

This is to certify that \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ of class XII \_\_\_\_\_ has completed the project “\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_” during session 2017-18 under my guidance and supervision.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ms. Yashika Budhraja Malhotra

PGT Computer Science

I would like to take this opportunity to express my profound sense of gratitude and respect to all those who helped me throughout the duration of this project. Amity International School, Pushp Vihar in particular has been the source of inspiration for me. I acknowledge the effort of those who have contributed significantly to my project.

I express my sincere gratitude and thankfulness towards **Ms. Yashika Malhotra**, **Head, Department of Computer Science, Amity International School, Pushp Vihar** for her valuable time and guidance throughout this course especially her skillful Teaching, precious suggestions and encouragements.

Last but not the least I would like to mention the name of group mates for their tremendous helping nature. I regret any inadvertent omissions.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Shivangi Gupta

I hereby declare that this project entitled **“RESORT MANAGEMENT SYSTEM”** is done by me at **“Amity International School, Pushp Vihar”** for the partial fulfillment of the requirement for the award of the Class XII Board Examination 2017-18 and no part of the project has been submitted by me for any other purpose

.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Shivangi Gupta

This project entitled *“RESORT MANAGEMENT SYSTEM”* has been designed towards improving the Management of a Resort and its billing system. It maintains room details, booking, cancellation and report generation depending on various criteria. It consists of functionally related GUI (application program) and database. The choice of the programming tools is individual and particular. The project has been an enriching experience for me in the field of programming and Enterprise Application development. Database used in this system is helpful to find out performance of resort, sales per year, and other information. This system is very user-friendly. User with basic computer knowledge can use this system. Databases in Resort management are very dynamic and scalable.



* The main objective of this project is to make Resort Management System simple, reliable, user friendly and corrective. Moreover less time consuming as compared to manual work.
* The basic idea is to reduce time in maintaining guest data, calculating the total amount, availability of rooms and generation of reports.
* This will lead to increase in the efficiency with little throughput.
* The project will reduce the tedious job of paperwork by keeping all the details of guests stored in the “resort” database in computer’s hard disk.
* Up to date information of guest can be provided regarding his personal and travel details.
* Handling inquiries from prospective guests is made very easy, which also makes the guests happy and satisfied.
* It helps in reducing redundancy.

**FRONT END**  : NET BEANS IDE 8.0.2

**BACK END** : MYSQL 5.5 Command line client

**OPERATING SYSTEM**  : WINDOWS 7

This project has been complied using Net Beans 8.0.2 as Front End Graphical User Interface(GUI) and MYSQL Server 5.5 as Back End database.



***Net Beans (Java)***

NetBeans IDE is the Smart Way to Code. It is a GUI programming language which provides the user with an interface that is easy and friendly to use. In developing such interface using Java Net Beans IDE, the programmer employs user friendly features such as windows, menus, buttons and list boxes.

Following are some features of Java Net Beans IDE:-

* **Simple:** Java was designed to be easy to use and is therefore easy to write, compile, debug, and learn than other programming languages. The reason that why Java is much simpler than C++ is because Java uses automatic memory allocation and garbage collection where else C++ requires the programmer to allocate memory and to collect garbage.
* **Object Oriented:** Java is object-oriented because programming in Java is centered on creating objects, manipulating objects, and making objects work together. This allows you to create modular programs and reusable code.
* **Platform Independence:** One of the most significant advantages of Java is its ability to move easily from one computer system to another. The ability to run the same program on many different systems is crucial to World Wide Web software, and Java succeeds at this by being platform-independent at both the source and binary levels.
* **Interpreted:** An interpreter is needed in order to run Java programs. The programs are compiled into Java Virtual Machine code called bytecode. The bytecode is machine independent and is able to run on any machine that has a Java interpreter. With Java, the program need only be compiled once, and the bytecode generated by the Java compiler can run on any platform.
* **Distributed:** Distributed computing involves several computers on a network working together. Java is designed to make distributed computing easy with the networking capability that is inherently integrated into it. Writing network programs in Java is like sending and receiving data to and from a file. For example, the diagram below shows three programs running on three different systems, communicating with each other to perform a joint task.

***MYSQL***

MySQL is characterised as a free, fast, reliable open source relational database. It does lack some sophistication and facilities, but it has an active development team and, as it goes from release to release, more capabilities are added. At certain times there will be a trade-off between speed and capabilities, and the MySQL team intend to keep their database engine fast and reliable.

Following are some features of MySQL command client:

* MySQL is released under an open-source license so it is customizable. It requires no cost or payment for its usage.
* MySQL has superior speed, is easy to use and is reliable.
* MySQL uses a standard form of the well-known ANSI-SQL standards.
* MySQL is a platform independent application which works on many operating systems like Windows, UNIX, LINUX, etc. and has compatibility with many languages including JAVA, C++, PHP, PERL, etc.
* MySQL is an easy to install RDBMS and is capable of handling large data sets.

An entity-relationship diagram (ERD) shows the relationships of entity sets stored in a database. ER diagrams often use symbols to represent three different types of information. Boxes are commonly used to represent entities. Diamonds are normally used to represent relationships and ovals are used to represent attributes.

Considering the example of a database that contains information on the guest’s personal details and travel/stay time according to Resort Management System. The ER Diagram shown in the image below contains entities: Guest, Reservation and Rooms; actions: Book, Reserve, Request; attributes: login id, password, Fname(first name), Lname(last name), e-mail, phone, checkin, type, number, etc.

GUEST

Book

Request

Reserve

Reservations

Rooms

In this project of Resort Management System we have 6 main modules and their sub-modules in Net Beans IDE, which is a general purpose GUI programming language that is powerful, efficient and compact.

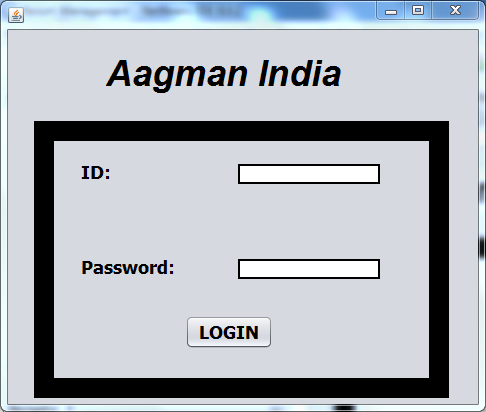
**Module 1:**

Module 1 is the introduction screen displaying a picture of the resort saying “welcome”.



**Module 2:**

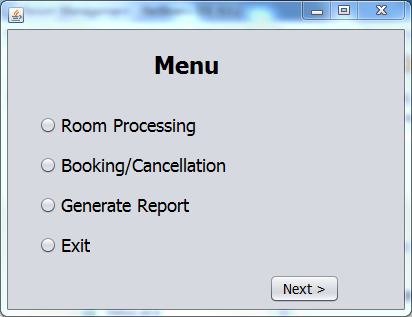
It is an Integrity Screen which ask for Password entry and display a JPassword Field where user enters the password, if it is correct he is allowed to move to the next module otherwise a message “Incorrect Password” is flashed.



**Module 3:**

Module 3 is an opening screen for navigating through the main menu. There are four Radio Buttons-“Room Processing”, “Booking/Cancellation”, “Generate Report”, “Exit”.

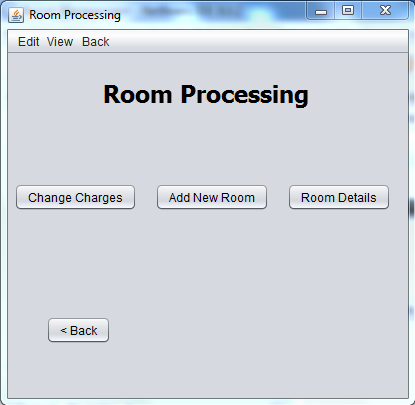
On clicking each of the radio buttons, new frame gets further opened related to their perspectives.



**Module 4:**

Module 4 is for ‘Room Processing’. It has four buttons-“change charges”, “add new rooms”, “room details”, “back”.

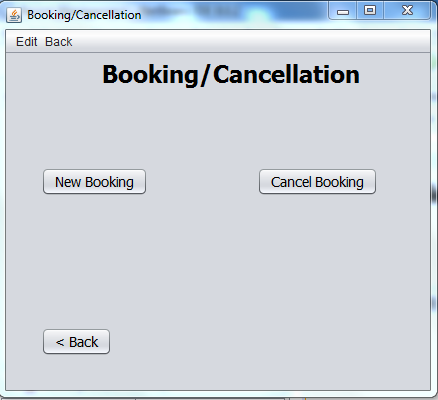
On clicking the first button, charges can be changed. When second button is clicked,new rooms are added, third button gives room details and last button opens the main menu again.



**Module 5:**

Module 5 is for ‘Booking/Cancellation’. It has three buttons-“make booking”, “cancel booking” and “back”.

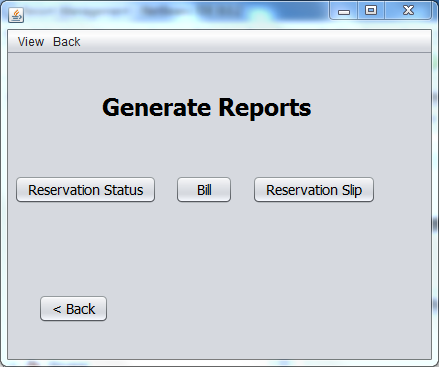
On clicking the first button, booking is made. The second button cancels the booking and last button opens the main menu again.



**Module 6:**

Module 6 is for ‘generating reports’ . it consists of four buttons-“reservation status”, “bill”, “reservation slip” and “back”.

On clicking the first button, the program shows the reservation status, the second button commands for the bill, third button commands to print the reservation slip for the reception.



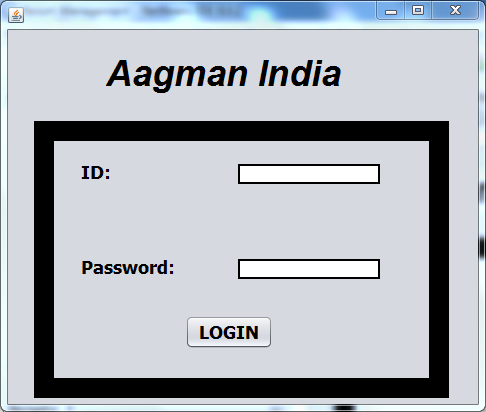


private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

Login l=new Login();

l.setVisible(true);

this.setVisible(false);



private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

String id=jTextField1.getText();

String pword=jPasswordField1.getText();

if (pword.contains("7578")&&(id.contains("Aagman India")))

{

JOptionPane.showMessageDialog(this,"welcome!!");

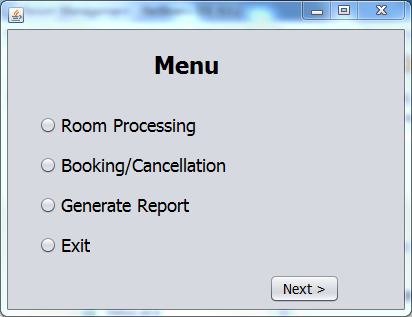
Menu m= new Menu();

m.setVisible(true);

this.setVisible(false);

}

else { JOptionPane.showMessageDialog(this,"Incorrect ID or Password"); }



private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

if (jRadioButton1.isSelected())

{

RoomProcessings rp= new RoomProcessings();

rp.setVisible(true);

this.setVisible(false);

}

else if (jRadioButton2.isSelected())

{

BookingCancellation bc= new BookingCancellation();

bc.setVisible(true);

this.setVisible(false);

}

else if (jRadioButton3.isSelected())

{

ReportGeneration rg= new ReportGeneration();

rg.setVisible(true);

this.setVisible(false);

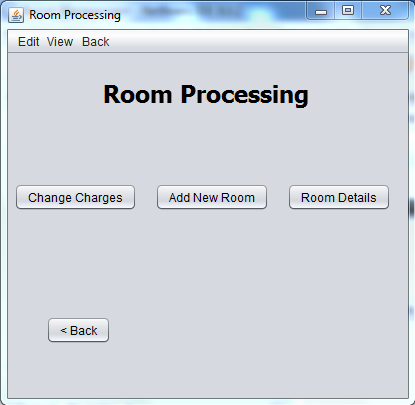
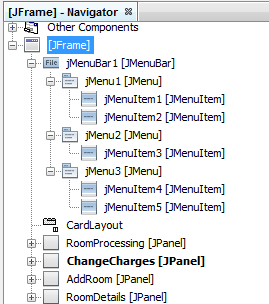
}

else if (jRadioButton4.isSelected())

{

System.exit(0);

}

import java.sql.DriverManager;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.Statement;

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

AddRoom.setVisible(false);

RoomProcessing.setVisible(false);

RoomDetails.setVisible(false);

ChangeCharges.setVisible(true);

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

AddRoom.setVisible(true);

RoomProcessing.setVisible(false);

RoomDetails.setVisible(false);

ChangeCharges.setVisible(false);

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

AddRoom.setVisible(false);

RoomProcessing.setVisible(false);

RoomDetails.setVisible(true);

ChangeCharges.setVisible(false);

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

Menu m= new Menu();

m.setVisible(true);

this.setVisible(false);

private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {

AddRoom.setVisible(false);

RoomProcessing.setVisible(false);

RoomDetails.setVisible(true);

ChangeCharges.setVisible(false);

private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) {

AddRoom.setVisible(false);

RoomProcessing.setVisible(false);

RoomDetails.setVisible(false);

ChangeCharges.setVisible(true);

private void jMenuItem3ActionPerformed(java.awt.event.ActionEvent evt) {

AddRoom.setVisible(true);

RoomProcessing.setVisible(false);

RoomDetails.setVisible(false);

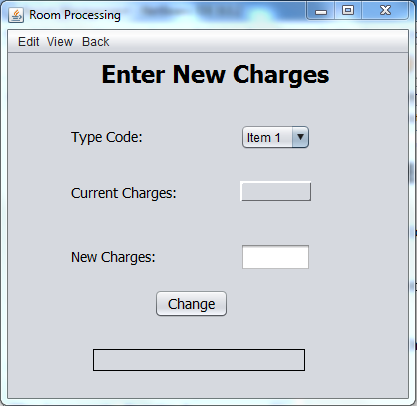
ChangeCharges.setVisible(false);

private void jMenuItem4ActionPerformed(java.awt.event.ActionEvent evt) {

Menu m= new Menu();

m.setVisible(true);

this.setVisible(false);



private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {

String code=jComboBox1.getSelectedItem().toString();

int charges=Integer.parseInt(jTextField1.getText());

try

{

Class.forName("java.sql.DriverManager");

Connection con=(Connection)

DriverManager.getConnection("jdbc:mysql://localhost:3306/resort","root","7578");

Statement stmt=(Statement)con.createStatement();

String sql="update type set charges=" + charges + "where typecode='" + code + "'";

stmt.executeUpdate(sql);

stmt.close();

con.close();

jLabel7.setText("Information Added");

}

catch (Exception e)

{

jLabel7.setText("Invalid Data");

}

private void jComboBox1ActionPerformed(java.awt.event.ActionEvent evt) {

String code=jComboBox1.getSelectedItem().toString();

try

{

Class.forName("java.sql.Driver");

Connection con=(Connection)

DriverManager.getConnection("jdbc:mysql://localhost:3306/resort","root","7578");

Statement stmt=(Statement)con.createStatement();

String sql="select charges from type where typecode='"+code+"'";

ResultSet rs= stmt.executeQuery(sql);

rs.next();

String str=rs.getString("charges");

rs.close();

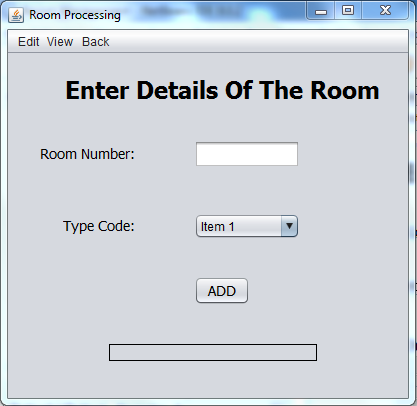
stmt.close();

con.close();

jLabel6.setText("Rs"+str);

}

catch (Exception e) {}



private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {

String roomno=jTextField2.getText();

String code=jComboBox2.getSelectedItem().toString();

try

{

Class.forName("java.sql.DriverManager");

Connection con=(Connection)

DriverManager.getConnection("jdbc:mysql://localhost:3306/resort","root","7578");

Statement stmt=(Statement)con.createStatement();

String sql="insert into room values('"+roomno+"','"+code+"');";

stmt.executeUpdate(sql);

for(int i=1;i<13;i++)

{

String sql2="insert into Status1"+i+"(roomNo,typecode) values('"+roomno+"','"+code+"')";

Statement stmt2=(Statement)con.createStatement();

stmt2.executeUpdate(sql2);

}

jLabel11.setText("Information Added");

stmt.close();

con.close();

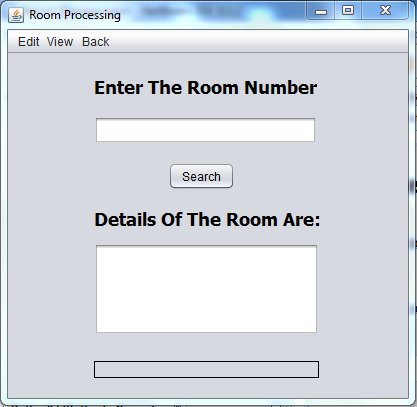
}

catch (Exception e)

{

jLabel11.setText("Invalid Informtion");

}



private void jButton7ActionPerformed(java.awt.event.ActionEvent evt) {

int roomno=Integer.parseInt(jTextField3.getText());

try

{

Class.forName("java.sql.DriverManager");

Connection con=(Connection)

DriverManager.getConnection("jdbc:mysql://locallhost:3306/resort","root","7578");

Statement stmt=(Statement) con.createStatement();

String sql="select type.typecode,description,charges from type,room where roomNo="+roomno+" and type.typecode=room.typecode;";

ResultSet rs= stmt.executeQuery(sql);

rs.next();

String str= rs.getString("typecode");

jTextArea2.setText(str);

String str1=rs.getString("description");

String str2=rs.getString("charges");

jTextArea2.setText("Room Number: "+roomno+ "\nType: "+str+ "\nDescription: "+str1+ "\nCharges: "+str2);

rs.close();

stmt.close();

con.close();

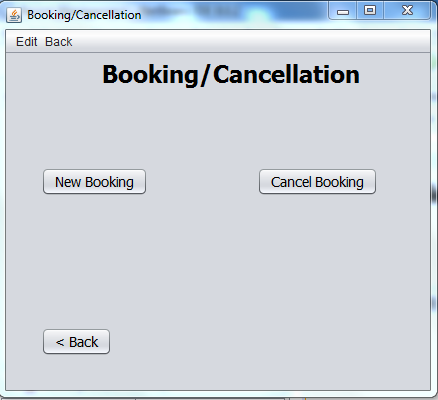
}

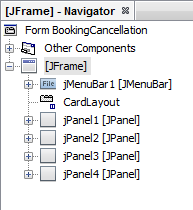
catch (Exception e)

{

jLabel14.setText("Room Number Not Found");

}





private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

jPanel1.setVisible(false);

jPanel2.setVisible(true);

jPanel3.setVisible(false);

jPanel4.setVisible(false);

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

jPanel1.setVisible(false);

jPanel2.setVisible(false);

jPanel3.setVisible(false);

jPanel4.setVisible(true);

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

Menu m= new Menu();

m.setVisible(true);

this.setVisible(false);



private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

jLabel6.setText("");

boolean found=false;

jButton5.setVisible(false);

int roomno=0;

int Month=Integer.parseInt(month.getSelectedItem().toString());

aMonth=Month;

int Month2=Integer.parseInt(month1.getSelectedItem().toString());

dMonth=Month2;

int Day=Integer.parseInt(date.getSelectedItem().toString());

aDay=Day;

int Day2=Integer.parseInt(date1.getSelectedItem().toString());

dDay=Day2;

char avail='n';

String code=jComboBox1.getSelectedItem().toString();

roomno=search(Month,Month2,Day,Day2,code);

if(roomno==0)

jLabel6.setText("No Room Available");

else

{

jLabel6.setText("Room Number: "+roomno+"is available. Click Continue to confirm the booking");

jButton5.setVisible(true);

jLabel8.setText(roomno+"");

jTextField4.setText("2017"+"/"+Month+"/"+Day);

jTextField5.setText("2017"+"/"+Month2+"/"+Day2);

}

Calendar c=Calendar.getInstance();

c.set(2017,Month,Day);

Calendar c2=Calendar.getInstance();

c2.set(2017,Month2,Day2);

stayTime=(int)((c2.getTimeInMillis()-c.getTimeInMillis())/(1000\*60\*60\*24));

private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {

String code=jComboBox1.getSelectedItem().toString();

try {

Class.forName("java.sql.DriverManager");

Connection con=(Connection)

DriverManager.getConnection("jdbc:mysql://localhost:3306/resort","root","7578");

Statement stmt=(Statement)con.createStatement();

String sql="select charges from type where typecode='"+code+"'";

ResultSet rs=stmt.executeQuery(sql);

rs.next();

int charge=rs.getInt("charges");

charge=charge\*stayTime;

jLabel10.setText(charge+ "");

rs.close();

stmt.close();

con.close();

}

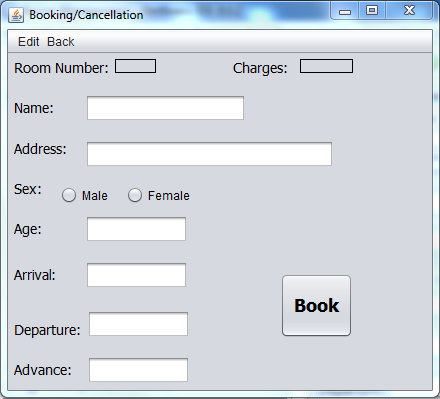
catch (Exception e) {

}

jPanel2.setVisible(false);

jPanel4.setVisible(false);

jPanel3.setVisible(true);



private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {

char gender='n';

int Month=aMonth;

if(jRadioButton1.isSelected())

gender='m';

else if(jRadioButton2.isSelected())

gender='f';

int age=Integer.parseInt(jTextField3.getText());

int roomno=Integer.parseInt(jLabel8.getText());

int advance=Integer.parseInt(jTextField6.getText());

int total=Integer.parseInt(jLabel10.getText());

try {

Class.forName("java.sql.DriverManager");

Connection con=(Connection)

DriverManager.getConnection("jdbc:mysql://localhost:3306/resort","root","7578");

Statement stmt2=(Statement)con.createStatement();

String sql2="select max(bookingNo) from booking";

ResultSet rs=stmt2.executeQuery(sql2);

rs.next();

int bno=rs.getInt(1)+1;

Statement stmt=con.createStatement();

String sql="insert into booking values("+bno+",'"+jTextField1.getText()+"','"+jTextField2.getText()+"','"+gender+"',"+age+","+roomno+",'"+jTextField4.getText()+"','"+jTextField5.getText()+"',"+advance+","+total+")";

stmt.executeUpdate(sql);

while (Month<=dMonth) {

int max=getmax(Month,dMonth,dDay);

int min=0;

if (Month==aMonth)

min=aDay;

else

min=1;

for(int d=min;d<=max;d++){

String sql3="update status"+Month+"set d"+d+"='b' where roomNo="+roomno;

stmt.executeUpdate(sql3);

}

Month++;

}

JOptionPane.showMessageDialog(null,"Room Booked. Booking Numbr is: "+bno);

stmt.close();

stmt.close();

con.close();

new ReservationSlip(bno).setVisible(true);

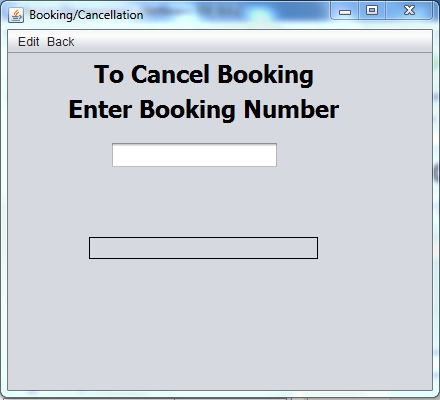
this.setVisible(false);

}

catch(Exception e){

JOptionPane.showMessageDialog(null,"Invalid data "+e);

}



private void jButton7ActionPerformed(java.awt.event.ActionEvent evt) {

jButton8.setVisible(false);

int bno=Integer.parseInt(jTextField7.getText());

try

{

Class.forName("java.sql.DriverManager");

Connection con=(Connection)

DriverManager.getConnection("jdbc:mysql://localhost:3306/resort","root","7578");

Statement stmt=(Statement)con.createStatement();

String sql="select roomNo,fromDate,toDate from booking where bookingNo="+bno;

ResultSet rs=stmt.executeQuery(sql);

if(rs.next())

{

int roomno=rs.getInt("roomNo");

Calendar c=Calendar.getInstance();

Calendar c2=Calendar.getInstance();

c.setTime(rs.getDate(2));

c2.setTime(rs.getDate(3));

int Month=c.get(Calendar.MONTH)+1;

int Month2=c2.get(Calendar.MONTH)+1;

int Day=c.get(Calendar.DATE);

int Day2=c2.get(Calendar.DATE);

while(Month<=Month2)

{

int max=getmax(Month,Month2,Day2);

int min=0;

if(Month==Month2)

min=Day;

else

min=1;

for(int d=min;d<=max;d++){

String sql3="update status"+Month+"set d"+d+"='f'where roomNo="+roomno;

stmt.executeUpdate(sql3);

}

Month++;

}

jLabel20.setText("Click continue to generate cancellation slip");

jButton8.setVisible(true);

}

else {

jLabel20.setText("Unable to find booking number");

}

stmt.close();

con.close();

}

catch(Exception e)

{

jLabel20.setText("Unable to find booking number");

JOptionPane.showMessageDialog(this,"Invalid data " + e);

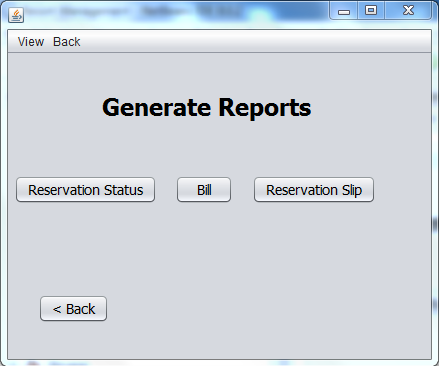
}

private void jButton8ActionPerformed(java.awt.event.ActionEvent evt) {

int bookingNo=Integer.parseInt(jTextField7.getText());

new CancellationSlip(bookingNo).setVisible(true);

this.setVisible(false);



private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

jPanel1.setVisible(false);

jPanel2.setVisible(true);

jPanel3.setVisible(false);

jPanel4.setVisible(false);

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

jPanel1.setVisible(false);

jPanel2.setVisible(false);

jPanel3.setVisible(false);

jPanel4.setVisible(true);

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

jPanel1.setVisible(false);

jPanel2.setVisible(false);

jPanel3.setVisible(true);

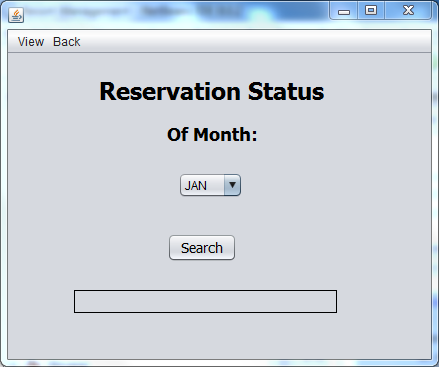
jPanel4.setVisible(false);

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

Menu m= new Menu();

m.setVisible(true);

this.setVisible(false);

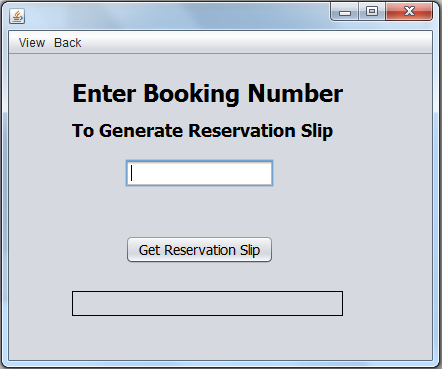


private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {

int month=Integer.parseInt(jComboBox1.getSelectedItem().toString());

new Status(month).setVisible(true);

this.setVisible(false);



private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {

int bno=Integer.parseInt(jTextField1.getText());

if(search(bno))

{

new ReservationSlip(bno).setVisible(true);

this.setVisible(false);

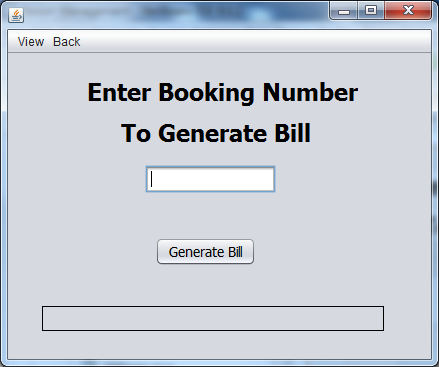
}

else

{

jLabel7.setText("unable to find booking number");

}



private void jButton7ActionPerformed(java.awt.event.ActionEvent evt) {

int bno=Integer.parseInt(jTextField2.getText());

if(search(bno))

{

new Bill(bno).setVisible(true);

this.setVisible(false);

}

else

{

jLabel10.setText("Unable to find booking number");

}

mysql> show tables;

+------------------+

| Tables\_in\_resort |

+------------------+

| booking |

| login |

| room |

| status1 |

| status2 |

| status3 |

| status4 |

| status5 |

| type |

+------------------+

9 rows in set (0.00 sec)

1. Booking Table

This table stores the details of all the bookings made. This table has following structure:

+-----------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+-----------+-------------+------+-----+---------+-------+

| bookingNo | int(11) | NO | PRI | NULL | |

| custName | varchar(20) | YES | | NULL | |

| address | varchar(30) | YES | | NULL | |

| sex | char(1) | YES | | NULL | |

| age | int(11) | YES | | NULL | |

| roomNo | int(11) | YES | | NULL | |

| fromDate | date | YES | | NULL | |

| toDate | date | YES | | NULL | |

| advance | int(11) | YES | | NULL | |

| total | int(11) | YES | | NULL | |

+-----------+-------------+------+-----+---------+-------+

2. Status Tables (status1, stauts2,….status5)

These are 5 tables to store reservation status of five rooms respectively . all tables have the following structure:

+----------+----------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+----------+----------+------+-----+---------+-------+

| roomNo | int(11) | NO | PRI | NULL | |

| typecode | char(20) | YES | | NULL | |

| D1 | char(1) | YES | | NULL | |

| D2 | char(1) | YES | | NULL | |

| D3 | char(1) | YES | | NULL | |

| D4 | char(1) | YES | | NULL | |

| D5 | char(1) | YES | | NULL | |

| D6 | char(1) | YES | | NULL | |

| D7 | char(1) | YES | | NULL | |

| D8 | char(1) | YES | | NULL | |

| D9 | char(1) | YES | | NULL | |

| D10 | char(1) | YES | | NULL | |

| D11 | char(1) | YES | | NULL | |

| D12 | char(1) | YES | | NULL | |

| D13 | char(1) | YES | | NULL | |

| D14 | char(1) | YES | | NULL | |

| D15 | char(1) | YES | | NULL | |

| D16 | char(1) | YES | | NULL | |

| D17 | char(1) | YES | | NULL | |

| D18 | char(1) | YES | | NULL | |

| D19 | char(1) | YES | | NULL | |

| D20 | char(1) | YES | | NULL | |

| D21 | char(1) | YES | | NULL | |

| D22 | char(1) | YES | | NULL | |

| D23 | char(1) | YES | | NULL | |

| D24 | char(1) | YES | | NULL | |

| D25 | char(1) | YES | | NULL | |

| D26 | char(1) | YES | | NULL | |

| D27 | char(1) | YES | | NULL | |

| D28 | char(1) | YES | | NULL | |

| D29 | char(1) | YES | | NULL | |

| D30 | char(1) | YES | | NULL | |

| D31 | char(1) | YES | | NULL | |

+----------+----------+------+-----+---------+-------+

3. Login Table

This table stores the details of all valid login details. This table has following structure:

+----------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+----------+-------------+------+-----+---------+-------+

| ID | varchar(20) | NO | PRI | NULL | |

| password | varchar(20) | YES | | NULL | |

+----------+-------------+------+-----+---------+-------+

4. Room table

This table stores the details of all the rooms in the resort. This table has following structure:

+----------+------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+----------+------------+------+-----+---------+-------+

| roomNo | int(11) | NO | PRI | NULL | |

| typecode | varchar(5) | YES | | NULL | |

+----------+------------+------+-----+---------+-------+

5. Type table

This table stores the details of room types and their charges. This table has following structure:

+-------------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+-------------+-------------+------+-----+---------+-------+

| typecode | varchar(5) | NO | PRI | NULL | |

| description | varchar(20) | YES | | NULL | |

| charges | int(11) | YES | | NULL | |

+-------------+-------------+------+-----+---------+-------+

At the time of developing this project, I have consulted various books of Java Net Beans IDE, Tutorials from various websites, these are the following:-

**Books:**

* NCERT CLASS XII Informatics Practices
* NCERT CLASS XI Informatics Practices
* Informatics Practices by Sumita Arora TextBook for Class XII

**Web Sites:**

* [http://www.javalobby.org](http://www.javalobby.org/)
* <http://javablogs.com/>
* <http://www.jguru.com/>
* <http://www.wikipedia.org/>

* **IDE:** An integrated development environment (IDE) also known as integrated design environment or integrated debugging environment is a [software application](http://en.wikipedia.org/wiki/Software_application) that provides comprehensive facilities to [computer programmers](http://en.wikipedia.org/wiki/Computer_programmer) for [software development](http://en.wikipedia.org/wiki/Software_development).
* **Window:** A window is an area on the screen that displays information for a specific program.
* **GUI:** It refers to the graphical interface of a computer that allows users to click and drag objects with a mouse instead of entering text at a command line.
* **Entity:** An entity is something that has a distinct, separate [existence](http://en.wikipedia.org/wiki/Existence), although it need not be a material existence.
* **Attribute:** Are represented by ovals. A key attribute is the unique, distinguishing characteristic of the entity
* **Frame:** A Frame is a top-level window with a title and a border. It is a container control that contains other controls inside it.
* **parseInt():**parseInt() is a method of Integer class which converts String to integer. If we want to enter a integer in a method or class using keyboard, then we have to use a method parseInt().
* **Table:** Table is a grid of rows and columns with information stored inside it. Columns of a table is also called ‘degree’ and the rows in a table is also called ‘cardinality’ in mysql.
* **Relational database:** A database in which the data is stored in the form of relations(also called tables) is called a relational database. In other words a relational Database is a collection of one or more tables.
* **Transaction:** A transaction is a unit o work that must be done in logical order and successfully as a group or not done at all.