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 U 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,

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.

Prof. Tejaswini Apte   Prof. Sunil Vernekar
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

(Anirudh Parvatikar)

(Vineeth Kemtur)

<u>ACKNOWLEDGEMENT</u>

ABSTRACT

Gen "Z" is too busy running after success and a career that they have no time to invest in other things. Gone are the days of waiting in long queues to dine at your favorite restaurant. As the digital world is extending its limits, so are the consumers extending their needs. Flight booking, taxi booking, bus booking, doctor's appointment, and with all these, the next is Hotel Reservation System.

With all of us going digital and online platforms being ruling in every arena, it is not uncommon to have Online restaurant & table booking apps. They perfectly meet the changing and growing demand of people across the globe and hence there are more and more firms joining this bandwagon.

It is learned from the reports that OpenTable, which is termed to be a leading player in this arena, is catering to over 16 million people per month all around the world.

Other prominent online restaurant booking platforms such as Eveve & Seat Me have similar statistics to flaunt. Now, these facts and figures are certainly alluring for the aspiring entrepreneurs out there and if you too consider yourself to be in that same zone, and want to enter the market with a restaurant table booking app then this post will prove to be a lot helpful for you. So, let's begin this journey!

Hotel Reservation 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

I.   Introduction

Hotel Reservation 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.

II.   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.

Market Analysis and Growth in Restaurant Industry

a) As per the Global Industry Analyst report, the market of food delivery service has seen a humongous response in the previous years with a revenue of $3 Trillion in 2020 and enjoying a growth rate of 5.4% in the coming years to $4.1 Trillion by 2026.
b) After the rise in vaccination rates, 45% of the customers preferred to choose restaurant dine than cook food at home.
c) The food industry faced the highest turnover of all time of around 75%.
d) The virtual platform increases the sales rate to 452%, supercharging the existing industry in many out-of-the-box ways.
e) One most interesting fact, 58% of consumers are likely to support local farms and food producers.

How Technology Influences Table Reservation Program Development?

As we try to remember the way online reservations started, it began with managers, hostesses, or other staff members taking calls and then writing the names and times on paper. This certainly was occupying a lot of time for the staff members who would rather be busy with other tasks like clearing tables or maybe assisting customers with something.

However, with the advancement of technology, things changed for the better. Today more and more people are having constant access to the internet and thus the restaurant table booking app makes the process quicker and more convenient for both customers and restaurant staff. With this software, the trend of online restaurant reservations emerged and today people can easily make bookings on a restaurant's personal website, or maybe via the third-party reservation system. Depending on the venue's location, staff, and traffic, one can choose the best online restaurant booking system for their restaurant.

If you think your restaurant is not being found easily, just make sure that you reach the right experts who can help you with everything. They would enhance your business visibility and thus reservations and profits making it easier for you to earn the best results.

How Does the Restaurant Booking System Prove to be Beneficial?

There was once a time when in the restaurant business "Open Table" was a much-hated term, as it implies an empty seat in the dining room. It was due to the reason that restaurants pay upfront for electricity, rent, food, utilities, etc. and in any case, if a seat remains unfilled it signifies a loss of profit, as the aforementioned bills don't lower depend on how many customers they had in a day.

To deal with this issue and to reduce the number of open tables in a restaurant, as well as to limit the waiting time for paying customers, restaurants started taking reservations.

This was how the Hotel Reservation System came into existence. Ever since some sort of reservation system began, things became easier for the restaurant owners and the customers.

For Business Owners:

One can never know how busy a dinner night will prove to be with open tables, as it remains an unknown fact, however, when it is to take reservations, it gives a better idea to owners of how busy or quiet the nights are going to be. And even in the case when there are a low number of reservations, the restaurant's owners will have enough time to come up with some special idea to market on social media accounts, in order to encourage customers to come out and spend money.

This also enables the business owner to prepare the staff in case of huge influx of customers into hotel.

On the lucky days, when the Hotel Reservation System shows that it will be packed, the chefs and kitchen staff have a fair warning and enough time to prepare themselves for an incoming rush at a particular time. At the same time, hostesses who are making reservations, or are seating walk-in customers, can space the seating times out enough as that way the kitchen won't get slammed all at once.

For the Customers:

It is not solely the owners and kitchen staff that benefit from the Hotel Reservation System, but it also made for the customers to make their dining experience more convenient and enjoyable. Like, when someone books a table for 6:00 p.m. on a Saturday, they can expect to be seated quite close to that time, thus reducing the waiting time. Getting the guests seated as close to their time slot will certainly result in happy customers who will surely return next time too.

There are a few other benefits of having a restaurant table order management system for restaurant owners as well as customers:

For restaurant Owner:
a) Keeps reservations all in one place

b) Chances to convert website visits into reservations by suggesting customers to download the program to make reservations.
c) Reduced time devoted to taking calls during operation hours
d) Allows to confirm or reject reservations via designated communication medium.

For Customers:
a) Need to make a phone call is eliminated
b) No requirement to wait for an employee to answer
c) Able to access 24/7 and not just during operating hours
d) Allows them to easily read reviews of your restaurants as well as compare prices online.

How does a restaurant booking app serve to potential market?

Every industry in order to revolve their sum, brush up their efforts in defining a successful business model. A restaurant owner allows their customers flexible table booking slots as per their preferences.

It is mainly divided into two categories:
1. Dedicated Type
2. Aggregator Type

1. Dedicated Type:

In such a model, users book their table within the application, it is highly focused on the user's choices and serves them to plan within their budgets. Basically, in this model, owners have all the authority to manage the whole app functionality.

2. Aggregator Type:

In such kind of model, the app owner shakes hands with the store owners to reflect their great services to the user. They combinedly grab the wholesome market of restaurant bookings within the platform. This mutual combination serves faster services. Here restaurant owners have the authority to offer deals and discounts on table booking on the app platform.

Top Features of Hotel Reservation System:

When we started writing Hotel Reservation System, we tried to make it a point that the customers have a real-time booking experience using our business software.

Some of the interesting features that must be a part of the development process and make things easier for you and your customers are:
a) **Reward Programs:** Can there be a more attractive way to lure customers to visit your restaurants.
b) **Gift Cards:** Gift tastefully, receive Happily. People nowadays really prefer Gift Cards over traditional gifts and we can't appreciate this move more.
c) **History of Bookings:** When having a Restaurant Booking System, it is a lot easier to check the previous booking whenever required.
d) **Online Payment System:** No more time wasted waiting for someone to get the bill cleared and that can be done by customers themselves only through the e-wallet app development. Hence, things became more hassle-free.

    e) **Find & Explore:** A lot easier to trace the restaurant's location on the maps provided on the restaurant's website or mobile app. It is also possible to explore easy routes to the location.

    f) **Get Rewarded:** Some restaurants have this amazing way of rewarding their customers from time to time for their loyalty and the support to extend the customer base.

    g) **Manage Restaurant Reservation:** Bookings and reservations can be made easily and conveniently 24/7 contrary to phone calls that can be made only during operational hours.

    h) **Gain Access:** The portal access can be given to staff members and employees as per your work requirements from them. This brings ease to the process.

    i) **Share Experience:** After visiting the restaurants, customers can share their experience in the review section based on their experience. In case of a bad review, restaurants can make efforts accordingly to please the customer.

    j) **Travel Ready:** The best thing about the world going digital is that you can manage work from just anywhere, anytime.

Objective of this program:

The project "Hotel Reservation System" is aimed to develop to maintain the day-to-day state of admission/vacation of hotel table, List of Bills etc…

    The following are the main objectives of the Hotel:

    a) Keeping user satisfaction as at most priority.

    b) Scheduling the allotment of user with room to make it convenient for user.

    c) Keeping records of user registration details accurately arranged order so that the treatment of

    d) Customers becomes quick and satisfactory.

    e) Keeping details about the users, their needs and payment detail reports etc.

    f) Keeping the best hotel facilities.

    g) Give the user choices from the cheapest rooms between the hotels.

    h) Keep the user connected by the hotel to receive the confirmation.

    i) Send the confirmation letter to the user about his bill and date.

Critical Success Factors:

    a) Simplifies Front Office Tasks – A hotel reservation system can provide the 'real time' status of the hotel tables.

    b) Enables distribution.

    c) Improves guest reservations.

    d) Increases multi-tasking.

    e) Centralizes hotel management.

Organization chart and responsibilities:

The Internet connection is also a constraint for the website. Since the website fetches data from the database over the Internet, it is crucial that there is an Internet connection for the website to function. The web portal will be constrained by the capacity of the database.

General rules (assumptions):

One assumption about the product is that it will always be used on computer that have enough performance. If the computer does not have enough hardware resources available it may cause irrelevant problems.

III. Technical Requirements (Hardware and Software)

System Requirements:

a) Intel i5 and above, AMD Ryzen 3 and above

b) 8GB RAM DDR4

c) Windows 10 64 bit

d) 50GB Hard Disk Space

Software Requirements:

a) JAVA 15+

b) Swing Package

c) AWT package

d) Internet connection

e) Microsoft Visual Studio Code

f) 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.

IV.   Project Description

The data of the internal and external aspects (strategic information) of many Hotels were obtained from the formulation of the questions list and from the interview. The results were then analyzed by us. The questionnaire was distributed to 8 respondents that responsible in hotel managerial. They are the General Manager, Head of the Department of Human Resources, Accounting, Housekeeping, Sales & Marketing, Front Office, Food & Beverage and Engineering. From the questionnaires, we can know the average existing condition from respondents of the hotel and the average urgency handling also from the respondents.

From the research, it is concluded that Hotels which were interviewed needs to have facility to upscale their booking methodologies. So, the hotel needs to implement diversification strategic. It is a strategic for collecting all internal powers and avoiding all company external threats in order to get all the long-term opportunity. This research also produces a strategic alternative for many hotels.

Literature Survey details:

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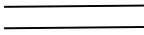
Bookings will be processed by the admin that can be viewed by the customer. Customer can know the currently processing status of their table booking.

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.

REFERENCES:

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/
5. Courtesy https://www.tutorialspoint.com/ for providing knowledge about ways to do documentation for project.
6. Courtesy https://www.octalsoftware.com/ for providing useful insights into current scenario of reservation system of hotel.

Following are the notations used in Data Flow Diagram

Notation	Meaning
⬭	Process
═══	Data Store
▭	External Entity
⟶	Data Flow

I.   System Design

UML design is the shortest form of "**Unified Modeling Language**". The purpose of this
modeling language is to visualize the design of the system. There are total 14 types of UML
diagram. They are:

a) Class Diagram.
b) Component Diagram.
c) Deployment Diagram.
d) Object Diagram.
e) Package Diagram.
f) Profile Diagram.
g) Composite Structure Diagram.
h) Use Case Diagram.
i) Activity Diagram.
j) State Machine Diagram.
k) Sequence Diagram
l) Communication Diagram.
m) Interaction Overview Diagram.
n) Timing Diagram.

Data Flow Diagram for
Client side Program

Customer
Opens
Program

Enter's Name,Contact Number

Enter's Starters if required

Processes
the given
requests and
generates bill

Enter Number of Seats Required

Bill is shown to customer

Customer
Pays
Outstanding
Bill

Provides Txn ID of Payment

Validates Txn
ID and
generates
TOKEN

Token is given

Customer
gets TOKEN

Sends all data to DB

All the
processed data
is stored in DB

Data Flow Diagram for
Server Side Program

Hotel
Admin/Manger
Logs In to
program

Requests data of pending booking

DB

◄— **Updates to DB**

Processes
the given
input

Requests data is given

Admin/Manager

Admin Either approves/rejects bookings
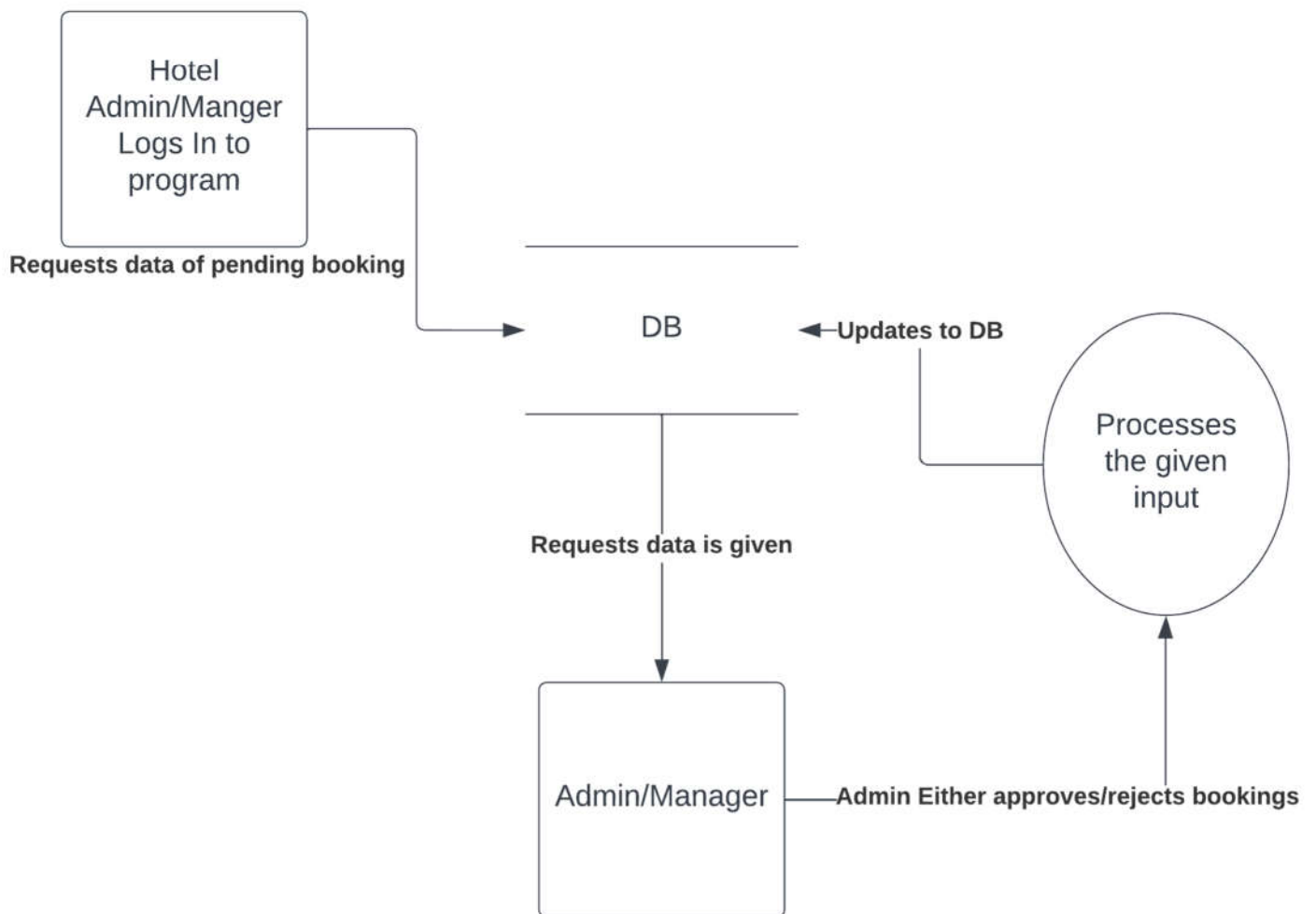
Data Flow Diagram for
Approving Seat Booking
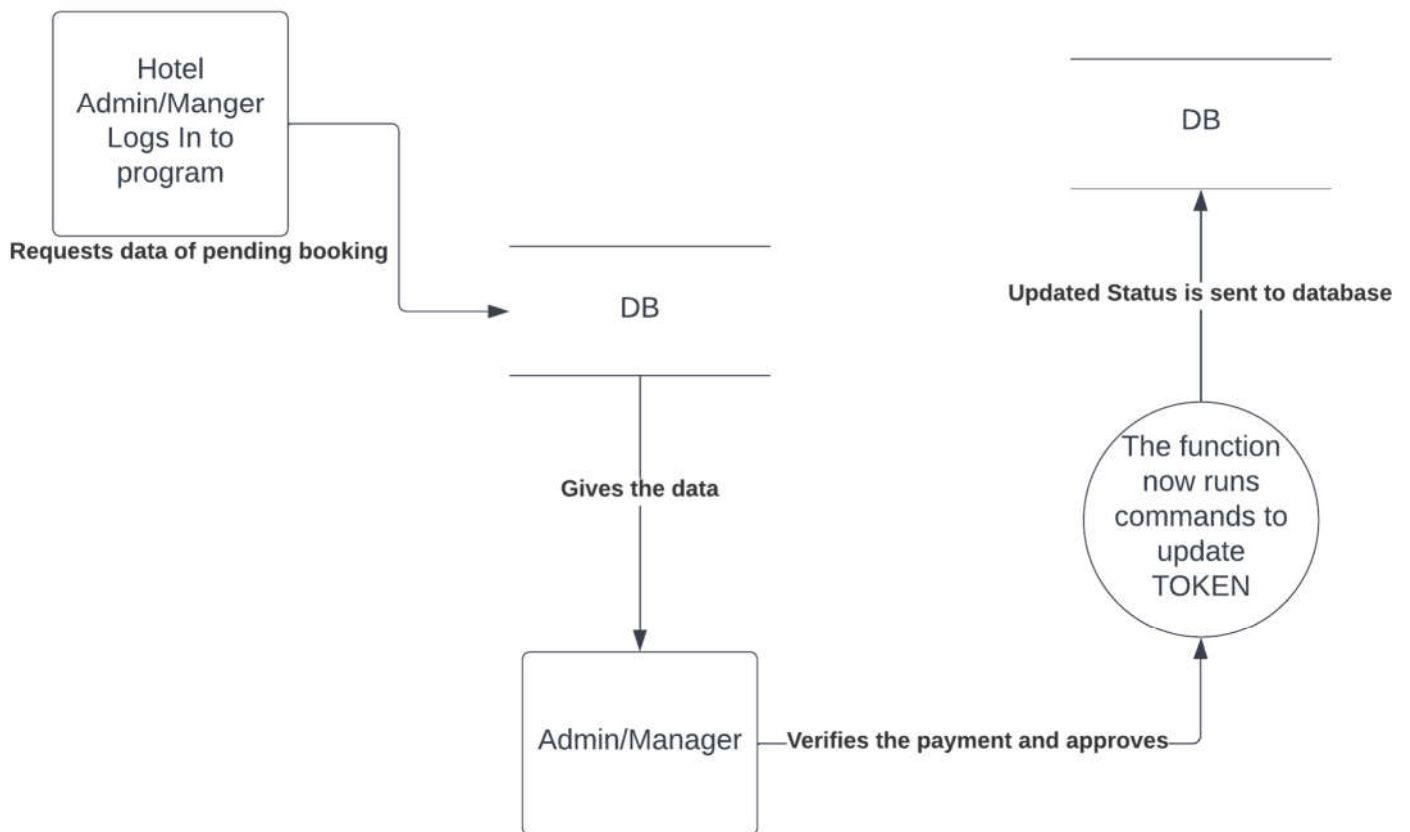
Data Flow Diagram for
Reject Seat Booking

Hotel
Admin/Manger
Logs In to
program

Requests data of pending/approved booking

DB

Gives the data

DB

Rejected Status is sent to database

Admin/Manager ——**Identifies inconsistencies and rejects**——

The function
now runs
commands to
update
TOKEN

Start

Enter Name
Enter Number of Seats
Enter Starters
Pay total Amount

Enter Transaction ID

Generate Token ID and Insert into DATABASE and Display it to Customer

STOP

Mechanism of
Generating TOKEN

Txn ID from Payment

Txn ID is
input for
TOKEN
generation

Generate
TOKEN
based on last
4 characters
of Txn ID

Check
if TOKEN
is already
present in
DB

Yes

No

Generate
TOKEN
based on first
4 char of Txn
ID

Display the
TOKEN

Next Operations are carried out

[Use Case Diagram for Hotel Reservation System]

August 12, 2022
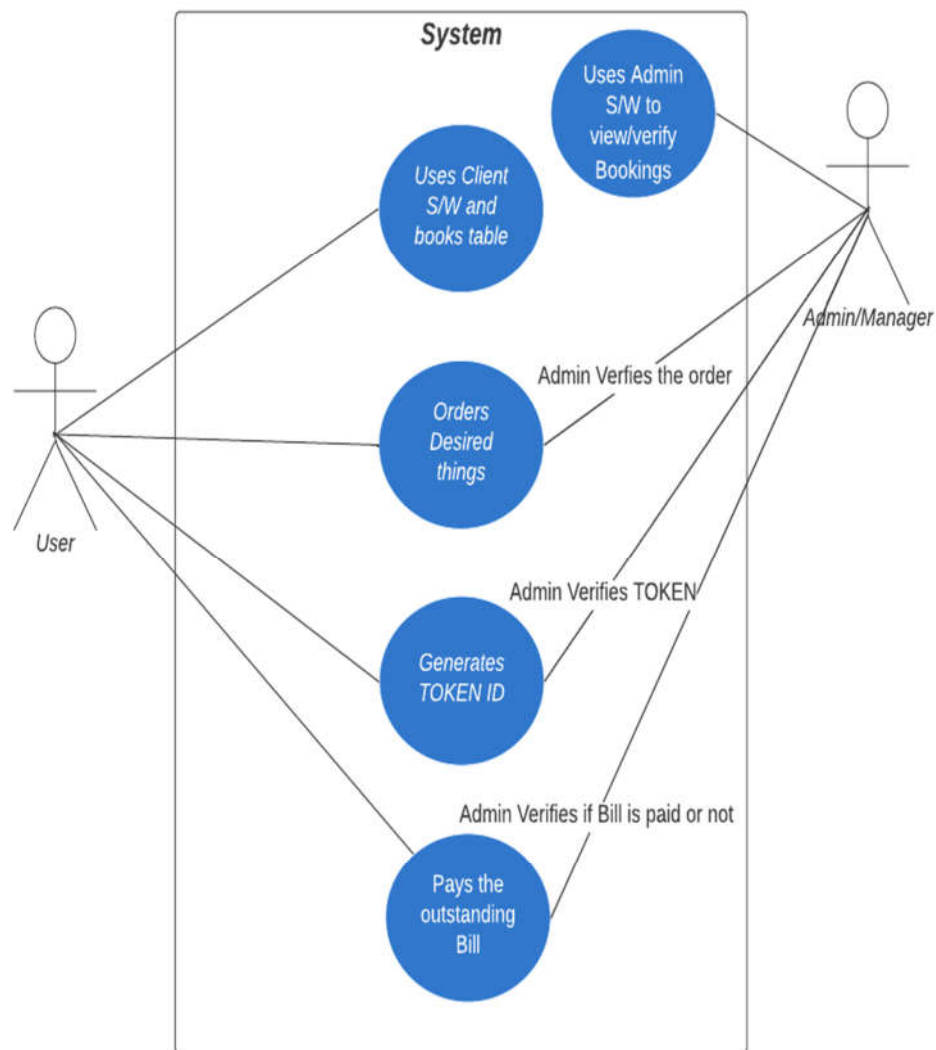
Database ER Diagram for Hotel Reservation System

Customer_details	
Name	Varchar
Contact	int(PK)

Orders	
order_id	varchar(PK)
particulars	varchar
seats	int
total_bill	int
status	varchar

Foreign Key to contact of Customer_detals

Foreign Key references to order_id of Orders table

Order_maps	
order_id	varchar(FK)
contact	int(FK)
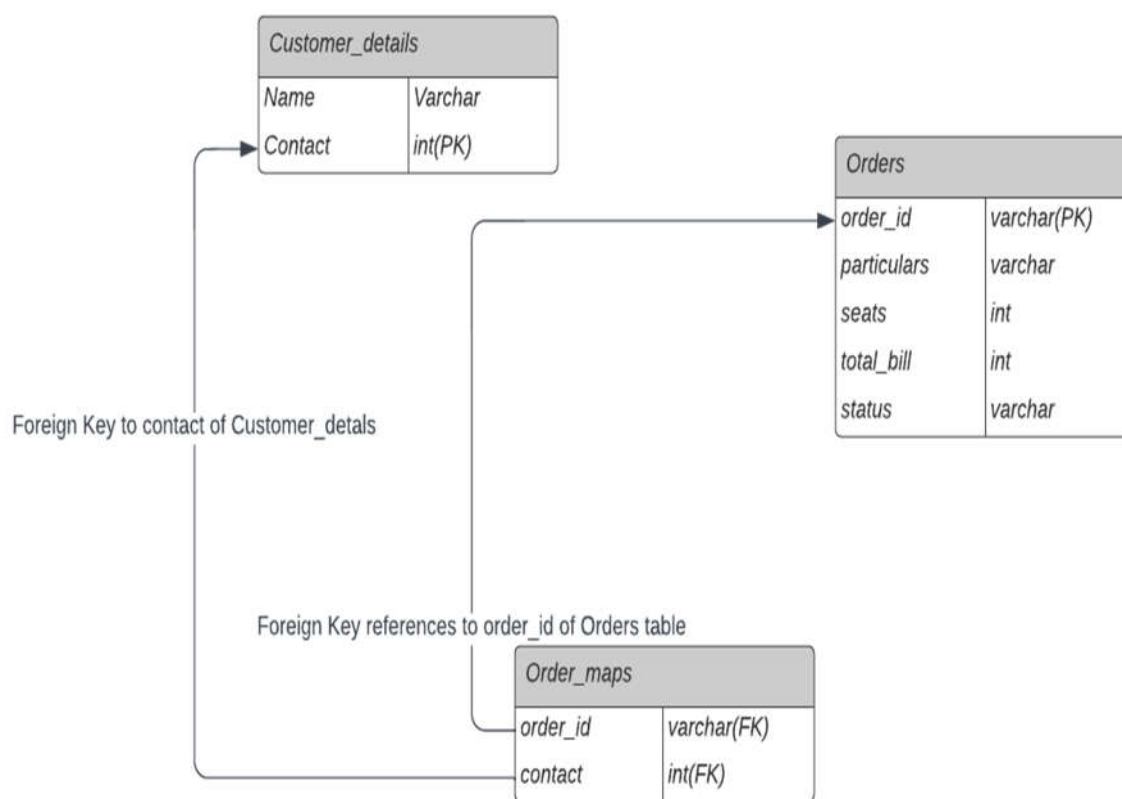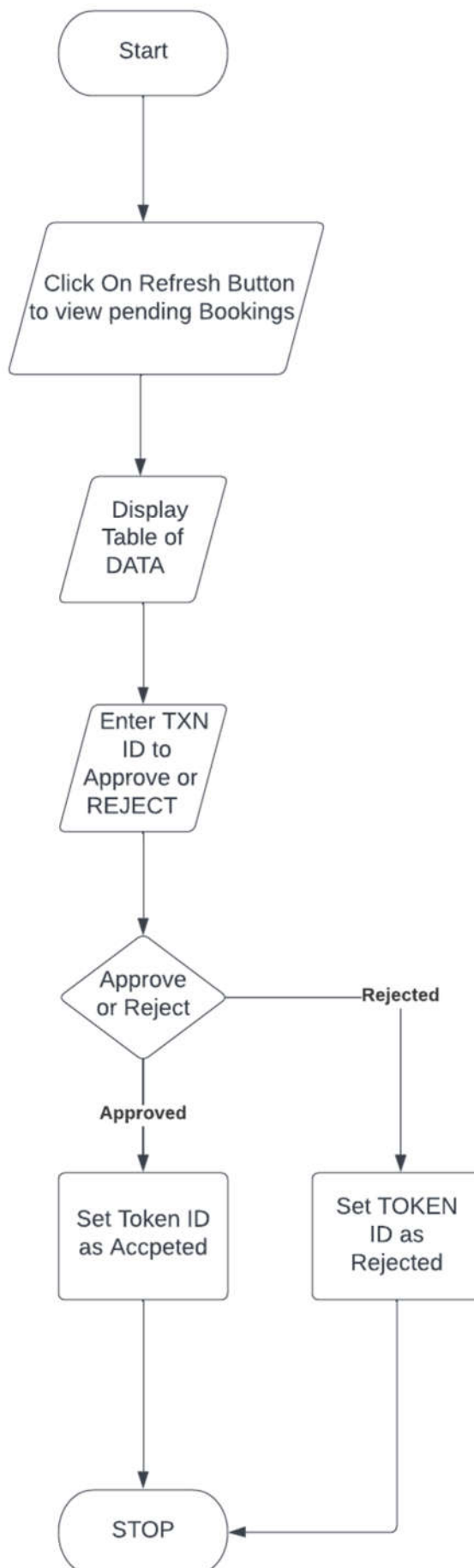
Database Schema:

Table Name: Customer_details

Field Name	Data-type
Name	varchar(200)
contact	int primary key

Table Name: orders

Field Name	Data-type
Order_id	varchar(200) primary key
Particulars	varchar(1000)
Seats	int
Total_bill	int
Status	varchcar(50)

Table Name: order_maps

Field Name	Data-Type
Order_id	varchar(200) foreign key to orders
Contact	int foreign key to customer_details

V.   Algorithm

Algorithm for Client-Side Program

```
Step 1: Request for the client's Name and Contact
Number.
Step 2: Request for number of people accompanying
him/her to hotel.
Step 3: Ask if the client needs to be served with
starters or not.
Step 4: Calculate Total bill based on selection client
made.
Step 5: Show/Redirect to the payment method of the
hotel to complete payment.
Step 6: Ask client to input transaction ID of the
completed payment.
```

Step 7: Generate Token ID based on the given transaction ID.

Step 8: Check if the TOKEN ID generated already exists in system or not.

Step 9: If TOKEN ID already exists then generate new one and repeat step 8 until unique TOKEN ID is generated.

Step 10: Display the generated TOKEN ID to user for his/her future reference.

Algorithm for Server-Side Program.

Step 1: Provide with options of viewing pending, approved, rejected bookings.

Step 2: If Pending bookings option is clicked, display all the pending table booking data to admin/hotel manager.

Step 3: If approved bookings option is clicked, display all the pending table booking data to admin/hotel manager.

Step 4: If rejected bookings option is clicked, display all the pending table booking data to admin/hotel manager.

Step 5: If admin/manager wants to approve pending booking, request for the TOKEN ID of pending booking.

Step 6: If admin/manager wants to reject pending booking, request for the TOKEN ID of pending booking.

Step 7: If admin/manager wants to reject already approved booking, request for the TOKEN ID of approved booking.

VI. Source Code

Client-Side

```java
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";
```

```java
            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().addVetoableChangeListe
ner( "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();
            }
        }
}
```

Database Code:

```
      create table customer_details
(
    name varchar(200),
    contact int primary key
);


create table orders
(
    order_id varchar(200) primary key,
    particulars varchar(1000),
    seats int,
    total_bill int,
    status varchar(50)
);
```

```
create table order_maps
(
    order_id varchar(200),
    contact int,
    foreign key(order_id) references orders(order_id),
    foreign key(contact) references customer_details(contact)
);
```

Source code is hosted at https://www.github.com/anirudhp06/Major-project/

Or you can scan this QR Code(To View/Download):
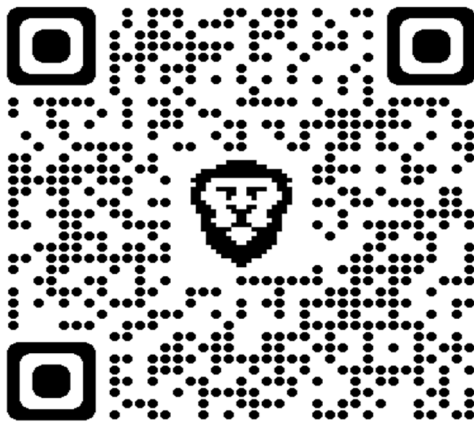
VII.   UI Design and Outputs

Client Side

Above picture is 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.
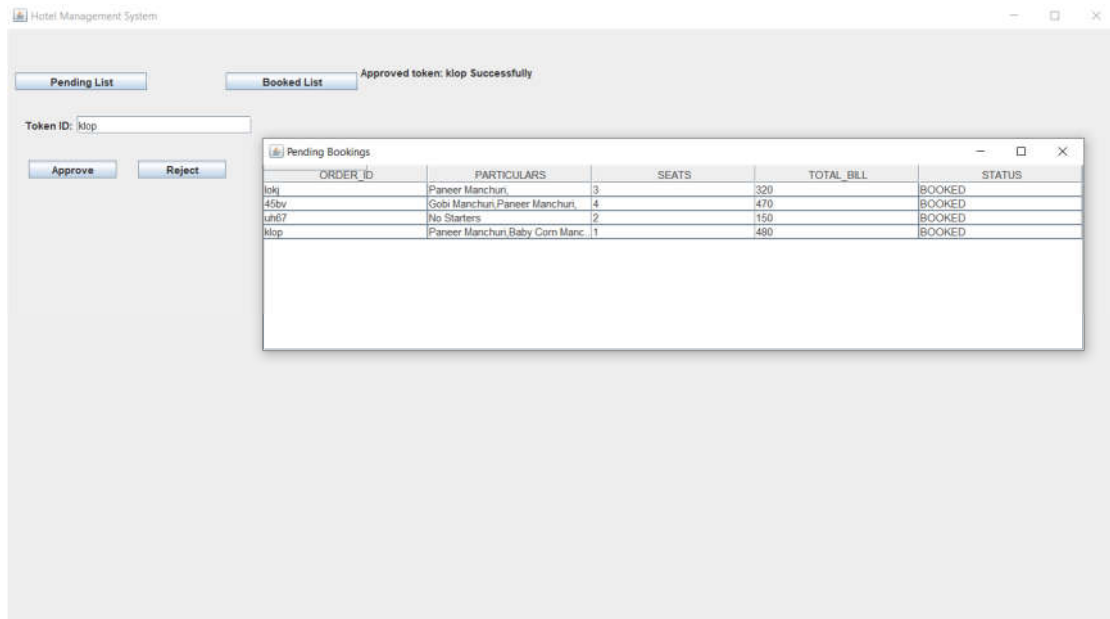
Demonstration of the program that runs on the administrator/manager to verify/reject bookings.

VIII. 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.

a) The Implementation stage consists of Making the necessary changes for the system as desired by the user.

b) Training the user personal prior to the implementation of two steps shown below as to be carried out.

c) Testing the developed program with the simple data.

d) 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.

We have used JAVA JDBC-ODBC connector api package to establish connection with the database as it provides rich functionalities and enables fetching and inserting data seamlessly to database.

In order to add user interface to the program we have use the swing package of java as it provides tools such as windows, labels, textboxes, buttons, checkboxes, radio buttons, and ability to draw table to display the fetched data from the database.

IX.   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:

    a) **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.

    b) **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.

Test Cases which have been used to fine tune the program:

1. User is able to access the Hotel Booking landing page
2. Validate the hotel booking landing page is rendered correctly for desktop as per the design specifications.
3. Validate the hotel booking landing page for admin is rendered correctly for Desktop as per the design specifications.
4. Validate hotel search fields are visible on screen.
5. User makes a successful payment for their table booking.
6. User makes an unsuccessful payment of their table booking.
7. Hotel table unavailability.
8. User cancels their booking and system refunds money – Test Refund Conditions
9. User cancels their booking and system does NOT refund money – Test Refund Conditions
10. User wants to make a group booking
11. Validate Booking Page displays correct booking data (eg that which was booked). – Visual check.
12. Confirm Payment Page is displayed when user selects "Make Payment".
13. End to End Test of Hotel Table Booking Engine.

Functional Tests

We have paid very good attention on validating the fields which are provided for the user to fill the information with various data and have fine-tuned them for accurate messages to be displayed if user makes any mistakes.

Some of the tests we carried out are:

1. Validate Location Field
2. Enter valid "From" Date in the date picker field
3. Enter invalid "From" Date in the date picker field – Eg from date is greater than to date.
4. Enter valid "To" Date in the date picker field.
5. Enter invalid "To" Date in the date picker field.
6. Enter valid data "Number of Tables" field.
7. Enter invalid "Number of Tables".
8. Enter valid "number of guests".
9. Enter invalid "number of guests".
10. Enter valid "number of adults".
11. Enter invalid "number of adults".
12. Enter valid "number of children".
13. Enter invalid "number of children".
14. Enter valid "number of infants".
15. Enter invalid data in "number of infants" field.
16. Enter valid "credit card details".
17. Enter invalid "credit card details".

Non-Functional Tests Which have been considered:

1. Does the landing page pass Google PageSpeed Insights? (Invalid as the program isn't web-based.)
2. Does landing page pass google web core vitals (Invalid as the program isn't web-based.)
3. Load Testing – The database must be able to handle [100] number of concurrent users. (Verified, it can handle)
4. The system must be able to handle [50] number of concurrent payment transactions. (Can handle as payment method is third party and not dependent on the performance of program's server)
5. Infrastructure Setup.
6. Load testing.

VALIDATION:

The process of evaluating software during the development process or at the end of the development process to determine whether it satisfies specified business requirements.

Validation Testing ensures that the product actually meets the client's needs. It can also be defined as to demonstrate that the product fulfills its intended use when deployed on appropriate environment.

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

Validation Testing - Workflow:

Validation testing can be best demonstrated using V-Model. The Software/product under test is evaluated during this type of testing.
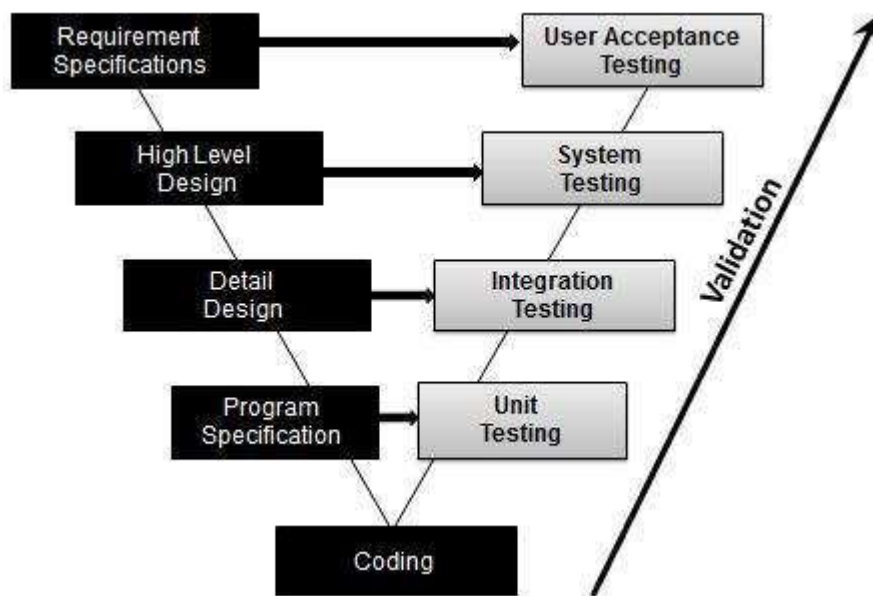
Activities in Validation Testing:

1. Unit Testing
2. Integration Testing
3. System Testing
4. User Acceptance Testing

➢ **Unit Testing**

What is Unit Testing?

Unit testing, a testing technique using which individual modules are tested to determine if there are any issues by the developer himself. It is concerned with functional correctness of the standalone modules.

Unit Testing Advantages:

a) Reduces Defects in the Newly developed features or reduces bugs when changing the existing functionality.
b) Reduces Cost of Testing as defects are captured in very early phase.
c) Improves design and allows better refactoring of code.
d) Unit Tests, when integrated with build gives the quality of the build as well.

Unit Testing Life-Cycle
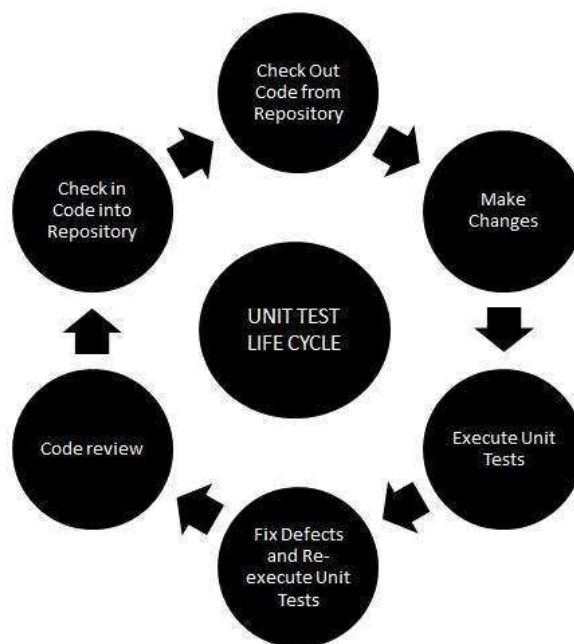
Unit testing Techniques

a) Black Box Testing - Using which the user interface, input and output are tested.
b) White Box Testing - used to test each one of those functions behaviour is tested.
c) Gray Box Testing - Used to execute tests, risks and assessment methods.

Black Box Testing:

Black-box testing is a method of software testing that examines the functionality of an application based on the specifications. It is also known as Specifications based testing. Independent Testing Team usually performs this type of testing during the software testing life cycle.

White Box Testing

White box testing is a testing technique, that examines the program structure and derives test data from the program logic/code. The other names of glass box testing are clear box testing, open box testing, logic driven testing or path driven testing or structural testing.

Gray Box Testing

Grey Box testing is testing technique performed with limited information about the internal functionality of the system. Grey Box testers have access to the detailed design documents along with information about requirements.

Grey Box tests are generated based on the state-based models, UML Diagrams or architecture diagrams of the target system.

➢ **Integration Testing**

Upon completion of unit testing, the units or modules are to be integrated which gives raise to integration testing. The purpose of integration testing is to verify the functional, performance, and reliability between the modules that are integrated.

There are several integration strategies:

a) Big-Bang Integration

b) Top-Down Integration

c) Bottom-Up Integration

d) Hybrid Integration

Big-Bang Integration

Big Bang Integration Testing is an integration testing strategy wherein all units are linked at once, resulting in a complete system. When this type of testing strategy is adopted, it is difficult to isolate any errors found, because attention is not paid to verifying the interfaces across individual units.

Big Bang Integration - WorkFlow Diagram
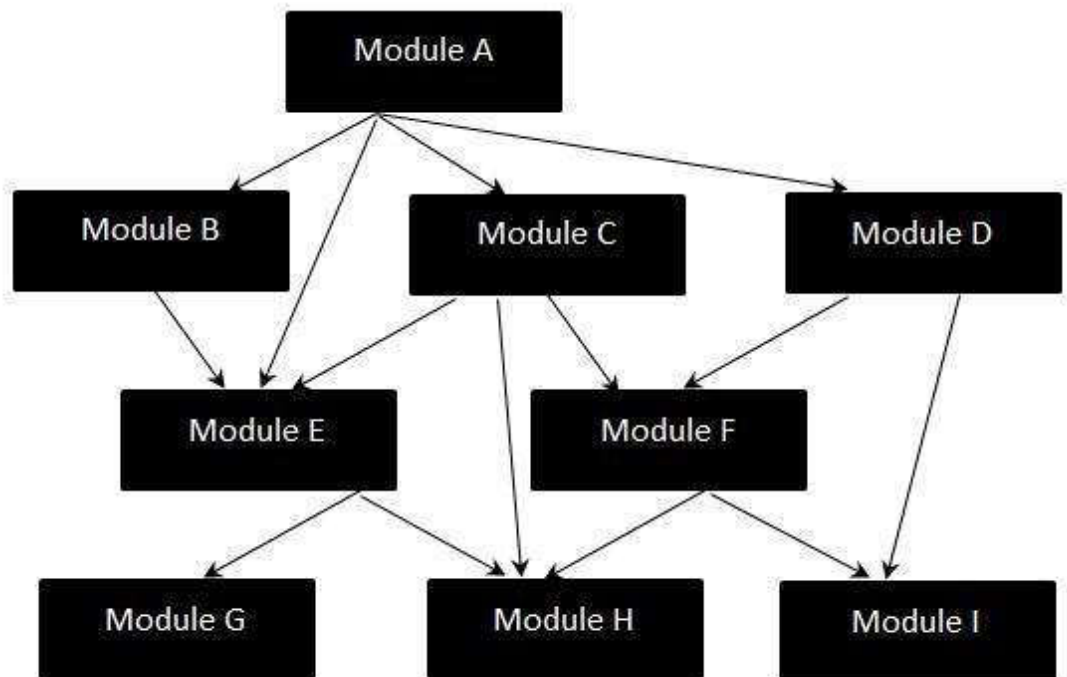
Disadvantages of Big-Bang Integration

a) It is very difficult to isolate the defects found.
b) There is high probability of missing some critical defects, which might pop up in the production environment.
c) It is very difficult to cover all the cases for integration testing without missing even a single scenario.
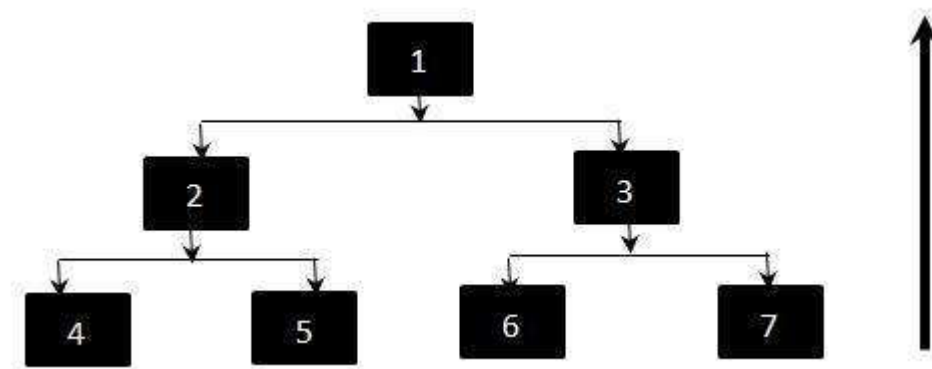
Top-Down Integration

Top-down integration testing is an integration testing technique used in order to simulate the behaviour of the lower-level modules that are not yet integrated. Stubs are the modules that act as temporary replacement for a called module and give the same output as that of the actual product.

Top-Down Integration Diagram
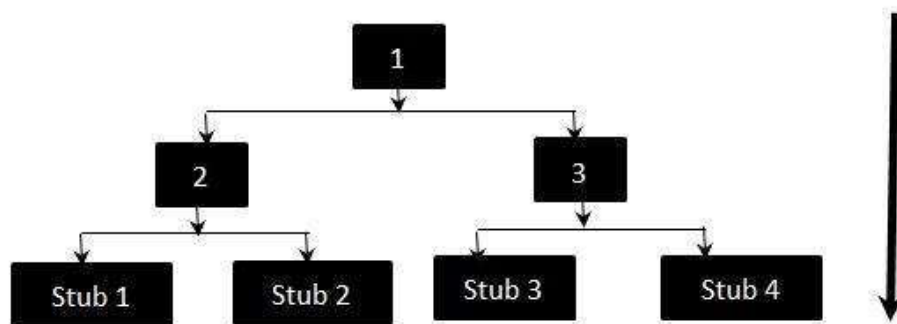
Bottom-Up Integration

Each component at lower hierarchy is tested individually and then the components that rely upon these components are tested.

Hybrid Integration

In Hybrid Integration Testing, we exploit the advantages of Top-down and Bottom-up approaches. As the name suggests, we make use of both the Integration techniques.

Features of Hybrid Integrations

a) It is viewed as three layers; viz - The Main Target Layer, a layer above the target layer and a layer below the target layer.
b) Testing is mainly focused for the middle level target layer and is selected on the basis of system characteristics and the structure of the code.
c) Hybrid Integration testing can be adopted if the customer wants to work on a working version of the application as soon as possible aimed at producing a basic working system in the earlier stages of the development cycle.

➢ **System Testing**

System Testing (ST) is a black box testing technique performed to evaluate the complete system the system's compliance against specified requirements. In System testing, the functionalities of the system are tested from an end-to-end perspective.

System Testing is usually carried out by a team that is independent of the development team in order to measure the quality of the system unbiased. It includes both functional and Non-Functional testing.

Types of System Tests
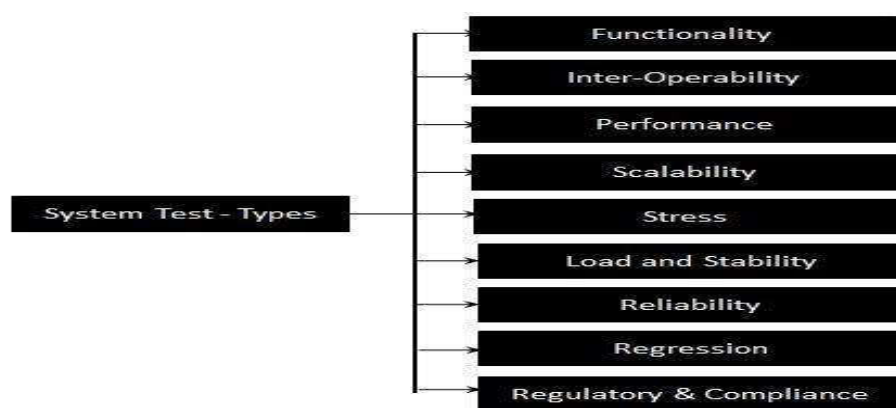
For this project we have used various testing procedures to make sure that all the modules imported/used function as they are intended to function. We distributed the Canary version of the program to early adopters and we got valuable feedbacks about correcting the UI to improve the User Experience. And made changes accordingly.

X.   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

In this era of high technology, everything is attaining more and more automation dependent. Hotel management system also should be involved in the realm of automation. This proposed intelligent management system provides high level privacy than the existing conventional manual system with greater reliability. To satisfy the customer's need, this project work provides a seamless and enjoyable experience for customers.

Introducing this automotive management system in any kind of reservation systems greatly reduced manpower and maintenance cost. This is our own concept and has been already successfully implemented as a project work in miniature version. In fact, this system is fast, comprehensive and flexible, but doesn't necessarily require ones to have that much skill in computer science.

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.

The project has a very vast scope in future. The project can be implemented on intranet in future. Project can be updated in near future as and when requirement for the same arises, as it is very flexible in terms of expansion.

With the proposed software of database Space Manager ready and fully functional the client is now able to manage and hence run the entire work in a much better, accurate and error free manner.

Bibliography:

[1]   Factors influencing online hotel booking intention: A conceptual framework from stimulus-organism-response perspective. A Emir, H Halim, A Hedre, D Abdullah Academic Research 2016.
[2]   Evaluation of Online Hotel Booking Services in O Khamidov.
[3]   Tourist behaviors in online booking: A new research agenda. H Peng, X Xu, W Chen - Communications in information, 2013
[4]   Choice behaviour in online hotel booking. L Masiero, JL Nicolau - Tourism Economics, 2016.
[5]   Factors influencing online hotel booking: Extending UTAUT2 with age, gender, and experience as moderators. CM Chang, LW Liu, HC Huang, HH Hsieh - Information, 2019.
[6]   Analyzing conversion rates in online hotel booking: The role of customer reviews, recommendations and rank order in search listings. A Cezar, H Ögüt - International Journal of Contemporary Hospitality, 2016.
[7]   Effects of haptic cues on consumers' online hotel booking decisions: the mediating role of mental imagery. X Lv, H Li, L Xia - Tourism management, 2020.

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