

HOTEL MANAGEMENT SYSTEM

Enrollment No. -101347

Name of Student - PRIYANKAR JAIN

Name of supervisor(s) - Dr. NITIN



MAY - 2014

Project report submitted in partial fulfilment of the requirement for the degree of

Bachelor of Technology

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY,

WAKNAGHAT

CERTIFICATE

This is to certify that the work titled “**HOTEL MANAGEMENT SYSTEM**” submitted by “**PRIYANKAR JAIN**” in partial fulfillment for the award of degree of B.Tech of Jaypee University of Information Technology, Waknaghat has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma.

Signature of Supervisor

Name of Supervisor

Designation

Date

ACKNOWLEDGEMENT

I would like to express my greatest gratitude to my mentor **Dr.Nitin** who has helped and supported me throughout our project. I'm grateful to him for his continuous support for the project, from the initial advice in the early stages of conceptual inception and through ongoing advice to this day.

Signature of the student

Name of Student

Date

Contents

CERTIFICATE.....	II
ACKNOWLEDGEMENT	III
Abstract.....	VI
Chapter 1.....	1
Introduction & Conceptual Framework	1
1.1 Introduction.....	1
1.2 Concept of Hotel Management	2
1.2.1 Definition of Hotel:-.....	2
1.2.2 Tourism and Hotel Industry	2
1.3 Computerization of services provided by Hotels and Hotel Management.....	3
CHAPTER 2	5
Need, Objectives and Advantages	5
2.1 Need or rationale of the study	5
2.2 ADVANTAGES	7
2.2.1 Performance	7
2.2.2 Efficiency	7
2.2.3 Control	7
2.2.4 Security	7
3.1 JAVA	8
3.1.1 The JAVA Buzzwords	9
3.1.2 Simple	10
3.1.3 Object-Oriented.....	10
3.1.4 Robust	10
3.1.5 Multithreaded	10
3.1.6 Dynamic	11
3.2 The Three OOP Principles	11
3.2.1 Encapsulation	11
3.2.2 Inheritance.....	12
3.2.3 Polymorphism.....	12
3.3 FRONT END	12
3.4 BACK END	12
CHAPTER 4	14

CODING	14
4.1 hm1.java.....	14
4.2 hm2.java.....	43
Chapter 5.....	60
5.1 DATABASE DESIGN.....	60
5.1.1 ROOM BOOKING	60
5.1.2 ROOMS	60
5.1.3 BILL.....	61
5.1.4 RESTAURANT ITEMS	61
Chapter 6.....	62
6.1 PROJECT EXECUTION.....	62
6.1.1 WELCOME SCREEN	62
6.1.2 LOGIN SCREEN	63
6.1.3 Available Facilities Screen.....	65
6.1.4 Room Booking Screen	66
6.1.5 Check In:	68
6.1.6 Restaurant	69
6.1.7 CHECK OUT.....	70
CONCLUSION.....	71
REFERENCES	72

Abstract

Hotel management System helps you manage reservations, bookings, guests and agents. This has been made in a user friendly interface. The user can search for the vacant rooms in the hotel and book for it. Booking can only be done by registered guests or through agents. The rooms will be added or updated by the admin and the same would be available for the guests to book them. A user is added by the admin, and is given the access for any of the rights that include adding guests and agents, reservation, booking, billing, viewing reports, etc.

An admin can view the booking details, transactions and coordinate the activity with the agents. He can track the entire site activity. Reservations can also be made by the admin.

Signature of Student

Name

Date

Signature of Supervisor

Name:

Date

Chapter 1

Introduction & Conceptual Framework

1.1 Introduction

The project “Computerization of services in Hotels and Hotel Management” is designed once and can be updated many times, so the burden of application maintenance is short listed by all the general guidelines. The main task ahead is to see all the things whether they are running smoothly or not, this software will be capable of doing or performing following tasks:-

1. Reservation.
2. Registration.
3. Billing/Clearance of Guest Folio.
4. Checkout.

This software will be developed keeping in mind the various aspects of hotels in public sector as well as in private sector. This software is intended to be developed for Deluxe Class / Super-Deluxe Class Hotels. A Comparison will be done of Deluxe Class Hotels in public sector and in private sector and then the resultant / desired software will be developed.

MS-Access database management system will be used as “Backend” which will be capable of recording information and high level language “Java” will serve as “Frontend” for the software. The database will be capable of doing following operations:

1. Keeping records of various customers that visit in the hotel.
2. Keeping records of various items that are served to the customers
3. Keeping records of customer’s Check-in/Check-out timings.

1.2 Concept of Hotel Management

1.2.1 Definition of Hotel

Hotel or inn is defined by British laws “as a place where a bonafide traveler can receive food and shelter, provided he is in a position to pay for it and in a fit condition to be received”. A hotel must provided food shelter to traveler on payments and have the right to refuse it if the traveler is drunk, disorderly kept or is not in position to pay for the services.

The industry is also known as hospitality industry the main purpose of the industry is to provide the guest’s quality service and make him feel at home.

1.2.2 Tourism and Hotel Industry

Tourism is the fastest growing and second largest industry of the world and has tremendous potentiality for carrying foreign exchange. The number of the foreign tourists visiting India is merely 0.30% of the total world tourist traffic. Besides promoting international understanding it helps the national economy and raises the standards of living of the people.

It has been genuinely said if there are no hotels there is no tourism. The hotel industry is an essential part of tourism development and vice versa. The hotel industry depends upon the demand for different types of accommodations, availability of suitable enfranchise and services and the economy feasibility of the developments. The concept of modern hotel business in India started late compared to the bold business elsewhere in Europe and America. Every hotel small or large has is own special atmosphere and ambience.

Tourism is the way in which all of us can favor the mysteries, plunder the unknown depths, acquire understanding and experience the world in its fullness.

1.3 Computerization of services provided by Hotels and Hotel Management

In a hotel, there are many functions that are to be performed to manage the hotel. Some of the main operations are:

- Reservation.
- Registration.
- Billing/Clearance of Guest Folio.

Reservation can be made in many ways :-

- 1) Walk-in Reservation
- 2) Instant Reservation
- 3) Online Reservation
- 4) Telephonic Reservation

As cleared by its name, Online Reservation can only be made through computers. On the other hand, other types of reservations like walk-in, telephonic, instant etc. reservations can be done by both ways i.e. manually entering the data into registers and computerized. Entering data into registers is a very complicated task and also very time-consuming. When there is a need to search any particular record, one should have to go through all the records which is a time-consuming task. Also the registration process is done by the same way. And finally, the billing is done. This also repeats the same process. As we can see, all the tasks can be done manually but which consumes time and labour.

These all are the services that every hotel provides and should be made computerized. By making these all tasks computerized, all the work can be done with ease and this also saves time. One major advantage of making these all services computerized is that if there is any need to search for a particular record, this can be done with very ease because the user has to just enter the name of the customer. So by making all these services computerized, all the records are maintained in an organized manner and can be retrieved very easily when required.

Hotel management means to manage all the resources of a hotel. Hotel management includes four major sections:-

- Front Office
- House Keeping
- Food and Beverage Production
- Food and Beverage Service.

Benefits of making these services computerized:-

- Security
- A large amount of data can be stored on a single disk.
- Accuracy
- Less chances of data loss.

CHAPTER 2

Need, Objectives and Advantages

2.1 Need or rationale of the study

During the past several decades personnel function has been transformed from a relatively obscure record keeping staff to central and top level management function. There are many factors that have influenced this transformation like technological advances, professionalism, and general recognition of human beings as most important resources.

- A computer based management system is designed to handle all the primary information required to calculate monthly statements. Separate database is maintained to handle all the details required for the correct statement calculation and generation.
- This project intends to introduce more user friendliness in the various activities such as record updation, maintenance, and searching.
- The searching of record has been made quite simple as all the details of the customer can be obtained by simply keying in the identification of that customer.
- Similarly, record maintenance and updation can also be accomplished by using the identification of the customer with all the details being automatically generated. These details are also being promptly automatically updated in the master file thus keeping the record absolutely up-to-date.
- The entire information has maintained in the database or Files and whoever wants to retrieve can't retrieve, only authorization user can retrieve the necessary information which can be easily be accessible from the file

• The main objective of the entire activity is to automate the process of day to day activities of Hotel like:

1. Room activities,
2. Admission of a New Customer,
3. Assign a room according to customer's demand,
4. Checkout of a computer and releasing the room
5. Finally compute the bill etc.
6. Packages available.
7. Advance online bookings.
8. Online Cancellation.
9. List of Regular customers.

This project have some features:

1. System connectivity
2. No data duplication
3. No paper work required
4. Time efficient
5. Cost efficient
6. Automatic data validation
7. User friendly environment
8. Fast data insertion & deletion
9. Easy performance check

2.2 ADVANTAGES

This project is useful for the authorities which keep track of all the users registered in a particular state .The authority can add hotel packages, room details, availability of rooms, online booking etc.

The following steps that give the detailed information of the need of proposed system are:

2.2.1 Performance

During past several decades, the records are supposed to be manually handled for all activities. The manual handling of the record is time consuming and highly prone to error. To improve the performance of the Hotel Management System, the computerized system is to be undertaken. This project is fully computerized and user friendly even that any of the members can see the report and status of the company.

2.2.2 Efficiency

The basic need of this website is efficiency. The website should be efficient so that whenever a new user submits his/her details the website is updated automatically. This record will be useful for other users instantly.

2.2.3 Control

The complete control of the project is under the hands of authorized person who has the password to access this project and illegal access is not supposed to deal with. All the control is under the administrator and the other members have the rights to just see the records not to change any transaction or entry.

2.2.4 Security

Security is the main criteria for the proposed system. Since illegal access may corrupt the database. So security has to be given in this project.

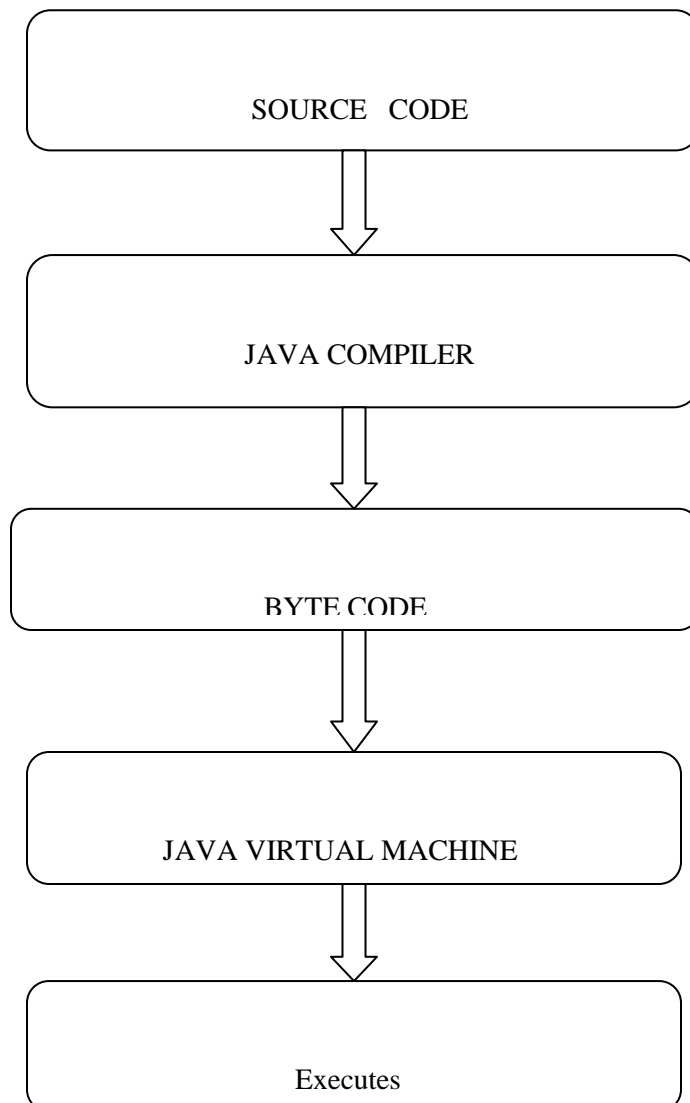
CHAPTER 3

3.1 JAVA

Java was conceived by James Gosling, Patrick Naughton, Chris Warth , Ed Frank ,and Mike Sheridan at Sun Microsystems,Inc. in 1991.It took 18 months to develop the first working version.This language was initially called “Oak” but was renamed “Java” in 1995.

Java has become the language of choice for implementing Internet-based applications and software for devices that communicate over a network.

PHASES OF JAVA



The key that allows Java to solve both the security and the portability is BYTECODE. Byte code is a highly optimized set of instructions designed to be executed by the Java run time system, which is called the Java Virtual Machine (JVM). That is in its standard form, the JVM is an interpreter for byte code.

Translating a Java program into byte code helps make it much easier to run a program in a wide variety of environments. The reason is straightforward : only the JVM needs to be implemented for each platform.

Java programs consist of pieces called classes. Classes include pieces called methods that perform tasks and return information when they complete them. Programmers can create each piece they need to form Java programs. However, most Java programmers take advantage of the rich collections of existing classes in the Java class libraries, which are also known as the Java APIs (Application Programming Interfaces). Thus, there are really two aspects to learning the Java world. The first is the Java language itself, so that you can program your own classes, and the second is the classes in the extensive Java class libraries.

3.1.1 The JAVA Buzzwords

Although the fundamental forces that necessitated the invention of Java are portability and security, other factors also played an important role in molding the final form of the language. The key considerations were summed up by the Java team in the following list of buzzwords:

- Simple
- Secure
- Portable
- Object-oriented
- Robust
- Multithreaded
- Dynamic

3.1.2 Simple

Java was designed to be easy for the professional programmer to learn and use effectively. If you already know the basic concepts of oops, learning Java will be even easier. Because Java inherits the C/C++ syntax and many of the oops features of C++, most programmers have little trouble learning Java. Also, some of the more confusing concepts from C++ are either left out of Java or implemented in a cleaner manner.

3.1.3 Object-Oriented

Although influenced by its predecessors, Java was not designed to be source code compatible. One outcome of this was a clean, usable, pragmatic approach to objects. Borrowing liberally from many seminal object-software environments of the last few decades, Java manages to strike a balance between the purist's "everything is an object" paradigm and the pragmatist's "stay out of my way" model. The object model in Java is simple and easy to extend, while simple types, such as integers, are kept as high-performance non-objects.

3.1.4 Robust

The multiplatformed environment of the Web places extraordinary demands on a program, because the program must execute reliably in a variety of systems. Thus, the ability to create robust programs was given a high priority in the design of Java. To gain reliability, Java restricts in a few key areas, to force you to find your mistakes early in program development. At the same time, Java frees you from having to worry about most of the programming errors. Because Java is a strictly typed language, it checks out your code at compile time. However, it also checks at run time. In fact, many hard-to-track-down bugs that often turn up in hard-to-reproduce run-time situations are simply impossible to create in Java. Knowing that what you have written will behave in a predictable way under diverse conditions is a key feature of Java.

3.1.5 Multithreaded

Java was designed to meet the real-world requirement of creating interactive, networked programs. To accomplish this, Java supports multithreaded programming, which allows you to write programs that do many things simultaneously. The Java run-time system comes with an elegant yet sophisticated solution for multiprocessor synchronization that enables you to construct smoothly running interactive systems. Java's easy-to-use approach to multithreading allows you to think about the specific behavior of your program, not the multitasking subsystem.

3.1.6 Dynamic

Java programs carry with them substantial amount of run time type information that is used to verify and resolve accesses to objects at run time. This makes it possible to dynamically link code in a safe and expedient manner. This is crucial to the robustness of the applet environment, in which small fragments of bytecode may be dynamically updated on a running system.

3.2 The Three OOP Principles

All object oriented programming languages provide mechanisms that help you implement the object oriented model. They are encapsulation, inheritance, and polymorphism.

3.2.1 Encapsulation

Encapsulation is the mechanism that binds together code and the data it manipulates, and keeps both safe from outside interference and misuse. One way to think about encapsulation is as a protective wrapper that prevents the code and data from being arbitrarily accessed by other code defined outside the wrapper. Access to the code and data inside the wrapper is tightly controlled through a well defined interface.

In java the basis of encapsulation is the class. A class defines the structure and behavior (data and code) that will be shared by a set of objects. Each object of a given class contains the structure and behavior by the class, as if it were stamped out by a mold in the shape of the class. For this reason, objects are sometimes referred to as instances of a class. Thus, a class is a logical construct; an object has physical reality.

When you create a class, you will specify the code and data that constitute that class. Collectively, these elements are called members of the class. Specifically, the data defined by the class are referred to as member variables. The code that operates on that data is referred to as member methods. In properly written java programs, the methods define how the member variables can be used. This means that the behavior and interface of a class are defined by the methods that operate on its instance data. Since the purpose of a class is to encapsulate complexity, there are mechanisms for hiding the complexity of the implementation inside the class. Each method or variable in a class may be marked private or public. The public interface of a class represent everything that external users of the class need to know. The private methods and data can only be accessed by code that is a member of the class. Therefore , any code that is not a member of the class cannot access a private method or variable.

3.2.2 Inheritance

Inheritance is the process by which one object acquires the properties of another object. This is important because it supports the concept of hierarchical classification.

3.2.3 Polymorphism

Polymorphism is a feature that allows one interface to be used for a general class of actions. The specific action is determined by the exact nature of the situation. Consider a stack (which is a last in, first out list). One stack is used for integer.

3.3 FRONT END

1. Software requirements :

- JDK 4.3.2
- ODBC Driver for MS-Access 2010
- Microsoft Access 2010
- Operating system- Windows 7/8/8.1

2. Hardware requirements

- RAM : Minimum 256 MB
- Hard Disk : Minimum 20 GB
- Intel i3 processor

3.4 BACK END

SQL is a standard language for accessing and manipulating databases.

➤ What is SQL?

- SQL stands for Structured Query Language.
- SQL lets you access and manipulate databases.
- SQL is an ANSI(American National Standard Institute) standard.

➤ Functions of SQL

- SQL can execute queries against a database
- SQL can retrieve data from a database
- SQL can insert records in a database
- SQL can update records in a database
- SQL can delete records from a database
- SQL can create new databases
- SQL can create new tables in a database
- SQL can create stored procedures in a database
- SQL can create views in a database
- SQL can set permissions on tables, procedures, and views

➤ Standards

Although SQL is an ANSI standard, there are many different versions of the SQL language. However, to be compliant with the ANSI standard, they all support at least the major commands (such as select, update, delete, insert, where) in a similar manner.

CHAPTER 4

CODING

4.1 hm1.java

```
import java.sql.*;
import java.applet.*;
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;
import java.util.*;
import java.sql.Statement;
import java.sql.Connection;
import java.sql.DriverManager;
//Front Page
class hm1 extends JFrame implements ActionListener
{
    JButton b1;
    hm1()
    {
        setSize(1024,700);
        setTitle("Hotel Swastik");
        setLayout(null);

        Font f1 =new Font("arial",Font.BOLD,20);
        java.util.Date date=new java.util.Date();
        JLabel ldate=new JLabel(String.valueOf(date));

        ldate.setFont(f1);
        ldate.setBounds(650,20,300,30);
        add(ldate);

        JLabel l2=new JLabel("****Hotel Swastik****");
        Font f =new Font("algerian",Font.BOLD,50);
        l2.setFont(f);
        l2.setBounds(330,60,650,50);
        l2.setForeground(Color.red);

        add(l2);

        JLabel l3=new JLabel("Feel Like Heaven..");

        l3.setBounds(580,110,280,40);
        l3.setForeground(Color.blue);
        l3.setFont(f1);
        add(l3);

        b1=new JButton("Continue....");
        b1.setBounds(600,150,120,30);
```

```

add(b1);
b1.addActionListener(this);

ImageIcon i1= new ImageIcon("C:\\Users\\Priyankar Jain\\Desktop\\Hotel Management
System\\abc.jpg");
JLabel l1=new JLabel("",i1,FlowLayout.LEFT);
l1.setBounds(0,0,1024,700);
add(l1);

setVisible(true);
}
public void actionPerformed(ActionEvent e)
{
if(e.getSource()==b1)

setVisible(false);
new hm2();
}

public static void main(String ar[])
{
new hm1();
}
}

// LOGIN Page
class hm2 extends JFrame implements ActionListener
{
String un,pass;

JTextField tun;
TextField tp;
JLabel lt;
JButton benter,bquit;
hm2()
{
un="a";
pass="a";
setSize(1024,700);
setLayout(null);
setTitle("Hotel Swastik:Login Form");

Font f1 =new Font("arial",Font.BOLD,20);
java.util.Date date=new java.util.Date();
JLabel ldate=new JLabel(String.valueOf(date));

ldate.setFont(f1);
ldate.setBounds(650,20,300,30);
add(ldate);

lt=new JLabel("Log in");
Font f =new Font("algerian",Font.BOLD,30);
lt.setFont(f);
lt.setBounds(500,50,300,40);

```

```

add(lt);

JLabel l2=new JLabel("User Name");
l2.setBounds(400,100,200,30);
l2.setFont(f1);
add(l2);
tun=new JTextField(20);
tun.setBounds(550,100,200,30);
tun.setFont(f1);
add(tun);

JLabel l3=new JLabel("Password");
l3.setFont(f1);
l3.setBounds(400,150,200,30);

tp=new TextField(20);
tp.setBounds(550,150,200,30);
tp.setFont(f1);
tp.setEchoChar('*');
add(tp);
add(l3);
benter=new JButton("Enter ");
add(benter);
benter.addActionListener(this);
benter.setBounds(400,200,180,30);

bquit=new JButton("Quit");
add(bquit);
bquit.addActionListener(this);
bquit.setBounds(600,200,180,30);

ImageIcon i1= new ImageIcon("C:\\Users\\Priyankar Jain\\Desktop\\Hotel Management
System\\abc.jpg");
JLabel li=new JLabel("",i1,FlowLayout.LEFT);
li.setBounds(0,0,1024,700);
add(li);

setVisible(true);
}
public void actionPerformed(ActionEvent e)
{
if(e.getSource()==benter)
{
    if((tun.getText().equals("admin"))&&(tp.getText().equals("admin")))
    {
        JOptionPane.showMessageDialog(this,"Login Successfull !");
        setVisible(false);
        new hm3();
    }
    else
    {
        JOptionPane.showMessageDialog(this,"User Name and Password not Correct !");
    }
}
}

```

```

}
if(e.getSource()==bquit)
{setVisible(false);}
}
}

// Available Fascilities Page
class hm3 extends JFrame implements ActionListener
{
JButton broom,bcin,bcout,bres,bquit;
hm3()
{
setSize(1024,700);
setLayout(null);
setTitle("Hotel Swastik:Menu");

Font f1 =new Font("arial",Font.BOLD,20);
java.util.Date date=new java.util.Date();
JLabel ldate=new JLabel(String.valueOf(date));

ldate.setFont(f1);
ldate.setBounds(650,14,300,30);
add(ldate);

JLabel l=new JLabel(" Available Fascilities");
Font f =new Font("algerian",Font.BOLD,30);
l.setFont(f);
l.setBounds(450,20,500,60);
add(l);

broom=new JButton("Room Booking ");
add(broom);
broom.addActionListener(this);
broom.setBounds(400,80,150,30);

bcin=new JButton("Check in");
add(bcin);
bcin.setBounds(800,80,150,30);
bcin.addActionListener(this);

bcout=new JButton("Check out");
add(bcout);
bcout.setBounds(800,150,150,30);
bcout.addActionListener(this);

bres=new JButton("Restaunt");
add(bres);
bres.addActionListener(this);
bres.setBounds(400,150,150,30);

bquit=new JButton("Quit");
bquit.addActionListener(this);
bquit.setBounds(830,610,150,30);
add(bquit);

```

```

ImageIcon i1= new ImageIcon("C:\\Users\\Priyankar Jain\\Desktop\\Hotel Management
System\\abc.jpg");
JLabel l1=new JLabel("",i1,FlowLayout.LEFT);
l1.setBounds(10,10,1024,700);
add(l1);

setVisible(true);
}
public void actionPerformed(ActionEvent e)
{
if(e.getSource()==broom)
{
setVisible(false);
new hm4();
}
if(e.getSource()==bcin)
{
setVisible(false);
new hmcin();
}
if(e.getSource()==bcout)
{
setVisible(false);
new hmcout();
}
if(e.getSource()==bres)
{
setVisible(false);
new hmres();
}
if(e.getSource()==bquit)
{
setVisible(false);
}
}
}
// Room Booking Page
class hm4 extends Frame implements ActionListener,ItemListener
{
JLabel lt,ln,lfn,ladd,lph,lcity,lstate,lbfor,lrt,lprice,lroomno;
JTextField tn,tfn,tadd,tph,tcity,tprice,troomno;
Choice cstate,cbd,cbford,cbform,cbfory;
JButton benter,bback;
Connection con;
int roomno;

hm4()
{
setSize(1024,700);
setLayout(null);
setTitle("Hotel Swastik:Room Booking");

Font f1 =new Font("arial",Font.BOLD,20);

```



```

java.util.Date date=new java.util.Date();
JLabel ldate=new JLabel(String.valueOf(date));

ldate.setFont(f1);
ldate.setBounds(650,30,300,30);
add(ldate);

Font f =new Font("algerian",Font.BOLD,30);

lt=new JLabel("Room Booking");
lt.setBounds(125,50,300,40);
lt.setFont(f);
ln=new JLabel("Name of Customer:");
ln.setBounds(100,100,200,30);
lfn=new JLabel("Father Name:");
lfn.setBounds(100,130,200,30);
ladd=new JLabel("Address:");
ladd.setBounds(100,160,200,30);
lph=new JLabel("Phone no:");
lph.setBounds(100,190,200,30);
lcity=new JLabel("City:");
lcity.setBounds(100,220,200,30);
lstate =new JLabel("State:");
lstate.setBounds(100,250,150,30);
lbfor=new JLabel("Date for Room Booked:");
lbfor.setBounds(100,280,310,30);
lrt=new JLabel("Room Type:");
lrt.setBounds(100,310,200,30);
lprice=new JLabel("Price:");
lprice.setBounds(100,340,200,30);
lroomno=new JLabel("Room no:");
lroomno.setBounds(100,370,200,30);
cstate =new Choice();
cstate.setBounds(250,250,200,30);
cstate.add("2");
cstate.add("3");

cbd=new Choice();
cbd.setBounds(310,310,200,30);
cbd.add("Single Bedroom");
cbd.add("Double Bedroom");
cbd.add("Single Bedroom AC");
cbd.add("Double Bedroom AC");
cbd.addItemListener(this);
cbford=new Choice();
cbford.setBounds(420,280,50,30);
cbford.add("1");
cbford.add("2");
cbford.add("3");
cbford.add("4");
cbford.add("5");
cbford.add("6");
cbford.add("7");
cbford.add("8");
cbford.add("9");

```

```
cbford.add("10");
cbford.add("11");
cbford.add("12");
cbford.add("13");
cbford.add("14");
cbford.add("15");
cbford.add("16");
cbford.add("17");
cbford.add("18");
cbford.add("19");
cbford.add("20");
cbford.add("21");
cbford.add("22");
cbford.add("23");
cbford.add("24");
cbford.add("25");
cbford.add("26");
cbford.add("27");
cbford.add("28");
cbford.add("29");
cbford.add("30");
cbford.add("31");
```

```
cbform=new Choice();
cbform.setBounds(480,280,50,30);
cbform.add("1");
cbform.add("2");
cbform.add("3");
cbform.add("4");
cbform.add("5");
cbform.add("6");
cbform.add("7");
cbform.add("8");
cbform.add("9");
cbform.add("10");
cbform.add("11");
cbform.add("12");
```

```
cbfory=new Choice();
cbfory.setBounds(540,280,95,30);
cbfory.add("2000");
```

```
cbfory.add("2010");
cbfory.add("2011");
cbfory.add("2012");
cbfory.add("2013");
cbfory.add("2014");
cbfory.add("2015");
cbfory.add("2016");
cbfory.add("2017");
cbfory.add("2018");
cbfory.add("2019");
cbfory.add("2020");
cbfory.add("2021");
cbfory.add("2022");
```

```

cbfory.add("2023");
cbfory.add("2024");
cbfory.add("2025");
cbfory.add("2026");
cbfory.add("2027");
cbfory.add("2028");
cbfory.add("2029");
cbfory.add("2030");
cbfory.add("2031");
cbfory.add("2032");
cbfory.add("2033");
cbfory.add("2034");
cbfory.add("2035");
cbfory.add("2036");
cbfory.add("2037");
cbfory.add("2038");
cbfory.add("2039");
cbfory.add("2040");
cbfory.add("2041");
cbfory.add("2042");
cbfory.add("2043");
cbfory.add("2044");
cbfory.add("2045");
cbfory.add("2046");
cbfory.add("2047");
cbfory.add("2048");
cbfory.add("2049");
cbfory.add("2050");
tn=new JTextField(15);
tn.setBounds(250,100,200,30);
tfn=new JTextField(15);
tfn.setBounds(250,130,200,30);
tadd=new JTextField(15);
tadd.setBounds(250,160,200,30);
tph=new JTextField(15);
tph.setBounds(250,190,200,30);
tcity=new JTextField(15);
tcity.setBounds(250,220,200,30);

tprice=new JTextField(15);
tprice.setEnabled(false);
tprice.setBounds(250,340,200,30);

troomno=new JTextField(15);
troomno.setEnabled(false);
troomno.setBounds(250,370,200,30);

benter=new JButton("Enter");
benter.setBounds(550,310,200,30);
bback=new JButton("Back");
bback.setBounds(745,310,185,30);

add(lt);add(ln);add(tn);add(lfn);add(tfn);add(ladd);add(tadd);add(lph);add(tph);add(l
city);

```

```

add(tcity);add(lstate);add(cstate);add(lbfor);add(cbford);add(cbform);add(cbfor);
add(lrt);add(cbd);add(lprice);add(tprice);add(benter);
add(bback);add(troomno);add(lroomno);

benter.addActionListener(this);
bback.addActionListener(this);

ImageIcon i1= new ImageIcon("C:\\Users\\Priyankar Jain\\Desktop\\Hotel Management
System\\bd.jpg");
JLabel l1=new JLabel("",i1,FlowLayout.LEFT);
l1.setBounds(550,80,380,212);
add(l1);

setVisible(true);

}
//for setting price
public void itemStateChanged(ItemEvent i)
{
    if(cbd.getSelectedItem().equals("Single Bedroom"))
    {tprice.setText("5000");}
    if(cbd.getSelectedItem().equals("Double Bedroom"))
    {tprice.setText("10000");}
    if(cbd.getSelectedItem().equals("Single Bedroom AC"))
    {tprice.setText("10000");}
    if(cbd.getSelectedItem().equals("Double Bedroom AC"))
    {tprice.setText("15000");}
}
public void actionPerformed(ActionEvent e)
{
    if(e.getSource()==bback)
    {
        setVisible(false);
        new hm3();
    }
    if(e.getSource()==benter)
    {
        try{

            //if(((tn.getText).equals(""))&&((tfn.getText).equals(""))&&((tadd.getText).eq
            uals(""))&&((tph.getText).equals(""))&&((tcity.getText).equals(""))&&((tpri
            ce.getText).equals(""))&&((troomno.getText).equals("")))
            //{JOptionPane.showMessageDialog(this,"fill all the fields" );}
            String driver = "sun.jdbc.odbc.JdbcOdbcDriver";
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            String database = "jdbc:odbc:Driver={Microsoft Access Driver (*.mdb)};DBQ=hm.accdb;";
            Connection con = DriverManager.getConnection(database, "system", "hm");
            Statement st=(Statement) con.createStatement();
            PreparedStatement ps1=con.prepareStatement("select min(roomno) from room
            where type=? and status=?");
            ps1.setString(1,cbd.getSelectedItem());
            ps1.setInt(2,0);

            ResultSet rs=ps1.executeQuery();
            if(rs.next())

```

```

        {
            roomno=rs.getInt(1);
            if(roomno==0)
            {
                JOptionPane.showMessageDialog(this,"Sorry!room is not
available");
            }
            else
            {
                PreparedStatement ps2=con.prepareStatement("update room
set status=? where roomno=?");
                ps2.setInt(1,1);
                ps2.setInt(2,roomno);
                ps2.executeUpdate();
                PreparedStatement ps=con.prepareStatement("insert into rb
values(?,?,?,?,?,?,?,?)");
                ps.setString(1,tn.getText());
                ps.setString(2,tfn.getText());
                ps.setString(3,tadd.getText());
                ps.setString(4,tph.getText());
                ps.setString(5,tcity.getText());
                ps.setString(6,cstate.getSelectedItem());

                ps.setString(7,cbform.getSelectedItem()+"/"+cbford.getSelectedItem()+"/"+cbfor
y.getSelectedItem());
                ps.setString(8,cbd.getSelectedItem());
                ps.setInt(9,Integer.parseInt(tprice.getText()));
                ps.setInt(10,roomno);
                ps.executeUpdate();
                troomno.setText(String.valueOf(roomno));
                JOptionPane.showMessageDialog(this,"Congrats Your Room no
is"+roomno);

                tn.setText("");
                tfn.setText("");
                tadd.setText("");
                tcity.setText("");
                tph.setText("");
                tprice.setText("");
                troomno.setText("");
                tn.setText("");
                tn.setText("");
                tn.setText("");
            }
        }
    }
    else
    {
        JOptionPane.showMessageDialog(this,"Sorry!room is not available");
    }

}
catch(Exception exc)
{
    System.out.println(exc);
}

```

```

    }
}

//Check in Page
class hmcin extends JFrame implements ActionListener
{
JLabel
lt,lcin,ln,lfn,ladd,lph,lcity,lstate,lbfor,lrt,lprice,lroomno,lln,llfn,lladd,llph,llc
ity,llstate,llbfor,llrt,llprice,llroomno;
JButton bback;
Label leroomno;
TextField troomno;
JButton benter,bok;
Connection con;
hmcin()
{
setTitle("Hotel Swastik:Check in");
setLayout(null);
setSize(1024,700);

Font f1 =new Font("arial",Font.BOLD,20);
java.util.Date date=new java.util.Date();
JLabel ldate=new JLabel(String.valueOf(date));

ldate.setFont(f1);
ldate.setBounds(650,20,300,30);
add(ldate);

Panel p=new Panel();

p.setBounds(150,80,700,70);
leroomno=new Label("Enter Room No");
leroomno.setBounds(100,80,200,30);
troomno=new TextField(30);
troomno.setBounds(512,80,200,30);
benter= new JButton("Enter");

bback=new JButton("Back");
bback.setBounds(512,490,300,30);
bback.addActionListener(this);
add(bback);

bok=new JButton("Ok");
bok.setBounds(200,490,300,30);
bok.addActionListener(this);
add(bok);

add(p);

p.add(leroomno);
p.add(troomno);

```

```
p.add(benter);
```

```
benter.addActionListener(this);
```

```
Font f =new Font("algerian",Font.BOLD,50);  
lt=new JLabel("Check-In");  
lt.setBounds(300,20,300,50);  
lt.setFont(f);
```

```
ln=new JLabel("Name of Customer:");  
ln.setBounds(100,150,300,30);  
lfn=new JLabel("Father Name:");  
lfn.setBounds(100,180,250,30);  
ladd=new JLabel("Address:");  
ladd.setBounds(100,210,250,30);  
lph=new JLabel("Phone no:");  
lph.setBounds(100,240,250,30);  
lcity=new JLabel("City:");  
lcity.setBounds(100,270,250,30);  
lstate =new JLabel("State:");  
lstate.setBounds(100,300,250,30);  
lbfor=new JLabel("Date for Room Booked:");  
lbfor.setBounds(100,330,350,30);  
lrt=new JLabel("Room Type:");  
lrt.setBounds(100,370,250,30);  
lprice=new JLabel("Price:");  
lprice.setBounds(100,400,250,30);  
lroomno=new JLabel("Room no:");  
lroomno.setBounds(100,430,250,30);
```

```
lln=new JLabel("");  
lln.setBounds(500,150,300,30);  
llfn=new JLabel("");  
llfn.setBounds(500,180,250,30);  
lladd=new JLabel("");  
lladd.setBounds(500,210,250,30);  
llph=new JLabel("");  
llph.setBounds(500,240,250,30);  
llcity=new JLabel("");  
llcity.setBounds(500,270,250,30);  
llstate =new JLabel("");  
llstate.setBounds(500,300,250,30);  
llbfor=new JLabel("");  
llbfor.setBounds(500,330,350,30);  
llrt=new JLabel("");  
llrt.setBounds(500,370,250,30);  
llprice=new JLabel("");  
llprice.setBounds(500,400,250,30);  
llroomno=new JLabel("");  
llroomno.setBounds(500,430,250,30);  
add(lln);add(llfn);add(lladd);add(llph);add(llcity);add(llstate);add(llbfor);add(llrt);  
add(llprice);add(llroomno);
```

```

add(lt);add(ln);add(lfn);add(ladd);add(lph);add(lcity);add(lstate);add(lbfor);add(lrt
);add(lprice);add(lroomno);

setVisible(true);
}
public void actionPerformed(ActionEvent e)
{
if(e.getSource()==bback)
{setVisible(false);new hm3(); }

if(e.getSource()==benter)
{
    try{
/*Class.forName("oracle.jdbc.OracleDriver");

        con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE","system"
,"hm");*/
        String driver = "sun.jdbc.odbc.JdbcOdbcDriver";
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
String database = "jdbc:odbc:Driver={Microsoft Access Driver (*.mdb)};DBQ=hm.accdb;";
Connection con= DriverManager.getConnection(database, "system", "hm");

PreparedStatement ps1=con.prepareStatement("select * from rb where roomno=?");
ps1.setInt(1,Integer.parseInt(troomno.getText()));
ResultSet rs=ps1.executeQuery();
if(rs.next())
    {

        lln.setText(rs.getString(1));
        llfn.setText(rs.getString(2));
        lladd.setText(rs.getString(3));
        llph.setText(rs.getString(4));
        llcity.setText(rs.getString(5));
        llstate.setText(rs.getString(6));
        llbfor.setText(rs.getString(7));
        llrt.setText(rs.getString(8));
        llprice.setText(rs.getString(9));
        llroomno.setText(rs.getString(10));

        JOptionPane.showMessageDialog(this,"Congrats! welcome To Hotel Swastik
");
    }
    else
    {
        JOptionPane.showMessageDialog(this,"Sorry!Room no is not
Incorrect.Please enter the correct Roomno");
    }
}
catch(Exception ex)
    {System.out.println(ex);}
}

if(e.getSource()==bok)
{
    java.util.Date d1=new java.util.Date();
    int dd=d1.getDate();

```



```

        int mm=d1.getMonth()+1;
        int yyyy=d1.getYear()+1900;
    try {
        String driver = "sun.jdbc.odbc.JdbcOdbcDriver";
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
        String database = "jdbc:odbc:Driver={Microsoft Access Driver (*.mdb)};DBQ=hm.accdb;";
        Connection con = DriverManager.getConnection(database, "system", "hm");
        PreparedStatement ps3=con.prepareStatement("insert into bill
values(?,?,?,?)" ) ;

        ps3.setInt(1,Integer.parseInt(troomno.getText()));
        ps3.setString(2,"checkin");
        ps3.setInt(3,Integer.parseInt(l1price.getText()));
        ps3.setString(4,mm+"/"+dd+"/"+yyyy);
        ps3.executeUpdate();
        JOptionPane.showMessageDialog(this,"Enjoy Our Services");
        setVisible(false);
        new hm3();

    }catch(Exception exc){System.out.println(exc);}
}
}

// Restraunt Page
class hmres extends JFrame implements ActionListener,ItemListener
{
    JLabel
    ltt,l1,litem,liprice,lqty,ls1,ls2,ls3,ls4,ls5,lsam,lat,lchat,lc,lm,lpc,lt,ldosa,lppak
    ,lbeer,lw,lr,lid,lsa,lsd,lbs,lms,lss,ltea,lcf;
    Checkbox csam,cat,chat,cppak;
    Checkbox cc,cm,ct,cpc;
    Checkbox cdosa,cid,csa;
    Checkbox csd,cbs,cms,css,ctea,ccf;
    Checkbox cbeer,cw,cr;
    JButton bback,bbill;
    JTextField
    tsam,tat,tchat,tppak,tc,tm,tpc,tt,tdosa,tid,tsa,tsd,tr,tw,tbs,tms,tss,ttea,tcf,tbeer;
    int a, i=50,j=390,k=70,l=20;

    hmres()
    {
        setTitle("Hotel Swastik:Restraunt Menu");
        setLayout(null);
        setSize(1024,700);

        Font f1 =new Font("arial",Font.BOLD,20);
        java.util.Date date=new java.util.Date();
        JLabel ldate=new JLabel(String.valueOf(date));

        ldate.setFont(f1);
        ldate.setBounds(650,20,300,30);
        add(ldate);

```

```

Font f =new Font("algerian",Font.BOLD,30);
l1t=new JLabel("Restaunt");
l1t.setBounds(125,50,300,40);
l1t.setFont(f);
add(l1t);

ImageIcon i1= new ImageIcon("C:\\Users\\Priyankar Jain\\Desktop\\Hotel Management
System\\res.jpg");
JLabel l1=new JLabel("",i1,FlowLayout.LEFT);
l1.setBounds(550,80,380,212);
add(l1);

bbill=new JButton("Bill");
bbill.setBounds(550,360,190,30);
add(bbill);
bbill.addActionListener(this);

litem=new JLabel("Item");
litem.setBounds(50,100,150,30);
litem.setFont(f1);
liprice=new JLabel("Price(in Rs)");
liprice.setBounds(240,100,150,30);
liprice.setFont(f1);
lqty=new JLabel("Quantity");
lqty.setBounds(390,100,150,30);
lqty.setFont(f1);
add(litem);add(liprice);add(lqty);
ls1=new JLabel("indian ");
ls1.setBounds(50,130,100,30);
ls1.setFont(f1);
add(ls1);

ls2=new JLabel("Chinese");
ls2.setBounds(50,235,100,30);
ls2.setFont(f1);
add(ls2);
ls3=new JLabel("South Indian ");
ls3.setBounds(50,340,100,30);
ls3.setFont(f1);
add(ls3);
csam=new Checkbox("Samosa");
csam.setBounds(50,155,100,20);
add(csam);
cat=new Checkbox("Aaloo Tikki");
cat.setBounds(50,175,100,20);
add(cat);
chat=new Checkbox("Chat");
chat.setBounds(50,195,100,20);
add(chat);
cppak=new Checkbox("Paneer Pakoda");
cppak.setBounds(50,215,100,20);
add(cppak);
cc=new Checkbox("Chowmine");

```

```

cc.setBounds(50,260,100,20);
add(cc);
cm=new Checkbox("Veg Momos");
cm.setBounds(50,280,100,20);
add(cm);
cpc=new Checkbox("Paneer Chowmine");
cpc.setBounds(50,300,120,20);
add(cpc);
ct=new Checkbox("Thupka");
ct.setBounds(50,320,100,20);
add(ct);
cdosa=new Checkbox("Dosa");
cdosa.setBounds(50,365,100,20);
add(cdosa);
cid=new Checkbox("Idlee");
cid.setBounds(50,385,100,20);
add(cid);
csa=new Checkbox("Samber");
csa.setBounds(50,405,100,20);
add(csa);

ls4=new JLabel("Drinks & Shakes ");
ls4.setBounds(50,425,100,30);
ls4.setFont(f1);
add(ls4);
csd=new Checkbox("Soft Drink");
csd.setBounds(50,450,100,20);
add(csd);
cbs=new Checkbox("Banana Shake");
cbs.setBounds(50,470,180,20);
add(cbs);
cms=new Checkbox("Mango Shake");
cms.setBounds(50,490,100,20);
add(cms);
css=new Checkbox("Strawbery Shake");
css.setBounds(50,510,100,20);
add(css);
ctea=new Checkbox("Tea");
ctea.setBounds(50,530,100,20);
add(ctea);
ccf=new Checkbox("Coffee");
ccf.setBounds(50,550,100,20);
add(ccf);

ls5=new JLabel("Liquor");
ls5.setBounds(50,570,100,30);
ls5.setFont(f1);
add(ls5);

cbeer=new Checkbox("Beer");
cbeer.setBounds(50,595,100,20);
add(cbeer);
cw=new Checkbox("Whisky");
cw.setBounds(50,615,100,20);
add(cw);

```

```
cr=new Checkbox("Rum");
cr.setBounds(50,635,100,20);
add(cr);

lsam=new JLabel(" 20.00");
lsam.setBounds(240,155,100,20);
add(lsam);
lat=new JLabel(" 20.00");
lat.setBounds(240,175,100,20);
add(lat);
lchat=new JLabel(" 50.00");
lchat.setBounds(240,195,100,20);
add(lchat);
lppak=new JLabel(" 50.00");
lppak.setBounds(240,215,100,20);
add(lppak);
lc=new JLabel(" 70.00");
lc.setBounds(240,260,100,20);
add(lc);
lm=new JLabel(" 50.00");
lm.setBounds(240,280,100,20);
add(lm);
lpc=new JLabel(" 100.00");
lpc.setBounds(240,300,100,20);
add(lpc);
lt=new JLabel(" 50.00");
lt.setBounds(240,320,100,20);
add(lt);
ldosa=new JLabel(" 50.00");
ldosa.setBounds(240,365,100,20);
add(ldosa);
lid=new JLabel(" 50.00");
lid.setBounds(240,385,100,20);
add(lid);
lsa=new JLabel(" 70.00");
lsa.setBounds(240,405,100,20);
add(lsa);
lsd=new JLabel(" 50.00");
lsd.setBounds(240,450,100,20);
add(lsd);
lbs=new JLabel(" 20.00");
lbs.setBounds(240,470,100,20);
add(lbs);
lms=new JLabel(" 30.00");
lms.setBounds(240,490,100,20);
add(lms);
lss=new JLabel(" 30.00");
lss.setBounds(240,510,100,20);
add(lss);
ltea=new JLabel(" 20.00");
ltea.setBounds(240,530,100,20);
add(ltea);
lcf=new JLabel(" 30.00");
lcf.setBounds(240,550,100,20);
add(lcf);
```

```
lbeer=new JLabel(" 80.00");
lbeer.setBounds(240,595,100,20);
add(lbeer);
lw=new JLabel(" 500.00");
lw.setBounds(240,615,100,20);
add(lw);
lr=new JLabel(" 300.00");
lr.setBounds(240,635,100,20);
add(lr);
tsam=new JTextField(i);
tsam.setBounds(j,155,k,1);
add(tsam);
tat=new JTextField(i);
tat.setBounds(j,175,k,1);
add(tat);
tchat=new JTextField(i);
tchat.setBounds(j,195,k,1);
add(tchat);
tppak=new JTextField(i);
tppak.setBounds(j,215,k,1);
add(tppak);
tc=new JTextField(i);
tc.setBounds(j,260,k,1);
add(tc);
tm=new JTextField(i);
tm.setBounds(j,280,k,1);
add(tm);
tpc=new JTextField(i);
tpc.setBounds(j,300,k,1);
add(tpc);
tt=new JTextField(i);
tt.setBounds(j,320,k,1);
add(tt);
tdosa=new JTextField(i);
tdosa.setBounds(j,365,k,1);
add(tdosa);
tid=new JTextField(i);
tid.setBounds(j,385,k,1);
add(tid);
tsa=new JTextField(i);
tsa.setBounds(j,405,k,1);
add(tsa);
tsd=new JTextField(i);
tsd.setBounds(j,450,k,1);
add(tsd);
tbs=new JTextField(i);
tbs.setBounds(j,470,k,1);
add(tbs);
tms=new JTextField(i);
tms.setBounds(j,490,k,1);
add(tms);
tss=new JTextField(i);
tss.setBounds(j,510,k,1);
add(tss);
ttea=new JTextField(i);
```

```

ttea.setBounds(j,530,k,1);
add(ttea);
tcf=new JTextField(i);
tcf.setBounds(j,550,k,1);
add(tcf);
tbeer=new JTextField(i);
tbeer.setBounds(j,595,k,1);
add(tbeer);
tw=new JTextField(i);
tw.setBounds(j,615,k,1);
add(tw);
tr=new JTextField(i);
tr.setBounds(j,635,k,1);
add(tr);

bback=new JButton("Back");
bback.setBounds(740,360,190,30);
bback.addActionListener(this);
add(bback);
csam.addItemListener(this);
cat.addItemListener(this);
chat.addItemListener(this);
cppak.addItemListener(this);
cc.addItemListener(this);
cm.addItemListener(this);
cpc.addItemListener(this);
ct.addItemListener(this);
cdosa.addItemListener(this);
cid.addItemListener(this);
csa.addItemListener(this);
csd.addItemListener(this);
cbs.addItemListener(this);
cms.addItemListener(this);
css.addItemListener(this);
ctea.addItemListener(this);
ccf.addItemListener(this);
cbeer.addItemListener(this);
cw.addItemListener(this);
cr.addItemListener(this);

setVisible(true);
}
public void actionPerformed(ActionEvent e)
{
String name[]=new String[20];
String p[]=new String[20];
int q=0,r=0;

if(e.getSource()==bback)
{setVisible(false);new hm3(); }
if(e.getSource()==bbill)
{if((csam.getState())==true)
{
name[q]=csam.getLabel();
q=q+1;

```

```

}
if((cat.getState())==true)
{
name[q]=cat.getLabel();
q=q+1;
}
if((chat.getState())==true)
{
name[q]=chat.getLabel();
q=q+1;
}
if((cppak.getState())==true)
{
name[q]=cppak.getLabel();
q=q+1;
}
if((cc.getState())==true)
{
name[q]=cc.getLabel();
q=q+1;
}
if((cm.getState())==true)
{
name[q]=csam.getLabel();
q=q+1;
}
if((cpc.getState())==true)
{
name[q]=cpc.getLabel();
q=q+1;
}
if((ct.getState())==true)
{
name[q]=ct.getLabel();
q=q+1;
}
if((cdosa.getState())==true)
{
name[q]=cdosa.getLabel();
q=q+1;
}
if((cid.getState())==true)
{
name[q]=cid.getLabel();
q=q+1;
}
if((csa.getState())==true)
{
name[q]=csa.getLabel();
q=q+1;
}
if((csd.getState())==true)
{
name[q]=csd.getLabel();
q=q+1;
}

```

```

}
if((cbs.getState())==true)
{
name[q]=cbs.getLabel();
q=q+1;
}
if((cms.getState())==true)
{
name[q]=cms.getLabel();
q=q+1;
}
if((css.getState())==true)
{
name[q]=css.getLabel();
q=q+1;
}
if((ctea.getState())==true)
{
name[q]=ctea.getLabel();
q=q+1;
}
if((ccf.getState())==true)
{
name[q]=ccf.getLabel();
q=q+1;
}
if((cbeer.getState())==true)
{
name[q]=cbeer.getLabel();
q=q+1;
}
if((cw.getState())==true)
{
name[q]=cw.getLabel();
q=q+1;
}
if((cr.getState())==true)
{
name[q]=cr.getLabel();
q=q+1;
}

//text
if(!(tsam.getText().equals("")))
{
p[r]=tsam.getText();
r=r+1;
}
if(!(tat.getText().equals("")))
{
p[r]=tat.getText();
r=r+1;
}
if(!(tchat.getText().equals("")))
{

```



```

p[r]=tchat.getText();
r=r+1;
}
if(!(tppak.getText().equals("")))
{
p[r]=tppak.getText();
r=r+1;
}
if(!(tc.getText().equals("")))
{
p[r]=tc.getText();
r=r+1;
}
if(!(tm.getText().equals("")))
{
p[r]=tm.getText();
r=r+1;
}
if(!(tpc.getText().equals("")))
{
p[r]=tpc.getText();
r=r+1;
}
if(!(tt.getText().equals("")))
{
p[r]=tt.getText();
r=r+1;
}
if(!(tdosa.getText().equals("")))
{
p[r]=tdosa.getText();
r=r+1;
}
if(!(tid.getText().equals("")))
{
p[r]=tid.getText();
r=r+1;
}
if(!(tsa.getText().equals("")))
{
p[r]=tsa.getText();
r=r+1;
}

if(!(tsd.getText().equals("")))
{
p[r]=tsd.getText();
r=r+1;
}

if(!(tbs.getText().equals("")))
{
p[r]=tbs.getText();
r=r+1;
}

```

```

if(!(tms.getText().equals(""))
{
p[r]=tms.getText();
r=r+1;
}
if(!(tss.getText().equals(""))
{
p[r]=tss.getText();
r=r+1;
}
if(!(ttea.getText().equals(""))
{
p[r]=ttea.getText();
r=r+1;
}
if(!(tcf.getText().equals(""))
{
p[r]=tcf.getText();
r=r+1;
}
if(!(tbeer.getText().equals(""))
{
p[r]=tbeer.getText();
r=r+1;
}
if(!(tw.getText().equals(""))
{
p[r]=tw.getText();
r=r+1;
}
if(!(tr.getText().equals(""))
{
p[r]=tr.getText();
r=r+1;
}

new resbill(name,p,r);
setVisible(false);

}
}
public void itemStateChanged(ItemEvent x)
{

}
}

//Restaunt Bill Frame
class resbill extends Frame implements ActionListener
{
JButton bcontinue;
int tbill;
JLabel ttbill;

```

```

resbill(String name[],String p[],int r)
{
    Font f =new Font("algerian",Font.BOLD,30);
        JLabel ltt=new JLabel("Restaunt Bill Detail");
        ltt.setBounds(125,50,300,40);
        ltt.setFont(f);
        add(ltt);

        Font f1 =new Font("arial",Font.BOLD,20);
        java.util.Date date=new java.util.Date();
        JLabel ldate=new JLabel(String.valueOf(date));

        ldate.setFont(f1);
        ldate.setBounds(650,20,300,30);
        add(ldate);

        ImageIcon i1= new ImageIcon("C:\\Users\\Priyankar
Jain\\Desktop\\Hotel Management System\\res.jpg");
        JLabel l1=new JLabel("",i1,FlowLayout.LEFT);
        l1.setBounds(550,80,380,212);
        add(l1);

try{
    String driver = "sun.jdbc.odbc.JdbcOdbcDriver";
    Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
    String database = "jdbc:odbc:Driver={Microsoft Access Driver (*.mdb)};DBQ=hm.accdb;";
    Connection con= DriverManager.getConnection(database, "system", "hm");

        setSize(1024,700);
        setLayout(null);
        setTitle("Hotel Swastik: Restaunt Bill Details");

        Font f2 =new Font("arial",Font.BOLD,20);
        java.util.Date pdate=new java.util.Date();
        JLabel psldate=new JLabel(String.valueOf(date));

        ldate.setFont(f2);
        ldate.setBounds(650,30,300,30);
        add(ldate);

        int sno=1;
        String s[]={"Sr no","Item Name","Quantity","Price"};
        String s1[][]=new String[r][4];
        for(int i=0;i<r;i++)
        {
            s1[i][0]=String.valueOf(sno);
            s1[i][1]=name[i];
            s1[i][2]=p[i];
            PreparedStatement ps=con.prepareStatement("select price from
resitem where item=?");
            ps.setString(1,name[i]);
            ResultSet rs=ps.executeQuery();
            if(rs.next())

```

```

        {
            s1[i][3]=String.valueOf(rs.getInt(1)*Integer.parseInt(p[i]));
            tbill=tbill+Integer.parseInt(s1[i][3]);
        }

        sno++;
    }

    JTable jt=new JTable(s1,s);

    int h=ScrollPaneConstants.HORIZONTAL_SCROLLBAR_ALWAYS;
    int v=ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS;
    JScrollPane jp=new JScrollPane(jt,v,h);
    jp.setBounds(100,100,400,500);
    add(jp);
    JLabel l2=new JLabel("Your Total Amount Is: ");
    l2.setBounds(550,310,150,30);
    add(l2);
    ttbill=new JLabel("");
    ttbill.setText(String.valueOf(tbill));
    ttbill.setBounds(700,310,150,30);
    add(ttbill);
    bcontinue=new JButton("Continue");
    bcontinue.setBounds(770,310,160,30);
    bcontinue.addActionListener(this);
    add(bcontinue);
    setVisible(true);
}
catch(Exception e)
{
    System.out.println(e);
}

}
public void actionPerformed(ActionEvent a)
{
    if(a.getSource()==bcontinue)
    {setVisible(false);new resbill2(tbill);}
}
}

// resbill2
class resbill2 extends JFrame implements ActionListener
{
    JLabel l1;
    JButton b1;
    JTextField t1;
    int tb;

    resbill2(int tbill)
    {
        setSize(400,250);
        setLayout(null);
        setTitle("Hotel Swastik:Saving Restaunt Bill");
    }
}

```

```

Font f=new Font("ALGERIAN",Font.BOLD,25);
JLabel lt=new JLabel("Saving Restraunt Bill");
lt.setBounds(30,20,400,40);
lt.setFont(f);
add(lt);
l1=new JLabel("Enter Room no :");
l1.setBounds(50,80,150,30);
add(l1);
t1=new JTextField(15);
t1.setBounds(210,80,150,30);
add(t1);
b1=new JButton("Enter(Save Bill)");
b1.setBounds(100,130,200,30);
add(b1);
b1.addActionListener(this);
tb=tbill;
setVisible(true);
}
public void actionPerformed(ActionEvent ee)
{
    if(ee.getSource()==b1)
    {
        java.util.Date d1=new java.util.Date();
        int dd=d1.getDate();
        int mm=d1.getMonth()+1;
        int yyyy=d1.getYear()+1900;
        try {
            String driver = "sun.jdbc.odbc.JdbcOdbcDriver";
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            String database = "jdbc:odbc:Driver={Microsoft Access Driver (*.mdb)};DBQ=hm.accdb;";
            Connection con = DriverManager.getConnection(database, "system", "hm");

            PreparedStatement ps1=con.prepareStatement("select * from rb where
            roomno=?");
            ps1.setInt(1,Integer.parseInt(t1.getText()));
            ResultSet rs=ps1.executeQuery();
            if(rs.next())
            {
                PreparedStatement ps3=con.prepareStatement("insert into bill
            values(?,?,?,?)") ;

                ps3.setInt(1,Integer.parseInt(t1.getText()));
                ps3.setString(2,"Restraunt");
                ps3.setInt(3,tb);
                ps3.setString(4,mm+"/"+dd+"/"+yyyy);
                ps3.executeUpdate();

                JOptionPane.showMessageDialog(this,"Your Bill is added to your
            Account.Thanks ");
                new hm3();
            }
            else
            {

```

```

        JOptionPane.showMessageDialog(this,"Sorry!Room no is not
Incorrect,Please pay your Bill");
        new hm3();
    }
    }catch(Exception exc){System.out.println(exc);}
    }
    setVisible(false);

}
}

class hmcout extends JFrame implements ActionListener
{
String roomno;
JButton b2;
String s1[][];
String s[];
JButton b1;
String indate,outdate;
JLabel l1;
int count,a=0,i=1;
long sum=0;
JTable jt;
    hmcout()
    {
        roomno=JOptionPane.showInputDialog("Enter your room No");

        try {
            setSize(1024,700);
            setLayout(null);
            String driver = "sun.jdbc.odbc.JdbcOdbcDriver";
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
String database = "jdbc:odbc:Driver={Microsoft Access Driver (*.mdb)};DBQ=hm.accdb;";
Connection con = DriverManager.getConnection(database, "system", "hm");

            PreparedStatement ps1=con.prepareStatement("select
count(*) from bill where roomno=?");
            ps1.setInt(1,Integer.parseInt(roomno));

            ResultSet rs1=ps1.executeQuery();
            if(rs1.next())
            {
                count=rs1.getInt(1);
                a=1;
            }
            else
            {

                JOptionPane.showMessageDialog(this,"room no is not
correct");

                setVisible(false);
                new hm3();
            }

            if(a==1)

```

```

        {
            Font f1 =new Font("arial",Font.BOLD,20);
            java.util.Date date=new java.util.Date();
            JLabel ldate=new JLabel(String.valueOf(date));

            ldate.setFont(f1);
            ldate.setBounds(650,20,300,30);
            add(ldate);

            Font f =new Font("algerian",Font.BOLD,30);
            JLabel ltt=new JLabel("Check-out");
            ltt.setBounds(125,50,300,40);
            ltt.setFont(f);
            add(ltt);

            ImageIcon i1= new ImageIcon("C:\\Users\\Priyankar
Jain\\Desktop\\Hotel Management System\\res.jpg");
            JLabel l1=new JLabel("",i1,FlowLayout.LEFT);
            l1.setBounds(550,80,380,212);
            add(l1);

            String s[]{"BillType","Amount","Date"};
            String s1[][]=new String[count][3];
            PreparedStatement ps2=con.prepareStatement("select * from
bill where roomno=? and billtype='checkin'");
            ps2.setInt(1,Integer.parseInt(roomno));
            ResultSet rs=ps2.executeQuery();
            if(rs.next())
            {
                s1[0][0]="Rent";
                indate=rs.getString("date");
                java.util.Date ddd=new java.util.Date(indate);
                int dd=ddd.getDate();

                java.util.Date outdate=new java.util.Date();
                int outdd=outdate.getDate();

                int diff=outdd-dd;

                int price=rs.getInt("amount")*diff;
                sum=sum+price;
                s1[0][1]=String.valueOf(price);
                s1[0][2]=indate;
                PreparedStatement ps3=con.prepareStatement("select *
from bill where roomno=?");
                ps3.setInt(1,Integer.parseInt(roomno));
                ResultSet rs2=ps3.executeQuery();
                while(rs2.next())
                {
                    String tt=rs2.getString("billtype").trim();
                    if(!tt.equals("checkin"))
                    {
                        s1[i][0]=tt;
                        int pp=rs2.getInt("amount");

```

```

        s1[i][1]=String.valueOf(pp);
        sum=sum+pp;
        s1[i][2]=rs2.getString("date");
        i++;
    }
}

jt=new JTable(s1,s);
jt.setBounds(100,100,380,500);
int
h=ScrollPaneConstants.HORIZONTAL_SCROLLBAR_ALWAYS;
int v=ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS;
JScrollPane jp=new JScrollPane(jt,v,h);
JLabel l2=new JLabel("Your Total Amount Is: ");
l2.setBounds(550,310,150,30);
l1=new JLabel(String.valueOf(sum));
l1.setBounds(700,310,150,30);
jp.setBounds(100,100,400,500);
add(jp);
add(l1);
add(l2);
b1=new JButton("checkout");
b1.setBounds(770,310,160,30);
add(b1);
b1.addActionListener(this);
b2=new JButton("Back");
b2.setBounds(550,350,380,30);
add(b2);
b2.addActionListener(this);
setVisible(true);
    }
}
}
catch(Exception exc)
{System.out.println(exc);
}
}
public void actionPerformed(ActionEvent e)
{
    if(e.getSource()==b1)
    {
        try {
            String driver = "sun.jdbc.odbc.JdbcOdbcDriver";
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            String database = "jdbc:odbc:Driver={Microsoft Access Driver (*.mdb)};DBQ=hm.accdb;";
            Connection con = DriverManager.getConnection(database, "system", "hm");

            PreparedStatement ps1=con.prepareStatement("delete
from bill where roomno=?");
            ps1.setInt(1,Integer.parseInt(roomno));
            ps1.executeUpdate();
            PreparedStatement ps2=con.prepareStatement("update
room set status=? where roomno=?");
            ps2.setInt(1,0);

```



```

        ps2.setInt(2,Integer.parseInt(roomno));
        ps2.executeUpdate();
        PreparedStatement ps3=con.prepareStatement("delete
from Rb where roomno=?");
        ps3.setInt(1,Integer.parseInt(roomno));
        ps3.executeUpdate();
        JOptionPane.showMessageDialog(this,"Thanks! You are
Successfully Checked Out");
        setVisible(false);
        new hm3();
    }
    catch(Exception exc)
    {System.out.println(exc);
    }
}
if(e.getSource()==b2)
{setVisible(false);new hm3();}
}
}

```

4.2 hm2.java

```

import java.sql.*;
import java.applet.*;
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;
import java.util.*;

//Front Page
class hm1 extends JFrame implements ActionListener
{
    JButton b1;
    hm1()
    {
        setSize(1024,700);
        setTitle("Hotel Swastik");
        setLayout(null);

        JLabel l2=new JLabel("****Hotel Swastik****");
        Font f =new Font("algerian",Font.BOLD,50);
        l2.setFont(f);
        l2.setBounds(330,60,650,50);
        l2.setForeground(Color.yellow);

        add(l2);

        JLabel l3=new JLabel("Feel Like Heaven..");

        Font f1 =new Font("arial",Font.BOLD,25);

```

```

13.setBounds(580,110,280,40);
13.setForeground(Color.green);
13.setFont(f1);
add(13);

    b1=new JButton("Continue");
b1.setBounds(600,150,120,30);
add(b1);
b1.addActionListener(this);

ImageIcon i1= new ImageIcon("C:\\Users\\Priyankar Jain\\Desktop\\Hotel Management
System\\abc.jpg");
JLabel l1=new JLabel("",i1,FlowLayout.LEFT);
l1.setBounds(0,0,1024,700);
add(l1);

setVisible(true);
}
public void actionPerformed(ActionEvent e)
{
if(e.getSource()==b1)

setVisible(false);
new hm2();
}

public static void main(String ar[])
{
new hm1();
}
}

// LOGIN Page
class hm2 extends JFrame implements ActionListener
{
String un,pass;

JTextField tun;
TextField tp;
JLabel lt;
JButton benter,bcp,bquit;
hm2()
{
un="a";
pass="a";
setSize(1024,700);
setLayout(null);
setTitle("Hotel Swastik:Login Form");

lt=new JLabel("Log in");
Font f =new Font("algerian",Font.BOLD,30);
lt.setFont(f);
lt.setBounds(500,50,300,40);
add(lt);

```

```

Font f1 =new Font("calibri",Font.BOLD,25);

JLabel l2=new JLabel("User Name");
l2.setBounds(400,100,200,30);
l2.setFont(f1);
add(l2);
tun=new JTextField(20);
tun.setBounds(550,100,200,30);
tun.setFont(f1);
add(tun);

JLabel l3=new JLabel("Password");
l3.setFont(f1);
l3.setBounds(400,150,200,30);
add(l3);

tp=new TextField(20);
tp.setBounds(550,150,200,30);
tp.setFont(f1);
tp.setEchoChar('*');
add(tp);

benter=new JButton("Enter ");
add(benter);
benter.addActionListener(this);
benter.setBounds(400,200,180,30);

bcp=new JButton("Change Password ");
add(bcp);
bcp.addActionListener(this);
bcp.setBounds(600,200,180,30);

bquit=new JButton("Quit");
add(bquit);
bquit.addActionListener(this);
bquit.setBounds(820,610,180,30);

ImageIcon i1= new ImageIcon("C:\\Users\\Priyankar Jain\\Desktop\\Hotel Management
System\\abc.jpg");
JLabel li=new JLabel("",i1,FlowLayout.LEFT);
li.setBounds(0,0,1024,700);
add(li);

setVisible(true);
}
public void actionPerformed(ActionEvent e)
{
if(e.getSource()==benter)
{
    if((tun.getText().equals(un))&&(tp.getText().equals(pass)))
    {
        JOptionPane.showMessageDialog(this,"Login Successfull !");
        setVisible(false);
        new hm3();
    }
}
}

```

```

    }
    else
    {
        JOptionPane.showMessageDialog(this,"User Name and Password not Correct !");
    }
}
}
if(e.getSource()==bquit)
{setVisible(false);}
}
}

```

```

// Availiable Fascilities Page
class hm3 extends JFrame implements ActionListener
{
    JButton broom,bcin,bcout,bres,bquit;
    hm3()
    {
        setSize(1024,700);
        setLayout(null);
        setTitle("Hotel Swastik:Menu");

        JLabel l=new JLabel("    Available Fascilities");
        Font f =new Font("algerian",Font.BOLD,30);
        l.setFont(f);
        l.setBounds(450,20,500,60);
        add(l);

        broom=new JButton("Room Booking ");
        add(broom);
        broom.addActionListener(this);
        broom.setBounds(400,80,150,30);

        bcin=new JButton("Check in");
        add(bcin);
        bcin.setBounds(800,80,150,30);
        bcin.addActionListener(this);

        bcout=new JButton("Check out");
        add(bcout);
        bcout.setBounds(800,150,150,30);
        bcout.addActionListener(this);

        bres=new JButton("Restraunt");
        add(bres);
        bres.addActionListener(this);
        bres.setBounds(400,150,150,30);

        bquit=new JButton("Quit");
        add(bquit);
        bquit.addActionListener(this);
        bquit.setBounds(830,610,150,30);

        ImageIcon i1= new ImageIcon("C:\\Users\\Priyankar Jain\\Desktop\\Hotel Management System\\abc.jpg");
    }
}

```

```

JLabel l1=new JLabel("",i1,FlowLayout.LEFT);
l1.setBounds(10,10,1024,700);
add(l1);

setVisible(true);
}
public void actionPerformed(ActionEvent e)
{
if(e.getSource()==broom)
{
setVisible(false);
new hm4();
}
if(e.getSource()==bcin)
{
setVisible(false);
new hmcin();
}
if(e.getSource()==bcout)
{
setVisible(false);
new hmcout();
}
if(e.getSource()==bres)
{
setVisible(false);
new hmres();
}
if(e.getSource()==bquit)
{
setVisible(false);
}
}
}
// Room Booking Page
class hm4 extends Frame implements ActionListener,ItemListener
{
JLabel lt,ln,lfn,ladd,lph,lcity,lstate,lbfor,lrt,lprice;
JTextField tn,tfn,tadd,tph,tcity,tprice;
Choice cstate,cbd,cbford,cbform,cbfory;
JButton benter,bback;

hm4()
{
setSize(1024,700);
setLayout(null);
setTitle("Hotel Swastik:Room Booking");

Font f =new Font("algerian",Font.BOLD,30);

lt=new JLabel("Room Booking");
lt.setBounds(125,50,300,40);
lt.setFont(f);
ln=new JLabel("Name of Customer:");

```

```
ln.setBounds(100,100,200,30);
lfn=new JLabel("Father Name:");
lfn.setBounds(100,130,200,30);
ladd=new JLabel("Address:");
ladd.setBounds(100,160,200,30);
lph=new JLabel("Phone no:");
lph.setBounds(100,190,200,30);
lcity=new JLabel("City:");
lcity.setBounds(100,220,200,30);
lstate =new JLabel("State:");
lstate.setBounds(100,250,200,30);
lbfor=new JLabel("Date for Room Booked:");
lbfor.setBounds(100,280,310,30);
lrt=new JLabel("Room Type:");
lrt.setBounds(100,310,200,30);
lprice=new JLabel("Price:");
lprice.setBounds(100,340,200,30);
cstate =new Choice();
cstate.setBounds(250,250,200,30);
cstate.add("2");
cstate.add("3");
```

```
cbd=new Choice();
cbd.setBounds(250,310,200,30);
cbd.add("Single Bedroom");
cbd.add("Double Bedroom");
cbd.add("Single Bedroom AC");
cbd.add("Double Bedroom AC");
cbd.addItemListener(this);
cbford=new Choice();
cbford.setBounds(250,280,50,30);
cbford.add("1");
cbford.add("2");
cbford.add("3");
cbford.add("4");
cbford.add("5");
cbford.add("6");
cbford.add("7");
cbford.add("8");
cbford.add("9");
cbford.add("10");
cbford.add("11");
cbford.add("12");
cbford.add("13");
cbford.add("14");
cbford.add("15");
cbford.add("16");
cbford.add("17");
cbford.add("18");
cbford.add("19");
cbford.add("20");
cbford.add("21");
cbford.add("22");
cbford.add("23");
cbford.add("24");
```

```
cbford.add("25");
cbford.add("26");
cbford.add("27");
cbford.add("28");
cbford.add("29");
cbford.add("30");
cbford.add("31");

cbform=new Choice();
cbform.setBounds(300,280,50,30);
cbform.add("1");
cbform.add("2");
cbform.add("3");
cbform.add("4");
cbform.add("5");
cbform.add("6");
cbform.add("7");
cbform.add("8");
cbform.add("9");
cbform.add("10");
cbform.add("11");
cbform.add("12");

cbfory=new Choice();
cbfory.setBounds(355,280,95,30);
cbfory.add("2000");

cbfory.add("2010");
cbfory.add("2011");
cbfory.add("2012");
cbfory.add("2013");
cbfory.add("2014");
cbfory.add("2015");
cbfory.add("2016");
cbfory.add("2017");
cbfory.add("2018");
cbfory.add("2019");
cbfory.add("2020");
cbfory.add("2021");
cbfory.add("2022");
cbfory.add("2023");
cbfory.add("2024");
cbfory.add("2025");
cbfory.add("2026");
cbfory.add("2027");
cbfory.add("2028");
cbfory.add("2029");
cbfory.add("2030");
cbfory.add("2031");
cbfory.add("2032");
cbfory.add("2033");
cbfory.add("2034");
cbfory.add("2035");
cbfory.add("2036");
cbfory.add("2037");
```

```

cbfory.add("2038");
cbfory.add("2039");
cbfory.add("2040");
cbfory.add("2041");
cbfory.add("2042");
cbfory.add("2043");
cbfory.add("2044");
cbfory.add("2045");
cbfory.add("2046");
cbfory.add("2047");
cbfory.add("2048");
cbfory.add("2049");
cbfory.add("2050");
tn=new JTextField(15);
tn.setBounds(250,100,200,30);
tfn=new JTextField(15);
tfn.setBounds(250,130,200,30);
tadd=new JTextField(15);
tadd.setBounds(250,160,200,30);
tph=new JTextField(15);
tph.setBounds(250,190,200,30);
tcity=new JTextField(15);
tcity.setBounds(250,220,200,30);

tprice=new JTextField(15);
tprice.setEnabled(false);
tprice.setBounds(250,340,200,30);

benter=new JButton("Enter");
benter.setBounds(550,310,200,30);
bback=new JButton("Back");
bback.setBounds(745,310,185,30);

add(lt);add(ln);add(tn);add(lfn);add(tfn);add(ladd);add(tadd);add(lph);add(tph);add(l
city);
add(tcity);add(lstate);add(cstate);add(lbfor);add(cbford);add(cbform);add(cbfory);
add(lrt);add(cbd);add(lprice);add(tprice);add(benter);
add(bback);

benter.addActionListener(this);
bback.addActionListener(this);

ImageIcon i1= new ImageIcon("C:\\Users\\Priyankar Jain\\Desktop\\Hotel Management
System\\bd.jpg");
JLabel l1=new JLabel("",i1,FlowLayout.LEFT);
l1.setBounds(550,80,380,212);
add(l1);

setVisible(true);

}
//for setting price
public void itemStateChanged(ItemEvent i)
{

```



```

if(cbd.getSelectedItem().equals("Single Bedroom"))
{tprice.setText("50006765");}
if(cbd.getSelectedItem().equals("Double Bedroom"))
{tprice.setText("10000");}
if(cbd.getSelectedItem().equals("Single Bedroom AC"))
{tprice.setText("10000");}
if(cbd.getSelectedItem().equals("Double Bedroom AC"))
{tprice.setText("15000");}
}
public void actionPerformed(ActionEvent e)
{
if(e.getSource()==bback)
{
setVisible(false);
new hm3();
}
if(e.getSource()==benter)
{
try{
String driver = "sun.jdbc.odbc.JdbcOdbcDriver";
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

String accessFileName = "C:\\hm.accdb";

String connURL="jdbc:odbc:DRIVER={Microsoft Access Driver (*.mdb,
*.accdb)};DBQ="+accessFileName+".accdb;";

Connection con = DriverManager.getConnection(connURL, "system","hm");
PreparedStatement ps=con.prepareStatement("insert into rb
values(?,?,?,?,?,?,?,?)");
ps.setString(1,tn.getText());
ps.setString(2,tfn.getText());
ps.setString(3,tadd.getText());
ps.setInt(4,Integer.parseInt(tph.getText()));
ps.setString(5,tcity.getText());
ps.setString(6,cstate.getSelectedItem());
ps.setString(7,cbford.getSelectedItem()+"/"+cbform.getSelectedItem()+"/"+cbfory.getSe
lectedItem());
ps.setString(8,cbd.getSelectedItem());
ps.setInt(9,Integer.parseInt(tprice.getText()));
ps.executeUpdate();
con.close();
}
catch(Exception exc){
System.out.println(exc);}
}
}

//Check in Page
class hmcin extends JFrame implements ActionListener
{
JLabel lt,lcin;
JButton bback;

```

```

hmcin()
{
setTitle("Hotel Swastik:Check in");
setLayout(null);
setSize(1024,700);

bback=new JButton("Back");
bback.setBounds(500,460,200,30);
bback.addActionListener(this);
add(bback);

setVisible(true);
}
public void actionPerformed(ActionEvent e)
{
if(e.getSource()==bback)
{setVisible(false);new hm3(); }
}
}

//Checkout Page
class hmcout extends JFrame implements ActionListener
{
JLabel lt,lcin;
JButton bback;
hmcout()
{
setTitle("Hotel Swastik:Check out");
setLayout(null);
setSize(1024,700);

bback=new JButton("Back");
bback.setBounds(500,460,200,30);
bback.addActionListener(this);
add(bback);

setVisible(true);
}
public void actionPerformed(ActionEvent e)
{
if(e.getSource()==bback)
{setVisible(false);new hm3(); }
}
}

// Restraunt Page
class hmres extends JFrame implements ActionListener,ItemListener
{
JLabel
l1t,l1,litem,liprice,lqty,ls1,ls2,ls3,ls4,ls5,lsam,lat,lchat,lc,lm,lpc,lt,ldosa,lppak
,lbeer,lw,lr,lid,lsa,lsd,lbs,lms,lss,ltea,lcf,lbill;
Checkbox csam,cat,chat,cppak;
Checkbox cc,cm,ct,cpc;
Checkbox cdosa,cid,csa;

```

```

Checkbox csd,cbs,cms,css,ctea,ccf;
Checkbox cbeer,cw,cr;
JButton bback,bbill;
JTextField
tsam,tat,tchat,tpak,tc,tm,tpc,tt,tdosa,tid,tsa,tsd,tr,tw,tbs,tms,tss,ttea,tcf,tbeer;
JTextField txtbill;
int a, i=50,j=390,k=70,l=20;
int p[],q=0;
hmres()
{
setTitle("Hotel Swastik:Restraunt");
setLayout(null);
setSize(1024,700);

Font f =new Font("algerian",Font.BOLD,30);
l1t=new JLabel("Restraunt");
l1t.setBounds(125,50,300,40);
l1t.setFont(f);
add(l1t);

ImageIcon i1= new ImageIcon("C:\\Users\\Priyankar Jain\\Desktop\\Hotel Management
System\\res.jpg");
JLabel l1=new JLabel("",i1,FlowLayout.LEFT);
l1.setBounds(550,80,380,212);
add(l1);

lbill=new JLabel("Total Bill");
lbill.setBounds(550,310,190,30);
add(lbill);

txtbill=new JTextField(20);
txtbill.setBounds(740,310,190,30);
add(txtbill);
bbill=new JButton("Bill");
bbill.setBounds(550,360,190,30);
add(bbill);
bbill.addActionListener(this);

Font f1 =new Font("microsoft sans serif",Font.BOLD,16);
litem=new JLabel("Item");
litem.setBounds(50,100,150,30);
litem.setFont(f1);
liprice=new JLabel("Price(in Rs)");
liprice.setBounds(240,100,150,30);
liprice.setFont(f1);
lqty=new JLabel("Quantity");
lqty.setBounds(390,100,150,30);
lqty.setFont(f1);
add(litem);add(liprice);add(lqty);
ls1=new JLabel("indian ");
ls1.setBounds(50,130,100,30);
ls1.setFont(f1);
add(ls1);

```

```

ls2=new JLabel("Chinese");
ls2.setBounds(50,235,100,30);
ls2.setFont(f1);
add(ls2);
ls3=new JLabel("South Indian ");
ls3.setBounds(50,340,100,30);
ls3.setFont(f1);
add(ls3);
csam=new Checkbox("Samosa");
csam.setBounds(50,155,100,20);
add(csam);
cat=new Checkbox("Aaloo Tikki");
cat.setBounds(50,175,100,20);
add(cat);
chat=new Checkbox("Chat");
chat.setBounds(50,195,100,20);
add(chat);
cppak=new Checkbox("Paneer Pakoda");
cppak.setBounds(50,215,100,20);
add(cppak);
cc=new Checkbox("Chowmine");
cc.setBounds(50,260,100,20);
add(cc);
cm=new Checkbox("Veg Momos");
cm.setBounds(50,280,100,20);
add(cm);
cpc=new Checkbox("Paneer Chowmine");
cpc.setBounds(50,300,120,20);
add(cpc);
ct=new Checkbox("Thupka");
ct.setBounds(50,320,100,20);
add(ct);

cdosa=new Checkbox("Dosa");
cdosa.setBounds(50,365,100,20);
add(cdosa);
cid=new Checkbox("Idlee");
cid.setBounds(50,385,100,20);
add(cid);
csa=new Checkbox("Samber");
csa.setBounds(50,405,100,20);
add(csa);

ls4=new JLabel("Drinks & Shakes ");
ls4.setBounds(50,425,100,30);
ls4.setFont(f1);
add(ls4);
csd=new Checkbox("Soft Drink");
csd.setBounds(50,450,100,20);
add(csd);
cbs=new Checkbox("Banana Shake");
cbs.setBounds(50,470,180,20);
add(cbs);
cms=new Checkbox("Mango Shake");

```

```

cms.setBounds(50,490,100,20);
add(cms);
css=new Checkbox("Strawbery Shake");
css.setBounds(50,510,100,20);
add(css);
ctea=new Checkbox("Tea");
ctea.setBounds(50,530,100,20);
add(ctea);
ccf=new Checkbox("Coffee");
ccf.setBounds(50,550,100,20);
add(ccf);

ls5=new JLabel("Liquor");
ls5.setBounds(50,570,100,30);
ls5.setFont(f1);
add(ls5);

cbeer=new Checkbox("Beer");
cbeer.setBounds(50,595,100,20);
add(cbeer);
cw=new Checkbox("Whisky");
cw.setBounds(50,615,100,20);
add(cw);
cr=new Checkbox("Rum");
cr.setBounds(50,635,100,20);
add(cr);

lsam=new JLabel(" 20.00");
lsam.setBounds(240,155,100,20);
add(lsam);
lat=new JLabel(" 20.00");
lat.setBounds(240,175,100,20);
add(lat);
lchat=new JLabel(" 50.00");
lchat.setBounds(240,195,100,20);
add(lchat);
lppak=new JLabel(" 50.00");
lppak.setBounds(240,215,100,20);
add(lppak);
lc=new JLabel(" 70.00");
lc.setBounds(240,260,100,20);
add(lc);
lm=new JLabel(" 50.00");
lm.setBounds(240,280,100,20);
add(lm);
lpc=new JLabel(" 100.00");
lpc.setBounds(240,300,100,20);
add(lpc);
lt=new JLabel(" 50.00");
lt.setBounds(240,320,100,20);
add(lt);
ldosa=new JLabel(" 50.00");
ldosa.setBounds(240,365,100,20);
add(ldosa);
lid=new JLabel(" 50.00");

```

```

lid.setBounds(240,385,100,20);
add(lid);
lsa=new JLabel(" 70.00");
lsa.setBounds(240,405,100,20);
add(lsa);
lsd=new JLabel(" 50.00");
lsd.setBounds(240,450,100,20);
add(lsd);
lbs=new JLabel(" 20.00");
lbs.setBounds(240,470,100,20);
add(lbs);
lms=new JLabel(" 30.00");
lms.setBounds(240,490,100,20);
add(lms);
lss=new JLabel(" 30.00");
lss.setBounds(240,510,100,20);
add(lss);
ltea=new JLabel(" 20.00");
ltea.setBounds(240,530,100,20);
add(ltea);
lcf=new JLabel(" 30.00");
lcf.setBounds(240,550,100,20);
add(lcf);
lbeer=new JLabel(" 80.00");
lbeer.setBounds(240,595,100,20);
add(lbeer);
lw=new JLabel(" 500.00");
lw.setBounds(240,615,100,20);
add(lw);
lr=new JLabel(" 300.00");
lr.setBounds(240,635,100,20);
add(lr);
tsam=new JTextField(i);
tsam.setBounds(j,155,k,l);
add(tsam);
tat=new JTextField(i);
tat.setBounds(j,175,k,l);
add(tat);
tchat=new JTextField(i);
tchat.setBounds(j,195,k,l);
add(tchat);
tppak=new JTextField(i);
tppak.setBounds(j,215,k,l);
add(tppak);
tc=new JTextField(i);
tc.setBounds(j,260,k,l);
add(tc);
tm=new JTextField(i);
tm.setBounds(j,280,k,l);
add(tm);
tpc=new JTextField(i);
tpc.setBounds(j,300,k,l);
add(tpc);
tt=new JTextField(i);
tt.setBounds(j,320,k,l);

```

```

add(tt);
tdosa=new JTextField(i);
tdosa.setBounds(j,365,k,1);
add(tdosa);
tid=new JTextField(i);
tid.setBounds(j,385,k,1);
add(tid);
tsa=new JTextField(i);
tsa.setBounds(j,405,k,1);
add(tsa);
tsd=new JTextField(i);
tsd.setBounds(j,450,k,1);
add(tsd);
tbs=new JTextField(i);
tbs.setBounds(j,470,k,1);
add(tbs);
tms=new JTextField(i);
tms.setBounds(j,490,k,1);
add(tms);
tss=new JTextField(i);
tss.setBounds(j,510,k,1);
add(tss);
ttea=new JTextField(i);
ttea.setBounds(j,530,k,1);
add(ttea);
tcf=new JTextField(i);
tcf.setBounds(j,550,k,1);
add(tcf);
tbeer=new JTextField(i);
tbeer.setBounds(j,595,k,1);
add(tbeer);
tw=new JTextField(i);
tw.setBounds(j,615,k,1);
add(tw);
tr=new JTextField(i);
tr.setBounds(j,635,k,1);
add(tr);

bback=new JButton("Back");
bback.setBounds(740,360,190,30);
bback.addActionListener(this);
add(bback);
csam.addItemListener(this);

setVisible(true);
}
public void actionPerformed(ActionEvent e)
{
if(e.getSource()==bback)
{setVisible(false);new hm3(); }
if(e.getSource()==bbill)
{

if((csam.getState())==true)
{

```

```

int z;
z=Integer.parseInt(tsam.getText());
p[0]=20*z;
txtbill.setText(String.valueOf(p[0]));
//q=q+1;

}
if((cat.getState()==true)
{
}
if((chat.getState()==true)
{
}
if((cppak.getState()==true)
{
}
if((cc.getState()==true)
{
}
if((cm.getState()==true)
{
}
if((cpc.getState()==true)
{
}
if((ct.getState()==true)
{
}
if((cdosa.getState()==true)
{
}
if((cid.getState()==true)
{
}
if((csa.getState()==true)
{
}
if((csd.getState()==true)
{
}
if((cbs.getState()==true)
{
}
if((cms.getState()==true)
{
}
if((css.getState()==true)
{
}
if((ctea.getState()==true)
{
}
if((ccf.getState()==true)
{
}
}

```



```
if((cbeer.getState())==true)
{
}
if((cw.getState())==true)
{
}
if((cr.getState())==true)
{
}
}
}
}
public void itemStateChanged(ItemEvent x)
{

}
}
```

Chapter 5

5.1 DATABASE DESIGN

5.1.1 ROOM BOOKING

FIELD NAME	TYPE	SIZE
CUSTOMER NAME	VARCHAR2	25
FATHER NAME	VARCHAR2	25
ADDRESS	VARCHAR2	30
PHONE	NUMBER	15
CITY	VARCHAR2	15
STATE	VARCHAR2	15
DATE OF ROOM BOOK	DATE	-
ROOM TYPE	VARCHAR2	10
PRICE	NUMBER	10
ROOM NO.	NUMBER	5

5.1.2 ROOMS

FIELD NAME	TYPE	SIZE
ROOM NO.	NUMBER	5
TYPE	VARCHAR2	10
STATUS	VARCHAR2	3

5.1.3 BILL

FIELD NAME	TYPE	SIZE
ROOM NO.	NUMBER	5
BILL TYPE	VARCHAR2	10
AMOUNT	NUMBER	10
DATES	DATE	-

5.1.4 RESTAURANT ITEMS

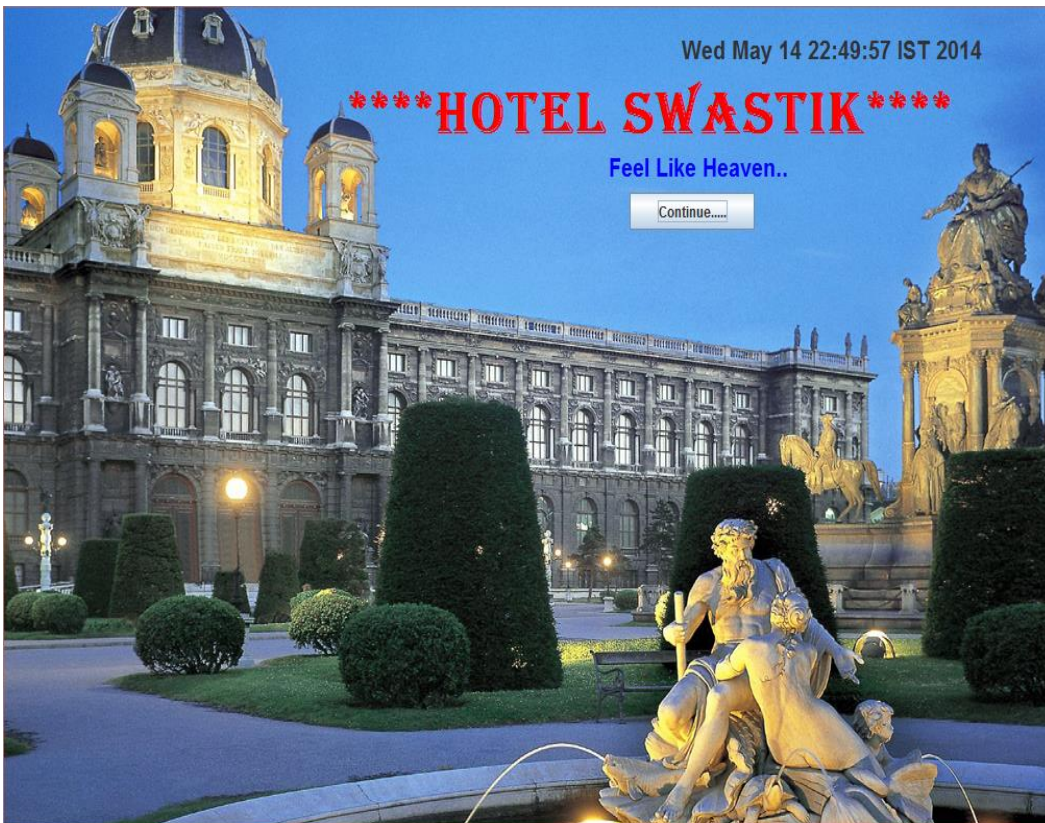
FIELD NAME	TYPE	SIZE
ITEM	VARCHAR2	10
PRICE	NUMBER	10

Chapter 6

6.1 PROJECT EXECUTION

6.1.1 WELCOME SCREEN

This is the welcome page of HOTEL SWASTIK. To proceed further, press the button CONTINUE.

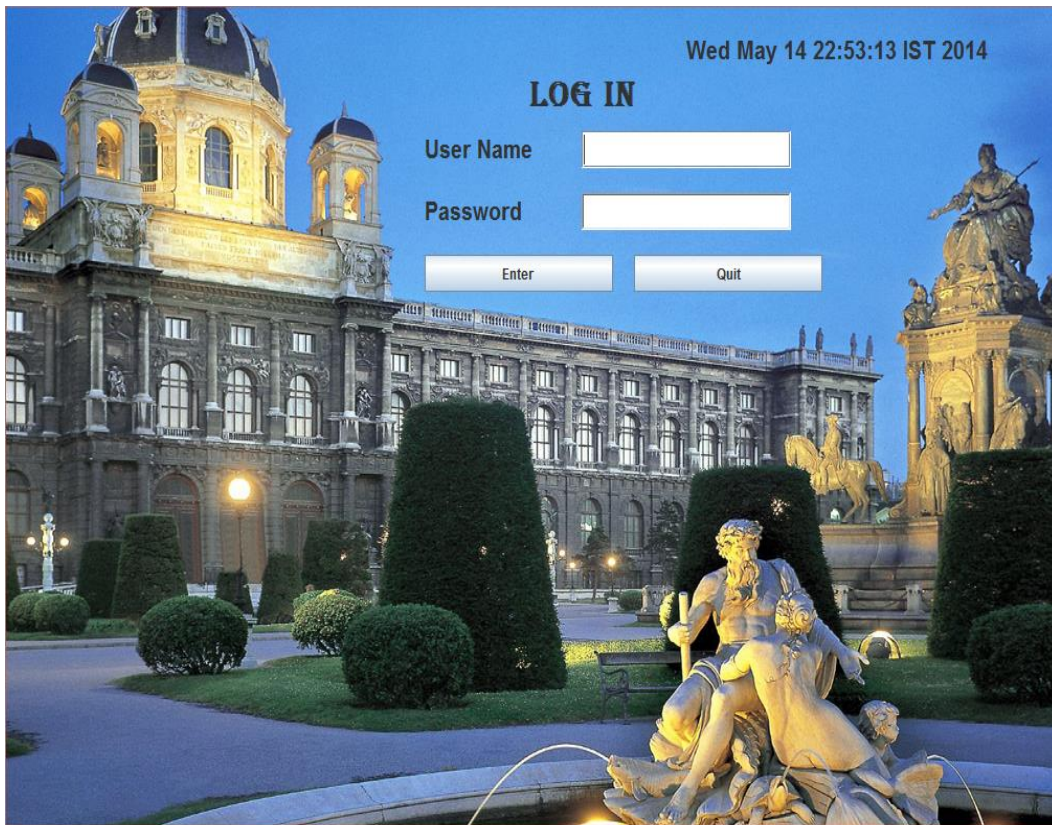


6.1.2 LOGIN SCREEN

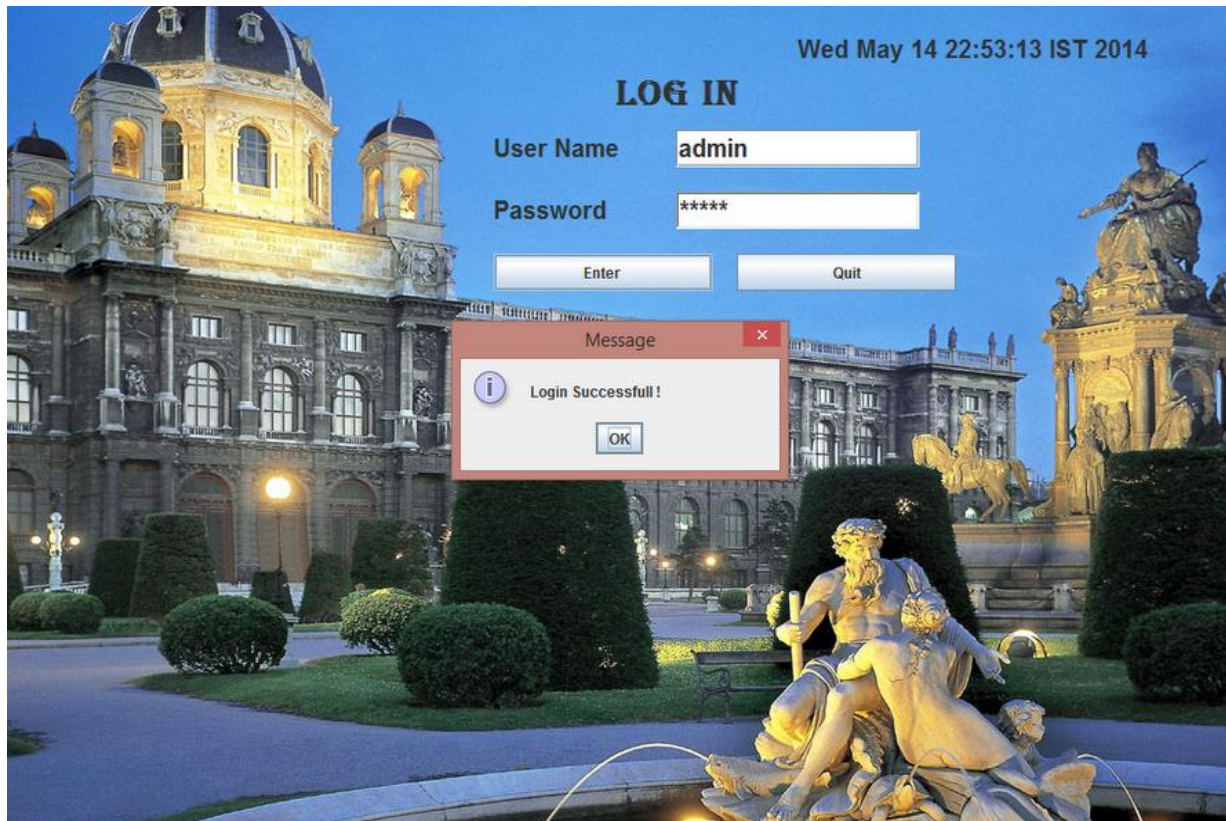
Available to the hotel manager...

Prompts for the name and password of the administrator for authentication purpose.

The username and password are already fed in the database in table called login.



- After filling the correct username and password, and pressing ENTER button on the screen you get a message “Login Successful” as follows, otherwise you will get a message “username and password not correct”.



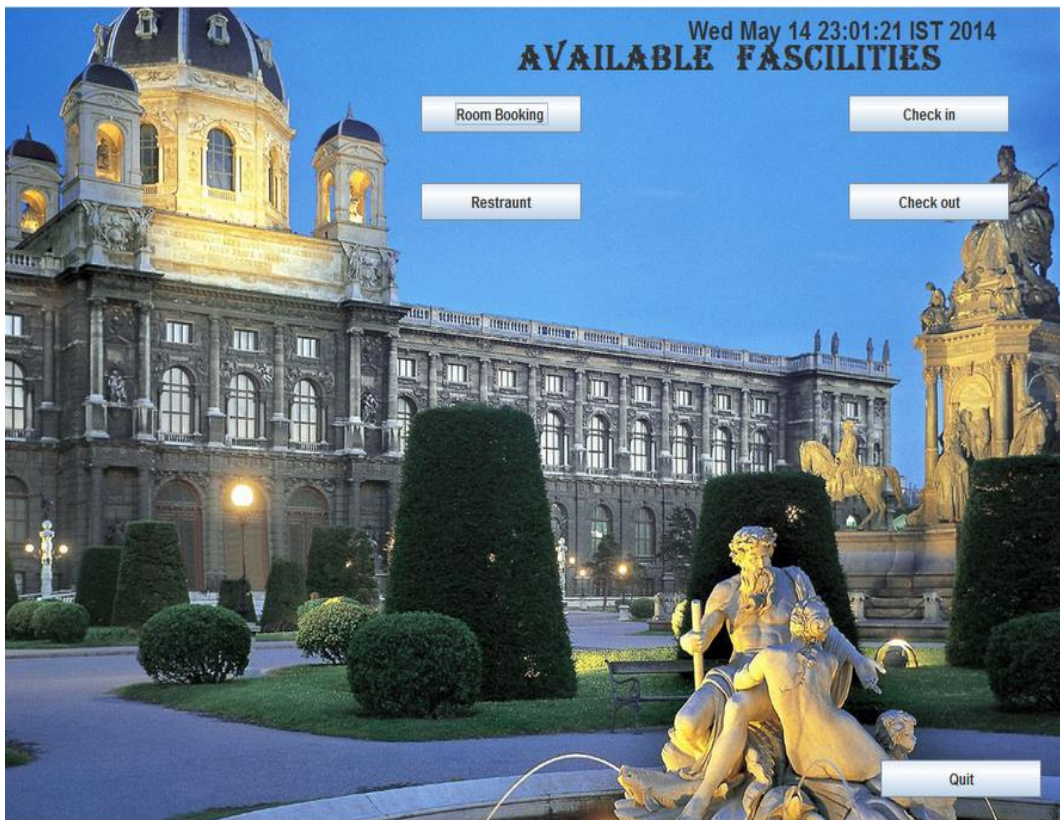
After pressing “OK”, AVAILABLE FACILITIES SCREEN will be opened..

6.1.3 Available Facilities Screen

This screen shows four options:

- Room booking
- Restaurant
- Check in
- Check out

Click any of the above buttons as per the requirement.



There is one QUIT button available at the right bottom corner, to quit from the screen.


6.1.4 Room Booking Screen

On clicking <Room Booking> button , a new screen will be displayed containing following text fields:

- ✓ Name of Customer
- ✓ Father Name
- ✓ Address
- ✓ Phone No.
- ✓ City
- ✓ State
- ✓ Date for room booked
- ✓ Room type
- ✓ Price
- ✓ Room no.

Wed May 14 23:03:10 IST 2014

ROOM BOOKING

Name of Customer:	<input type="text"/>	
Father Name:	<input type="text"/>	
Address:	<input type="text"/>	
Phone no:	<input type="text"/>	
City:	<input type="text"/>	
State:	2 <input type="text"/>	
Date for Room Booked:	1 <input type="text"/> 1 <input type="text"/> 2000 <input type="text"/>	
Room Type:	Single Bedroom <input type="text"/>	
Price:	<input type="text"/>	
Room no:	<input type="text"/>	

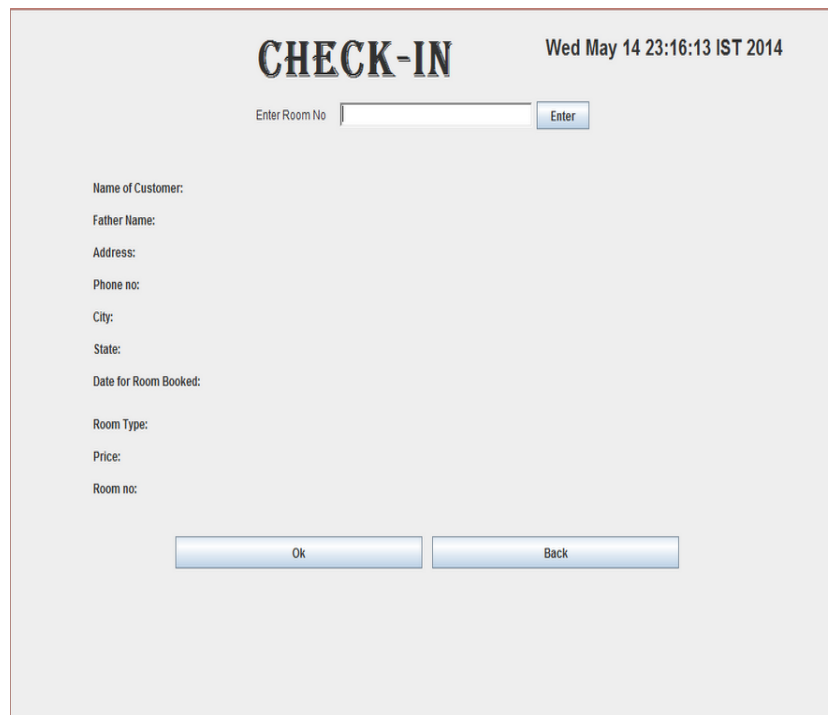
- After filling all the fields, on pressing Enter ,if the room is available, all the entries are sent to the database(MS Access) and saved over there and room will be allotted with a message “Congratz, your room no. is < room no>”,if its not available ,a message “sorry, room is not available” is shown.
- To go back to available fascilities screen, press ‘Back’.

The entries at the database are represented as:

name	fname	add	phone	city	state	drb	rt	price	roomno
Ram Singh	sham	1524,vasant viahr	87687433	mumbai	maharashtra	1/1/2014	Single Bedroom	5000	102
Kishan Kumar	hari kumar	12,raman colony	77857444	delhi	delhi	15/2/2014	Double Bedroom	5000	103
Suresh Jain	manish jain	34,harinagar	33389309	pritampura	delhi	11/3/2014	Single Bedroom	5000	105
Karan Kumar	varun kumar	89,devinagar	51541254	hisar	haryana	4/4/2014	Single Bedroom	5000	104
Sahil Goyal	mohit goyal	354,sector4	68544416	panchkula	haryana	10/5/2014	Double Bedroom AC	15000	210
Rahul Gupta	Rajpal Gupta	house # 16,sec 12	98973782	chandigarh	chandigarh	10/05/2014	Double Bedroom	5000	101
Ramesh Garg	Vijay Garg	12,34	25647858	ambala	haryana	10/5/2014	Single Bedroom AC	10000	106
Sunil Kumar	Suresh Kumar	12	34566633	hisar	haryana	11/15/2014	Single Bedroom AC	10000	107

6.1.5 Check In:

- To get information about any check ins, click Check In button on Available Fascalities and following screen will be displayed.
- Type the room no. in Enter Room No. field, press Enter and you will get the required info about the check in.
- To leave press OK, to go back press Back.



The screenshot displays a web interface titled "CHECK-IN" with a timestamp "Wed May 14 23:16:13 IST 2014". At the top, there is a text input field labeled "Enter Room No." followed by an "Enter" button. Below this, a list of labels is shown, each followed by an empty input field: "Name of Customer:", "Father Name:", "Address:", "Phone no:", "City:", "State:", "Date for Room Booked:", "Room Type:", "Price:", and "Room no:". At the bottom of the form, there are two buttons: "Ok" and "Back".


6.1.6 Restaurant

- To see and order the menu of restaurant of Hotel Mittals, press Restaurant button, and following menu will be shown.
- Just check the box of the required item and mention the quantity, and press Bill.

Wed May 14 23:17:50 IST 2014

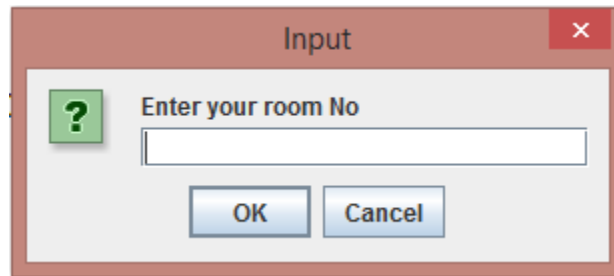
RESTRAUNT

Item	Price(in Rs)	Quantity
indian		
<input type="checkbox"/> Samosa	20.00	<input type="text"/>
<input type="checkbox"/> Aaloo Tikki	20.00	<input type="text"/>
<input type="checkbox"/> Chat	50.00	<input type="text"/>
<input type="checkbox"/> Paneer Pakoda	50.00	<input type="text"/>
Chinese		
<input type="checkbox"/> Chowmine	70.00	<input type="text"/>
<input type="checkbox"/> Veg Momos	50.00	<input type="text"/>
<input type="checkbox"/> Paneer Chowmine	100.00	<input type="text"/>
<input type="checkbox"/> Thupka	50.00	<input type="text"/>
South In...		
<input type="checkbox"/> Dosa	50.00	<input type="text"/>
<input type="checkbox"/> Idlee	50.00	<input type="text"/>
<input type="checkbox"/> Samber	70.00	<input type="text"/>
Drinks &...		
<input type="checkbox"/> Soft Drink	50.00	<input type="text"/>
<input type="checkbox"/> Banana Shake	20.00	<input type="text"/>
<input type="checkbox"/> Mango Shake	30.00	<input type="text"/>
<input type="checkbox"/> Strawberry Shak	30.00	<input type="text"/>
<input type="checkbox"/> Tea	20.00	<input type="text"/>
<input type="checkbox"/> Coffee	30.00	<input type="text"/>
Liquor		
<input type="checkbox"/> Beer	80.00	<input type="text"/>
<input type="checkbox"/> Whisky	500.00	<input type="text"/>
<input type="checkbox"/> Rum	300.00	<input type="text"/>



6.1.7 CHECK OUT

To check out click the check out button, it will ask for your room no.



Press OK, and you will successfully check out.

CONCLUSION

Hotel Management System is a Windows Application based software system specializing in providing custom solutions for small businesses. It provides facilities to the user with user friendly modules with sub modules. This system is developed in understandable approach which can be easier to the layman of the computers. This system is developed totally GUI based and with smart links. The main aim of this project was to manage and systemize the bulky records of the customers who visits in a hotel.

REFERENCES

Bruce Eckel (2003). *Thinking in Java: Third Edition*. Upper Saddle River, New Jersey: Prentice Hall.

Cay S. Horstmann (2000). *Computing Concepts with Java 2: Second Edition*. New York: John Wiley & Sons, Inc

H.M. Deitel & P.J. Deitel (2002). *Java How to Program: Fourth Edition*. Upper Saddle River, New Jersey: Prentice Hall

H.M. Deitel, P.J. Deitel & S.E. Santry (2002). *Advanced Java How to Program: First Edition*. Upper Saddle River, New Jersey: Prentice Hall.

David M. Geary (1999). *Graphic Java 2, Volume II: Swing*. Upper Saddle River, New Jersey: Prentice Hall.

George Reese (1997, June). *Database Programming with JDBC and Java: First Edition*. Sebastopol, CA.

Larry Ullman (2003). *PHP and MySQL for Dynamic Web Sites*. Peachpit Press.

<https://www.w3schools.com>

<https://www.java2s.com>

http://www.tutorialspoint.com/java/java_quick_guide.htm

<https://www.eclipse.org/>