





**lational model as introduced by E. F. Codd.**

1. **INTRODUCTION TO MYSQL**

RDBMS stands for **R**elational **D**atabase **M**anagement **S**ystem. RDBMS is the basis for SQL, and for all modern database systems like MS SQL Server, IBM DB2, Oracle, MySQL, and Microsoft Access.

A Relational database management system (RDBMS) is a database management system (DBMS) that is based on the relational model as introduced by E. F. Codd.

**What is a table ?**

**The data in an RDBMS is stored in database objects which are called as tables. This table is basically a collection of related data entries and it consists of numerous columns and rows.**

**Remember, a table is the most common and simplest form of data storage in a relational database. The following program is an example of a CUSTOMERS table –**

**What is a field ?**

**Every table is broken up into smaller entities called fields. The fields in the CUSTOMERS table consist of ID, NAME, AGE, ADDRESS and SALARY.**

**A field is a column in a table that is designed to maintain specific information about every record in the table.**

**What is a Record or a Row ?**

**A record is also called as a row of data is each individual entry that exists in a table. For example, there are 7 records in the above CUSTOMERS table.**

**2. INTRODUCTION TO**

**NET BEANS IDE 6.7.1**

The NetBeans IDE is an award-winning integrated development environment available for Windows, Mac, Linux, and Solaris. The NetBeans project consists of an [open-source IDE](https://netbeans.org/features/index.html)and an application platform that enable developers to rapidly create web, enterprise, desktop, and mobile applications using the Java platform, as well as JavaFX, PHP, JavaScript and Ajax, Ruby and Ruby on Rails, Groovy and Grails, and C/C++.

The NetBeans project is supported by a vibrant [developer community](https://netbeans.org/community/index.html) and offers extensive [documentation and training](https://netbeans.org/kb/index.html) resources as well as a diverse selection of third-party [plugins](http://plugins.netbeans.org/).

NetBeans IDE 6.7 is integrated with [Project Kenai](http://kenai.com/), a collaborative environment for developers to host open-source projects. With Kenai and the NetBeans IDE, a team of developers is able to create projects, check out, edit, debug, build, discuss, and commit code, all through one easy-to-use interface!

The release also builds on the success of NetBeans 6.5 with native support for Maven; GlassFish, issue tracker and Hudson integrations; and enhancements to Java, PHP, Ruby, Groovy and C/C++. Highlights of the 6.7 release include support for JavaScript 1.7, Ruby Remote Debugging, and integration of the Java ME SDK 3.0.

**3.** Objective of proposed system

**Hotel management system – booking details and room rent details are the main functions of the system we have designed.**

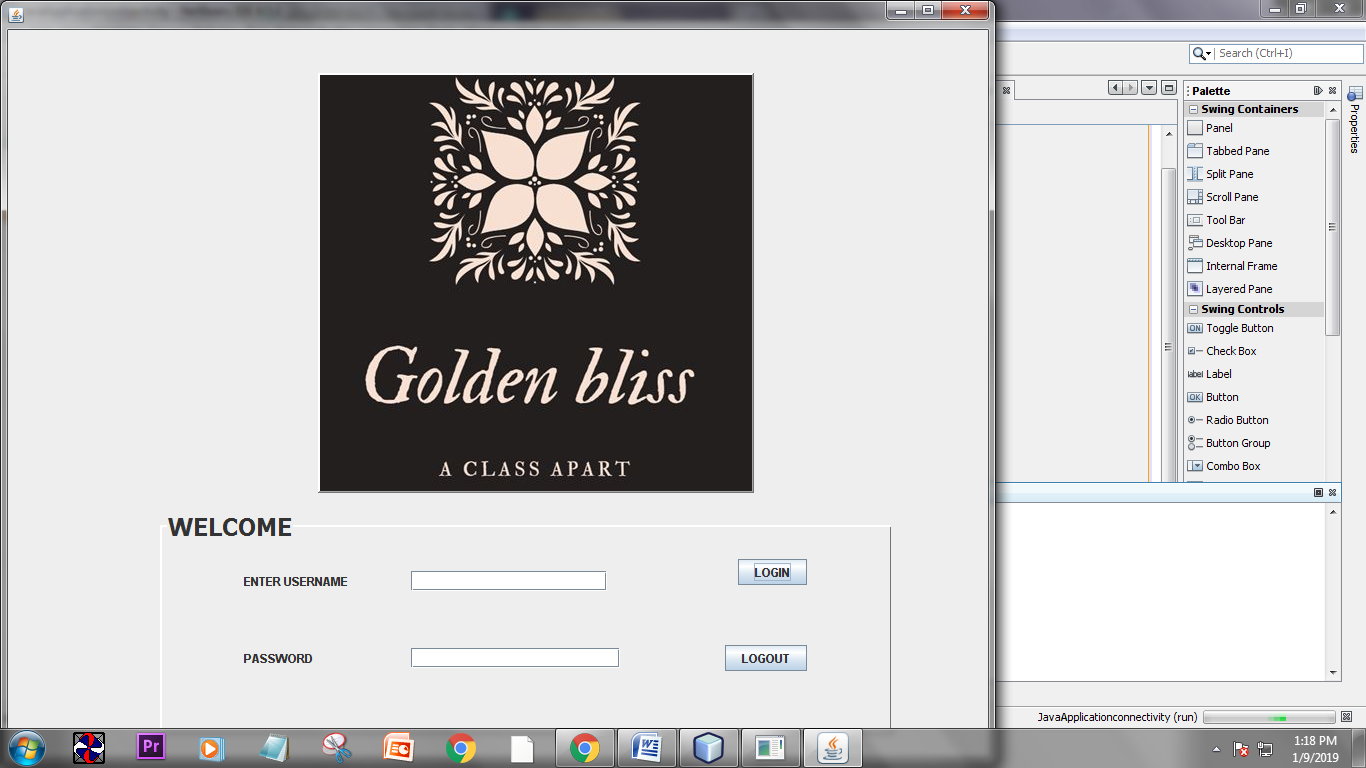
**In this software we will be having a basic functions of hotel management system. Adding of record, deleting of records, updating of records, searching of records. We are having a table in our data base named GOLDEN. Which will have all the necessary details of the customer. There**

**will be a second table in which all the necessary room details will be provided.**

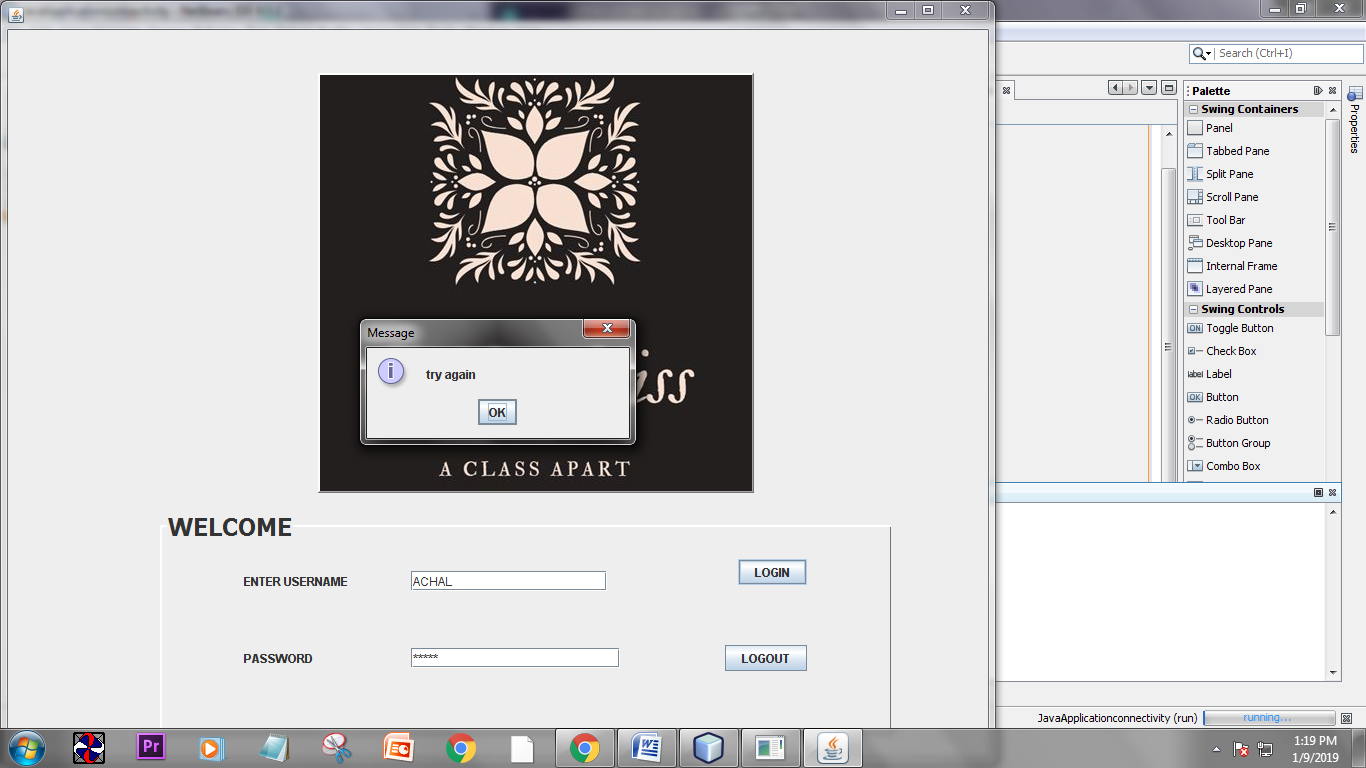
**Our main objective of this project is to show how real life intricacies are used to design a software.**

**4. DESIGN WINDOW**

**Welcome page for the user.**

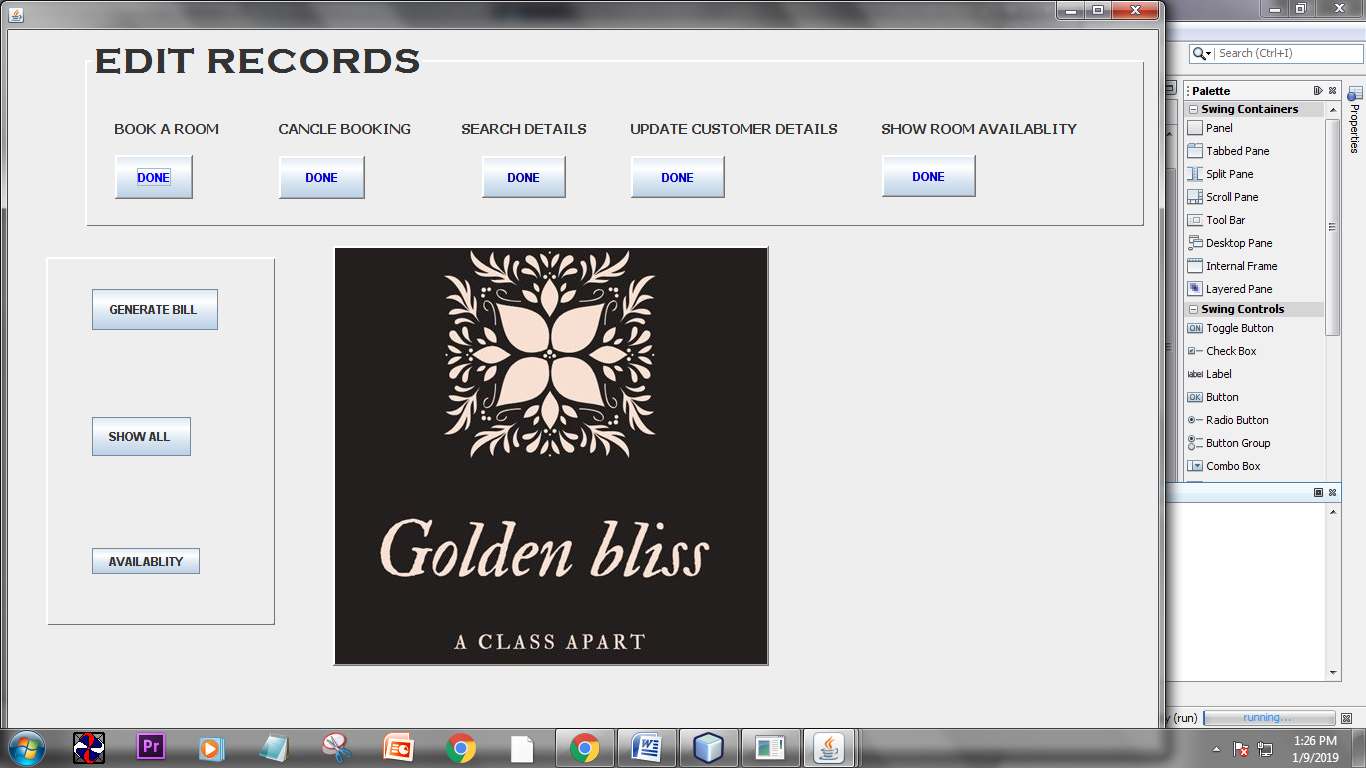
****

**Now when the user gives wrong password.**

****

**When user enters right password.**

**Now when user will click on “ PROCEED TO MENU” then automatically he/she will jump on main menu page.**

****

**As we can see there are eight button for different functions to perform.**

**1. add button**

**2. delete button.**

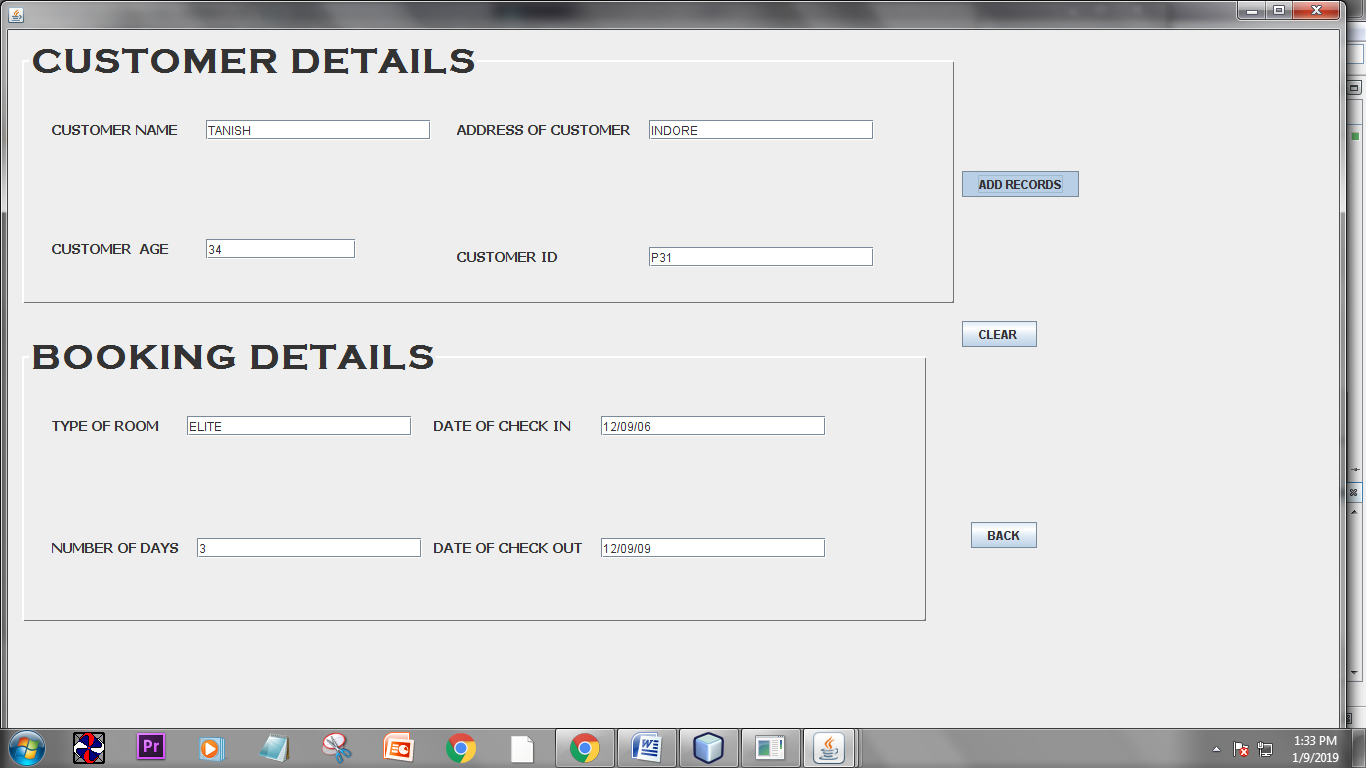
**3. search button.**

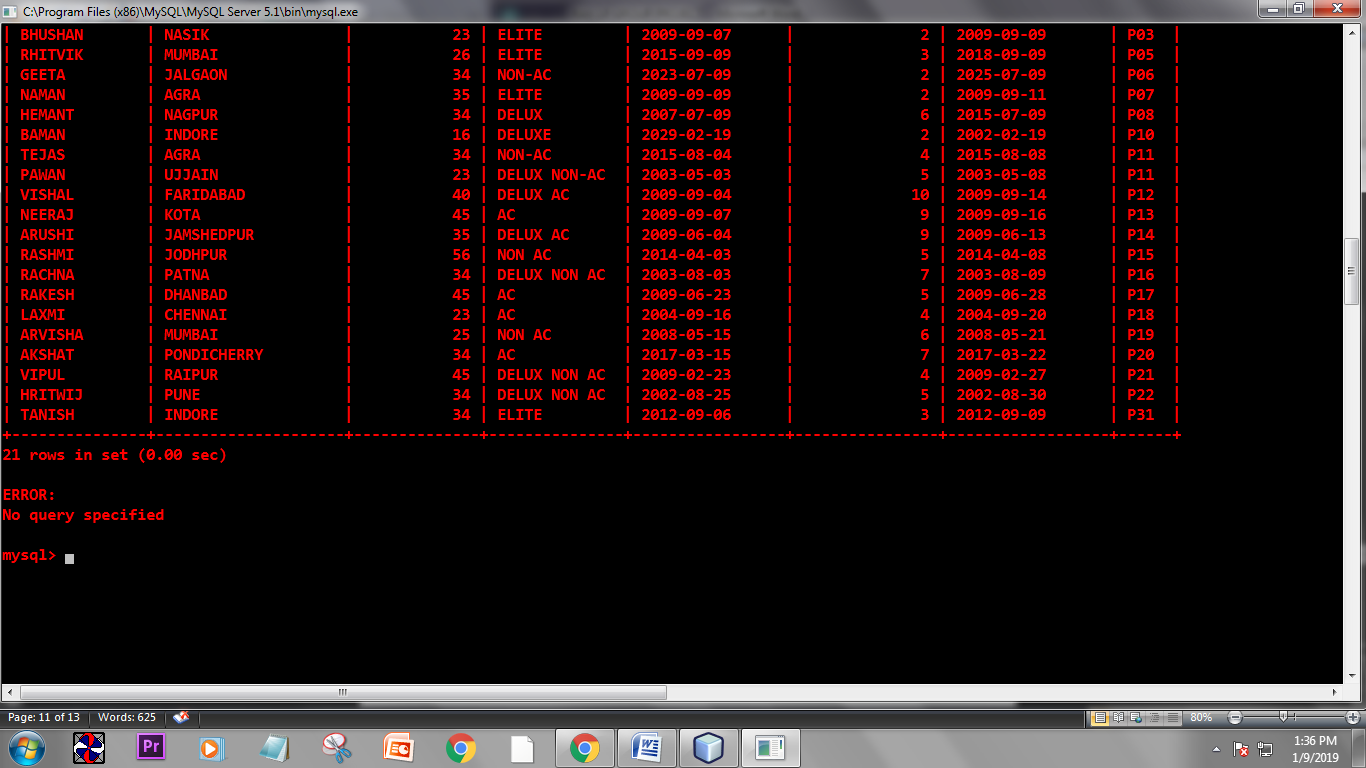
**4. update button.**

**5. show room availability button**

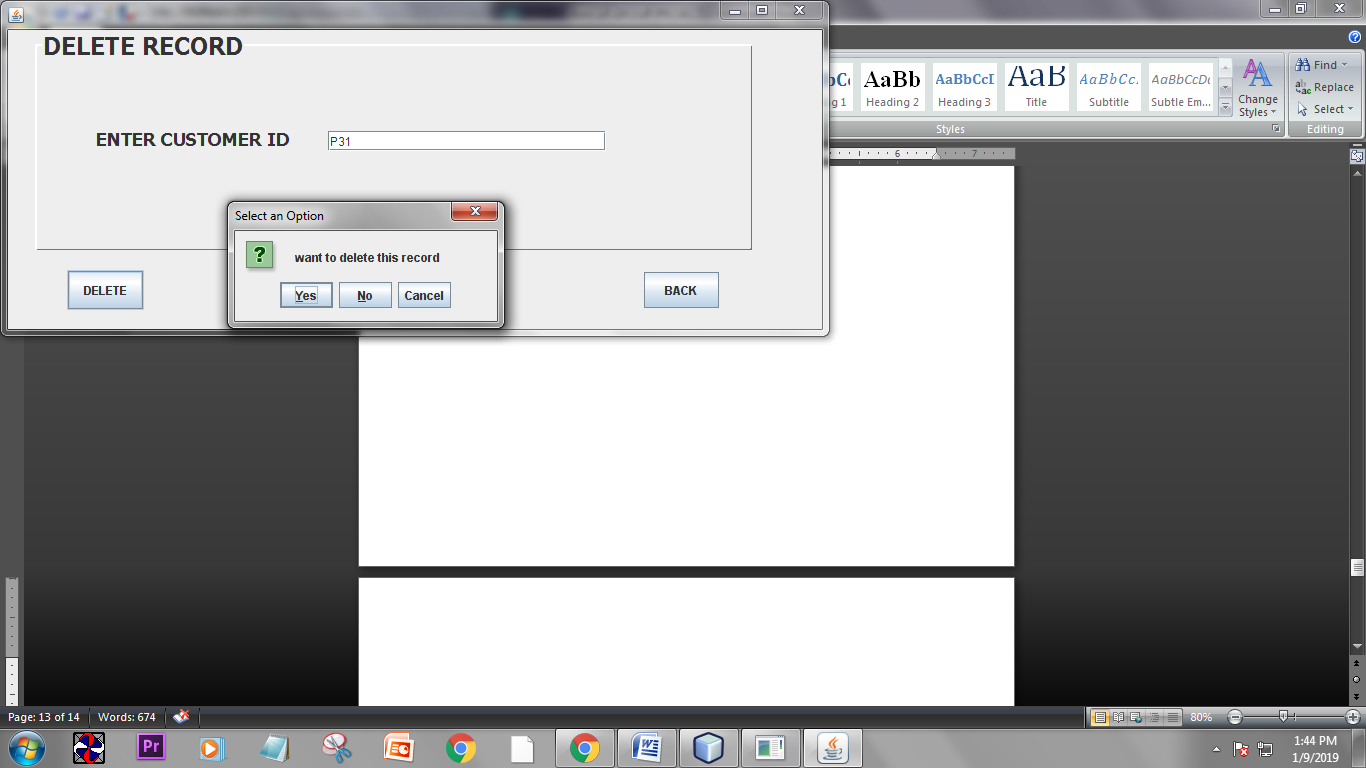
**6. generate bill button**

**7. show all records button.**

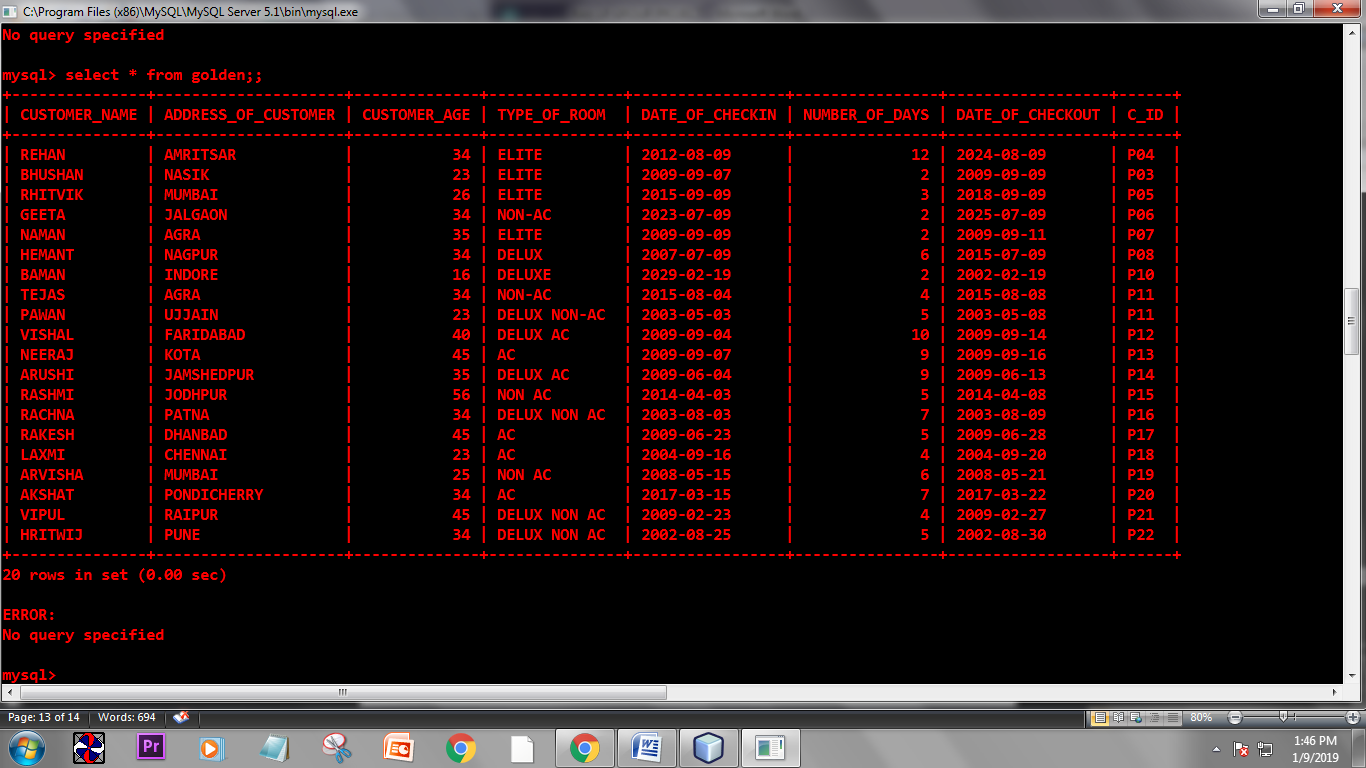
**Window for booking of room**

**Now as we click on ‘ADD RECORDS’ the following records are entered. In the Mysql database. Here is the screenshot of the table.**

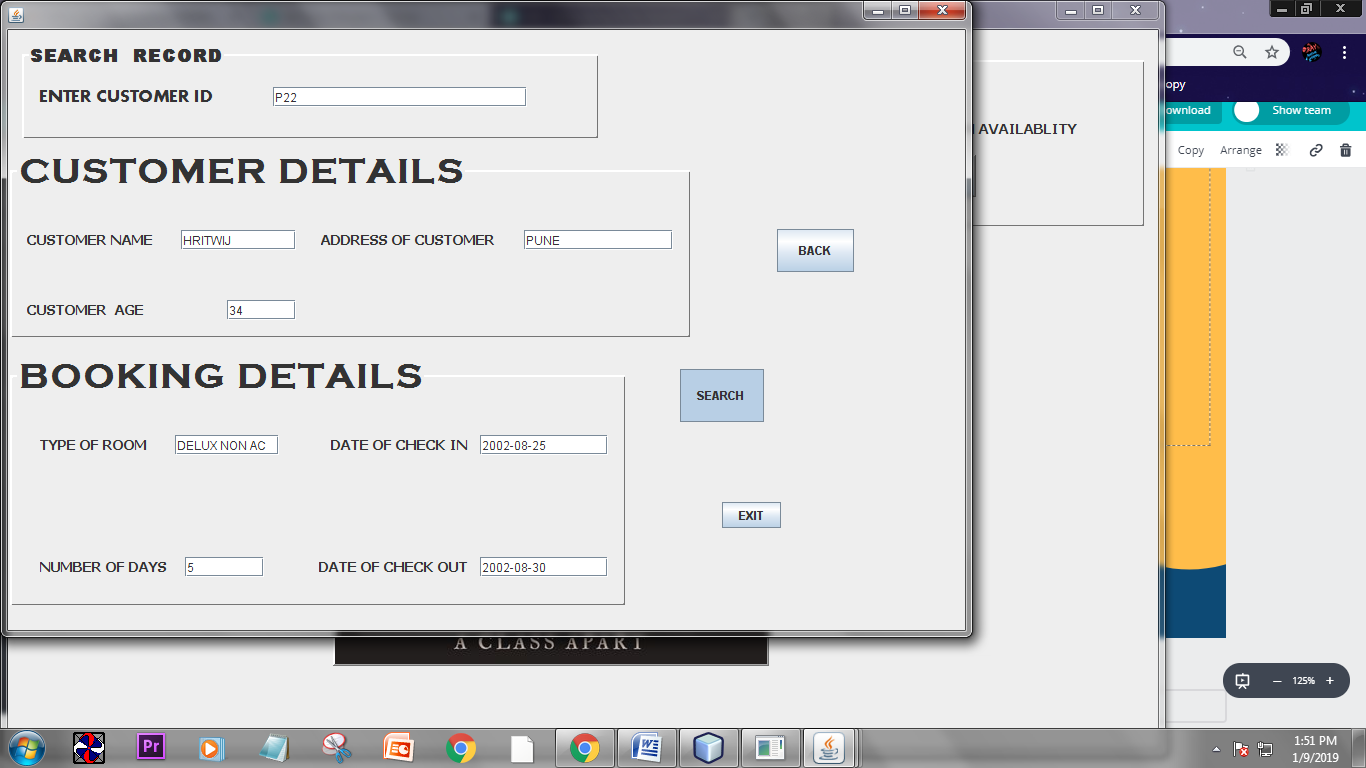
**Window for canceling the record which was entered.**

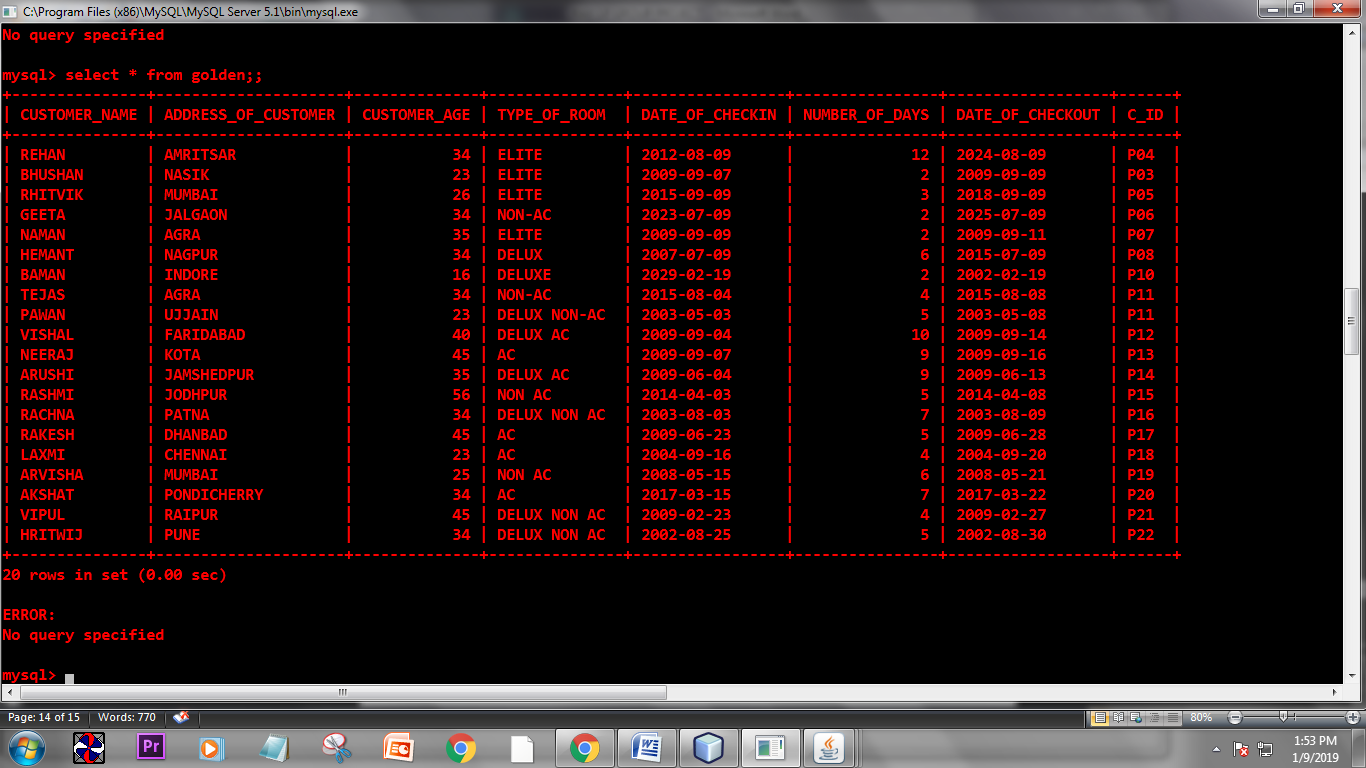
****

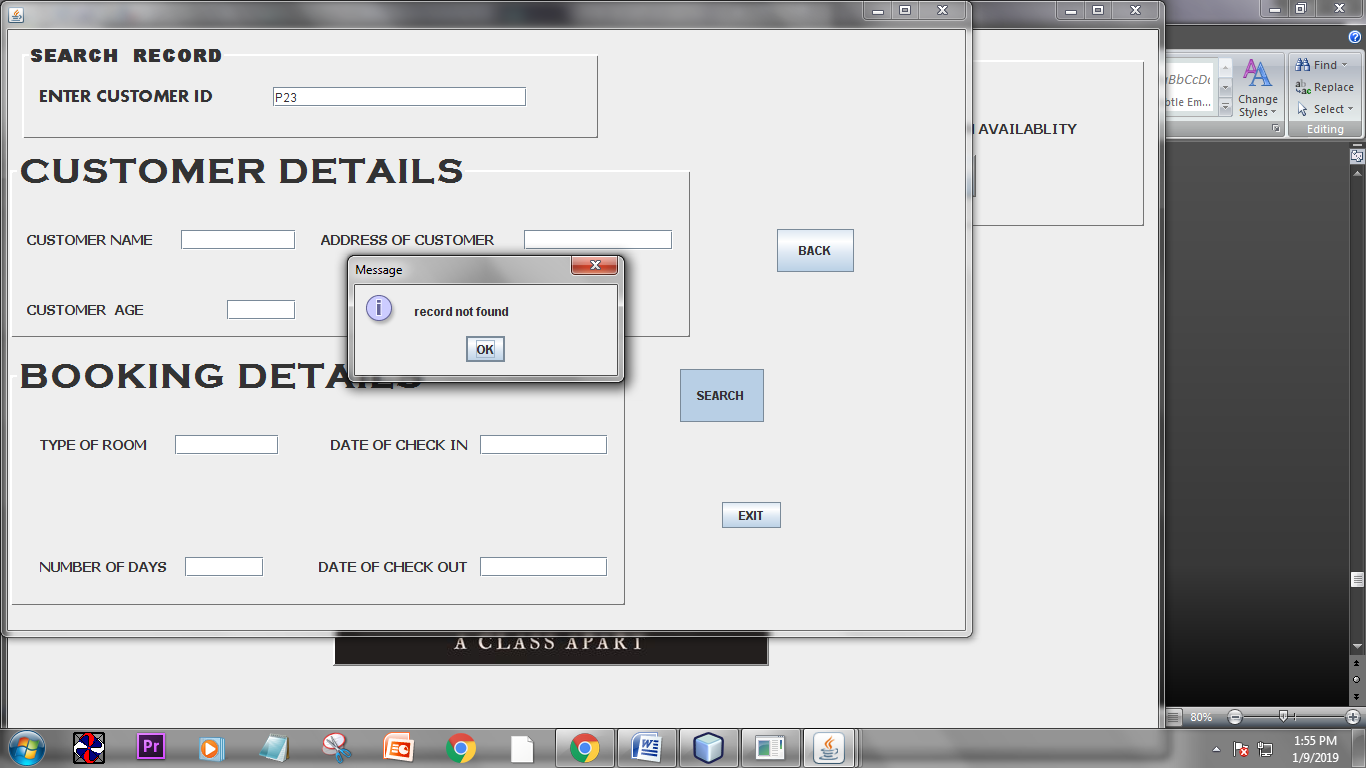
**As we click on delete button then there will a confirmation box asking whether we have to delete the record or not. As we click on yes the record which we want to delete is deleted from database. Here in this software we will use ‘C\_ID’ as a basis for finding out the desired record. As “C\_ID” is the primary key.**

**As we can see that the last record we entered was “ tanish” and below is the screen shot of the table. we can see that “ tanish” record is deleted.**

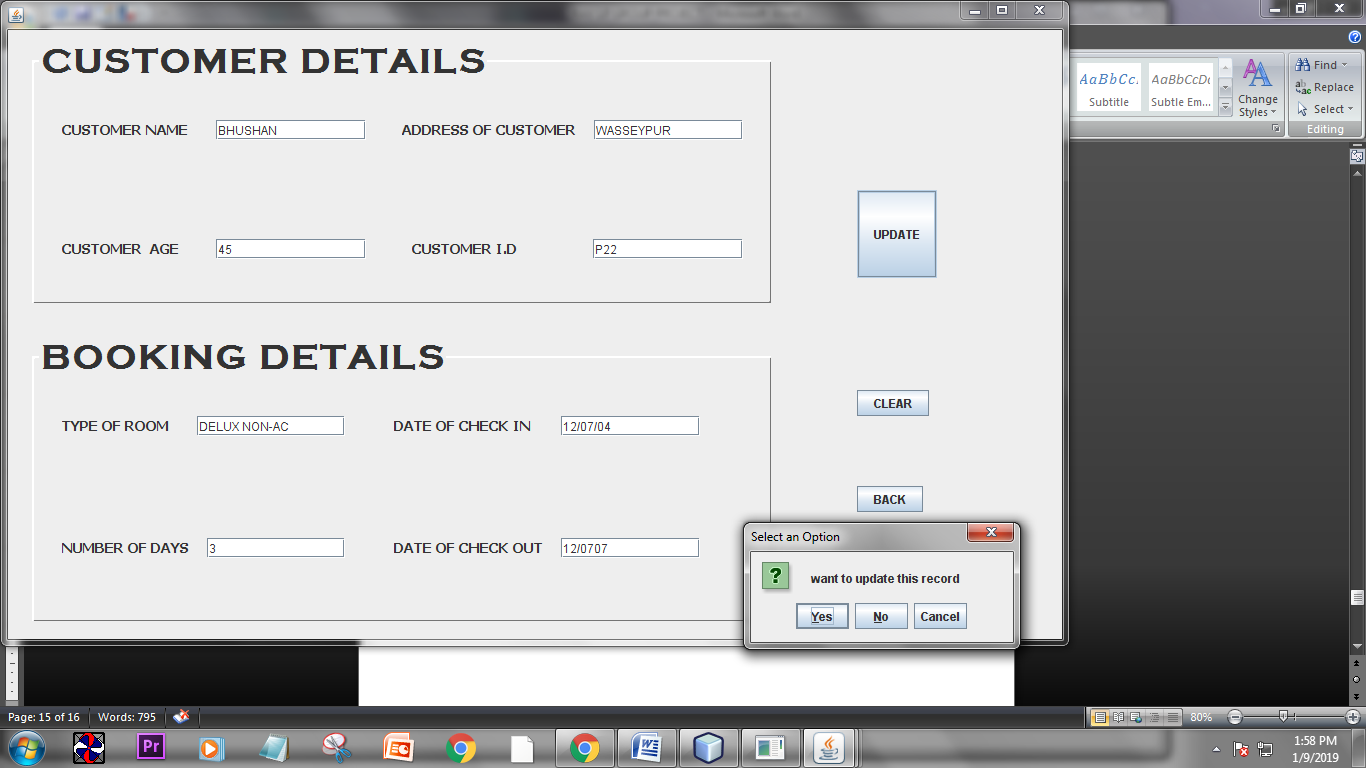
**Window for searching a record.**

**Here also we will be searching records on the basis of our primary key “ C\_ID”.**

**Below is the screen shot of the table which shows that the records are brought in the textfields of the jform of net beans.**

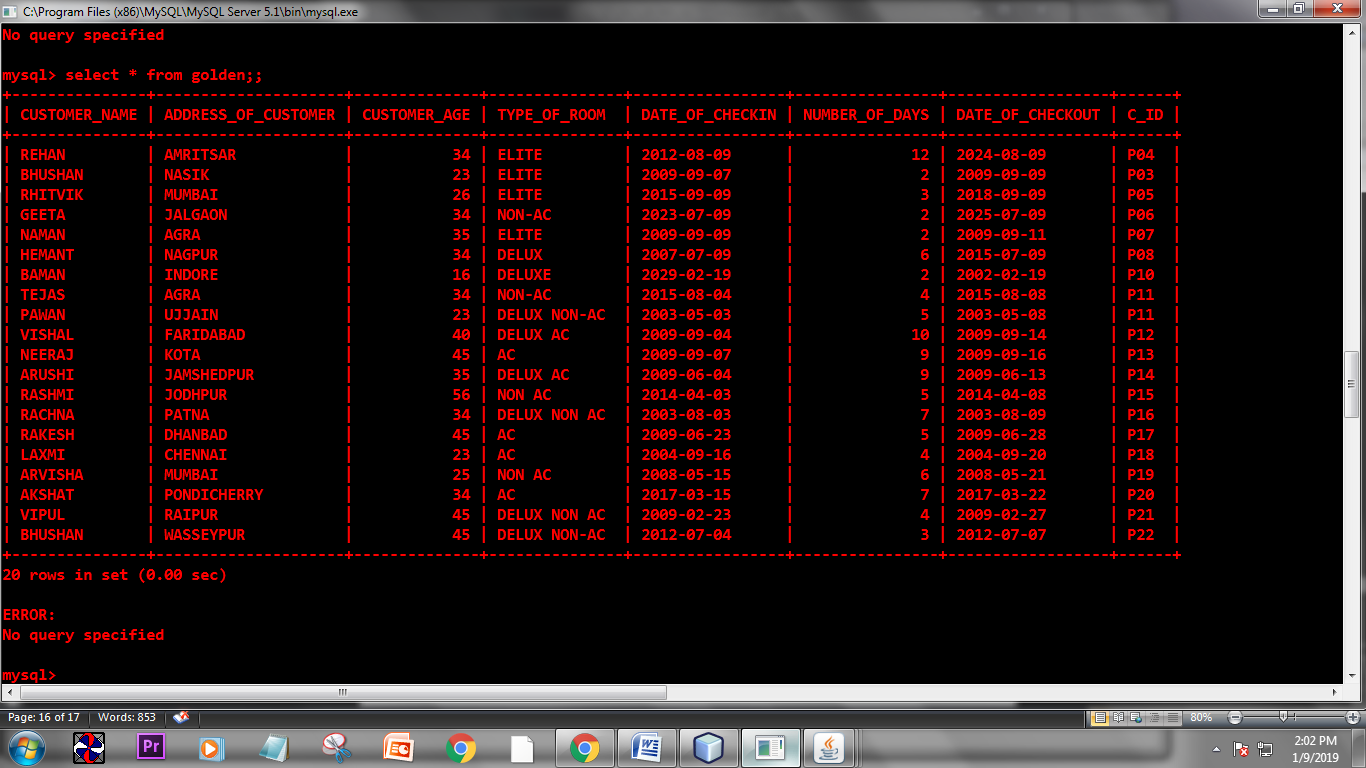
**If the user enters wrong “C\_ID” then.**

**There will be a pop-up screen which will show “ record not found”.**

**Window for updating customer details.**

**We can see in the above screen shot that we have entered the desired record (on the basis of primary key)**

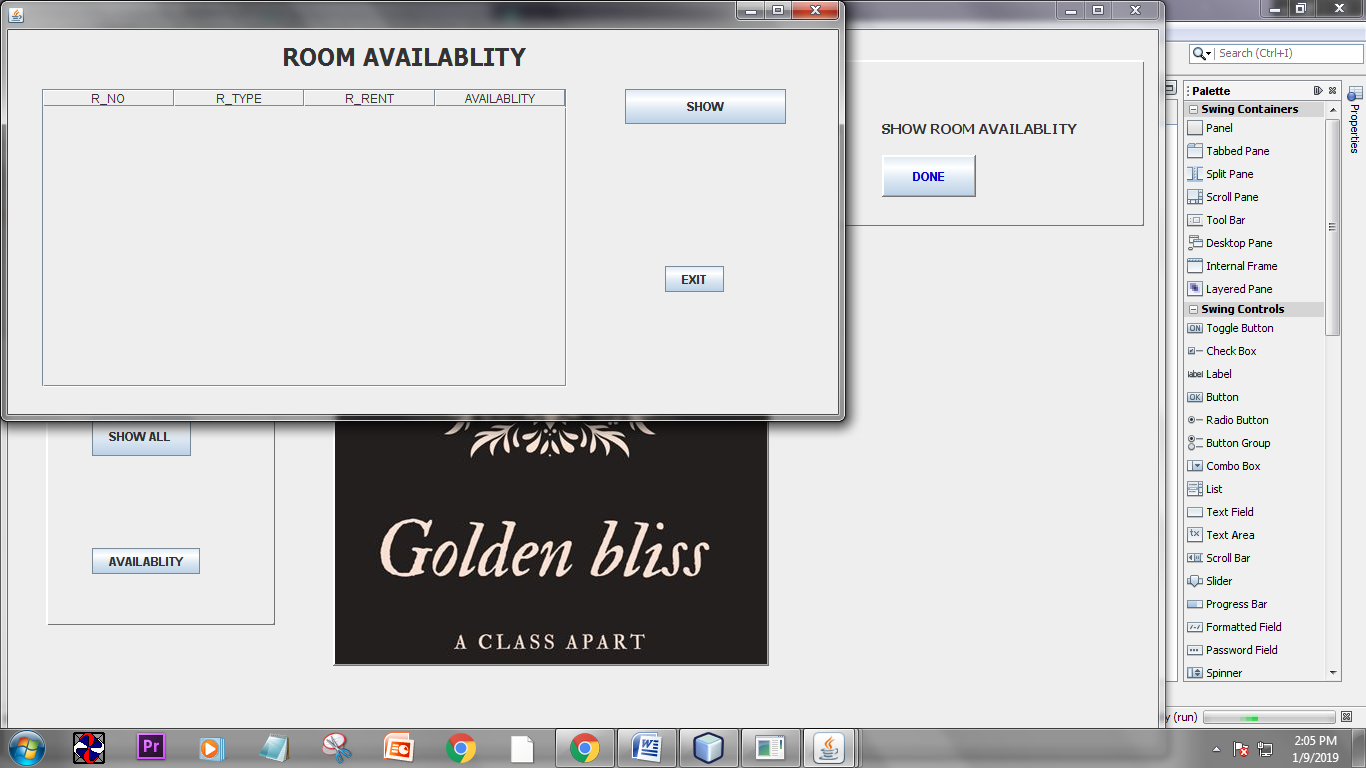
**To be updated and we have made all necessary changes in the record. Now a pop-up screen will come asking**

****

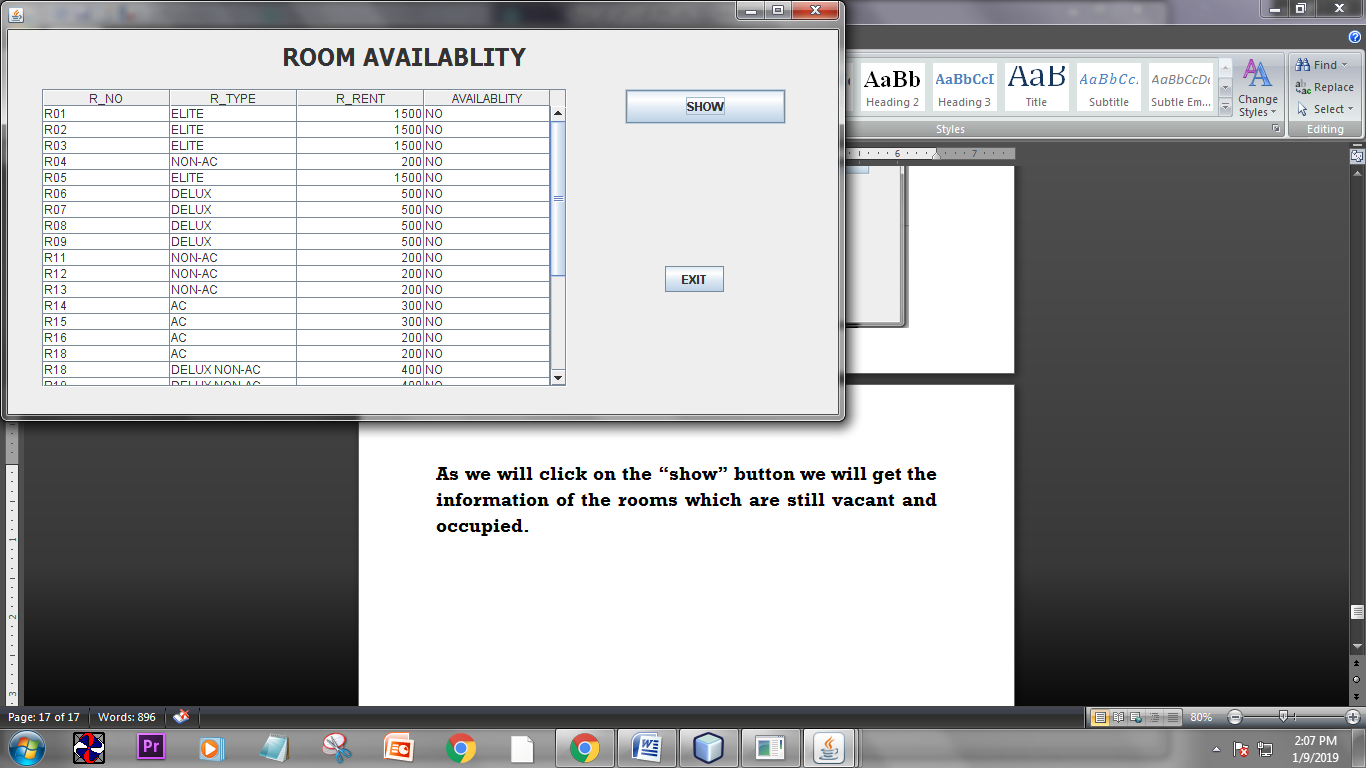
**Whether we need to update this record. As we click on ok the record will be updated.**

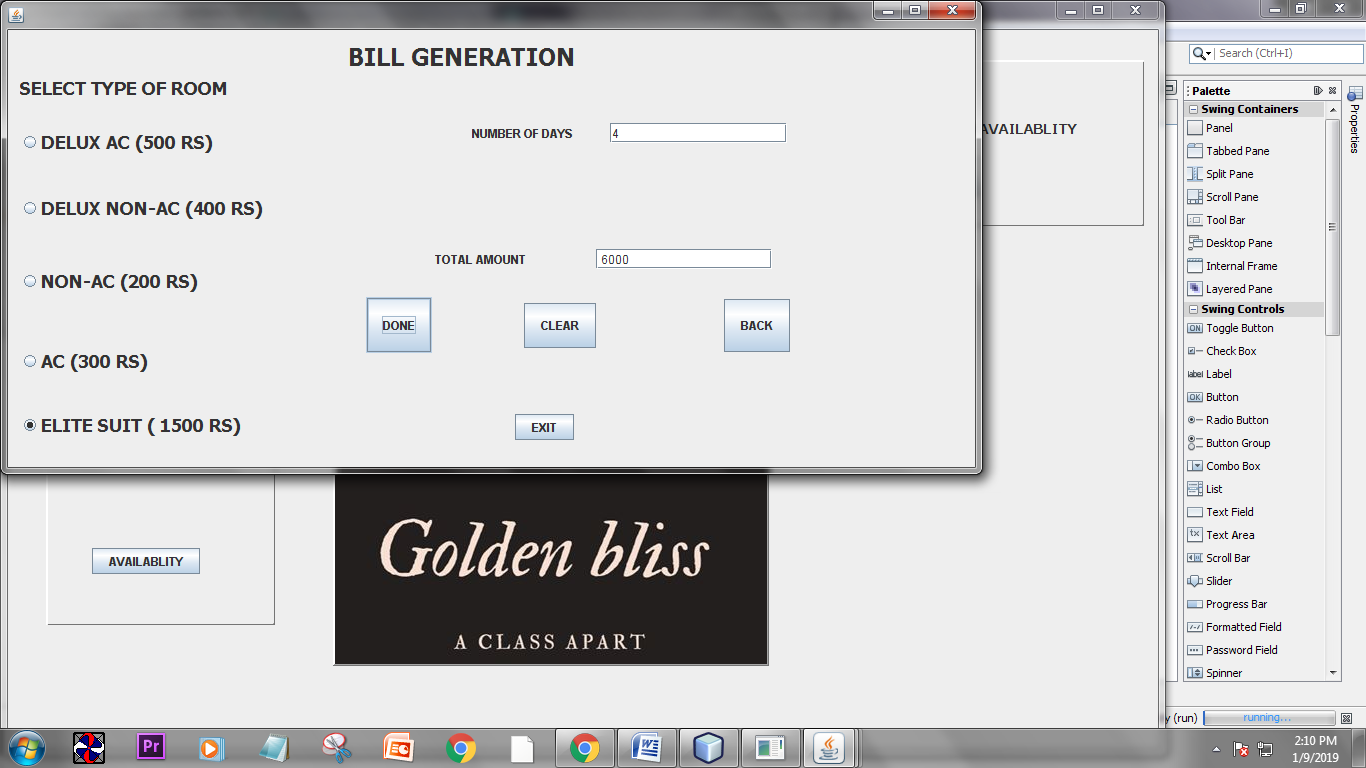
**We can see that the record is update with “ bhushan” which was earlier “ hritwij”.**

**Window for showing room availability**

****

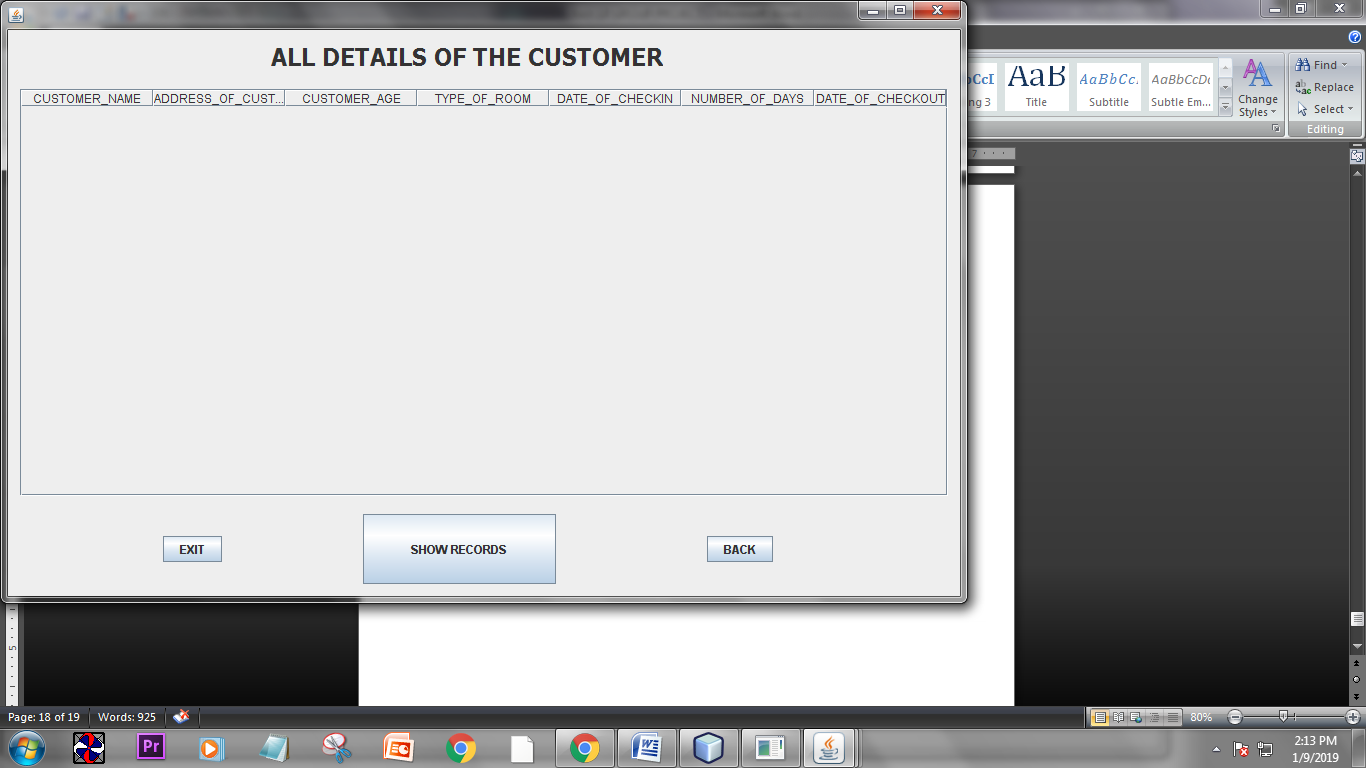
**As we will click on the “show” button we will get the information of the rooms which are still vacant and occupied.**

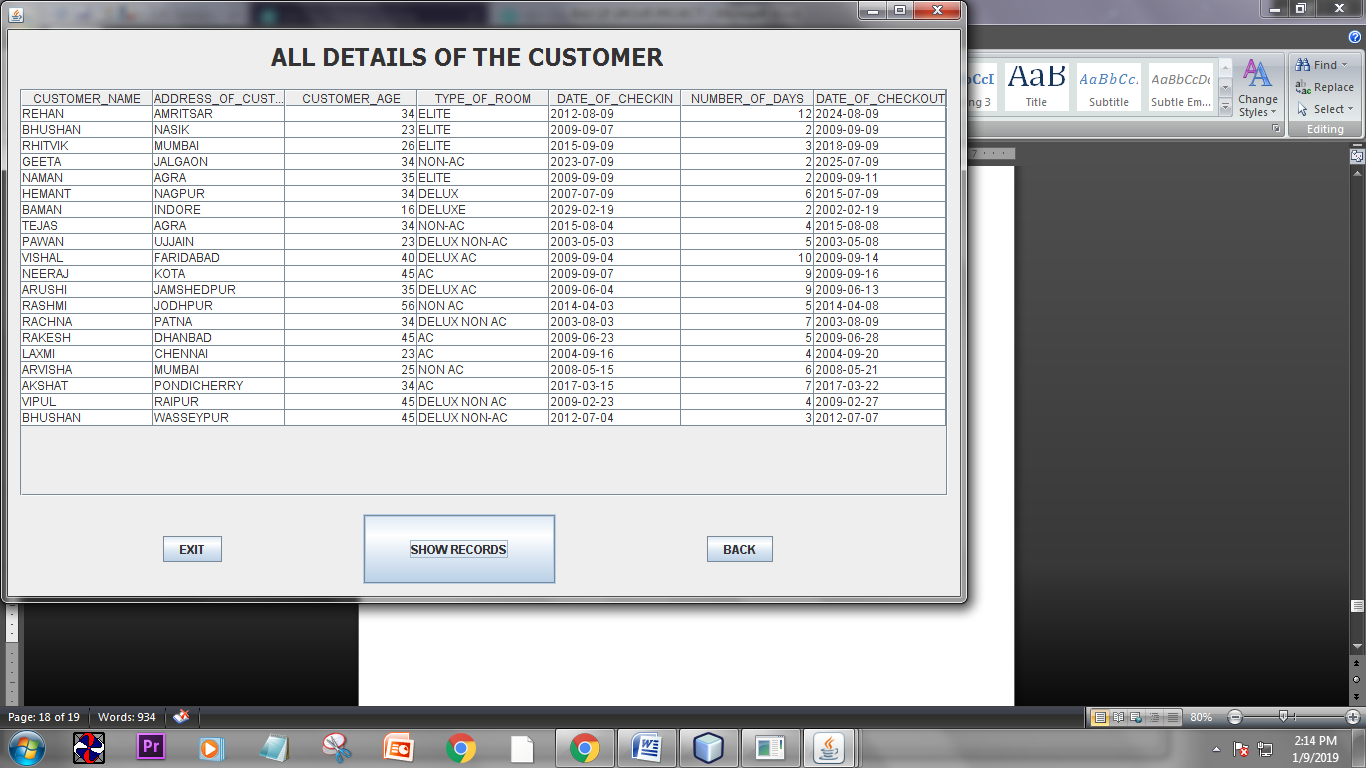
****

**Window for generating bill**

**It is seen that the bill is generated on the basis of days the customer has stayed in the hotel.**

**Window for showing all record**

****

**Now we will click on the button show records.**

**5. SOURCE CODE**

**INSERT BUTTON:**

String CUSTOMER\_NAME= tf3.getText();

Connection con = null;

int CUSTOMER\_AGE = Integer.parseInt(tf4.getText());

String TYPE\_OF\_ROOM = tf6.getText();

String DATE\_OF\_CHECKOUT = tf10.getText();

int NUMBER\_OF\_DAYS =Integer.parseInt (tf9.getText());

String DATE\_OF\_CHECKIN= tf8.getText();

String ADDRESS\_OF\_CUSTOMER = tf5.getText();

String C\_ID = tf11.getText();

try {

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

con = (Connection)

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

Statement stmt = (Statement)

con.createStatement();

String query = " insert into golden values('"+CUSTOMER\_NAME+"','"+ADDRESS\_OF\_CUSTOMER+"',"+CUSTOMER\_AGE+",'"+TYPE\_OF\_ROOM+"','"+DATE\_OF\_CHECKIN+"',"+NUMBER\_OF\_DAYS+",'"+DATE\_OF\_CHECKOUT+"','"+C\_ID+"');";

stmt.executeUpdate(query);

}

catch(Exception e )

{

JOptionPane.showMessageDialog(this, e.getMessage());

}

**DELETE BUTTON:**

Connection con = null;

String a = tf1.getText();

if (a.isEmpty())

{

JOptionPane.showMessageDialog(this, " enter customer id");

}

else

{

int ans = JOptionPane.showConfirmDialog(this, " want to delete this record");

if (ans==JOptionPane.YES\_OPTION)

{

try {

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

con = (Connection)

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

Statement stmt = (Statement)

con.createStatement();

String query ="delete from golden " + "where C\_ID = '"+tf1.getText()+"' " ;

stmt.executeUpdate(query);

}

catch(Exception e )

{

JOptionPane.showMessageDialog(this, e.getMessage());

}

}

**}**

**SEARCH BUTTON:**

Connection con = null;

String C\_ID = tf1.getText();

if ( C\_ID.isEmpty())

{

JOptionPane.showMessageDialog(this, "enter C\_ID");

}

else

{

try{

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

con = (Connection)

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

Statement stmt = (Statement)

con.createStatement();

String query =" select \* from golden " + " where C\_ID = '"+ tf1.getText()+"' ; " ;

ResultSet rs = stmt.executeQuery(query);

if (rs.next())

{

String CUSTOMER\_NAME = rs.getString(1);

String ADDRESS\_OF\_CUSTOMER = rs.getString(2);

int CUSTOMER\_AGE = rs.getInt(3);

String TYPE\_OF\_ROOM = rs.getString(4);

String DATE\_OF\_CHECKIN = rs.getString(5);

int NUMBER\_OF\_DAYS = rs.getInt(6);

String DATE\_OF\_CHECKOUT = rs.getString(7);

tf2.setText(CUSTOMER\_NAME);

tf3.setText(ADDRESS\_OF\_CUSTOMER);

// tf4.setText(CUSTOMER\_AGE);

tf6.setText(TYPE\_OF\_ROOM);

tf7.setText(DATE\_OF\_CHECKIN);

// tf8.setText(NUMBER\_OF\_DAYS);

tf9.setText(DATE\_OF\_CHECKOUT);

}

else

{

JOptionPane.showMessageDialog(this, " record not found");

}

rs.close();

stmt.close();

con.close();

}

catch(Exception e )

{

JOptionPane.showMessageDialog(this, e.getMessage());

}

}

**UPDATE BUTTON:**

Connection con = null;

String C\_ID= tf1.getText();

if (C\_ID.isEmpty())

{

JOptionPane.showMessageDialog(this, " enter customer id");

}

else

{

int ans = JOptionPane.showConfirmDialog(this, " want to update this record");

if (ans==JOptionPane.YES\_OPTION)

{

try {

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

con = (Connection)

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

Statement stmt = (Statement)

con.createStatement();

String query ="update golden set CUSTOMER\_NAME = '"+tf11.getText()+"', ADDRESS\_OF\_CUSTOMER = '" +tf2.getText()+"', CUSTOMER\_AGE = '" +tf3.getText()+"',TYPE\_OF\_ROOM = '" +tf4.getText()+"', DATE\_OF\_CHECKIN = '" +tf8.getText()+"', NUMBER\_OF\_DAYS = '" +tf9.getText()+"', DATE\_OF\_CHECKOUT = '" +tf10.getText()+"' " + "where C\_ID = '"+tf1.getText()+"' " ;

stmt.executeUpdate(query);

JOptionPane.showMessageDialog(this, " RECORD UPDATED");

}

catch(Exception e )

{

JOptionPane.showMessageDialog(this, e.getMessage());

}

}

}

**ROOM AVAILABLITY BUTTON:**

Connection con = null;

DefaultTableModel model = (DefaultTableModel)tb1.getModel();

try{

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

con = (Connection)

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

Statement stmt = (Statement)

con.createStatement();

String query =" select \* from room";

ResultSet rs = stmt.executeQuery(query);

while (rs.next())

{

String R\_NO = rs.getString("R\_NO");

String R\_TYPE = rs.getString("R\_TYPE");

int R\_RENT = rs.getInt("R\_RENT");

String AVAILABLITY = rs.getString("AVAILABLITY");

model.addRow(new Object[] {R\_NO, R\_TYPE, R\_RENT, AVAILABLITY});

}

rs.close();

stmt.close();

con.close();

}

catch (Exception e )

{

JOptionPane.showMessageDialog(this, e.getMessage());

}

}

**GENERATE BILL BUTTON:**

int a = Integer.parseInt(tf1.getText());

int b = 0;

if(rb1.isSelected())

{

b = a \* 500;

}

else if(rb2.isSelected())

{

b = a \* 400;

}

else if(rb3.isSelected())

{

b = a \* 200;

}

else if(rb4.isSelected())

{

b = a \* 300;

}

else if(rb5.isSelected())

{

b = a \* 1500;

}

tf2.setText(" " + b);

}

**SHOW ALL RECORD BUTTON:**

Connection con = null;

DefaultTableModel model = (DefaultTableModel)tb1.getModel();

try{

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

con = (Connection)

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

Statement stmt = (Statement)

con.createStatement();

String query =" select \* from golden";

ResultSet rs = stmt.executeQuery(query);

while (rs.next())

{

String CUSTOMER\_NAME = rs.getString("CUSTOMER\_NAME");

String ADDRESS\_OF\_CUSTOMER = rs.getString("ADDRESS\_OF\_CUSTOMER");

int CUSTOMER\_AGE = rs.getInt("CUSTOMER\_AGE");

String TYPE\_OF\_ROOM = rs.getString("TYPE\_OF\_ROOM");

String DATE\_OF\_CHECKIN = rs.getString("DATE\_OF\_CHECKIN");

int NUMBER\_OF\_DAYS = rs.getInt("NUMBER\_OF\_DAYS");

String DATE\_OF\_CHECKOUT = rs.getString("DATE\_OF\_CHECKOUT");

model.addRow(new Object[] {CUSTOMER\_NAME, ADDRESS\_OF\_CUSTOMER, CUSTOMER\_AGE, TYPE\_OF\_ROOM, DATE\_OF\_CHECKIN, NUMBER\_OF\_DAYS, DATE\_OF\_CHECKOUT});

}

rs.close();

stmt.close();

con.close();

}

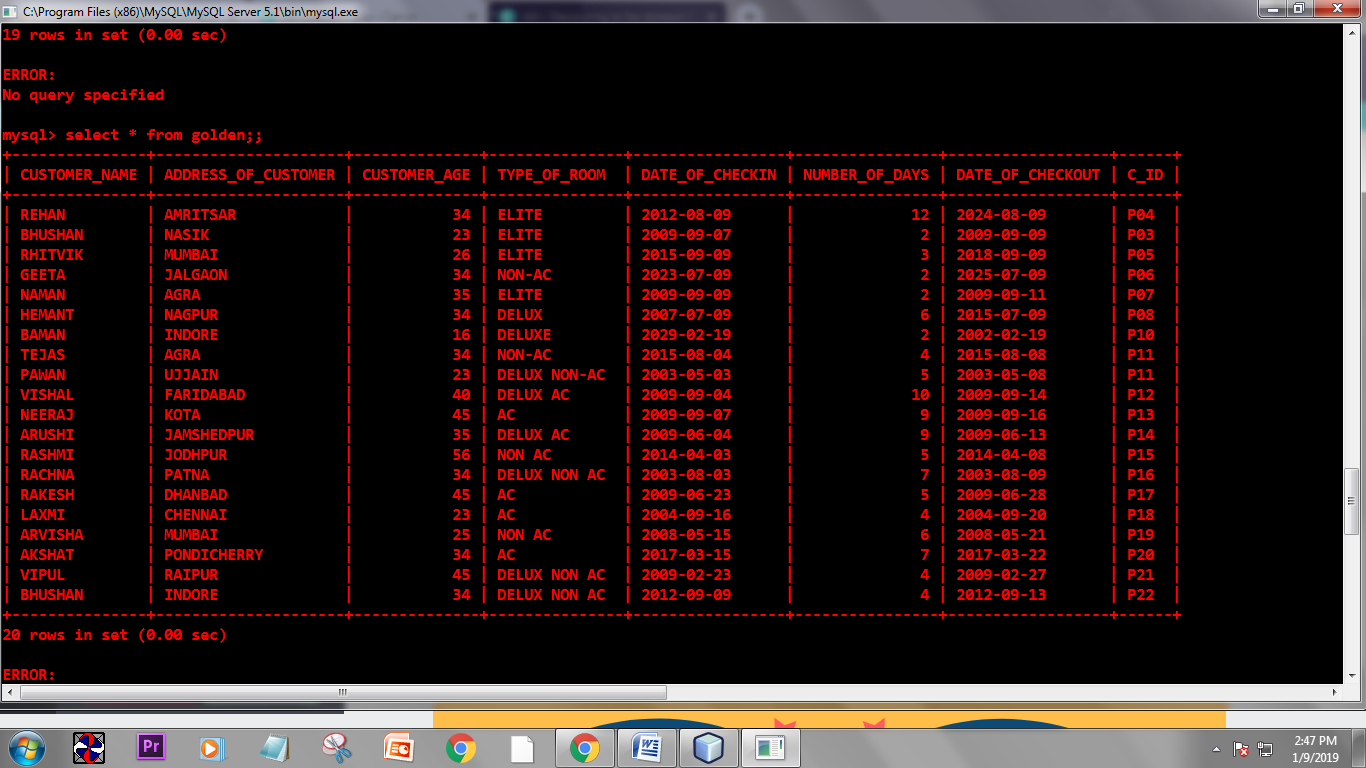
catch (Exception e )

{

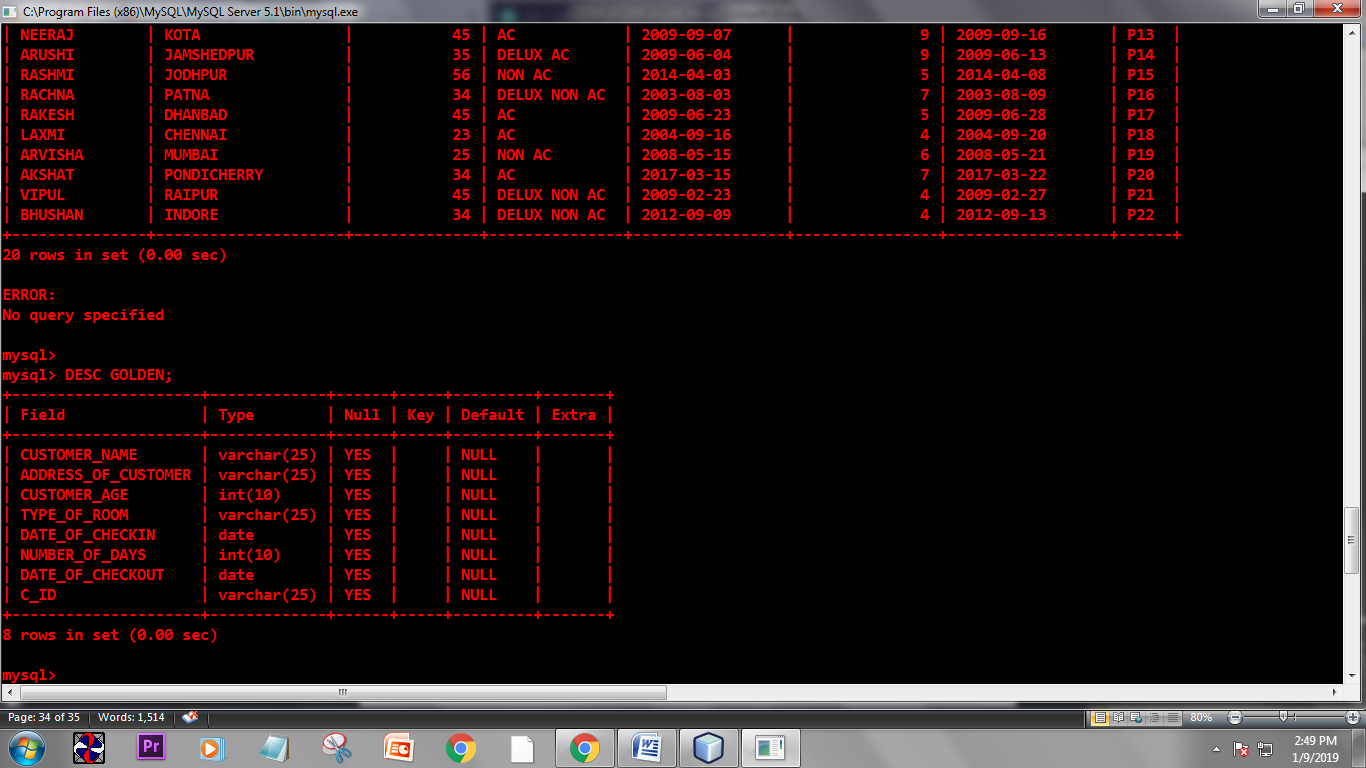
JOptionPane.showMessageDialog(this, e.getMessage());

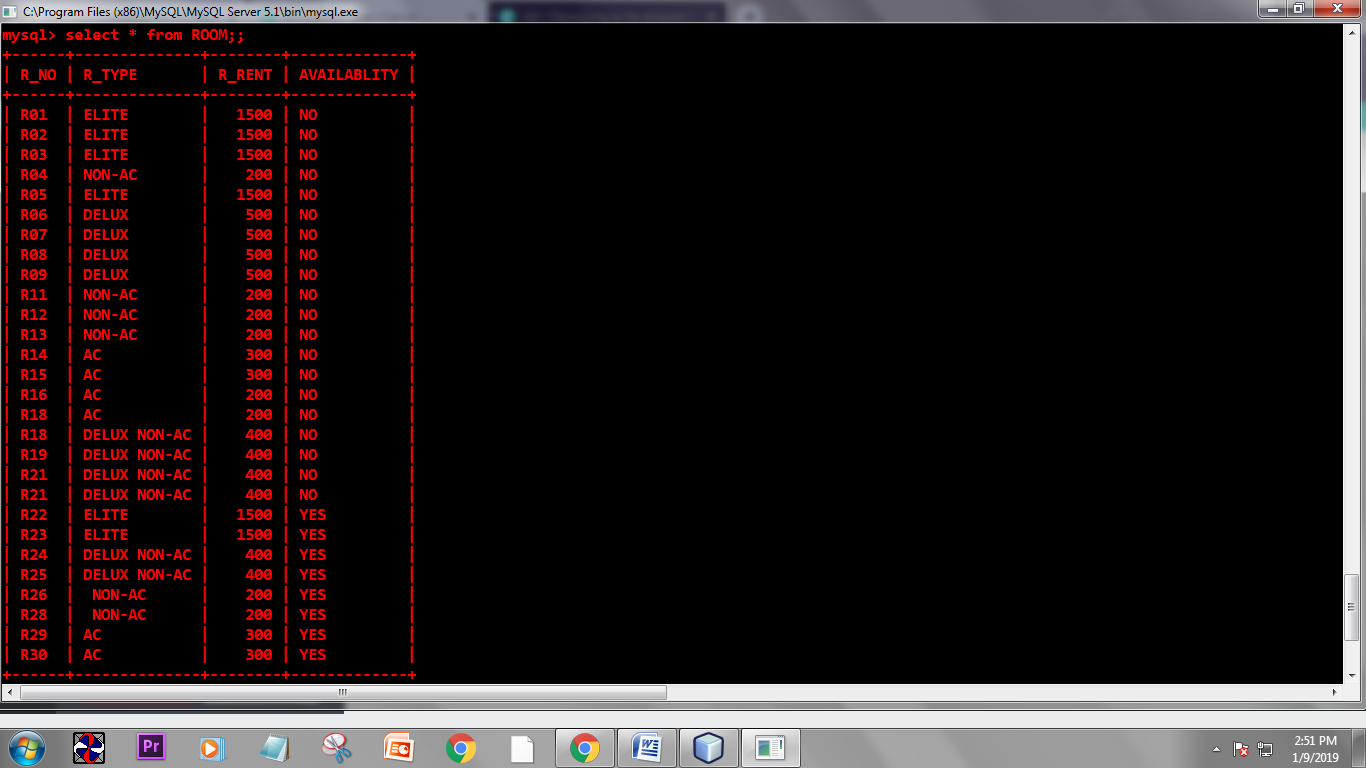
}

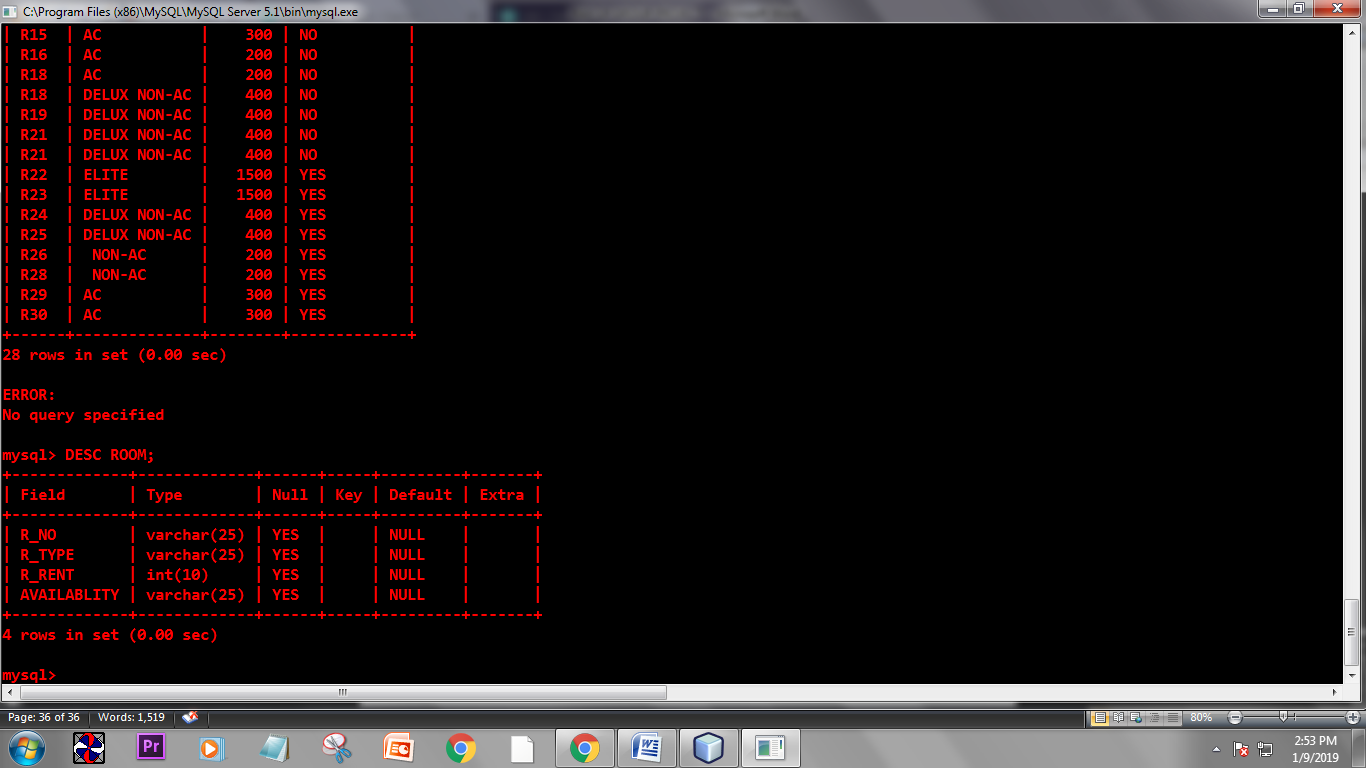
**6. Tables and its details.**

**TABLE 1:GOLDEN**

**TABLE STRUCTURE :**

****

**TABLE 2: ROOM**

**TABLE STRUCTURE:**