface



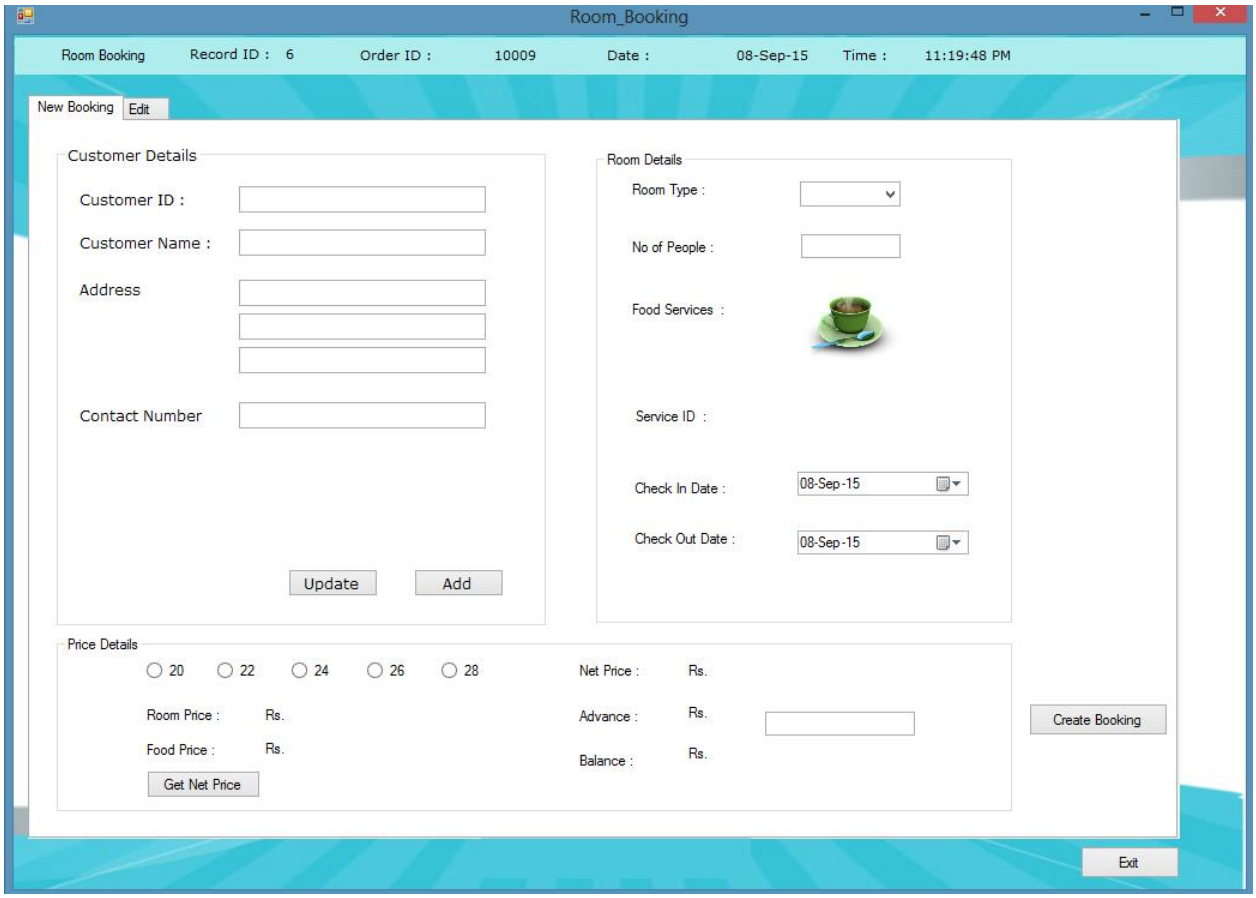
Login interface is used to login to the system using username and password for three different users

```
namespace Hotel_Ordering_System
{
   public partial class Loging : Form
   {
      public Loging()
      {
         InitializeComponent();
      }
      DBOperationLoging log = new DBOperationLoging();

      private void buttonOk_Click(object sender, EventArgs e)
      {

         string US =textBoxUsName. Text;
         string PS = textBoxPass.Text;


         SqlDataReader da = log.Login(US, PS);

         if (US == "")
         {
```

```
            labelUSname1.Text = "*";
            labelUSName2.Text = "Enter the User Name
  }
         else if (PS == "")
         {
            labelUSname1.Text = "";
            labelPass.Text = "*";
            labelUSName2.Text = "Enter Password";

         }
         else if (US == "" && PS == "")
         { labelUSName2.Text = "Enter the User Name and Password"; }
   else
         {
            labelUSname1.Text = "";
            //labelUSName2.Text = "";
            labelPass.Text = ""
          try
            {
               if (da.Read())
                  if (US == (da["UserName"].ToString()) && PS ==
(da["Password"].ToString()))
                  {
                     this.Hide();

                     MainInterface hm = new MainInterface(US);
                     hm.ShowDialog();


                  }
                  else

                   //(US != (da["UserName"].ToString()) && PS ==
(da["Password"].ToString()))
                  {
                     labelUSName2.Text = "The User Name And the Password did not
match";
                  }
               //if (da.Read())
               //    if
               //    (US != (da["UserName"].ToString()) && PS !=
(da["Password"].ToString()))
               //    {
               //        labelUSName2.Text = "The User Name And the Password did not
match";
               //    }
```

```
            //    else
            //    { }

        }
        catch (Exception et)
        {
            MessageBox.Show(et.Message.ToString());
        }
        textBoxUsName.Clear();
        textBoxPass.Clear();
    }

}

private void Loging_Load(object sender, EventArgs e)
{

}

private void button1_Click(object sender, EventArgs e)
{
    UpdateLogin otc = new UpdateLogin();
    otc.Show();
}

private void buttonExit_Click(object sender, EventArgs e)
{
    Application.Exit();
}


}
}
```

13.3 Room Booking




```
{
   public partial class Room_Booking : Form
   {
      public Room_Booking()
      {
         InitializeComponent();



   }



      public static Label si = new Label();
      public static Label fp = new Label();
```

```
DBOperationRoomBooking d3 = new DBOperationRoomBooking();
DBOperationCatering cT = new DBOperationCatering();




private void pictureBoxRoomSer_Click(object sender, EventArgs e)
{
   RoomBooking_Services r1 = new RoomBooking_Services();
   r1.ShowDialog();
}

//float price;
private void buttonAddRoom_Click(object sender, EventArgs e)
{

}

// int reRmD;
private void buttonRemvRoom_Click(object sender, EventArgs e)
{

}

private void buttonShw_Click(object sender, EventArgs e)
{

   dataGridViewRD.Visible = true;
   DataSet drn = d3.showRoomDetail();
   dataGridViewRD.DataSource = drn.Tables["Room_Details"];
}

private void pictureBoxRoomSer_Click_1(object sender, EventArgs e)
{
   RoomBooking_Services r1 = new RoomBooking_Services();
   r1.ShowDialog();
}

private void buttonNetPrice_Click(object sender, EventArgs e)
{
   float roomPrice = 0;
   float foodPrice = 0;
   float netPrice;

   roomPrice = float.Parse(labelRoomPrice.Text);
```

```
    foodPrice = float.Parse(labelRoomServicePrice.Text);

    netPrice = roomPrice + foodPrice;

    labelTotalRoomPrice.Text = netPrice.ToString();
}

int HOid;
private void Room_Booking_Load(object sender, EventArgs e)
{
    si = labelServiceId;
    fp = labelRoomServicePrice;

    labelDate.Text = DateTime.Now.ToShortDateString();
    //labelTime.Text = DateTime.Now.ToShortTimeString();

    try
    {
        string oI = d3.Order_Id();
        if (oI.Equals("") || oI == null)
        {
            HOid = 000010000;
        }
        else
        {
            HOid = Convert.ToInt32(oI);
            HOid = HOid + 1;

        }
        labelOrderID.Text = Convert.ToString(HOid);
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }

    try
    {
        string rI = d3.Record_Id();
        if (rI.Equals("") || rI == null)
        {
            HOid = 00001;
        }
        else
        {
            HOid = Convert.ToInt32(rI);
```

```
            HOid = HOid + 1;

        }
        labelRecordID.Text = Convert.ToString(HOid);
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}


int custp;
int people;
float rmPri;
float foodPri;
float netPri;
float adva;
float bal;
int orderId;
int recordId;


private void buttonCreateRoomBooking_Click_1(object sender, EventArgs e)
{
    //try
    //{
    custp = int.Parse(textBoxRmCusTp.Text);

    people = int.Parse(textBoxNoOfPersonsRoom.Text);
    rmPri = float.Parse(labelRoomPrice.Text);
    foodPri = float.Parse(labelRoomServicePrice.Text);
    netPri = float.Parse(labelTotalRoomPrice.Text);
    adva = float.Parse(textBoxAdvance.Text);
    bal = float.Parse(labelBalance.Text);
    orderId = int.Parse(labelOrderID.Text);
    recordId = int.Parse(labelRecordID.Text);


        d3.insertRoomBooking(recordId, labelDate.Text, labelTime.Text,
textBoxRmCusId.Text, textBoxRmCusName.Text, orderId, comboBoxRm.Text,
people, dateTimePickerCheckInDate.Text, dateTimePickerCheckOutDate.Text,
labelServiceId.Text, rmPri, foodPri, netPri, adva, bal);
        d3.insertRoomOrder(orderId, textBoxRmCusId.Text, labelDate.Text,
labelTime.Text, labelOrderType.Text);
        MessageBox.Show("New Booking is Created");
```

```
    //}
    //catch (Exception ex)
    //{
    //    MessageBox.Show(ex.Message);
    //}
}

private void radioButtonR1_CheckedChanged(object sender, EventArgs e)
{
    DataSet drpc = d3.showRoomPriceWithRadio1();
    labelRoomPrice.Text = drpc.Tables["Room_Details"].Rows[0][0].ToString();
}

private void radioButtonR2_CheckedChanged(object sender, EventArgs e)
{
    DataSet drpc = d3.showRoomPriceWithRadio2();
    labelRoomPrice.Text = drpc.Tables["Room_Details"].Rows[0][0].ToString();
}

private void radioButtonR3_CheckedChanged(object sender, EventArgs e)
{
    DataSet drpc = d3.showRoomPriceWithRadio3();
    labelRoomPrice.Text = drpc.Tables["Room_Details"].Rows[0][0].ToString();

}

private void radioButtonR4_CheckedChanged(object sender, EventArgs e)
{
    DataSet drpc = d3.showRoomPriceWithRadio4();
    labelRoomPrice.Text = drpc.Tables["Room_Details"].Rows[0][0].ToString();
}

private void radioButtonR5_CheckedChanged(object sender, EventArgs e)
{
    DataSet drpc = d3.showRoomPriceWithRadio5();
    labelRoomPrice.Text = drpc.Tables["Room_Details"].Rows[0][0].ToString();
}

float Rprice;
private void buttonAddRoom_Click_1(object sender, EventArgs e)
{
    try
    {

        Rprice = float.Parse(textBoxRoomPrice.Text);
```

```
            d3.insertRoomDetails(textBoxRoomNo.Text,        textBoxRoomStatus.Text,
Rprice, labelDate.Text, labelTime.Text);
            MessageBox.Show("Items are Added");
            DataSet dr = d3.showRoomDetail();
            dataGridViewRD.DataSource = dr.Tables["Room_Details"];

        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
        textBoxRoomNo.Clear();
        textBoxRoomStatus.Clear();
        textBoxRoomPrice.Clear();
    }

    private void buttonRemvRoom_Click_1(object sender, EventArgs e)
    {
        try
        {
            d3.removRmD(textBoxRoomNo.Text);
            MessageBox.Show("Record is Removed");

        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }

    float Rpr;
    private void buttonUpdateRoom_Click(object sender, EventArgs e)
    {
        Rpr = float.Parse(textBoxRoomPrice.Text);

        try
        {
            d3.updateRoomDetail(textBoxRoomStatus.Text,Rpr,
textBoxRoomNo.Text);
            MessageBox.Show("Record is Updated");
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
        textBoxRoomNo.Clear();
```

```
            textBoxRoomStatus.Clear();
            textBoxRoomPrice.Clear();


        }

        private void button2_Click(object sender, EventArgs e)
        {
            dataGridViewRD.Visible = true;
            DataSet drn = d3.showRoomDetail();
            dataGridViewRD.DataSource = drn.Tables["Room_Details"];
        }

        private void radioButtonR1_CheckedChanged_1(object sender, EventArgs e)
        {
            DataSet drpc = d3.showRoomPriceWithRadio1();
            labelRoomPrice.Text = drpc.Tables["Room_Details"].Rows[0][0].ToString();
        }

        private void radioButtonR2_CheckedChanged_1(object sender, EventArgs e)
        {
            DataSet drpc = d3.showRoomPriceWithRadio2();
            labelRoomPrice.Text = drpc.Tables["Room_Details"].Rows[0][0].ToString();
        }

        private void radioButtonR3_CheckedChanged_1(object sender, EventArgs e)
        {
            DataSet drpc = d3.showRoomPriceWithRadio3();
            labelRoomPrice.Text = drpc.Tables["Room_Details"].Rows[0][0].ToString();
        }

        private void showRoomPriceWithRadio4_CheckedChanged(object sender,
EventArgs e)
        {
            DataSet drpc = d3.showRoomPriceWithRadio4();
            labelRoomPrice.Text = drpc.Tables["Room_Details"].Rows[0][0].ToString();
        }

        private void showRoomPriceWithRadio5_CheckedChanged(object sender,
EventArgs e)
        {
            DataSet drpc = d3.showRoomPriceWithRadio5();
            labelRoomPrice.Text = drpc.Tables["Room_Details"].Rows[0][0].ToString();
        }

        float roomPrice = 0;
```

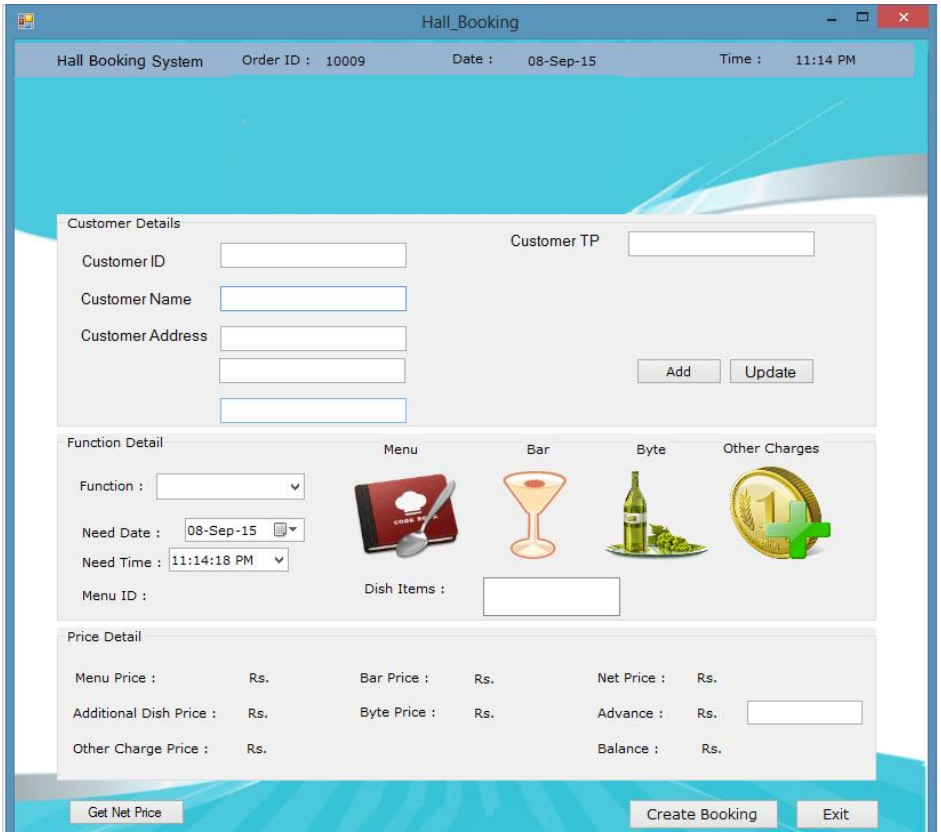
```
float foodPrice = 0;
float netPrice = 0;

private void buttonNetPrice_Click_1(object sender, EventArgs e)
{
   if (labelRoomPrice.Text == "")
   {
      labelRoomPrice.Text = ("0.00");
   }
   else if (labelRoomServicePrice.Text == "")
   {
      labelRoomServicePrice.Text = ("0.00");
   }
   roomPrice = float.Parse(labelRoomPrice.Text);
   foodPrice = float.Parse(labelRoomServicePrice.Text);

   netPrice = roomPrice + foodPrice;

   labelTotalRoomPrice.Text = netPrice.ToString();

}

private void textBoxAdvance_KeyPress(object sender, KeyPressEventArgs e)
{
   if (e.KeyChar == (char)Keys.Enter)
   {
      float netPri, advance, balance;


      netPri = float.Parse(labelTotalRoomPrice.Text);
      advance = float.Parse(textBoxAdvance.Text);




      balance = netPri - advance;

      labelBalance.Text = balance.ToString();
   }
}

private void pictureBoxRoomSer_Click_2(object sender, EventArgs e)
{
   RoomBooking_Services r1 = new RoomBooking_Services();
   r1.ShowDialog();
}
```

```
    private void textBoxRmCusName_MouseClick(object sender, MouseEventArgs
e)
    {
       try
       {

          SqlDataReader dr = cT.select(textBoxRmCusId.Text);
          if (dr.Read())
          {
             MessageBox.Show("Record exists in the database ", "Conformation",
MessageBoxButtons.OK, MessageBoxIcon.Information);
             textBoxRmCusId.Text = dr[3].ToString();
             textBoxRmCusName.Text = dr[4].ToString();
             textBoxRmAdd1.Text = dr[5].ToString();
             textBoxRmAdd2.Text = dr[6].ToString();
             textBoxRmAdd3.Text = dr[7].ToString();
             textBoxRmCusTp.Text = dr[8].ToString();


          }
       }
       catch (Exception ex)
       {
          MessageBox.Show(ex.Message);
       }
    }

    private void buttonUpdate_Click(object sender, EventArgs e)
    {
       try
       {


          int tp = int.Parse(textBoxRmCusTp.Text);
          Regex TP = new Regex(@"^[0-9]{10}$");
          {
             MessageBox.Show("Not a Valid Telephone Number");
          }


          cT.update_cat(textBoxRmCusId.Text,                textBoxRmAdd1.Text,
textBoxRmAdd2.Text, textBoxRmAdd3.Text, tp);
          MessageBox.Show("Record updated successfully");

       }
```

```
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void buttonAdd_Click(object sender, EventArgs e)
{
    try
    {

        string p = textBoxRmCusId.Text;
        string n = textBoxRmCusName.Text;
        string ad = textBoxRmAdd1.Text;
        string co = textBoxRmCusTp.Text;

        if (p == "")
        {


            labelStarId.Text = "*";
            labelMsg.Text = "Enter the Customer NIC";
        }
        else if (n == "")
        {
            labelStarId.Text = "";

            labelStarName.Text = "*";
            labelMsg.Text = "Enter the Customer Name";
        }
        else if (ad == "")
        {
            labelStarId.Text = "";
            labelStarName.Text = "";

            labelStarAdd1.Text = "*";
            labelStarAdd2.Text = "*";
            labelStarAdd3.Text = "*";
            labelMsg.Text = "Enter the Customer Address";
        }

        else if (co == "")
        {
            labelStarAdd1.Text = "";
            labelStarAdd2.Text = "";
            labelStarAdd3.Text = "";
```

```
                labelStarcN.Text = "*";
                labelMsg.Text = "Enter the  Contact Number";
            }

            else
            {
                int tp = int.Parse(textBoxRmCusTp.Text);


                cT.insert1(labelDate.Text, labelTime.Text, textBoxRmCusId.Text,
textBoxRmCusName.Text, textBoxRmAdd1.Text, textBoxRmAdd2.Text,
textBoxRmAdd3.Text, tp);
                MessageBox.Show("Record Added Successfully.");
                labelStarcN.Text = "";
                labelMsg.Text = "";
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }

    }

    private void timer1_Tick(object sender, EventArgs e)
    {
        DateTime t = DateTime.Now;
        labelTime.Text = t.ToLongTimeString();
    }

    private void groupBox1_Enter(object sender, EventArgs e)
    {

    }


  }
}
```

13.4 Hall Booking




```
public partial class Hall_Booking : Form
   {
      public Hall_Booking()
      {
         InitializeComponent();
      }

      public static Label l1 = new Label();
      public static Label l2 = new Label();
      public static Label l3 = new Label();
      public static Label l4 = new Label();

      public static ListBox l5 = new ListBox();
      public static bool byt_clic,bar_clic,menu_hall,add_dish =false;
```

```
//public static Label t1 = new Label();
public static Label t2 = new Label();
public static Label t3 = new Label();
//public static Label t4 = new Label();
//public static Label t5 = new Label();

public static ListBox k1 = new ListBox();
public static Label l6 = new Label();


DBOperationHallBooking d1 = new DBOperationHallBooking();
DBOperationCatering cot=new DBOperationCatering();


int custp;
float menupri;
float addish;
float och;
float net;
float adva;
float bal;
float bar_p;
float Bite_p;

int HOid;

private void buttonCreateBooking_Click(object sender, EventArgs e)
{

   //try
   //{

       custp = int.Parse(textBoxTP.Text);
       menupri = float.Parse(labelMenuPrice.Text);
       addish = float.Parse(labelAddDishPrice.Text);
       och = float.Parse(labelOtherCprice.Text);
       net = float.Parse(labelNetPrice.Text);
       adva = float.Parse(textBoxAddvance.Text);
       bar_p = float.Parse(labelBarPrice.Text);
       Bite_p = float.Parse(labelBitePrice.Text);

       bal = float.Parse(labelBalance.Text);
       HOid = int.Parse(LabelOID.Text);
```

```
            d1.insertHallBooking(labelDate.Text, labelTime.Text, HOid,
textBox1cid.Text, textBoxName.Text, comboBoxFunction.Text,
dateTimePickerHallBDate.Text, dateTimePickerHallBTime.Text, labelMenuId.Text,
menupri, och, addish, bar_p, Bite_p, net, adva, bal);
            //              d1.insertOrderHall(HOid, textBox1cid.Text,
dateTimePickerHallBDate.Text, dateTimePickerHallBTime.Text, labelOType.Text);
            d1.insertOrder(HOid, textBox1cid.Text, dateTimePickerHallBDate.Text,
dateTimePickerHallBTime.Text, labelOType.Text);
            //d1.insertOrderedMenuHall();
            MessageBox.Show("New Booking is Created");


          // }
           //catch (Exception ex)
           //{
           //    MessageBox.Show(ex.Message);
           //}
           //textBoxCustomerId.Clear();
           //textBoxCustomerName.Clear();
           //textBoxAdd1.Clear();
           //textBoxAdd2.Clear();
           //textBoxAdd3.Clear();
           //textBoxTP.Clear();


       }

       private void buttonCancelHall_Click(object sender, EventArgs e)
       {
           this.Close();
       }

       private void pictureMenu_Click(object sender, EventArgs e)
       {
           menu_hall = true;
           add_dish = true;

           CateringMenu f1 = new CateringMenu();
           f1.ShowDialog();
       }

       private void pictureOCharge_Click(object sender, EventArgs e)
       {
           Hall_Other_Charges f2 = new Hall_Other_Charges();
           f2.ShowDialog();
```

```
}

private void Hall_Booking_Load(object sender, EventArgs e)
{
   l1 = labelMenuPrice;
   l2 = labelMenuId;
   l3 = labelAddDishPrice;
   l4 = labelOtherCprice;

   l5 = listBoxDishI;
   l6 = labelAddDishPrice;


   t2 = labelBarPrice;
   t3 = labelBitePrice;

   k1 = listBoxDishI;



   labelDateH.Text = DateTime.Now.ToShortDateString();
   labelTimeH.Text = DateTime.Now.ToShortTimeString();


   //int HOid;
   try
   {

      string oI = d1.Order_Id();
      if (oI.Equals("") || oI == null)
      {
         HOid = 000010000;
      }
      else
      {
         HOid = Convert.ToInt32(oI);
         HOid = HOid + 1;

      }
      LabelOID.Text = Convert.ToString(HOid);
   }

   catch (Exception ex)
   {
      MessageBox.Show(ex.Message);
   }
```

```
}


private void buttonAdd_Click(object sender, EventArgs e)
{

   try
   {

      string p = textBox1cid.Text;
      string n = textBoxName.Text;
      string ad = textBoxAddress1.Text;
      string co = textBoxTP.Text;

      if (p == "")
      {
         label5n.Text = "*";
         labelms.Text = "Enter the Customer NIC";
      }
      else if (n == "")
      {
         label5n.Text = "";

         label21c.Text = "*";
         labelms.Text = "Enter the Customer Name";
      }
      else if (ad == "")
      {
         label5n.Text = "";
         label21c.Text = "";

         label5ca.Text = "*";
         label30ad.Text = "*";
         label21ad.Text = "*";
         labelms.Text = "Enter the Customer Address";
      }

      else if (co == "")
      {
         label5ca.Text = "";
         label30ad.Text = "";
         label21ad.Text = "";
         label4n.Text = "*";
```

```
            labelms.Text = "    Enter the  Contact Number";
        }

        else
        {



            int tp = int.Parse(textBoxTP.Text);


            cot.insert1(labelDateH.Text, labelTimeH.Text, textBox1cid.Text,
textBoxName.Text, textBoxAddress1.Text, textBoxAddress2.Text,
textBoxAddress3.Text, tp);
            MessageBox.Show("Record Added Successfully.");
            label4n.Text = "";
            labelms.Text = "";
        }


    }

    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void button1_Click(object sender, EventArgs e)
{
    try
    {


        int tp = int.Parse(textBoxTP.Text);
        Regex TP = new Regex(@"^[0-9]{10}$");
        {
            MessageBox.Show("Not a Valid Telephone Number");
        }


        cot.update_cat(textBox1cid.Text, textBoxAddress1.Text,
textBoxAddress2.Text, textBoxAddress3.Text, tp);
        MessageBox.Show("Record updated successfully");

    }
```

```
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }

    }

private void groupBox2_Enter(object sender, EventArgs e)
    {

    }

    private void groupBox3_Enter(object sender, EventArgs e)
    {

    }

    private void label25_Click(object sender, EventArgs e)
    {

    }


    private void label3_Click(object sender, EventArgs e)
    {

    }

    private void panel1_Paint(object sender, PaintEventArgs e)
    {

    }

    private void buttonGetNetP_Click(object sender, EventArgs e)
    {
        float menuPrice = 0;
        float ocPrice = 0;
        float addPrice = 0;
        float netPrice = 0;

        menuPrice = float.Parse(labelMenuPrice.Text);
        ocPrice = float.Parse(labelOtherCprice.Text);
        addPrice = float.Parse(labelAddDishPrice.Text);

        netPrice = menuPrice + ocPrice + addPrice;
```

```
        labelNetPrice.Text = netPrice.ToString();
    }



    private void label8_Click(object sender, EventArgs e)
    {

    }

    //private void textBoxName_TextChanged(object sender, EventArgs e)
    //{

    //}

    private void textBoxName_MouseClick_1(object sender, MouseEventArgs e)
    {
        try
        {

            SqlDataReader dr = cot.select(textBox1cid.Text);
            if (dr.Read())
            {
                MessageBox.Show("Record exists in the database ", "Conformation",
MessageBoxButtons.OK, MessageBoxIcon.Information);
                textBox1cid.Text = dr[3].ToString();
                textBoxName.Text = dr[4].ToString();
                textBoxAddress1.Text = dr[5].ToString();
                textBoxAddress2.Text = dr[6].ToString();
                textBoxAddress3.Text = dr[7].ToString();
                textBoxTP.Text = dr[8].ToString();


            }
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }

    private void buttonGetNetP_Click_1(object sender, EventArgs e)
    {
        try
        {
```

```
        float menuPrice = 0;
        float ocPrice = 0;
        float addPrice = 0;
        float barPrice = 0;
        float BitePrice = 0.00f;
        float netPrice = 0;

        if (labelMenuPrice.Text == "")
        {
            labelMenuPrice.Text = ("0.00");
        }
        else if (labelAddDishPrice.Text == "")
        {
            labelAddDishPrice.Text = ("0.00");
        }
        else if (labelOtherCprice.Text == "")
        {
            labelOtherCprice.Text = ("0.00");
        }
        else if (labelBarPrice.Text == "")
        {
            labelBarPrice.Text = ("0.00");
        }
        else if (labelBitePrice.Text == "")
        {
            labelBitePrice.Text = ("0.00");
        }

        menuPrice = float.Parse(labelMenuPrice.Text);
        ocPrice = float.Parse(labelOtherCprice.Text);
        addPrice = float.Parse(labelAddDishPrice.Text);
        barPrice = float.Parse(labelBarPrice.Text);
        BitePrice = float.Parse(labelBitePrice.Text);

        netPrice = menuPrice + ocPrice + addPrice + barPrice + BitePrice;

        labelNetPrice.Text = netPrice.ToString();
    }

    catch(Exception)
    {

    }
}
```

```
    private void CheckEnter1(object sender,
System.Windows.Forms.KeyPressEventArgs e)
    {


       if (e.KeyChar == (char)Keys.Enter)
       {
          float toNetPrice;
          float advance;
          float balance;

          toNetPrice = float.Parse(labelNetPrice.Text);
          advance = float.Parse(textBoxAddvance.Text);

          balance = toNetPrice - advance;
          labelBalance.Text = balance.ToString();
       }
    }

    private void groupBox3_Enter_1(object sender, EventArgs e)
    {

    }

    private void pictureBar_Click(object sender, EventArgs e)
    {
       string td = dateTimePickerHallBDate.Text;
       string td1 = dateTimePickerHallBTime.Text;
       string td2 = LabelOID.Text;
       string td3 = textBoxName.Text;

       bar_clic = true;

       CateringBar otc =new CateringBar(td,td1,td2,td3);
       otc.ShowDialog();
    }

    private void pictureBoxBite_Click(object sender, EventArgs e)
    {
       string td = dateTimePickerHallBDate.Text;
       string td1 = dateTimePickerHallBTime.Text;
       string td2 = LabelOID.Text;
       string td3 = textBoxName.Text;

       byt_clic = true;
```

```
        CateringBite otc = new CateringBite(td, td1, td2, td3);
        otc.Show();

    }

    private void groupBox2_Enter_1(object sender, EventArgs e)
    {

    }

    private void groupBox1_Enter(object sender, EventArgs e)
    {

    }

    private void Hall_Booking_FormClosing(object sender, FormClosingEventArgs
e)
    {
        byt_clic = false;
        bar_clic = false;
        menu_hall = false;
        add_dish = false;
    }


    }
    }
```

14.Sql Based Codes

14.1 Operation Login

```
{
    class DBOperationLoging
    {
        public SqlDataReader Login(string UN, string PS)
        {
            string sql = "SELECT * FROM Login_Table WHERE UserName='" +
UN + "' AND Password='" + PS + "' ";
            SqlCommand com = new SqlCommand(sql,
ConnectionManager.connection());
            SqlDataReader dr = com.ExecuteReader();

            return dr;

        }

        public SqlDataReader add(string DEP)
        {
            string sql = "SELECT Department FROM Login_Table WHERE
Department='" + DEP + "'";
            SqlCommand de = new SqlCommand(sql,
ConnectionManager.connection());
            SqlDataReader dp = de.ExecuteReader();
            return dp;
        }
        //public void update(string dep, string UN, string PS)
        //{
        //    {
        //        string sql = "UPDATE Login_Table SET UserName='" + UN
+ "',Password'" + PS + "' WHERE Department='" + dep + "'";
        //        SqlCommand com = new SqlCommand(sql,
ConnectionManager.connection());
        //        com.ExecuteNonQuery();
        //    }
        //}
        public void insert(string dep, string UN, string PS)
        {
```

```
            string sql = "INSERT INTO
Login_table(Department,UserName,Password)VALUES('" + dep + "','" + UN +
"','" + PS + "')";
            SqlCommand com = new SqlCommand(sql,
ConnectionManager.connection());
            com.ExecuteNonQuery();
        }

        public void update(string dep, string UN, string PS)
        {
            string sql = "UPDATE Login_table SET UserName='" + UN +
"',Password='" + PS + "' WHERE Department='" + dep + "'";
            SqlCommand com = new SqlCommand(sql,
ConnectionManager.connection());
            com.ExecuteNonQuery();
        }

        public SqlDataReader Log(string TN)
        {
            string pr = "SELECT * FROM Login_Table WHERE UserName='" +
TN + "'";
            SqlCommand com = new SqlCommand(pr,
ConnectionManager.connection());
            SqlDataReader pri = com.ExecuteReader();
            return pri;
        }
    }
}
```

14.2 Operation Hall Booking

```
    class DBOperationHallBooking
    {

        //-----------to insert data to menu_list, menu_price, customer,
other_charge, additional_dish, hall_booking, order, ordered_menu-------
-----------------------------------------------------------------------
        public void insertHallCustomer(string date, string time, string
cusId, string cusName, string cusAdd1, string cusAdd2, string cusAdd3,
int cusTp)
        {
            String sql = "INSERT INTO Customer
(Date,Time,Customer_Id,Customer_Name,Customer_Address_1,Customer_Addres
s_2,Customer_Address_3,Customer_Tp) VALUES ('" + date + "','" + time +
"','" + cusId + "','" + cusName + "','" + cusAdd1 + "','" + cusAdd2 +
"','" + cusAdd3 + "','" + cusTp + "')";
            SqlCommand com = new SqlCommand(sql,
ConnectionManager.connection());
            com.ExecuteNonQuery();
```

```
        }

        public void insertMenuList(string rn_ml, string date, string
time, string menu_id, string menu_type, string menu_items)
        {
            String sql = "INSERT INTO Menu_List
(Record_No_ML,Date,Time,Menu_Id,Menu_Type,Menu_Items) VALUES ('" +
rn_ml + "','" + date + "','" + time + "','" + menu_id + "','" +
menu_type + "','" + menu_items + "')";
            SqlCommand com = new SqlCommand(sql,
ConnectionManager.connection());
            com.ExecuteNonQuery();
        }

        public void insertOtherDish(string o_dish_id, string date,
string time, string type, float price)
        {
            String sql = "INSERT INTO Other_Dishes (Other_Dish_Id,
Date, Time, Type, Price) VALUES ('" + o_dish_id + "','" + date + "','"
+ time + "','" + type + "','" + price + "')";
            SqlCommand com = new SqlCommand(sql,
ConnectionManager.connection());
            com.ExecuteNonQuery();
        }

        public void insermenuprice(string rn_mpl, string date, string
time, string m_id, float price)
        {
            String sql = "INSERT INTO Menu_Price_List (Record_No_MPL,
Date, Time, Menu_Id, Price) VALUES ('" + rn_mpl + "','" + date + "','"
+ time + "','" + m_id + "','" + price + "')";
            SqlCommand com = new SqlCommand(sql,
ConnectionManager.connection());
            com.ExecuteNonQuery();
        }

        public void insertOtherCharge(string oc_id, string oc_type,
float oc_price, string date, string time)
        {
            String sql = "INSERT INTO Other_Charges_Hall
(Other_Charge_Id, Other_Charge_Type, Other_Charge_Price, Date, Time)
VALUES ('" + oc_id + "','" + oc_type + "','" + oc_price + "','" + date
+ "','" + time + "')";
            SqlCommand com = new SqlCommand(sql,
ConnectionManager.connection());
            com.ExecuteNonQuery();
        }

        public void insertHallBooking(string date, string time, int
order_id, string cus_id, string cus_name, string func_type, string
need_date, string need_time, string menu_id, float menu_price, float
```

```
oc_price, float add_d_price, float net_price, float bar_pri, float
Bite_pri, float advance, float balance)
        {
            String sql = "INSERT INTO Hall_Booking (Date, Time,
Order_Id, Customer_Id, Customer_Name, Function_Type, Need_Date,
Need_Time, Menu_Id, Menu_Price, Other_Charges_Price,
Additional_Dish_Price, Bar_Price, Bite_Price, Net_Price, Advance,
Balance) VALUES ('" + date + "','" + time + "','" + order_id + "','" +
cus_id + "','" + cus_name + "','" + func_type + "','" + need_date +
"','" + need_time + "','" + menu_id + "','" + menu_price + "','" +
oc_price + "','" + add_d_price + "','" + bar_pri + "','" + Bite_pri +
"','" + net_price + "','" + advance + "','" + balance + "')";
            SqlCommand com = new SqlCommand(sql,
ConnectionManager.connection());
            com.ExecuteNonQuery();
        }


        public void insertOrder(int oi, string cid, string dt, string
tm, string otyp)
        {
            String sq = "INSERT INTO
Order1(Order_Id,Customer_Id,Date,Time,Order_Type) VALUES ('" + oi +
"','" + cid + "','" + dt + "','" + tm + "','" + otyp + "')";
            SqlCommand CO = new SqlCommand(sq,
ConnectionManager.connection());
            CO.ExecuteNonQuery();
        }
        public void insertOrderedMenuHall(string menu_id, string date,
string time, string need_date, string need_time, string od, string
menu_pri)
        {
            String sql = "INSERT INTO Ordered_Menu (Menu_Id, Date,
Time, Need_Date, Need_Time,Other_Dishes, Menu_Price) VALUES ('" +
menu_id + "','" + date + "','" + time + "','" + need_date + "','" +
need_time + "','" + od + "','" + menu_pri + "',)";
            SqlCommand com = new SqlCommand(sql,
ConnectionManager.connection());
            com.ExecuteNonQuery();
        }


        //----------to add otherdish table to datagridview-------------
------------------------------
        public DataSet otherDishTbl()
        {
            DataSet ds = new DataSet();
            string sql = "SELECT Other_Dish_Id,Date,Time,Type,Price
FROM Other_Dishes";
            SqlDataAdapter da = new SqlDataAdapter(sql,
ConnectionManager.connection());
```

```
            da.Fill(ds, "Other_Dishes");
            return ds;
        }

        //----------to add menu price to label-------------------------
--------------------------------
        public DataSet addMenuPrice1()
        {
            DataSet dp = new DataSet();
            string sql = "SELECT Price FROM Menu_Price_List WHERE
menu_Id='menu1'";
            SqlDataAdapter da = new SqlDataAdapter(sql,
ConnectionManager.connection());
            da.Fill(dp, "Menu_Price_List");
            return dp;
        }

        public DataSet addMenuPrice2()
        {
            DataSet dp = new DataSet();
            string sql = "SELECT Price FROM Menu_Price_List WHERE
menu_Id='menu2'";
            SqlDataAdapter da = new SqlDataAdapter(sql,
ConnectionManager.connection());
            da.Fill(dp, "Menu_Price_List");
            return dp;
        }

        public DataSet addMenuPrice3()
        {
            DataSet dp = new DataSet();
            string sql = "SELECT Price FROM Menu_Price_List WHERE
menu_Id='menu3'";
            SqlDataAdapter da = new SqlDataAdapter(sql,
ConnectionManager.connection());
            da.Fill(dp, "Menu_Price_List");
            return dp;
        }

        public DataSet addMenuPrice4()
        {
            DataSet dp = new DataSet();
            string sql = "SELECT Price FROM Menu_Price_List WHERE
menu_Id='menu4'";
            SqlDataAdapter da = new SqlDataAdapter(sql,
ConnectionManager.connection());
            da.Fill(dp, "Menu_Price_List");
            return dp;
        }

        public DataSet addMenuPrice5()
```

```
        {
            DataSet dp = new DataSet();
            string sql = "SELECT Price FROM Menu_Price_List WHERE
menu_Id='menu5'";
            SqlDataAdapter da = new SqlDataAdapter(sql,
ConnectionManager.connection());
            da.Fill(dp, "Menu_Price_List");
            return dp;
        }


        // to add menu_id to label-------------------------------------
----------------------------

        public DataSet addMenuId1()
        {
            DataSet dd = new DataSet();
            string sql = "SELECT menu_Id FROM Menu_Price_List WHERE
Menu_Id='menu1'";
            SqlDataAdapter da = new SqlDataAdapter(sql,
ConnectionManager.connection());
            da.Fill(dd, "Menu_Price_List");
            return dd;
        }

        public DataSet addMenuId2()
        {
            DataSet dd = new DataSet();
            string sql = "SELECT menu_Id FROM Menu_Price_List WHERE
Menu_Id='menu2'";
            SqlDataAdapter da = new SqlDataAdapter(sql,
ConnectionManager.connection());
            da.Fill(dd, "Menu_Price_List");
            return dd;
        }

        public DataSet addMenuId3()
        {
            DataSet dd = new DataSet();
            string sql = "SELECT menu_Id FROM Menu_Price_List WHERE
Menu_Id='menu3'";
            SqlDataAdapter da = new SqlDataAdapter(sql,
ConnectionManager.connection());
            da.Fill(dd, "Menu_Price_List");
            return dd;
        }

        public DataSet addMenuId4()
        {
            DataSet dd = new DataSet();
```

```
            string sql = "SELECT menu_Id FROM Menu_Price_List WHERE
Menu_Id='menu4'";
            SqlDataAdapter da = new SqlDataAdapter(sql,
ConnectionManager.connection());
            da.Fill(dd, "Menu_Price_List");
            return dd;
        }

        public DataSet addMenuId5()
        {
            DataSet dd = new DataSet();
            string sql = "SELECT menu_Id FROM Menu_Price_List WHERE
Menu_Id='menu5'";
            SqlDataAdapter da = new SqlDataAdapter(sql,
ConnectionManager.connection());
            da.Fill(dd, "Menu_Price_List");
            return dd;
        }

        //-----------to show menu items in the datagridview------------
---------------------------
        public DataSet showMenu1()
        {
            DataSet dm = new DataSet();
            string sql = "SELECT Record_No_ML,Menu_Type,Menu_Items FROM
Menu_List WHERE menu_Id='menu1'";
            SqlDataAdapter da = new SqlDataAdapter(sql,
ConnectionManager.connection());
            da.Fill(dm, "Menu_List");
            return dm;
        }


        public DataSet showMenu2()
        {
            DataSet dm = new DataSet();
            string sql = "SELECT Record_No_ML,Menu_Type,Menu_Items FROM
Menu_List WHERE menu_Id='menu2'";
            SqlDataAdapter da = new SqlDataAdapter(sql,
ConnectionManager.connection());
            da.Fill(dm, "Menu_List");
            return dm;
        }


        public DataSet showMenu3()
        {
            DataSet dm = new DataSet();
            string sql = "SELECT Record_No_ML,Menu_Type,Menu_Items FROM
Menu_List WHERE menu_Id='menu3'";
```

```
            SqlDataAdapter da = new SqlDataAdapter(sql,
ConnectionManager.connection());
            da.Fill(dm, "Menu_List");
            return dm;
        }

        public DataSet showMenu4()
        {
            DataSet dm = new DataSet();
            string sql = "SELECT Record_No_ML,Menu_Type,Menu_Items FROM
Menu_List WHERE menu_Id='menu4'";
            SqlDataAdapter da = new SqlDataAdapter(sql,
ConnectionManager.connection());
            da.Fill(dm, "Menu_List");
            return dm;
        }

        public DataSet showMenu5()
        {
            DataSet dm = new DataSet();
            string sql = "SELECT Record_No_ML,Menu_Type,Menu_Items FROM
Menu_List WHERE menu_Id='menu5'";
            SqlDataAdapter da = new SqlDataAdapter(sql,
ConnectionManager.connection());
            da.Fill(dm, "Menu_List");
            return dm;
        }

        //to show other charges in datagridview-----------------------
-------------------------------------------------------------
        public DataSet showOtherCharge()
        {
            DataSet dm = new DataSet();
            string sql = "SELECT * FROM Other_Charges_Hall";
            SqlDataAdapter da = new SqlDataAdapter(sql,
ConnectionManager.connection());
            da.Fill(dm, "Other_Charges_Hall");
            return dm;
        }
        //show full menu table in datagridview------------------------
-------------------------------------------------------------
        public DataSet showMenuAll()
        {
            DataSet du = new DataSet();
            string sql = "SELECT * FROM Menu_List";
            SqlDataAdapter da = new SqlDataAdapter(sql,
ConnectionManager.connection());
            da.Fill(du, "Menu_List");
            return du;
        }
        //update menu
```

```
        public void updateMenu(string mid, string mtype, string mitems,
int Rno)
        {
            string sql = "UPDATE Menu_List SET Menu_Id='" + mid +
"',Menu_Type='" + mtype + "',Menu_Items='"+mitems+"' WHERE
Record_No_ML='" + Rno + "'";
            SqlCommand com = new SqlCommand(sql,
ConnectionManager.connection());
            com.ExecuteNonQuery();
        }
        //update menu_price
        public void updateMenuPrice(string mpid, float mpri, int Rn)
        {
            string sql = "UPDATE Menu_Price_List SET Menu_Id='" + mpid
+ "',Price='"+mpri+"'  WHERE Record_No_MPL='" + Rn + "'";
            SqlCommand com = new SqlCommand(sql,
ConnectionManager.connection());
            com.ExecuteNonQuery();
        }

        //show full menu table in datagridview-------------------------
------------------------------------------------------------------
        public DataSet showMenuPriAll()
        {
            DataSet dh = new DataSet();
            string sql = "SELECT * FROM Menu_Price_List";
            SqlDataAdapter da = new SqlDataAdapter(sql,
ConnectionManager.connection());
            da.Fill(dh, "Menu_Price_List");
            return dh;
        }
        public void update_cat(string no, string ca1, string ca2,
string ca3, int ctp)
        {
            string sql = "UPDATE Customer SET Customer_Address_1='" +
ca1 + "',Customer_Address_2='" + ca2 + "',Customer_Address_3='" + ca3 +
"',Customer_Tp='" + ctp + "' WHERE Customer_Id='" + no + "'";
            SqlCommand cot = new SqlCommand(sql,
ConnectionManager.connection());
            cot.ExecuteNonQuery();
        }
        public SqlDataReader select_cus(string no)
        {
            string sql = "SELECT*FROM Customer WHERE Customer_Id='" +
no + "'";
            SqlCommand com = new SqlCommand(sql,
ConnectionManager.connection());
            SqlDataReader dr = com.ExecuteReader();
            return dr;
        }
```

```
        string no;
        public string Order_Id()
        {

            string ato = "SELECT MAX(Order_Id) FROM Hall_Booking";
            SqlCommand com = new SqlCommand(ato,
ConnectionManager.connection());
            SqlDataReader dr = com.ExecuteReader();
            while (dr.Read() == true)
            {
                no = dr[0].ToString();
            }
            return no;
        }

          public void inner(string no, string ca1, string ca2, string
ca3, int ctp)
         {
             string sql = " SELECT Hall_Booking.Order_Id,
Customers.CustomerName, Orders.OrderDate FROM Orders INNER JOIN
Customers ON Orders.CustomerID=Customers.CustomerID;";
             SqlCommand cot = new SqlCommand(sql,
ConnectionManager.connection());
             cot.ExecuteNonQuery();
         }

          public SqlDataReader Name(string TN)
          {
              string pr = "SELECT * FROM Other_Dishes WHERE Type='" + TN
+ "'";
              SqlCommand com = new SqlCommand(pr,
ConnectionManager.connection());
              SqlDataReader pri = com.ExecuteReader();
              return pri;
          }
          public SqlDataReader id(string odi)
          {
              string pr = "SELECT Other_Dish_Id,Date,Time,Type,Price
FROM Other_Dishes WHERE Other_Dish_Id='" + odi + "'";
              SqlCommand com = new SqlCommand(pr,
ConnectionManager.connection());
              SqlDataReader pri = com.ExecuteReader();
              return pri;
          }
    }
}
```

14.3 Operation Bar Order

```
{
    class DBOperationBarOrder
    {
        //-----------------Get Price From Bar Stock Table-------------
-----
        public SqlDataReader price(string TN)
        {
            string pr = "SELECT * FROM Bar_Stock WHERE Product_Name='"
+ TN + "'";
            SqlCommand com = new SqlCommand(pr,
ConnectionManager.connection());
            SqlDataReader pri = com.ExecuteReader();
            return pri;
        }
        //-----------------Insert Bar Stock Table---------------------
------
        public void insert(string Date, string Time, string BID, string
Type, string Name, int StoQty,string cap, float POB)
        {
            string st = "INSERT INTO
Bar_Stock(Date,Time,Bar_Item_Id,Product_Type,Product_Name,Stock_Quantit
y,Capasity,Price_Of_25ml)VALUES('" + Date + "','" + Time + "','" + BID
+ "','" + Type + "','" + Name + "','" + StoQty + "','" + cap + "','" +
POB + "')";
            SqlCommand dbl = new SqlCommand(st,
ConnectionManager.connection());
            dbl.ExecuteNonQuery();
        }
        //-----------------Insert Bar Order Table---------------------
------
        public void insert_Or(string Date, string Time,string
CID,string CNm,string oi,string NeeDat,string neeTim, string Type,
string Name, int StoQty, float NP,float adv, float Bal)
        {
            string st = "INSERT INTO
Bar_Order(Date,Time,Customer_Id,Customer_Name,Order_Id,Need_Date,Need_T
ime,Product_Type,Product_Name,Quantity,Net_Price,Advance,Balance)
VALUES('" + Date + "','" + Time + "','" + CID + "','" + CNm + "','" +
oi + "','" + NeeDat + "','" + neeTim + "','" + Type + "','" + Name +
"','" + StoQty + "','" + NP + "','" + adv + "','" + Bal + "')";

            SqlCommand dbl = new SqlCommand(st,
ConnectionManager.connection());
            dbl.ExecuteNonQuery();
        }
        //-----------------Date Set-----------------------------------
-------
        public DataSet all()
```

```
        {
            DataSet sa = new DataSet();
            string sql = "SELECT * FROM Bar_Stock";
            SqlDataAdapter da = new SqlDataAdapter(sql,
ConnectionManager.connection());
            da.Fill(sa, "Bar_Stock");
            return sa;
        }
        //------------------After Order Bar Stock Update---------------
-------
        public void updatest(string nm,string stoc)
        {
            string sto = "UPDATE Bar_Stock SET Stock_Quantity='" + stoc
+ "' WHERE Product_Name='" + nm + "'";
            SqlCommand com = new SqlCommand(sto,
ConnectionManager.connection());
            com.ExecuteNonQuery();
        }
        //------------------Update Bar Stock---------------------------
-------
        public void update(string dte, string tme, float pri, int qt,
string id)
        {
            string st = "UPDATE Bar_Stock SET Date= '" + dte +
"',Time='" + tme + "',Price_Of_25ml='" + pri + "',Stock_Quantity='" +
qt + "' WHERE Bar_Item_Id='" + id + "'";
            SqlCommand com = new SqlCommand(st,
ConnectionManager.connection());
            com.ExecuteNonQuery();
        }
        //--------------------Select-------------------------------------
-------
        public SqlDataReader select(string id)
        {
            string sql = "SELECT * FROM Bar_Stock WHERE Product_Name='"
+ id + "'";
            SqlCommand com = new SqlCommand(sql,
ConnectionManager.connection());
            SqlDataReader dr = com.ExecuteReader();
            return dr;
        }
        //---------------------Delete-------------------------------------
--------
        public void delete(string nid)
        {
            //string sql = "DELETE FROM Bar_Stock WHERE Record_No_BS=
'" + pid + "'AND Date='" + nid + "' AND Time='" + ty + "' AND
Product_Type='" + nm + "'AND Product_Name='" + pn + "'AND
Stock_Quantity='" + st + "'AND Capasity='" + ca + "'AND
Price_Of_25ml='" + pr + "'";
```

```
            string sql = "DELETE  FROM Bar_Stock WHERE Product_Name='"
+ nid + "'";
            SqlCommand com = new SqlCommand(sql,
ConnectionManager.connection());
            com.ExecuteNonQuery();
        }
        //---------------------Order Id---------------------------------
--------
        string no;
        public string Order_Id()
        {

            string ato = "SELECT MAX(Order_Id) FROM Bar_Order";
            SqlCommand com = new SqlCommand(ato,
ConnectionManager.connection());
            SqlDataReader dr = com.ExecuteReader();
            while (dr.Read() == true)
            {
                no = dr[0].ToString();
            }
            return no;
        }



        //-----------------Get Price From Bar Stock Table--------------
-----
        public SqlDataReader qty_descris(string TN,string tt)
        {
            string pr = "SELECT * FROM Bar_Stock WHERE Product_Name='"
+ TN + "'AND Product_Type='" + tt + "'";
            SqlCommand com = new SqlCommand(pr,
ConnectionManager.connection());
            SqlDataReader pri = com.ExecuteReader();
            return pri;
        }

    }
}
```

14.4 RoomBooking

```
{
    class DBOperationRoomBooking
    {
       //to insert room services to RoomBooking_Services table---------
-------------------------------------------------------------
        public void insertRoomBookingServices(int rn_rbs, string
ser_id, string ser_items, float price)
        {
            String sql = "INSERT INTO RoomBooking_Services
(Record_No_RBS, Service_Id, Service_Items, price) VALUES ('" + rn_rbs +
"','" + ser_id + "','" + ser_items + "','" + price + "')";
            SqlCommand com = new SqlCommand(sql,
ConnectionManager.connection());
            com.ExecuteNonQuery();
        }



//to show RoomBooking_Services table in datagridview-------------------
-----------------------------------------------------

        public DataSet showService()
        {
            DataSet dr = new DataSet();
            string sql = "SELECT * FROM RoomBooking_Services";
            SqlDataAdapter da = new SqlDataAdapter(sql,
ConnectionManager.connection());
            da.Fill(dr,"RoomBooking_Services");
            return dr;
        }

//to remove row of RoomBooking_Services table
        public void removSer(int rno)
        {
            string sql = "DELETE FROM RoomBooking_Services WHERE
Record_No_RBS='" + rno + "'";
            SqlCommand com = new SqlCommand(sql,
ConnectionManager.connection());
            com.ExecuteNonQuery();
        }

//to remove row of room_details
        public void removRmD(string rno)
        {
            string sql = "DELETE FROM Room_Details WHERE Room_No='" +
rno + "'";
            SqlCommand com = new SqlCommand(sql,
ConnectionManager.connection());
            com.ExecuteNonQuery();
```

```
        }


//to update RoomBooking_Services table

        public void updateSer(string sr_id, string sr_itm, float pri,
int Rno)
        {
            string sql = "UPDATE RoomBooking_Services SET Service_Id='"
+ sr_id + "', Service_Items='" + sr_itm + "',price='"+pri+"' WHERE
Record_No_RBS='" + Rno + "'";
            SqlCommand com = new SqlCommand(sql,
ConnectionManager.connection());
            com.ExecuteNonQuery();
        }
//to update Room_Details table
        public void updateRoomDetail(string rm_sta, float rm_pri,
string Rno)
        {
            string sql = "UPDATE Room_Details SET Room_Status='" +
rm_sta + "', Room_Price='" + rm_pri + "' WHERE Room_No='" + Rno + "'";
            SqlCommand com = new SqlCommand(sql,
ConnectionManager.connection());
            com.ExecuteNonQuery();
        }

//to add services to checkbox

        public DataSet showBreak()
        {
            DataSet dr = new DataSet();
            string sql = "SELECT Record_No_RBS, Service_Items,price
FROM RoomBooking_Services WHERE Service_Id='breakfast'";
            SqlDataAdapter da = new SqlDataAdapter(sql,
ConnectionManager.connection());
            da.Fill(dr, "RoomBooking_Services");
            return dr;
        }

        public DataSet showLunch()
        {
            DataSet dr = new DataSet();
            string sql = "SELECT  Record_No_RBS,Service_Items,price
FROM RoomBooking_Services WHERE Service_Id='lunch'";
            SqlDataAdapter da = new SqlDataAdapter(sql,
ConnectionManager.connection());
            da.Fill(dr, "RoomBooking_Services");
            return dr;
        }

        public DataSet showDinner()
```

```
        {
            DataSet dr = new DataSet();
            string sql = "SELECT  Record_No_RBS,Service_Items,price
FROM RoomBooking_Services WHERE Service_Id='dinner'";
            SqlDataAdapter da = new SqlDataAdapter(sql,
ConnectionManager.connection());
            da.Fill(dr, "RoomBooking_Services");
            return dr;
        }

//to add service record no to label
        public DataSet showBRe()
        {
            DataSet dL = new DataSet();
            string sql = "SELECT Record_No_RBS FROM
RoomBooking_Services ";
            SqlDataAdapter da = new SqlDataAdapter(sql,
ConnectionManager.connection());
            da.Fill(dL, "RoomBooking_Services");
            return dL;
        }

        public DataSet showLun()
        {
            DataSet dL = new DataSet();
            string sql = "SELECT Record_No_RBS FROM
RoomBooking_Services ";
            SqlDataAdapter da = new SqlDataAdapter(sql,
ConnectionManager.connection());
            da.Fill(dL, "RoomBooking_Services");
            return dL;
        }


//----------------------insert room details----------------------------
------------------

        public void insertRoomDetails(string rm_no, string rm_status,
float price, string date, string time)
        {
            String sql = "INSERT INTO Room_Details (Room_No,
Room_Status, Room_Price, Date, Time) VALUES ('" + rm_no + "','" +
rm_status + "','" + price + "','" + date + "','" + time + "')";
            SqlCommand com = new SqlCommand(sql,
ConnectionManager.connection());
            com.ExecuteNonQuery();
        }

        //to show room details---------------------------------------
-----------------------
```

```
        public DataSet showRoomDetail()
        {
            DataSet drn = new DataSet();
            string sql = "SELECT * FROM Room_Details";
            SqlDataAdapter da = new SqlDataAdapter(sql,
ConnectionManager.connection());
            da.Fill(drn, "Room_Details");
            return drn;
        }

        //to add room_no to combobox---------------------------------
        public DataSet showRoomNoCombo()
        {
            DataSet drnc = new DataSet();
            string sql = "SELECT Room_No FROM Room_Details";
            SqlDataAdapter da = new SqlDataAdapter(sql,
ConnectionManager.connection());
            da.Fill(drnc, "Room_Details");
            return drnc;
        }

        //add room_price to label when RADIO BUTTON select-------------
--------
        public DataSet showRoomPriceWithRadio1()
        {
            DataSet drpc = new DataSet();
            string sql = "SELECT Room_Price FROM Room_Details WHERE
Room_No='20'";
            SqlDataAdapter da = new SqlDataAdapter(sql,
ConnectionManager.connection());
            da.Fill(drpc, "Room_Details");
            return drpc;
        }

        public DataSet showRoomPriceWithRadio2()
        {
            DataSet drpc = new DataSet();
            string sql = "SELECT Room_Price FROM Room_Details WHERE
Room_No='22'";
            SqlDataAdapter da = new SqlDataAdapter(sql,
ConnectionManager.connection());
            da.Fill(drpc, "Room_Details");
            return drpc;
        }

        public DataSet showRoomPriceWithRadio3()
        {
            DataSet drpc = new DataSet();
            string sql = "SELECT Room_Price FROM Room_Details WHERE
Room_No='24'";
```

```
            SqlDataAdapter da = new SqlDataAdapter(sql,
ConnectionManager.connection());
            da.Fill(drpc, "Room_Details");
            return drpc;
        }

        public DataSet showRoomPriceWithRadio4()
        {
            DataSet drpc = new DataSet();
            string sql = "SELECT Room_Price FROM Room_Details WHERE
Room_No='26'";
            SqlDataAdapter da = new SqlDataAdapter(sql,
ConnectionManager.connection());
            da.Fill(drpc, "Room_Details");
            return drpc;
        }

        public DataSet showRoomPriceWithRadio5()
        {
            DataSet drpc = new DataSet();
            string sql = "SELECT Room_Price FROM Room_Details WHERE
Room_No='28'";
            SqlDataAdapter da = new SqlDataAdapter(sql,
ConnectionManager.connection());
            da.Fill(drpc, "Room_Details");
            return drpc;
        }

//remove room details-----------------------------------------------------
---------------------

        public void removRmDetail(int rno)
        {
            string sql = "DELETE FROM Room_Details WHERE
Record_No_RD='" + rno + "'";
            SqlCommand com = new SqlCommand(sql,
ConnectionManager.connection());
            com.ExecuteNonQuery();
        }

//----------------------insert room booking form details to Customer
tbl,Room_Booking tbl and Order tbl------
        public void insertRoomCustomer(string date, string time, string
cusId, string cusName, string cusAdd1, string cusAdd2, string cusAdd3,
int cusTp)
        {
            String sql = "INSERT INTO Customer
(Date,Time,Customer_Id,Customer_Name,Customer_Address_1,Customer_Addres
s_2,Customer_Address_3,Customer_Tp) VALUES ('" + date + "','" + time +
"','" + cusId + "','" + cusName + "','" + cusAdd1 + "','" + cusAdd2 +
"','" + cusAdd3 + "','" + cusTp + "')";
```

```
            SqlCommand com = new SqlCommand(sql,
ConnectionManager.connection());
            com.ExecuteNonQuery();
        }

        public void insertRoomBooking(int reId, string date, string
time, string cusId, string cusName, int orderId, string room_type, int
people, string checkindate, string checkOutdate, string ser_id, float
roomPrice, float foodPri, float netPrice, float adva, float bal)
        {
            String sql = "INSERT INTO Room_Booking ( Record_No_R,
Date,Time,Customer_Id,Customer_Name,Order_Id, Room_Type, No_Of_People,
Check_In_Date, Check_Out_Date, Service_ID, Room_Price,Food_Price,
Net_Price, Advance, Balance) VALUES ('" + reId + "','" + date + "','" +
time + "','" + cusId + "','" + cusName + "','" + orderId + "','" +
room_type + "','" + people + "','" + checkindate + "','" + checkOutdate
+ "','" + ser_id + "','" + roomPrice + "','" + foodPri + "','" +
netPrice + "','" + adva + "','" + bal + "')";
            SqlCommand com = new SqlCommand(sql,
ConnectionManager.connection());
            com.ExecuteNonQuery();
        }

        public void insertRoomOrder(int ROid, string cus_id,string
date, string time, string orderType)
        {
            String sql = "INSERT INTO Order1 (Order_Id, Customer_Id,
Date, Time, Order_Type) VALUES ('" + ROid + "','" + cus_id + "','" +
date + "','" + time + "','" + orderType + "')";
            SqlCommand com = new SqlCommand(sql,
ConnectionManager.connection());
            com.ExecuteNonQuery();
        }

        string no;
        string noR;
        public string Record_Id()
        {

            string ato = "SELECT MAX(Record_No_R) FROM Room_Booking";
            SqlCommand com = new SqlCommand(ato,
ConnectionManager.connection());
            SqlDataReader dr = com.ExecuteReader();
            while (dr.Read() == true)
            {
                noR = dr[0].ToString();
            }
            return noR;
        }

        public string Order_Id()
```

```
        {

            string ato = "SELECT MAX(Order_Id) FROM Hall_Booking";
            SqlCommand com = new SqlCommand(ato,
ConnectionManager.connection());
            SqlDataReader dr = com.ExecuteReader();
            while (dr.Read() == true)
            {
                no = dr[0].ToString();
            }
            return no;
        }
    }
}
```

14.5 OperationTakeAway

```
{
    class DBOperationTakeAway1
    {
        public void update_dish(string Date, string Time, string ItemId, string
productName, int st, float price)
        {
            string sql = "UPDATE Daily_Lunch_Sales SET Date='" + Date + "',Time='" +
Time + "',Stock='" + st + "',Price_Of_A_Packet='" + price + "'WHERE Item_Id='" +
ItemId + "' AND Pruduct_Name='" + productName + "' ";
            SqlCommand food = new SqlCommand(sql, ConnectionManager.connection());
            food.ExecuteNonQuery();
        }

        public SqlDataReader select(string id)
        {
            string sql = "SELECT*FROM Daily_Lunch_Sales WHERE Pruduct_Name='" + id +
"'";
            SqlCommand com = new SqlCommand(sql, ConnectionManager.connection());
            SqlDataReader dr = com.ExecuteReader();
            return dr;
        }

        public SqlDataReader select(float C1, float F1, float E1, float V1)
        {
            string sql = "SELECT*FROM Daily_Lunch_Sales WHERE
Pruduct_Name(C1,F1,E1,V1)VALUES('" + C1 + "','" + F1 + "','" + E1 + "','" + V1 + "')";
            SqlCommand pcom = new SqlCommand(sql, ConnectionManager.connection());
            SqlDataReader dr = pcom.ExecuteReader();
            return dr;
        }
        public DataSet all()
        {
            DataSet food = new DataSet();
            string sql = "SELECT * FROM Daily_Lunch_Sales";
            SqlDataAdapter di = new SqlDataAdapter(sql,
ConnectionManager.connection());
            di.Fill(food, "Daily_Lunch_Sales");
            return food;
        }
```

```
        public void insert(string Date, string Time, string Sales_Id,string ProName,
int Sto, float PriceOfPacket)
        {
            string foods = "INSERT INTO
Inventry(Date,Time,Sales_Id,Item_Name,Quantity,Net_Price) VALUES('" + Date + "','" +
Time + "','"+Sales_Id+"','" + ProName + "','" + Sto + "','" + PriceOfPacket + "')";
            SqlCommand db = new SqlCommand(foods, ConnectionManager.connection());
            db.ExecuteNonQuery();
        }

        string no;
        public string takeAway_Id()
        {

            string ato = "SELECT MAX(Sales_Id) FROM Inventry";
            SqlCommand com = new SqlCommand(ato, ConnectionManager.connection());
            SqlDataReader dr = com.ExecuteReader();
            while (dr.Read() == true)
            {
                no = dr[0].ToString();
            }
            return no;
        }
    }
}
```

15. Other Requirements

15.1 Performance Requirements

Performance requirements define acceptable response times for system functionality. Although the system is developed suiting for the least system performances, the performance of the system will highly depend on the performance of the hardware and software components of the installing computer. When consider about the timing relationships of the system the load time for user interface screens shall take no longer than two seconds. It makes fast access to system functions. The log in information shall be verified within five seconds causes' efficiency of the system. Returning query results within five seconds makes search function more accurate.

15.2 Safety

There are several user levels in hotel management system, Access to the various subsystems will be protected by a user log in screen that requires a user name and password. This gives different views and accessible functions of user levels through the system. Maintaining backups ensure the system database security. System can be restoring in any case of emergency.

15.3 Security

Customer Service Representatives and Managers and owner will be able to log in to the Hotel Management System. Customer Service Representatives will have access to the Reservation/Booking and subsystems. Managers will have access to the Management subsystem as well as the Reservation/Booking subsystems. Manager has the maximum privilege to all subsystems. Access to the various subsystems will be protected by a user log in screen that requires a user name and password.

15 Software Quality Attributes

- Availability: - The system shall be available during normal hotel operating hours
- Correctness: - extent to which program satisfies specifications, fulfills user's mission objectives
- Efficiency: - How much less number of resources and time are required to achieve a particular task through the system.
- Flexibility: - Ability to add new features to the system and handle them conveniently.
- Integrity: - How the system would insecure the information in the system and how it avoids the data losses. Referential integrity in database tables and interfaces
- Maintainability: - How easy is to keep the system as it is and correct defects with making changes.
- Portability: - The Hotel Management System shall run in any Microsoft Windows environment
- Reliability: - Specify the factors required to establish the required reliability of the software system at time of delivery. Mean time between failures and mean time to recovery
- Reusability: - What is the ability to use the available components of the system in other systems as well.
- Testability: - Effort needed to test to ensure performs as intended
- Usability: - How easily a person can be taken the benefits of the system and the user friendliness.

- Robustness: – Strength of the system to handle system functions accurately and maintain the database without facing to unexpected failures
- Maintainability: – What design, coding standards must be adhered to exclusions created

16. References

Books:

[1]Ian Sommerville, *Software engineering*, 9th edition, Pearson education, 2011.

[2]Rajib Mall, *Fundamentals of Software Engineering***,** 2nd edition, PHI Learning Pvt. Ltd, 2004.

[3]Ragu Ramakrishnan, Johnes Gehrke, *Database Management Systems*, 3rd edition, McGraw-Hill

Education, 2003.

[4]Gerald W. Latin, *Modern hotel management*, W.H.Freeman, 2011.

World Wide Web:

[5]"Hotel Management Case Study", March.6, 2010. [Online].Available: http://www.scribd.com/doc/27927992/Hotel-Management-Case-Study, [Accessed: June.28, 2014]

[6]"High-Level-Software Features", [Online].Available: http://www.high-level-software.com/features/, [Accessed: June.25, 2014]

[7]Fernandez & Yuan, X,(1999). *An analysis Pattern for Reservation and Use of Reusable Entities*. PloP 1999 conference, Retrieved from http://hillside.net/plop/plop99/proceedings/Fernandez2/reservanalysisPattern3.PDf

[8]Lauesen, S, (2003), *Task Descriptions as Functional Requirements*, IEEE Computer Society, Retrieved from http://www.itu.dk/~slauesen/Papers/IEEEtasks.pdf

[9]Louw,D,(2006, may 10).*Description with UML for a Hotel Reservation System*. Retrieved from