**K.L.E. SOCIETY’S**

P.C JABIN SCIENCE COLLEGE,

AUTONOMUS,

(Affiliated to KARNATAK UNIVERSITY, DHARWAD)

**HUBBALLI -580031**

**Bachelor of Computer Application**

**2021-22**

PROJECT REPORT

On

**HOTEL RESERVATION SYSTEM**

Submitted in partial fulfillment of the requirement for the award of the degree

**BACHELOR OF COMPUTER APPLICATION**

Submitted By

**Anirudh Parvatikar Vineeth Kemtur**

(219127) (219172)

Under The Guidance Of**Prof Tejaswini Apte**

Affiliated to

**Karnatak University, Dharwad.**

**K.L.E. SOCIETY’S**

P.C.JABIN SCIENCE COLLEGE CAMPUS,

AUTONOMUS,

(Affiliated to KARNATAK UNIVERSITY, DHARWAD)

**HUBBALLI -580031**

**BCA DEPARTMENT**

**2021-22**

**Certificate**

This is to certify that the project entitled **Hotel Reservation System** is a bonafied work carried out by the student team Mr. Anirudh U Parvatikar, Reg No 219127 and

Mr. Vineeth S Kemtur, Reg No 219172, in partial fulfillment of the award of degree of Bachelor of Computer Application during the year 2021 – 2022. The project report has been approved as it satisfies the academic requirement with respect to the project work prescribed for the award of BCA Degree.

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

**Guide Principal**

**External Examination:**

**Name of the Examiners Signature with date**

**1.**

**2.**

**DECLARATION**

We here by declared that the project report entitled **Hotel Reservation System,** submitted in fulfillment of requirement of BCA VI Sem Project work for the award of Degree in Bachelor of Computer Application of KARNATAK UNIVERSITY, Dharwad during the academic year 2021-22.

We further declare that this project report is the result of our original work and has not been submitted to any other organization or institute for the award of any degree or diploma.

Date:

Place: Hubballi

**Sign Sign**

**Vineeth Kemtur Anirudh Parvatikar**

**ACKNOWLEDGEMENT**

It’s our pleasure to thank all the individuals who have directly or indirectly helped and motivated us in the fulfilment of completion of the project work.

We thank **Prof** **Sunil Vernekar (Principal), KLE Society’s BCA, P C Jabin Science College, HUBBALLI** for having given us all encouragement and motivation for making this project work successful.

We thank our guide **Prof** **Your Guide name, KLE Society’s BCA, P C Jabin Science College, HUBBALLI** for giving us valuable suggestions and guidance for our project work, which are the background of the project.

Our gratitude also goes to all **Teaching and Non-Teaching staff** of **KLE Society’s BCA, P C Jabin Science College, HUBBALLI** who have helped us in completing this project work.

Finally, we would like to thank our family and friends for their constant motivation and inspiration that kept us going.

**Sign Sign**

**Vineeth Kemtur Anirudh Parvatikar**

**ABSTRACT**

Hotel Management System is developed to be used by hotel staff and their customers for better management of

the hotel table booking process.

It is used to- give facility to the user. It is also developed to replace the manual system that is used before.

There have been many problems during use manual system, like data redundancy, lost or damage.

The scopes that exist in this system are booking online table for breakfast, lunch, dinner & make payment using system.

Data used in this study was collected using observation, reviewing of existing documents and interview.

Interviews were conducted using interview guides. The tool was implemented using Oracle10g database software and Java Programming Language. The tool was tested and validated using sample data we created.

This tool can be useful at the Hotel in speeding up the process of

determining the status of client’s requests submitted.

**Dedicated**

**To**

**OUR PARENTS**

**\*\*\*\*\*\*\*\*\***

**CONTENTS**

|  |  |  |
| --- | --- | --- |
| Sl.no | Topic | Page No |
| 1 | Introduction | 1 |
| 2 | Literature Survey (Objective and Feasibility study) | 2 |
| 3 | Technical Requirements (Hardware and Software) | 3 |
| 4 | Project Description | 5 |
| 5 | System Design (Flow Charts / DFD / ER Diagrams …) | 6 |
| 6 | Source Code | 10 |
| 7 | UI Design and Outputs | 21 |
| 8 | Implementation (Deployment) | 22 |
| 9 | Testing Methods | 23 |
| 10 | Advantages of Project | 25 |
|  | Conclusion | 25 |
|  | Future Enhancement | 25 |
|  |  |  |

1. **Introduction**

Hotel Management system is a program which uses internet to connect potential customers,

to their favourite hotel and book table remotely so they can arrive at hotel hassle free and dine-in system.

This project aims to help user to book dining table for them and their family in advance to

avoid the issues of hotel being full when arrived to. This program allows users to book hotel table

at their suitable date and time and specify how many are going to accompany them. This also

enables user to also order quick snack as they arrive to the hotel…

This program is also helpful for the managers of hotel too, as this enables them to have

analytics about number of people using this service and estimate their business… It is also

equipped with billing system to make it easy to hotel management to track their earnings for a day.

User have to pay booking fees online + amount incurred for any meals they

ordered(optional), after the payment is done their table will be booked and they would get a token

number.

**Limitations of existing system**

Existing system allows you to only book seats in hotel either by call or in-person, if you book seats in-person that you would have to travel basically 2 or times to same place, Once to book, again to have dinner… which is not good use of our time. To overcome these things we have come up with a software through which you can book tables by sitting at home and just resume with any other work you have. No need to travel and no other things.

**Proposed system**

Based on given technological development we aim the users to have system with which is latest or maximum of 5 years old not more than that, as modules used in this software are compatible only in newer versions of operating systems & they might not execute properly in legacy system as the program lacks backwards compatability.

1. **Literature Survey (Objective and Feasibility study)**

Technology has made a considerable impact on the Hospitality industry in recent years and will continue to do so with the increasing use of computer, controlled equipment and the growth of information technology in general

The use of range of computer programs from everything to bookings, communications, security and payments.  If a hospitality establishment does not use some sort of advanced technological system in its operations, it is deemed to be out of date and disorganized.

In this project bookings of customers can be accessed easily by the Hotel Manager. Fast and dynamic data, latest data can be fetched (table booking) automatic sorting as per incoming bookings.

**Feasibility Study**

Feasibility study is an important phase in the software development process. It enables the developer to have an assessment of the product being developed. It refers to the feasibility study of the product in terms of outcomes of the product, operational use and technical support required for implementing it. Feasibility study should be performed on the basis of various criteria and parameters.

The various types of feasibility studies are:

1. Economic Feasibility
2. Technical Feasibility
3. Organizational Feasibility

**Economic Feasibility**

It refers to the benefits or outcomes we are deriving from the product as compared to the total cost we are spending for Developing the product. If the benefits are more or less the same as the older system, then it is not feasible to develop the product.

As of now the program isn’t intended to charge the customers for online booking, but the clients of it, i.e., Hotel management have to pay some amount to use this software & to maintain database.

**Technical Feasibility**

It refers to whether the software that is available in the market fully supports the present application. It studies the pros and cons of using particular software for the development.

This software doesn’t require much of technical requirements to run successfully, any average 64bit Processor is enough to run the program and even 4GB of RAM, Integrated Graphic are suffice to render the user interface and use it.

**Organizational Feasibility**

It refers to the feasibility of the product to be operational. Some products may work very well at design and implementation but may fail in the real environment. It includes the study of additional human resource required and their technical expertise and its feasibility. It also studies the additional training needed to be given to the people to make the application word.

As of now we are providing this charge very minimal to the organization to setup the server, software at client side & to setup some other things, over all the goal is to make the software as affordable for the organization so that their clients won’t be affected with different tariffs either online or offline booking.

1. **Technical Requirements (Hardware and Software)**

**Hardware Requirements:**

* Intel i5 and above, AMD Ryzen 3 and above
* 8GB RAM DDR4
* Windows 10 64 bit
* 50GB Hard Disk Space

**Software Requirements:**

* JAVA 15+
* Swing Package
* AWT package
* Internet connection
* Microsoft Visual Studio Code
* Oracle 10g

**Reason of Use:** We have tried our best to make the program as much compatible with almost most of the new & old/legacy devices keeping in the perspective on Indian market.

**IDE: Visual Studio Code**

**Reason of Use:** Visual Studio code provides with very useful tools to build software; it even offers many different extensions to make the code more legible for the reader & it provides feature of auto indentation which increases the overall quality of code.

**Database: Oracle 10g Express Edition**

**Reason of Use:** Very simple to use database management system, can be scaled in future according to requirements, has very good supporting API’s for JAVA language, Lightweight on Server.

1. **Project Description**

Major project intended for final year students to make a contribution to technological area by implementing new ideas they have and giving it to the community.

**Literature Survey details (Min 5 Papers or References):**

The customer has to visit hotel and has to book table in hotel. As the booking has to be done by being physically present in the location it might potentially consume lot of customers precious time as sometime it might take very long duration. Finally once table has been booked customer has to arrive again at hotel 2nd time to have dinner, totally customer has to commute 2 times at same location to have peacefully breakfast, lunch or dinner.

The idea is to automate the entire booking process from physical medium to electronic medium in order to increase efficiency and reduce the response time.

To provide simple speedy and inexpensive of customer disputes. Customer can order different types of staters on their specific arrival time, so that they can be served easily.

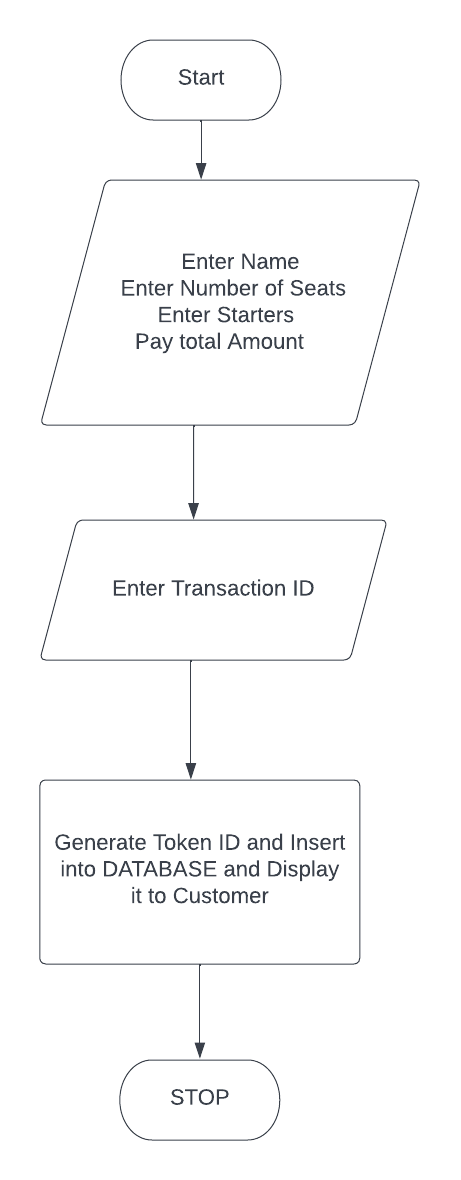
Bookings will be processed by the admin that can be viewed by the customer. Customer can know the currently processing status of their table booking.

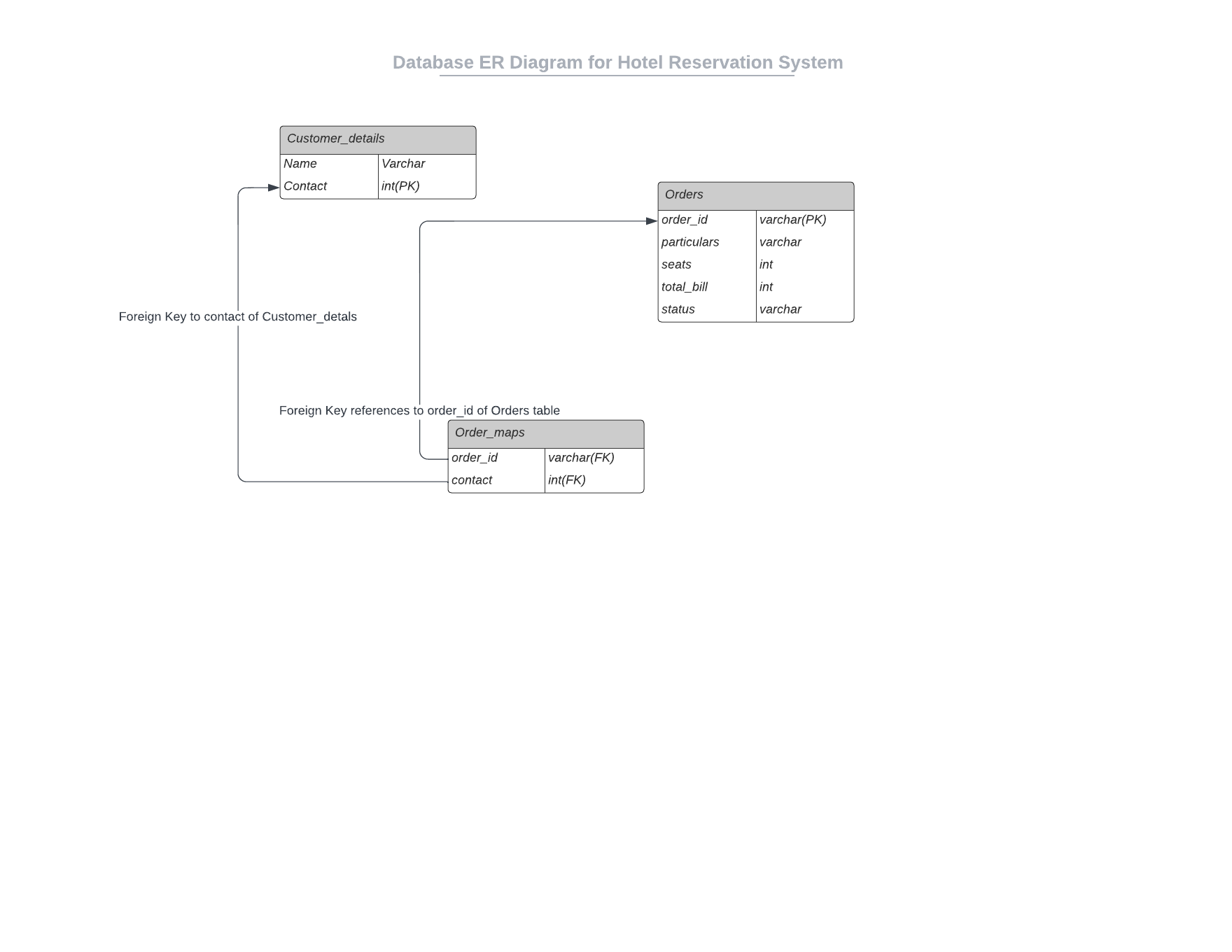
A Hotel Management system is tool which is used to increase performance of administrator, this software helps to handle the table bookings of customer, it provides fewer efforts in manual work. This is effective tool to resolve minimize the number of customers arriving at hotel just to book table.

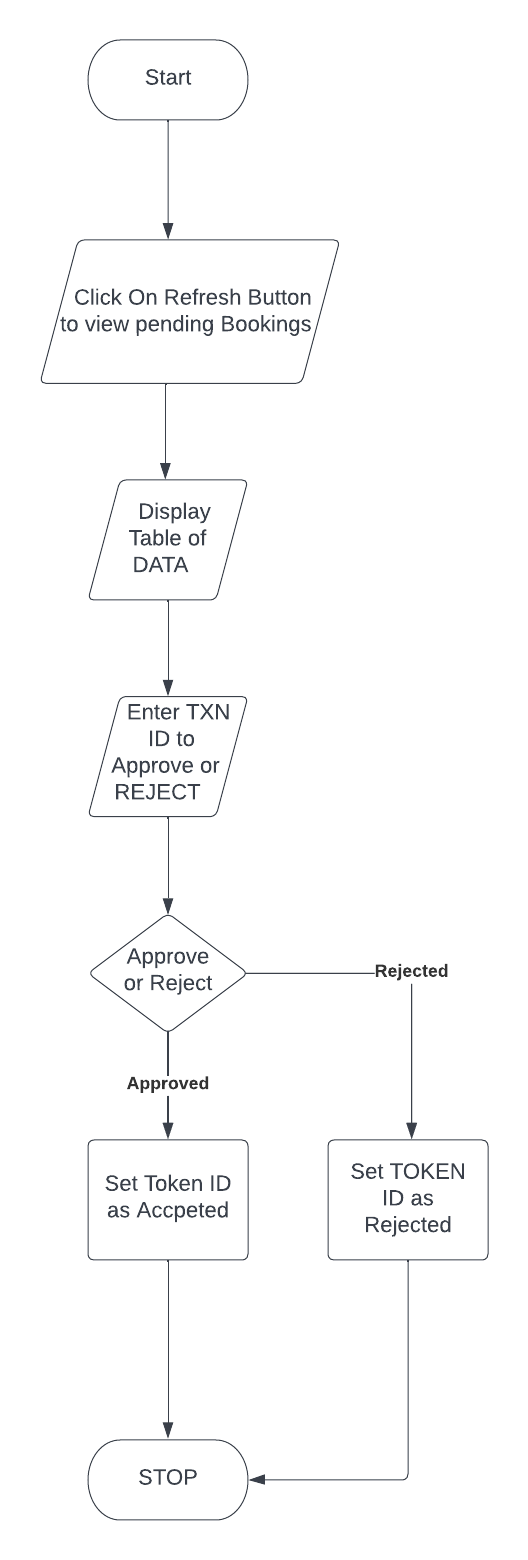
1. Java Swings: <https://docs.oracle.com/javase/tutorial/uiswing/>
2. Java Swings (practical usage): <https://www.javatpoint.com/java-swing>
3. JAVA ODBC-JDBC Bridge: <https://docs.oracle.com/javase/tutorial/jdbc/basics/index.html>
4. Java ODBC-JDBC(Practical Usage): <https://www.geeksforgeeks.org/introduction-to-jdbc/>

**Following are the notations used in Data Flow Diagram**

|  |  |
| --- | --- |
| **Notation** | **Meaning** |
|  | Process |
|  | Data Store |
|  | External Entity |
|  | Data Flow |

1. **System Design (Flow Charts/DFD/ ER Diagrams …)**

****

****

1. **Source Code**

**Client-Side**

import java.sql.\*;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.border.\*;

public class Client {

public static String gen(JTextField tkn){

String tk=tkn.getText().substring(tkn.getText().length()-4, tkn.getText().length());

return tk;

}

public static String gen\_again(JTextField tkn){

String tk=tkn.getText().substring(0,4);;

return tk;

}

public static void insertRecord(Connection con,JLabel l1,JCheckBox ch1,JCheckBox ch2,JCheckBox ch3,JCheckBox ch4,String token,JTextField nameInput,JTextField contactInput,JTextField seatsInput){

try {

int bill=150;

if(!(ch1.isSelected()||ch2.isSelected()||ch3.isSelected()||ch4.isSelected())){bill=150;}

if(ch1.isSelected()){bill=150;}

if(ch2.isSelected()){bill+=150;}

if(ch3.isSelected()){bill+=170;}

if(ch4.isSelected()){bill+=160;}

Statement stmt;

stmt=con.createStatement();

String ContactQuery="insert into customer\_details values('"+nameInput.getText()+"',"+contactInput.getText()+")";

stmt.executeUpdate(ContactQuery);

/\*

\* Required Things:

\* token->order\_id

\* particulars->Starters Selected

\* seats->total Seats required

\* total\_bill->Final bill

\* status->By default during insertion this has to be 'PENDING'

\*/

String chk1="No Starters",chk2="Gobi Manchuri",chk3="Paneer Manchuri",chk4="Baby Corn Manchuri";

String parts="";

if(ch1.isSelected()){parts+=chk1;}

else{

if(ch2.isSelected()){parts+=chk2+",";}

if(ch3.isSelected()){parts+=chk3+",";}

if(ch4.isSelected()){parts+=chk4;}

}

String InsertQuery="insert into orders(order\_id,particulars,seats,total\_bill,status)values(?,?,?,?,?)";

PreparedStatement pstmt=con.prepareStatement(InsertQuery);

pstmt.setString(1, token);

pstmt.setString(2, parts);

pstmt.setInt(3, Integer.parseInt(seatsInput.getText()));

pstmt.setInt(4,bill);

pstmt.setString(5, "PENDING");

pstmt.executeUpdate();

String OrderMaps="insert into order\_maps(order\_id,contact)values(?,?)";

PreparedStatement pt=con.prepareStatement(OrderMaps);

pt.setString(1, token);

pt.setLong(2, Long.parseLong(contactInput.getText()));

pt.executeUpdate();

l1.setText("Token ID:"+token+" inserted into db successfully");

System.out.println("Token ID:"+token+" inserted into db successfully");

} catch (Exception e) {

System.out.println("Please Generate Another Token ID and call function again");

e.printStackTrace();

}

}

public static void main(String[] args){

try{

Connection con;

Class.forName("oracle.jdbc.driver.OracleDriver");

String url="jdbc:oracle:thin:@localhost:1521:XE";

String username="project",password="project";

con=DriverManager.getConnection(url,username,password);

System.out.println("Connected to "+username+" database");

JFrame f1=new JFrame();

JLabel l1=new JLabel("Hello Welcome to the JAVA UI");

f1.setLayout(null);

f1.setTitle("Hotel Reservation System");

f1.setBounds(100, 50, 1280, 720);

f1.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

l1.setHorizontalAlignment(JLabel.CENTER);

l1.setSize(1000,100);

f1.add(l1);

JLabel name=new JLabel("Name:");

name.setBounds(10, 50, 100, 100);

f1.add(name);

JTextField nameInput=new JTextField("Enter your name",16);

nameInput.setBounds(50, 95, 200, 20);

f1.add(nameInput);

JLabel seats=new JLabel("Number of seats:");

seats.setBounds(10, 100, 100, 100);

f1.add(seats);

JTextField seatsInput=new JTextField("Number of Seats required");

seatsInput.setBounds(115, 140, 200, 20);

f1.add(seatsInput);

JLabel contact=new JLabel("Contact Number:");

contact.setBounds(10,200,100,10);

f1.add(contact);

JTextField contactInput=new JTextField("Enter Contact Number");

contactInput.setBounds(115, 200, 200, 20);

f1.add(contactInput);

JLabel starters=new JLabel("Starters");

starters.setBounds(10,200,100,100);

f1.add(starters);

JCheckBox ch1=new JCheckBox("No starters 0/-");

ch1.setBounds(10, 270, 250, 30);

f1.add(ch1);

JCheckBox ch2=new JCheckBox("Gobi Manchurian:150/-");

ch2.setBounds(10, 300, 250, 30);

f1.add(ch2);

JCheckBox ch3=new JCheckBox("Paneer Manchurian:170/-");

ch3.setBounds(10, 330, 250, 30);

f1.add(ch3);

JCheckBox ch4=new JCheckBox("Baby Corn Manchurian:160/-");

ch4.setBounds(10, 360, 250, 30);

f1.add(ch4);

JLabel txnID=new JLabel("Enter Transaction ID:");

txnID.setBounds(10,500,200,20);

f1.add(txnID);

JTextField txn=new JTextField("Transaction ID generated after payment",16);

txn.setBounds(150,500,300,20);

f1.add(txn);

JButton submit =new JButton("SUBMIT");

submit.setBounds(10,550,200,20);

f1.add(submit);

JTextArea ta1=new JTextArea(10,25);

ta1.setEditable(false);

ta1.setBounds(500, 150, 600, 200);

ta1.setLineWrap(true);

Border b3=BorderFactory.createLineBorder(Color.GREEN,10);

ta1.setBorder(b3);

ta1.setFont(new Font("Berlin Sans FB",Font.PLAIN,20));

f1.add(ta1);

JButton calc =new JButton("CALCULATE BILL");

calc.setBounds(10,450,200,20);

ch1.addItemListener(new ItemListener(){

public void itemStateChanged(ItemEvent e){

if(ch1.isSelected()){

ta1.setText(ch1.getText()+"");

ch2.setSelected(false);

ch3.setSelected(false);

ch4.setSelected(false);

}

}

});

ch4.addItemListener(new ItemListener(){

public void itemStateChanged(ItemEvent e){

if(ch4.isSelected()){

ch1.setSelected(false);

}

}

});

ch3.addItemListener(new ItemListener(){

public void itemStateChanged(ItemEvent e){

if(ch3.isSelected()){

ch1.setSelected(false);

}

}

});

ch2.addItemListener(new ItemListener(){

public void itemStateChanged(ItemEvent e){

if(ch2.isSelected()){

ch1.setSelected(false);

}

}

});

calc.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

int bill=150;

String msg="Online Seat Booking:150/-\n",part="";

ta1.setText(msg);

if(!(ch1.isSelected()||ch2.isSelected()||ch3.isSelected()||ch4.isSelected())){msg+="\n\nTotal Bill:150/-";ta1.setText(msg);}

if(ch1.isSelected()){bill=150;ta1.setText(msg+"\n Total Bill:"+bill+"/-");}

if(ch2.isSelected()){bill+=150;part+=ch2.getText()+"\n";ta1.setText(msg+part+"\n Total Bill:"+bill+"/-");}

if(ch3.isSelected()){bill+=170;part+=ch3.getText()+"\n";ta1.setText(msg+part+"\n Total Bill:"+bill+"/-");}

if(ch4.isSelected()){bill+=160;part+=ch4.getText()+"\n";ta1.setText(msg+part+"\n Total Bill:"+bill+"/-");}

//ta1.setText("Total Bill:"+bill);

}

});

f1.add(calc);

JButton payBill=new JButton("Pay Bill");

payBill.setBounds(300,450,200,20);

payBill.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

JFrame f2=new JFrame();

ImageIcon ii=new ImageIcon("images/cropped\_qr.png");

JLabel pic=new JLabel(ii);

pic.setBounds(10, 50, 590, 566);

JScrollPane jsp=new JScrollPane(pic);

f2.getContentPane().add(jsp);

f2.setBounds(750, 50, 600, 700);

f2.add(pic);

f2.setTitle("Scan to Pay");

f2.setVisible(true);

}

});

submit.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

if(!(txnID.getText().equals("")||txnID.getText().equals("Transaction ID generated after payment"))){

String token=gen(txn);

try {

insertRecord(con, l1, ch1, ch2, ch3, ch4, token, nameInput, contactInput, seatsInput);

} catch (Exception e1) {

System.out.println("Token Already exists, will insert modified one");

token=gen\_again(txn);

insertRecord(con, l1, ch1, ch2, ch3, ch4, token, nameInput, contactInput, seatsInput);

}

}

}

});

f1.add(payBill);

f1.setVisible(true);

}catch(Exception e){

e.printStackTrace();

}

}

}

**Server Side Code:**

import java.sql.\*;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

import java.beans.PropertyChangeEvent;

import java.beans.PropertyVetoException;

import java.beans.VetoableChangeListener;

import javax.swing.border.\*;

import javax.swing.table.\*;

import javax.swing.table.DefaultTableModel;

public class server\_side {

public static void createTable(String stats,Connection con,JButton buttonType,boolean bl){

final JFrame frame;

JTabbedPane myListTabs = null;

frame = new JFrame("Pending Bookings");

myListTabs = new JTabbedPane();

/\* myComicsListPane = new ComicsListPane();

myListTabs.add(myComicsListPane); \*/

/\* myListTabs.setTitleAt(myListTabs.getTabCount()-1, "Status"); \*/

JTable myComicsTable = null;

DefaultTableModel model=new DefaultTableModel();

myComicsTable = new JTable(model);

myComicsTable.setPreferredScrollableViewportSize(new Dimension(750, 110));

myComicsTable.setFillsViewportHeight(true);

myComicsTable.setFillsViewportHeight(true);

try {

System.out.println(buttonType.getText()+" button working");

Statement stmt;

stmt=con.createStatement();

String query;

if(bl)

query="select \* from orders where STATUS='"+stats+"' or STATUS='REJECTED'";

else

query="select \* from orders where STATUS='"+stats+"'";

ResultSet rs=stmt.executeQuery(query);

ResultSetMetaData rsmd=rs.getMetaData();

int col=rsmd.getColumnCount();

String[] colName=new String[col];

for(int i=0;i<col;i++)

colName[i]=rsmd.getColumnName(i+1);

model.setColumnIdentifiers(colName);

while(rs.next()){

String od=rs.getString(1);

String parts=rs.getString(2);

int seats=rs.getInt(3);

int bill=rs.getInt(4);

String final\_bill=Integer.toString(bill);

String seats\_conf=Integer.toString(seats);

String final\_status=rs.getString(5);

String[] row={od,parts,seats\_conf,final\_bill,final\_status};

model.addRow(row);

}

myComicsTable.setDefaultEditor(Object.class, null);

//myComicsTable.setEnabled(false);

} catch (Exception e1) {

e1.printStackTrace();

}

KeyboardFocusManager.getCurrentKeyboardFocusManager().addVetoableChangeListener( "focusedWindow",new VetoableChangeListener() {

private boolean gained = false;

@Override

public void vetoableChange( PropertyChangeEvent evt ) throws PropertyVetoException {

if ( evt.getNewValue() == frame ) {

gained = true;

}

if ( gained && evt.getNewValue() != frame ) {

frame.dispose();

}

}

} );

JScrollPane scrollPane = new JScrollPane(myComicsTable);

scrollPane.setPreferredSize(new Dimension(600, 110));

frame.getContentPane().add(myListTabs);

frame.setAlwaysOnTop(true);

frame.pack();

frame.setBounds(500, 150, 950, 250);

frame.add(scrollPane, BorderLayout.CENTER);

frame.setVisible(true);

}

public static void main(String[] args){

try{

Connection con;

Class.forName("oracle.jdbc.driver.OracleDriver");

String url="jdbc:oracle:thin:@localhost:1521:XE";

String username="project",password="project";

con=DriverManager.getConnection(url,username,password);

System.out.println("Connected to "+username+" database");

JFrame MainFrame=new JFrame();

MainFrame.setTitle("Hotel Management System");

MainFrame.setLayout(null);

MainFrame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

MainFrame.setBounds(100,50,1280,720);

JLabel wlcm=new JLabel("Welcome to the app");

wlcm.setHorizontalAlignment(JLabel.CENTER);

wlcm.setSize(1000, 100);

MainFrame.add(wlcm);

JButton pendingList=new JButton("Pending List");

pendingList.setBounds(10, 50, 150, 20);

MainFrame.add(pendingList);

JButton viewBooked=new JButton("Booked List");

viewBooked.setBounds(250,50,150,20);

MainFrame.add(viewBooked);

JLabel tkn=new JLabel("Token ID:");

tkn.setBounds(20, 60, 100, 100);

MainFrame.add(tkn);

JTextField tk\_Field=new JTextField("Enter the token Number");

tk\_Field.setBounds(80, 100, 200, 20);

MainFrame.add(tk\_Field);

JButton apprv=new JButton("Approve");

apprv.setBounds(25,150,100,20);

apprv.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

if(tk\_Field.getText().equals("")||tk\_Field.getText().equals(" ")||tk\_Field.getText().equals("Enter the token Number")){

wlcm.setText("Enter Valid Token Number");

}

else{

try {

System.out.println("Approve Button Clicked");

Statement stmn=con.createStatement();

String query="update orders set STATUS='BOOKED' where order\_id='"+tk\_Field.getText()+"'";

stmn.executeUpdate(query);

wlcm.setText("Approved token: "+tk\_Field.getText()+" Successfully");

} catch (SQLException Se) {

wlcm.setText("Error with TOKEN id please check again.");

Se.printStackTrace();

}

}

}

});

MainFrame.add(apprv);

JButton reject=new JButton("Reject");

reject.setBounds(150,150,100,20);

reject.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

if(tk\_Field.getText().equals("")||tk\_Field.getText().equals(" ")||tk\_Field.getText().equals("Enter the token Number")){

wlcm.setText("Enter Valid Token Number");

}

else{

try {

System.out.println("Reject Button Clicked");

Statement stmn=con.createStatement();

String query="update orders set STATUS='REJECTED' where order\_id='"+tk\_Field.getText()+"'";

stmn.executeUpdate(query);

wlcm.setText("Rejected token "+tk\_Field.getText()+" Successfully");

} catch (SQLException Se) {

wlcm.setText("Error with TOKEN id please check again.");

Se.printStackTrace();

}

}

}

});

MainFrame.add(reject);

pendingList.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

try {

createTable("PENDING", con,pendingList,true);

} catch (Exception e1) {

e1.printStackTrace();

}

}

});

viewBooked.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

try {

createTable("BOOKED", con,viewBooked,false);

} catch (Exception e1) {

e1.printStackTrace();

}

}

});

MainFrame.setVisible(true);

}catch(Exception e){

e.printStackTrace();

}

}

}

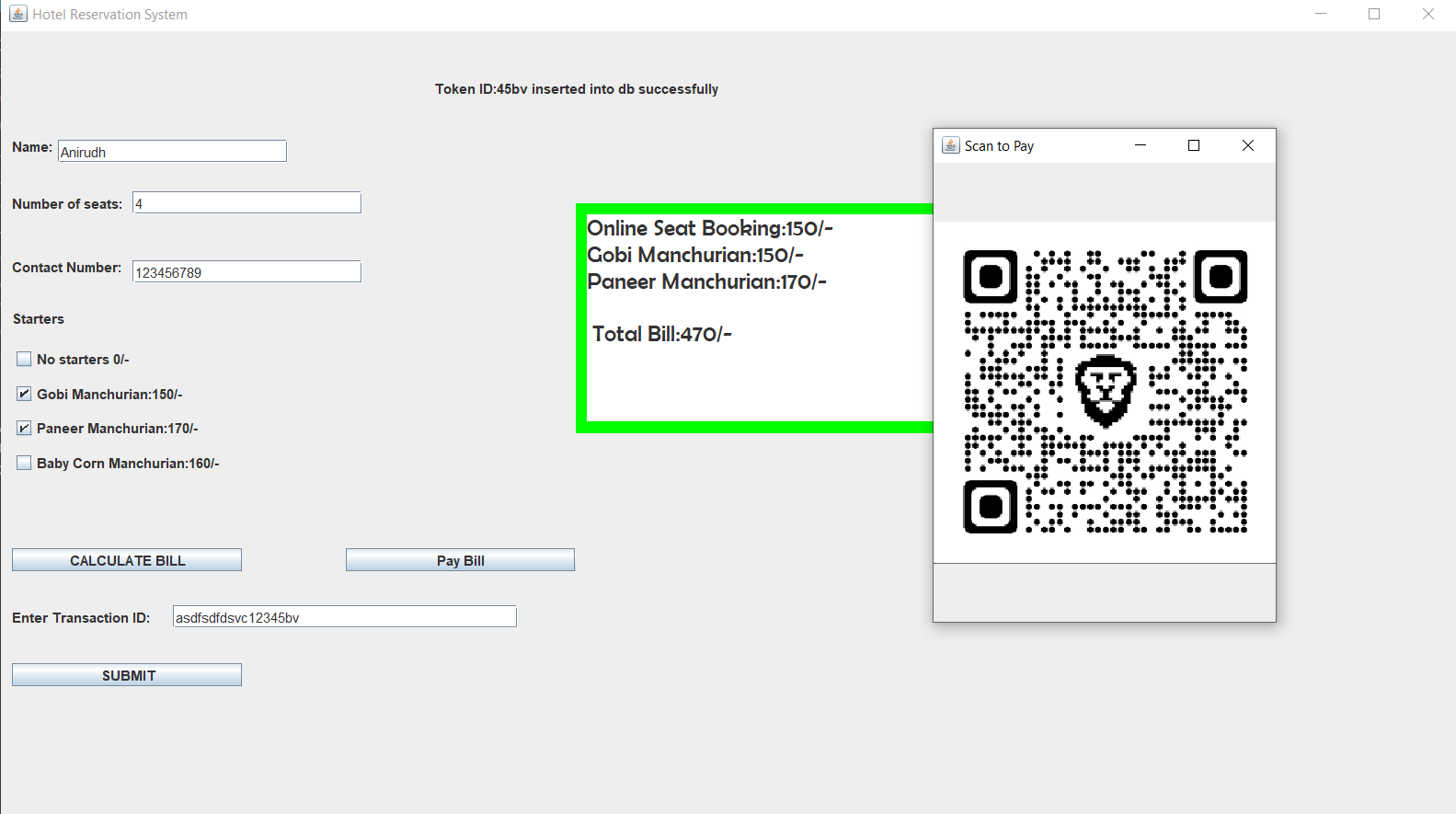
**Source code is hosted at** [**https://www.github.com/anirudhp06/Major-project/**](https://www.github.com/anirudhp06/Major-project/)

**Or you can scan this QR Code:**

****

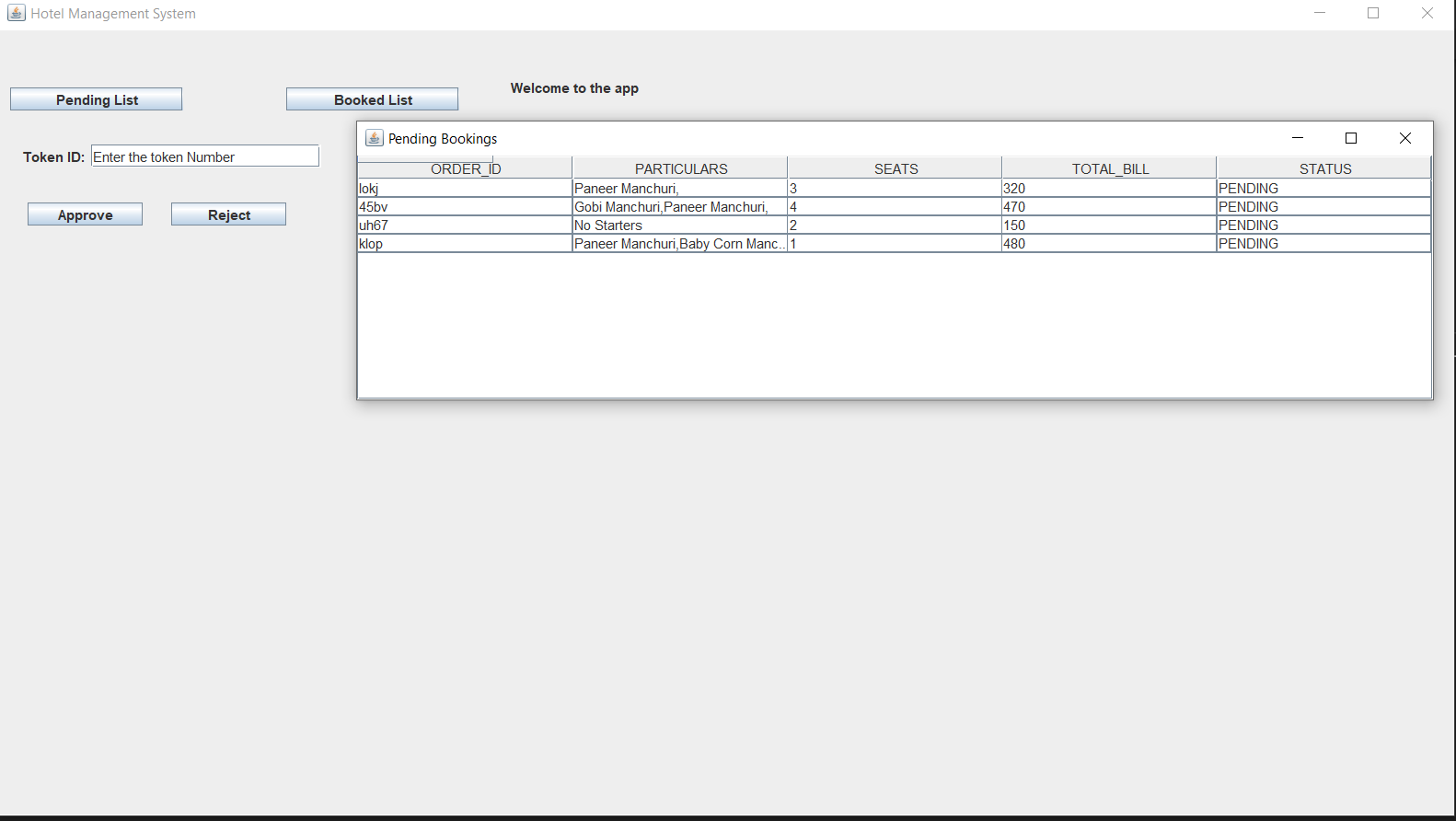
1. **UI Design and Outputs**

Client Side

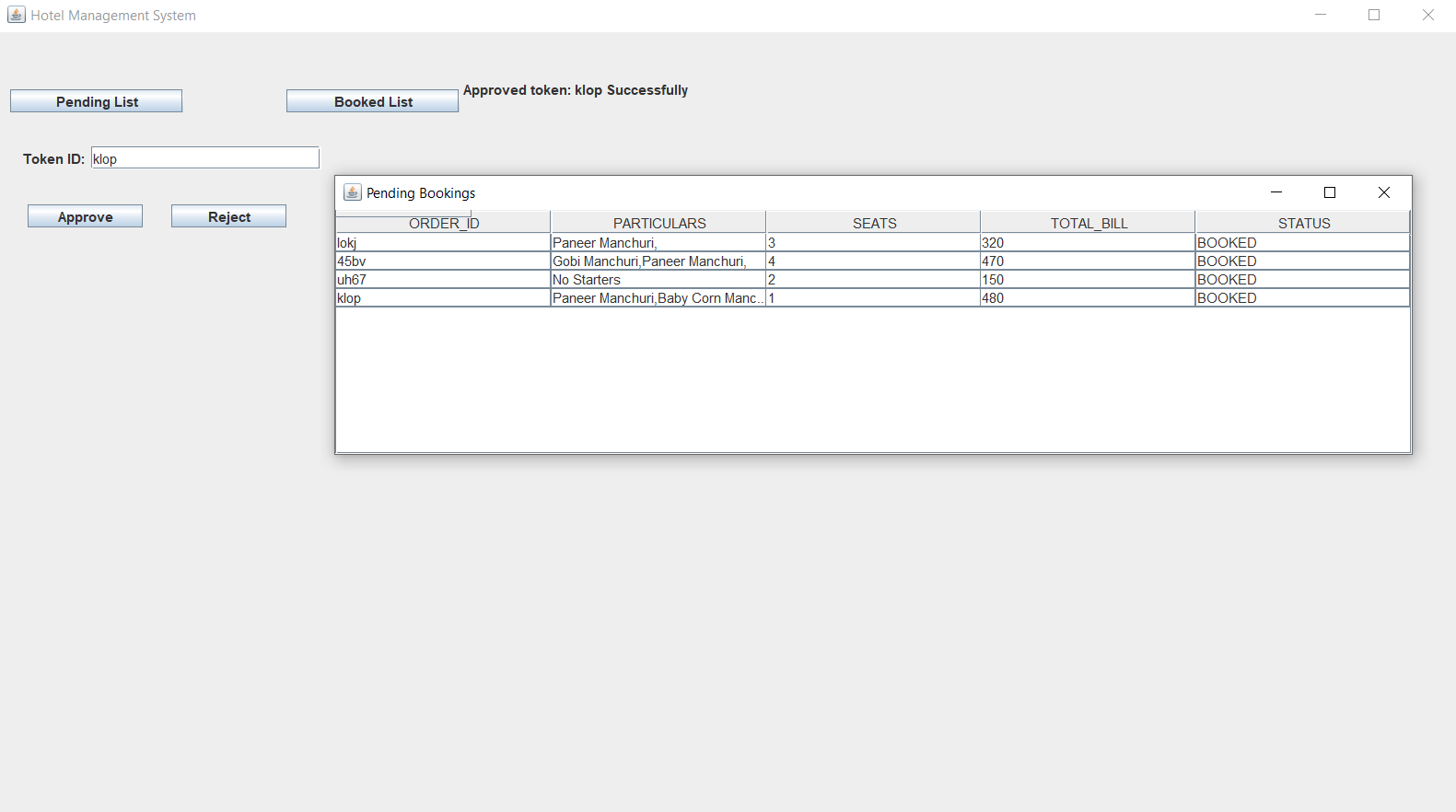
****

Demonstration of Program that runs on Client side, this shows details of customer & particulars they have ordered & the token number that has been generated for them for future references.

**Server Side**

****

Demonstration of the program that runs on administrator side of the hotel, this shows what all are necessary for the manager to verify the online bookings, the current picture shows which all booking are pending and awaiting approval from manager.

****

Another picture of the program that runs on the administrator/manager side of the hotel, this picture shows which all bookings have been approved and what all things they have ordered as starters which has to be served to them when they arrive.

1. **Implementation (Deployment)**

Implementation is the stage where the theoretical design is turned into a working system. Once the design is complete, most of the major decisions about the system have been made. The goal of the coding phase is to translate the design of the system into code in a given programming language. For a given design, the aim in this phase is to implement the design in the best possible manner.

The coding phase affects both testing and maintenance profoundly. Since the testing and maintenance costs of software are much higher than the coding cost, the goal of the coding should be to reduce the testing and maintenance effort. Hence, during coding the focus should be on developing the programs that are easy to read and understand, neither simply on developing the programs that are easy to read and understand, nor simply on developing programs that are easy to write.

1. The Implementation stage consists of Making the necessary changes for the system as desired by the user.
2. Training the user personal prior to the implementation of two steps shown below as to be carried out.
3. Testing the developed program with the simple data.
4. Detection and Correction of errors.

Oracle 10g was used as database in this project as its very easy to setup and has ability to serve multiple users at once. Java was the major programming language used in the development, it contributes to all the working of the code, it has rich modules which makes the project user friendly and also robust in nature.

1. **Testing Methods**

In a software development project, different errors can be incurred at any stage during development. There are techniques for detecting and eliminating errors that originate in that phase, However, no technique is perfect, and it is expected that some of the errors of the earlier phases will finally manifest themselves in the code. This is particularly true because in earlier phases of software development most of the verification techniques are manual because no executable code exists.

Ultimately, these remaining errors will be reflected in the code. Hence, the code developed during the coding activity is likely to have some designing errors, in addition to errors introduced during the coding activity.

Behavior can be observed, testing is the phase where the errors lingering from all the previous phases must be detected. Hence, testing performs a very critical role for quality assurance and for ensuring the reliability of software.

During testing, the program to be tested is executed with a set of test cases, and the output of the program for the test cases is evaluated to determine if the program is performing as expected.

Due to its approach, dynamic testing can only ascertain the presence of errors in the program: the exact nature of the errors is not usually decided by testing. Testing forms the first step in determining the errors in a program. Clearly, the success of testing to reveal errors in code depends critically on the test cases.

Testing a large system is a very complex activity, and like any complex activity it has to be broken into smaller activities. Due to this, for a project, incremental testing is generally performed, in which components and subsystems of the system are tested separately before integrating them to be called a complete system for system testing.

This form of testing, though necessary to ensure quality for a large system, introduces new issues of how to select components for testing and how to combine them to form subsystems and systems.

**Types Of Testing:**

1. **Integration Testing:**

Integration testing is of two kinds: Bottom-Up integration and Top Down Integration. For this system, Bottom-Up Integration Testing was carried out. Bottom-up Integration is the traditional strategy used to integrate the components of a software system into a functioning whole.

Bottom-up integration consists of a Unit Testing, followed by the Sub System Testing, and testing of the entire system.

Unit testing has the goal of discovering errors in the individual modules of the system. The primary function of Sub-System testing is to verify operation of the interface between the modules in the subsystem.

System testing is concerned with the decision-logic, control flow, recovery procedures, capacity.

1. **User Acceptance Testing:**

The user acceptance test verifies that the system's procedures operate up to system's specifications and that the integrity of the data is maintained. It involves the execution of Procedure Test, Performance Test and Peak-Load Test.

Unit Testing: Individual components are to ensure that they operate correctly. Each component is tested independently, without other system component. This system was tested with the set of proper test data for each module and the results were checked with the expected output.

Unit testing focuses on verification effort on the smallest unit of the software design module. This is also known as module testing.

The modules were integrated together and the new system tested by allowing users to enter samples

data.

This helped to verify that it accepts the data and processes it, in the manner desired.

**VALIDATION:**

This involved entering sample data into the new system, so as to compare its tracking functionalities with the existing systems.

1. **Advantages of Project**

This program is gives good advantage for both the hotel and for the potential customers too…

1) The hotel can know when a dine-in is booked and can make all necessary arrangements

to it prior to the customer arriving.

2) The user can properly plan their lunch or dinner.

3) This avoids waiting for seats when someone plans to go out on special occasions.

**Conclusion**

The aim of this project is to provide a facility for the users to book hotel table at ease of their fingertips without much hassle and also to provide platform for the businessmen to serve the people.

**Future Enhancement**

There is always room for improvement everywhere. We can further enhance the present software by extending support to mobile users, hence designing same software for mobile platforms. Can even keep updating the UI to keep the users engaged.

**K.L.E. SOCIETY’S**

P.C.JABIN SCIENCE COLLEGE,

AUTONOMUS,

(Affiliated to KARNATAK UNIVERSITY, DHARWAD)

**HUBBALLI -580031**

** Bachelor of Computer Application**

**e-mail:**klesbca@gmail.com Ph: 0836-2372298