

HOTEL MANAGEMENT

SYSTEM

PROJECT REPORT ON
HOTEL MANAGEMENT SYSTEM
UNDER THE GUIDANCE OF
Ms. NAWAL DANDEKAR

SUBMITTED BY
VAIBHAV ANAND JADHAV
SEAT NO. TYCS16

UNIVERSITY OF MUMBAI
T.Y.B. Sc (COMPUTER SCIENCE)
ACADEMIC YEAR: 2020-2021

BHARTIYA VIDYA BHAVAN'S
MM COLLEGE OF ARTS, NM COLLEGE OF COMMERCE &
HRJ COLLEGE OF COMMERCE
BHAVAN'S COLLEGE MUNSHI NAGAR,
ANDHERI (WEST),
MUMBAI – 400058

Vaibhav Jadhav



BHARTIYA VIDYA BHAVAN'S
MM COLLEGE OF ARTS, NM COLLEGE OF COMMERCE &
HRJ COLLEGE OF COMMERCE
BHAVAN'S COLLEGE MUNSHI NAGAR,
ANDHERI (WEST),
MUMBAI – 400058

BHARATIYA VIDYA BHAVAN

BHAVAN'S COLLEGE, ANDHERI (W)

CERTIFICATE

This is to certify that **Mr. Vaibhav Anand Jadhav, Seat No. TYCS16** of the class **T. Y. B. Sc. Computer Science** has satisfactorily completed the practical course in **Group 5** as prescribed by the University of Mumbai during the academic year **2020 – 21**.

Signature
Staff in charge

Signature
Computer Science Coordinators

Signature
Examiner

College Stamp

Vaibhav Jadhav

ACKNOWLEDGEMENT

I would like to express my sincere gratitude towards the Computer Science Department of Bhavan's College.

After months of hard work, finally I am very happy to present my Sem 5 Project. The Project making was full of new experiences and learning and difficult one too. Though a difficult job it was made simpler by the timely guidance received, which helped me greatly in the completion of my project. But it wouldn't be right to do so without thanking to those who have helped me in converting our thought into reality. So, I would like to take full advantage of this opportunity to thank each and every person who has helped me throughout the completion of our project.

I am obliged to my parents & family members who always support me greatly and encouraged me in each and every step. I give my special thanks and sincere gratitude towards the Principal Dr. Zarine Bhathena and Head of Department (Computer Science) Prof. Mrs. Krupa Balsara-Kamdar.

I owe my sincere thanks to our Project guide Ms. Nawal Dandekar for her constant support and encouragement without which the successful completion of this project would have been impossible. They have been instrumental for making me concentrate and focus my effort in this project. And last but not the least the entire college staff specially Mr. Prathamesh N. Rane.

I would like to give thanks to my friends for helping me in my project. Finally, I would like to thank each and every individual who was directly or indirectly contributing for this project.

-Thank you

Index

Sr. No.	Topics	Page No.
1.	Preliminary Investigation	2
	1.1. Title of Project	2
	1.2. Description of System	2
	1.3. Advantages of proposed system	2
	1.4. Number of Modules	3
	1.5. Feasibility Study and Stakeholders	3
	1.6. Requirement Specification: software/hardware/data require	4
	1.7. Gantt Chart	5
2.	System Analysis	
	2.1. Fact Finding Techniques (Questionnaire, Sample Reports, Forms...)	7
	2.2. Event Table	7
	2.3. Use Case Diagram	9
	2.4. ERD	10
	2.5. Class diagram	11
	2.6. Object Diagram	12
	2.7. Activity Diagram	13
	2.8. Sequence diagram	14
	2.9. State Diagram	15
3.	System Design	
	3.1 Component Diagram	17
	3.2 Package Diagram	18
4.	System Coding	
	4.1 Menu Tree	20
	4.2 List of tables with attributes and constraints	20
	4.3 Program Description [Programs /Classes and their responsibilities in brief] with Naming Conventions.	23
	4.5 Validations	29
	4.6 Test Cases, Test Data and Test Results	32
	4.7 Screen Layouts & Report Layouts	34
5.	Future Enhancement	42
6.	Conclusion	43
7.	References, Books, web links, research articles, etc.	44

BHAVAN'S COLLEGE, ANDHERI (W)
DEPARTMENT OF COMPUTER SCIENCE

Project Implementation

PHASE COMPLETION CHART

CLASS : T.Y.B.Sc. COMPUTER SCIENCE

ROLL NO. : TYCS16

STUDENT NAME : VAIBHAV ANAND JADHAV

TITLE OF THE PROJECT: HOTEL MANAGEMENT SYSTEM

Phase Title	Expected Date of Completion	Actual Time of Completion with Guide's Signature	Remarks
I. Preliminary investigation			
1.1. Title of Project	5/8/20		
1.2. Description of System	5/8/20		
1.3. Advantages of proposed system	5/8/20		
1.4. Number of Modules	5/8/20		
1.5. Feasibility Study and Stakeholders	5/8/20		
1.6. Requirement Specification: software/hardware/data requirement	5/8/20		
1.7. Gantt Chart	6/10/20		
II. System Analysis			

Hotel Management System**TYCS16**

2.1. Fact Finding Techniques (Questionnaire, Sample Reports, Forms...)	12/08/2020		
2.2. Event Table	25/10/2020		
2.3. Use Case Diagram	12/08/2020		
2.4. ERD	12/08/2020		
2.5. Class diagram	12/08/2020		
2.6. Object Diagram	12/08/2020		
2.7. Activity Diagram	12/08/2020		
2.8. Sequence diagram	12/08/2020		
2.9 State Diagram	25/10/2020		
III. System Design			
3.1 Component Diagram	12/9/2020		
3.2 Package Diagram	12/9/2020		
IV. System Coding			
4.1 Menu Tree	10/10/2020		
4.2 List of tables with attributes and constraints	10/10/2020		
4.3 Program Description [Programs /Classes and their responsibilities in brief] with Naming Conventions.	15/10/2020		
4.4 Validations	15/10/2020		
4.5 Test Cases, Test Data and Test Results	15/10/2020		
4.6 Screen Layouts & Report Layouts	25/10/2020		
V. Project Report Submission	3/12/2020		

Note:

Mrs. Krupa Kamdar
Co-Ordinator
Computer Science

Vaibhav Jadhav

1.

PRELIMINARY

INVESTIGATION

1.1 Title of Project : Hotel Management System

1.2 Project description:

- Hotel management system is a software system where the management of the entire hotel is computerized.
- It is a desktop application that allows the hotel manager to handle all the hotel activities easily.
- Interactive GUI and the ability to manage various hotel rooms of customer, staff management attendance, records of customer and the room allotment makes system very flexible and convenient.
- The details like customer details staff details and availability and reservation of rooms are maintained and management is done without any difficulties.
- It will reduce the current cost of maintaining a manual system.
- The response time for looking up particulars will be greatly reduced.
- It also has Records of all the guest that are in hotel as well as those who checked out which will reduce time than searching records on paper

1.3 Advantages of proposed system:

- This system is effective and saves time and cost
- Can easily make reports
- Better management
- Huge reduction in number of pages used for data storage
- Speedy retrieval of data
- Compatible with advance versions for future

1.4 Number of modules in proposed system:

- **Login:** This module can be used for admin login and receptionist login. If it is valid user can log in, otherwise invalid username and password message is displayed.

- **Room reservation:** This module deals with reservation enquiry and reservation. It will search for the unoccupied rooms and will display as result. If the Room is not available, then that Room cannot be booked, it will display Room not available message
- **Staff:** This module will contains information regarding all the staff, their designation and the salary and information can be updated. Staff can be removed as well from this module.
- **Records:** This module deals with the records of all the customers and room allotted to them. It is useful for future queries
- **Staff attendance:** This module deals with the regular attendance of all the staff
- **Customer:** This module will have all the information regarding customer

1.5 Feasibility study:

A feasibility study is designed to provide an overview of primary issue related to the project idea. It is concerned with expected benefits.

The types of Feasibility study of this system are: -

➤ **Organizational Feasibility**

This project provides considerable assistance to the manager to manage all activities of the hotel. It provides more accuracy. The project will be user friendly and will understand the working and functioning.

➤ **Technical Feasibility**

It deals with all the technical aspects relates to the project. It is evident that necessary hardware and software are available for implementation of proposed system. The programming language used in project is C# and SQL server

➤ **Operational Feasibility**

It deals with human organization aspects. The project has to handle all the staff details room allotment details customer details and all the records.

➤ **Economic Feasibility**

The cost of hardware and software requirements is feasible as it won't cost much. Cost and benefits with proposed system is economically feasible

1.5 Stakeholder's:

Stakeholder's are the people who have an interest in the successful implementation of system.

We categorize them into following groups.

➤ **Admin:**

It includes those users who are assigned as Admin in the system. This particular Application granting rights to admin to have full control over using of Application

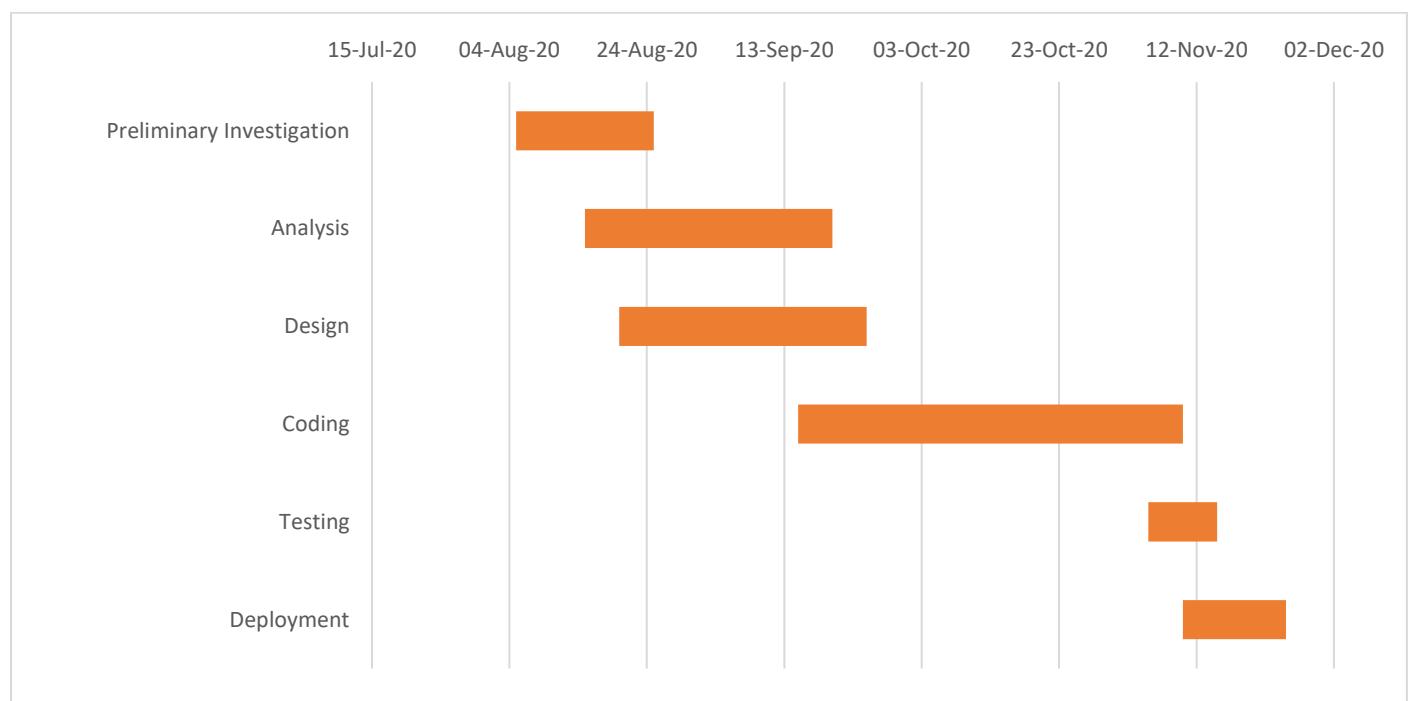
➤ **Receptionist:**

User of those who actually use the system on frequent basis and those are the end users of the system. These are the people who need Current information from the system and uses the system for doing their assigned tasks in particular environment

1.6 Technologies used:

- **Operating System:** Windows xp and above
- **Software:** Visual Studio 2019
- **Languages:** C#
- **Database System:** SQL Server
- **Documentation Tool:** MS-Word 2019, MS-Excel 2019, Smart Draw, Visual Paradigm 14.0

1.7 Gantt Chart



2.

SYSTEM ANALYSIS

2.1 FACT FINDING TECHNIQUE

Information and through knowledge as well as on job training plays a very important role in any organization. Acquiring valuable information is essential for developing a new system in an organization. There are many ways of gathering information. But the following things need to be kept in mind while gathering information.

- No system details are left out.
- Problems are correctly identified.
- Repetitive work is avoided.

2.2 EVENT TABLE

➤ **TRIGGER:**

Trigger an occurrence that tells the system that has occurred, either the arrival of data needing or of a point in time

➤ **SOURCE:**

An external agent or actor that supplies data to the system

➤ **ACTIVITY:**

Behavior that the system performs when an event occurs

➤ **RESPONSE:**

An output produced by the system that goes to a destination.

➤ **DESTINATION:**

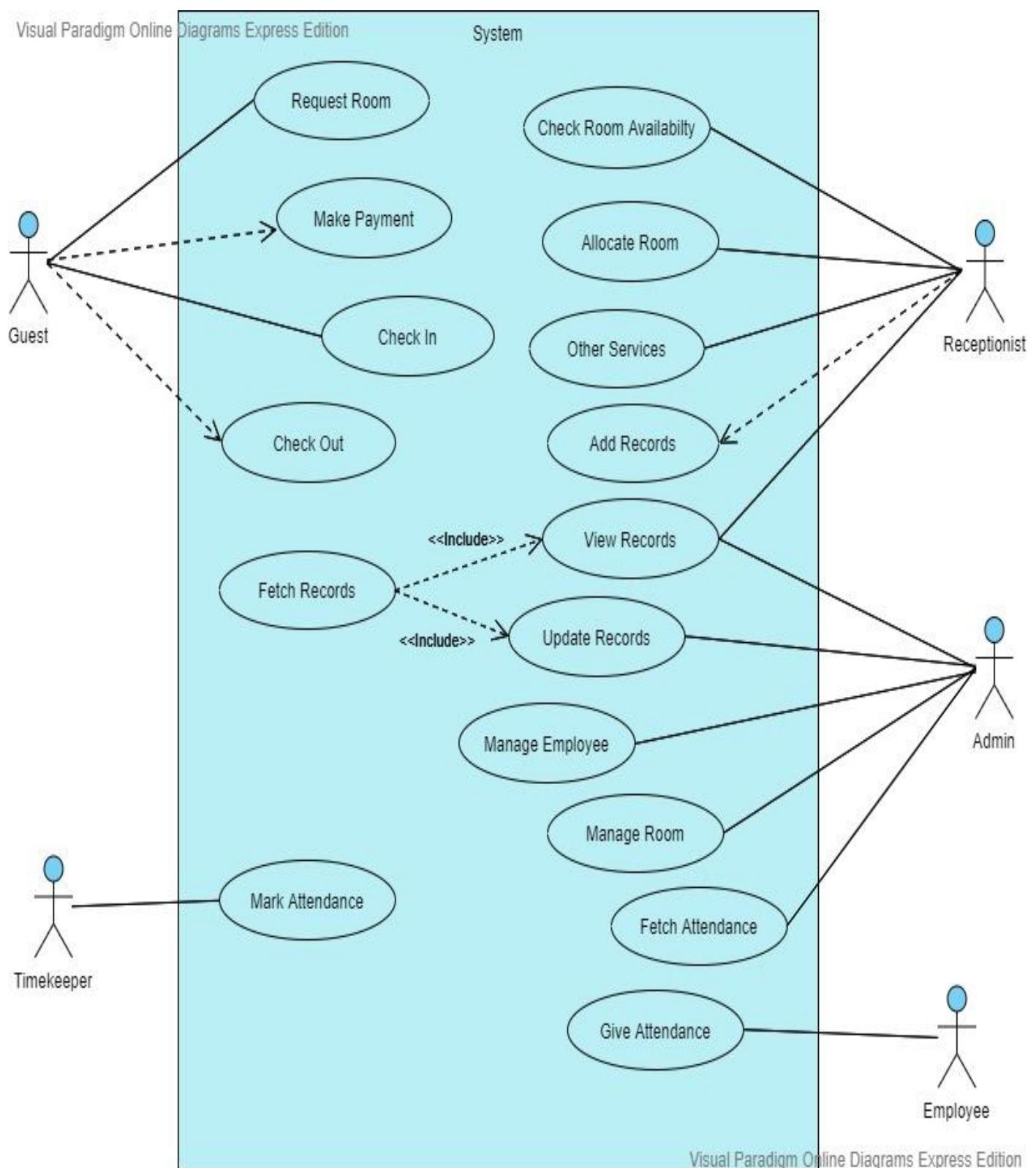
An external agent or actor that receives data from the system

Use Case Name	Log-in/Sign-in
Summary/Description	Use case depicting how an Employee or Customer would login.
Actors	Receptionist and Admin
Pre-Conditions	Log-in ID must be present in Login table for both Receptionist and Admin
Descriptions	Receptionist and Admin will have to login to land on their respective form windows. From here they can perform their respective actions.
Exception	No ID present in the Login table, which will give them an error
Post-Condition	Receptionist and Admin successfully Login.

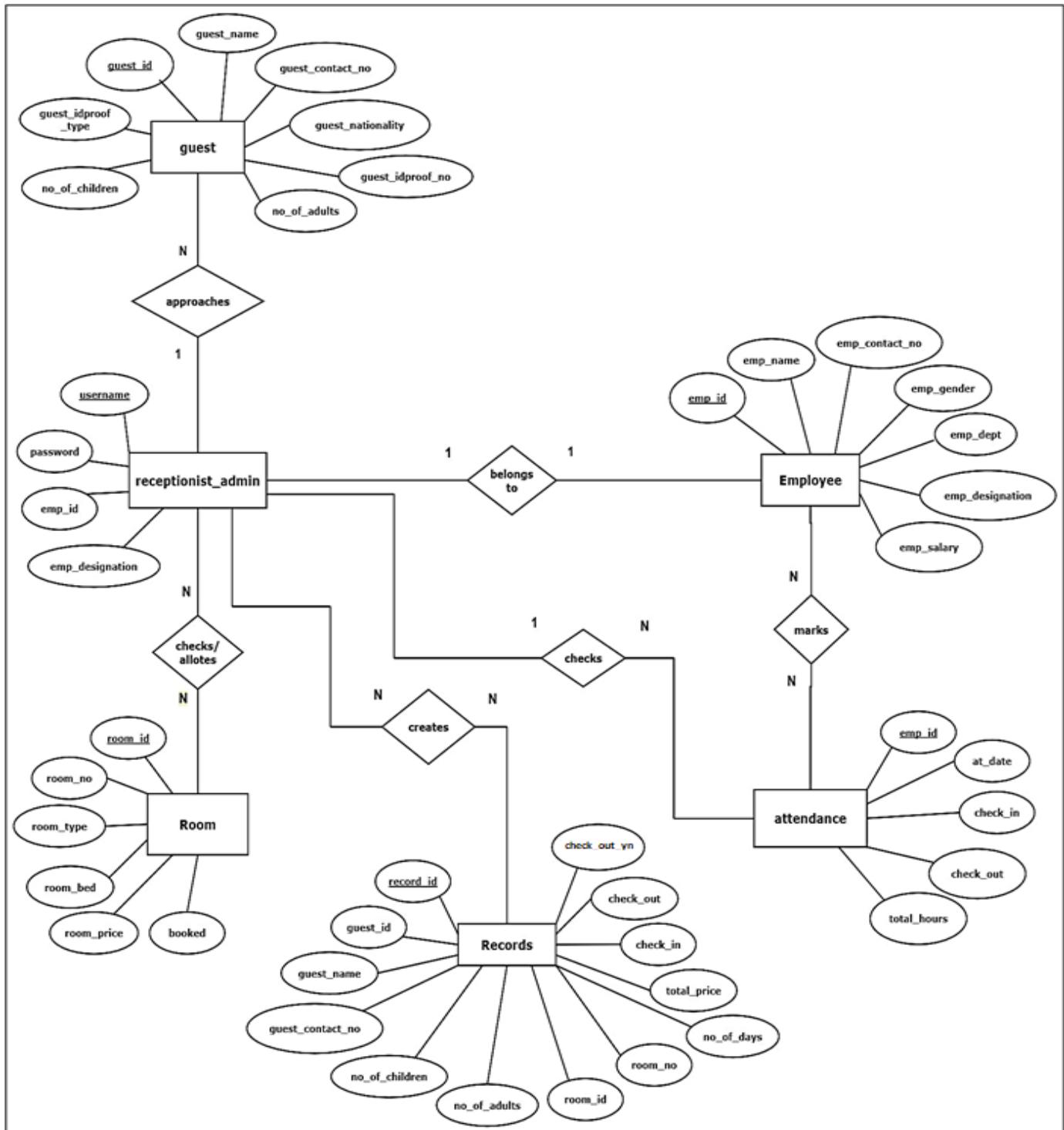
Use Case Name	Receptionist Use Case
Summary/Description	Use case depicting how an employee will log-in and then perform his respective tasks from the desktop application
Actors	Receptionist
Pre-Conditions	Receptionist Login ID must be present in the Login table enabling Receptionist to login
Descriptions	Once the Receptionist has logged in to the system, he can perform various tasks such as, Adding guest ,View Records, Allot rooms, update order progress etc.
Exception	Receptionist ID is not present
Post-Condition	Receptionist is able to successfully log-in and perform all the actions he is authorized to.

Use Case Name	Admin Use Case
Summary/Description	Use case depicting how a system admin will be able to log-in and perform his respective tasks in the web application
Actors	Admin
Pre-Conditions	Person trying to log-in must have administrative privileges.
Descriptions	Admin holds all the rights the system has to offer. He/she can perform all the tasks of the employee, add new employees, review records of all the guest, add new rooms to the system, and view attendance etc
Exception	Person trying to log-in does not have administrator privileges.
Post-Condition	Admin is successfully able to log-in and perform tasks.

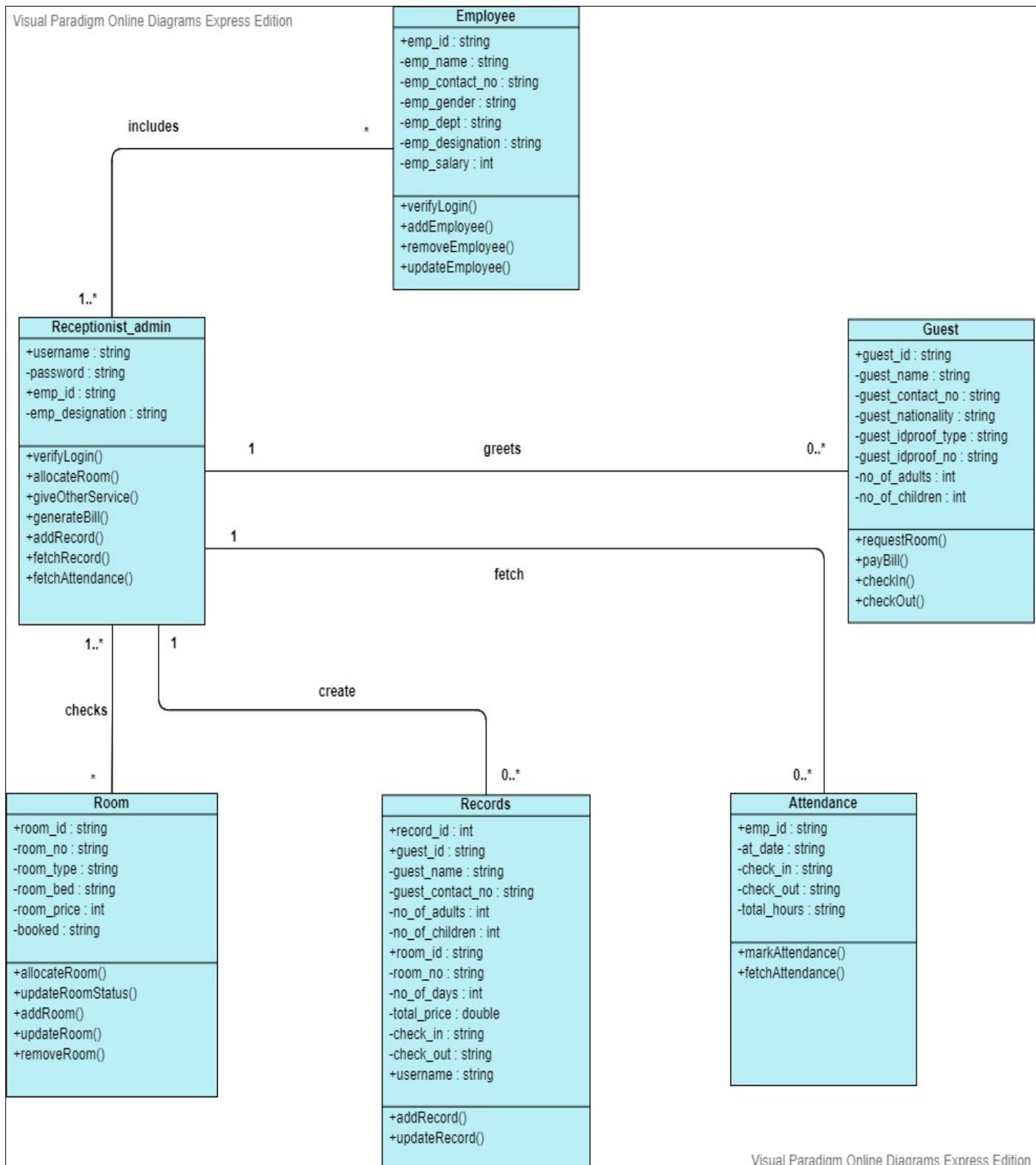
2.3 USE CASE DIAGRAMS



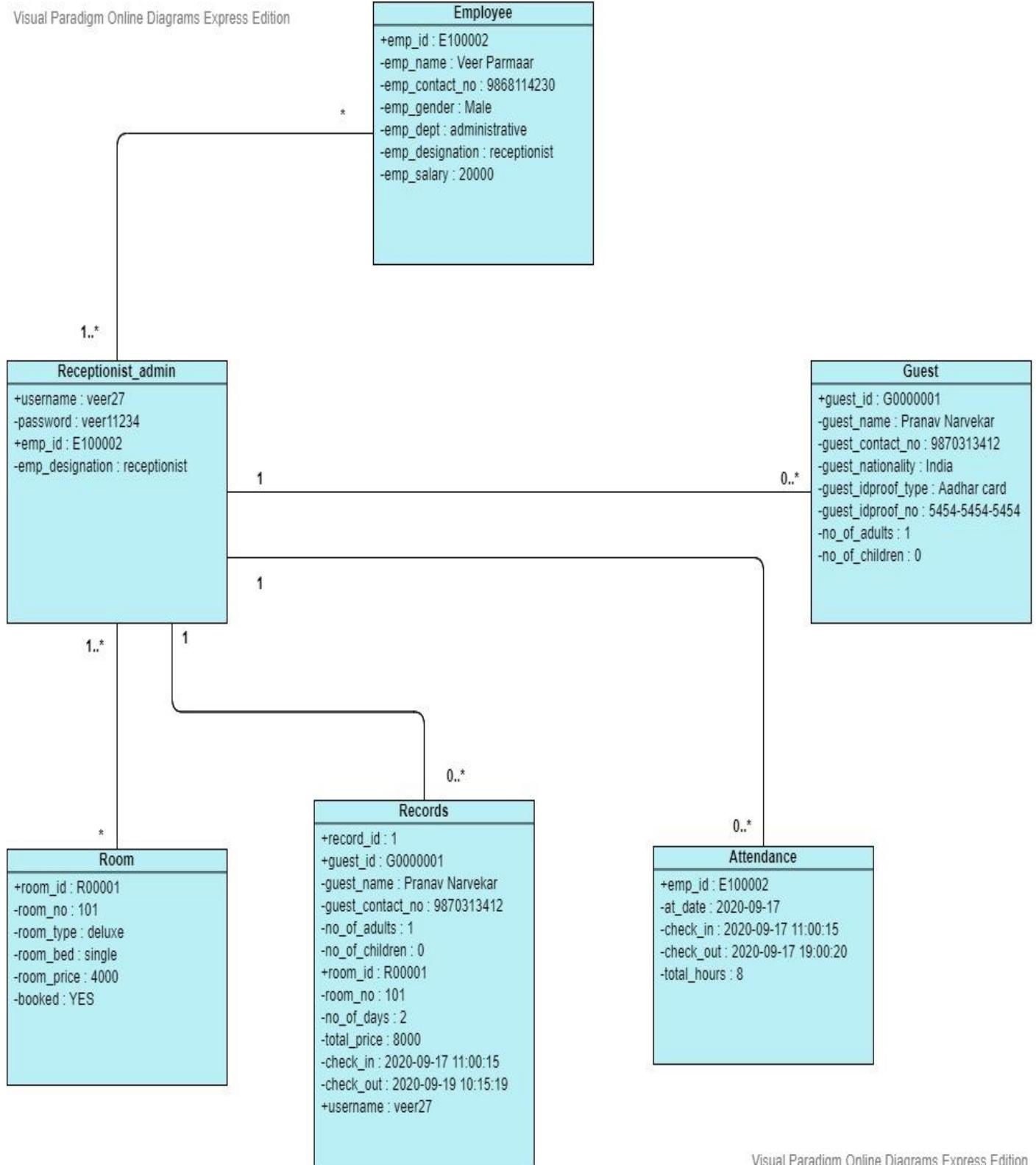
2.4 ENTITY RELATIONSHIP DIAGRAM



2.5 CLASS DIAGRAM

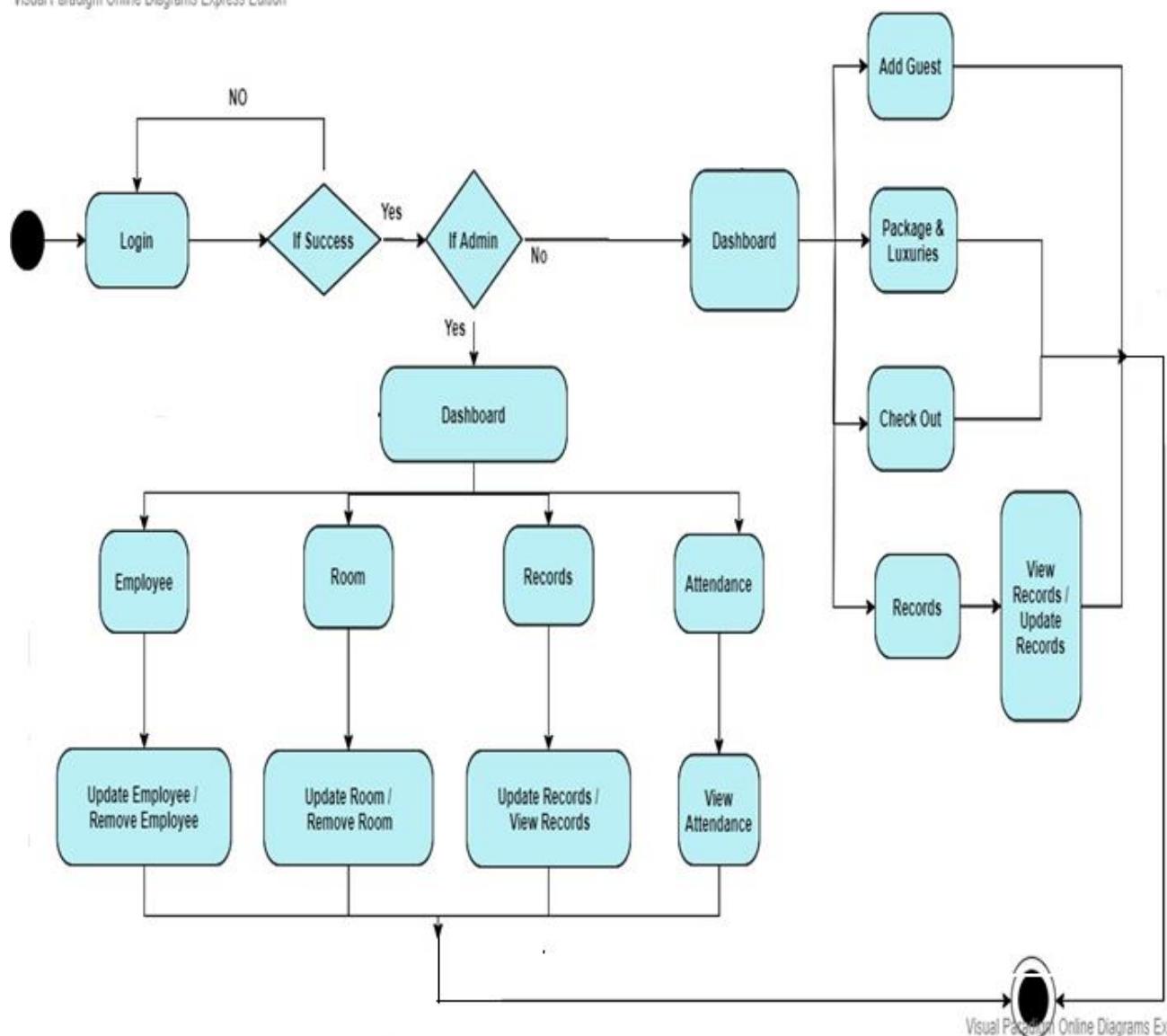


2.6 OBJECT DIAGRAM



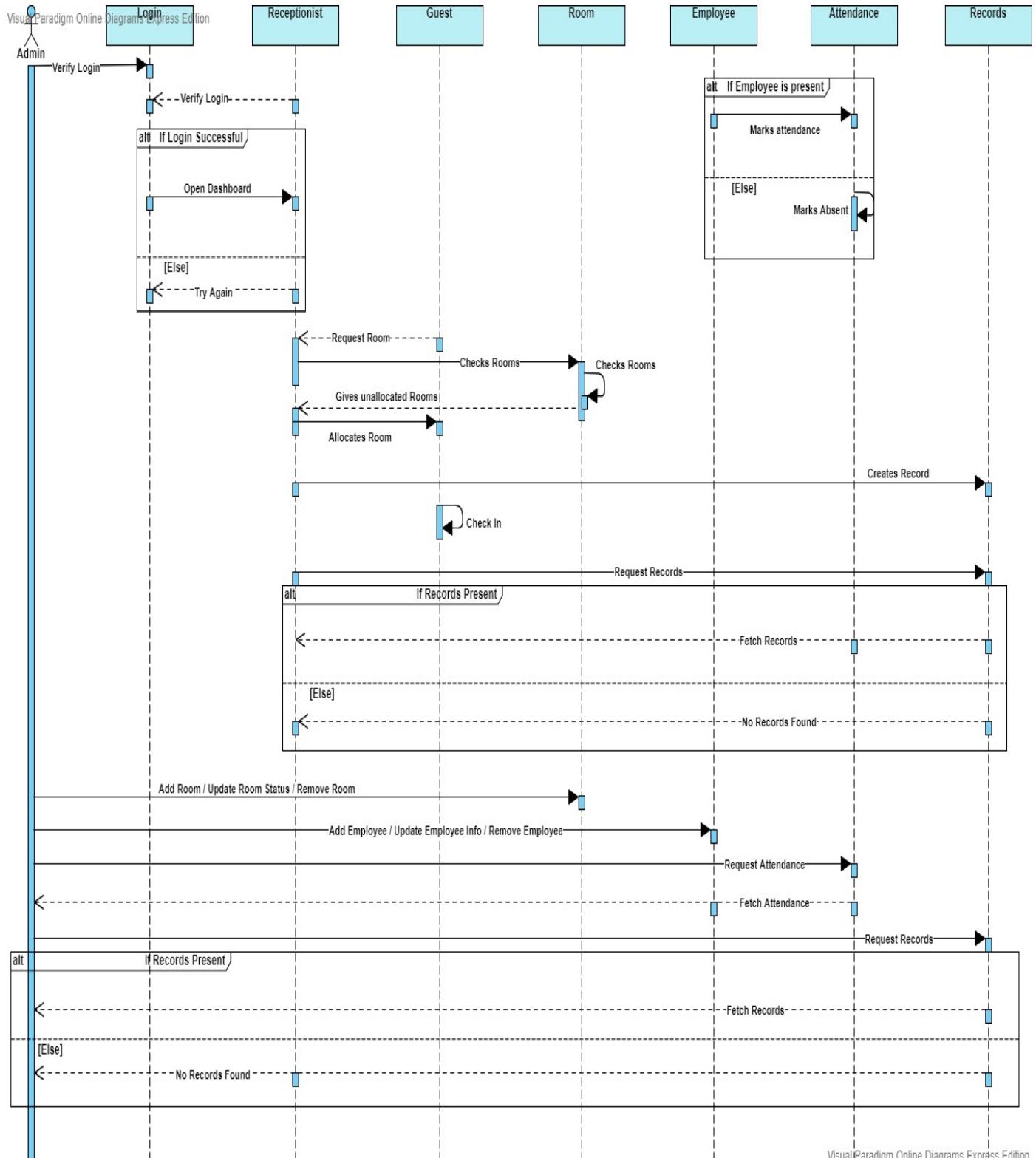
2.7 ACTIVITY DIAGRAM

Visual Paradigm Online Diagrams Express Edition

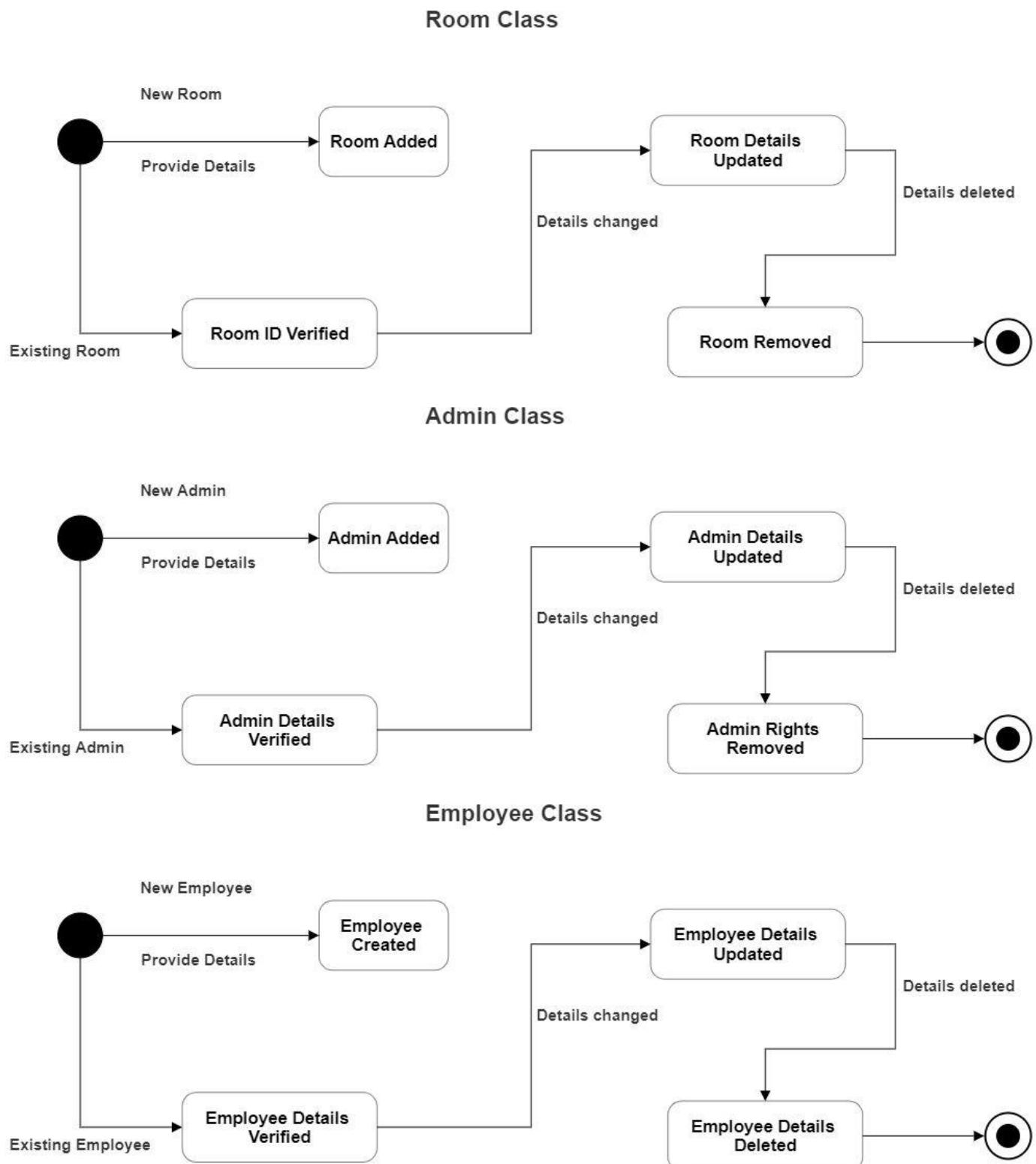


Visual Paradigm Online Diagrams Express Edition

2.8 SEQUENCE DIAGRAM

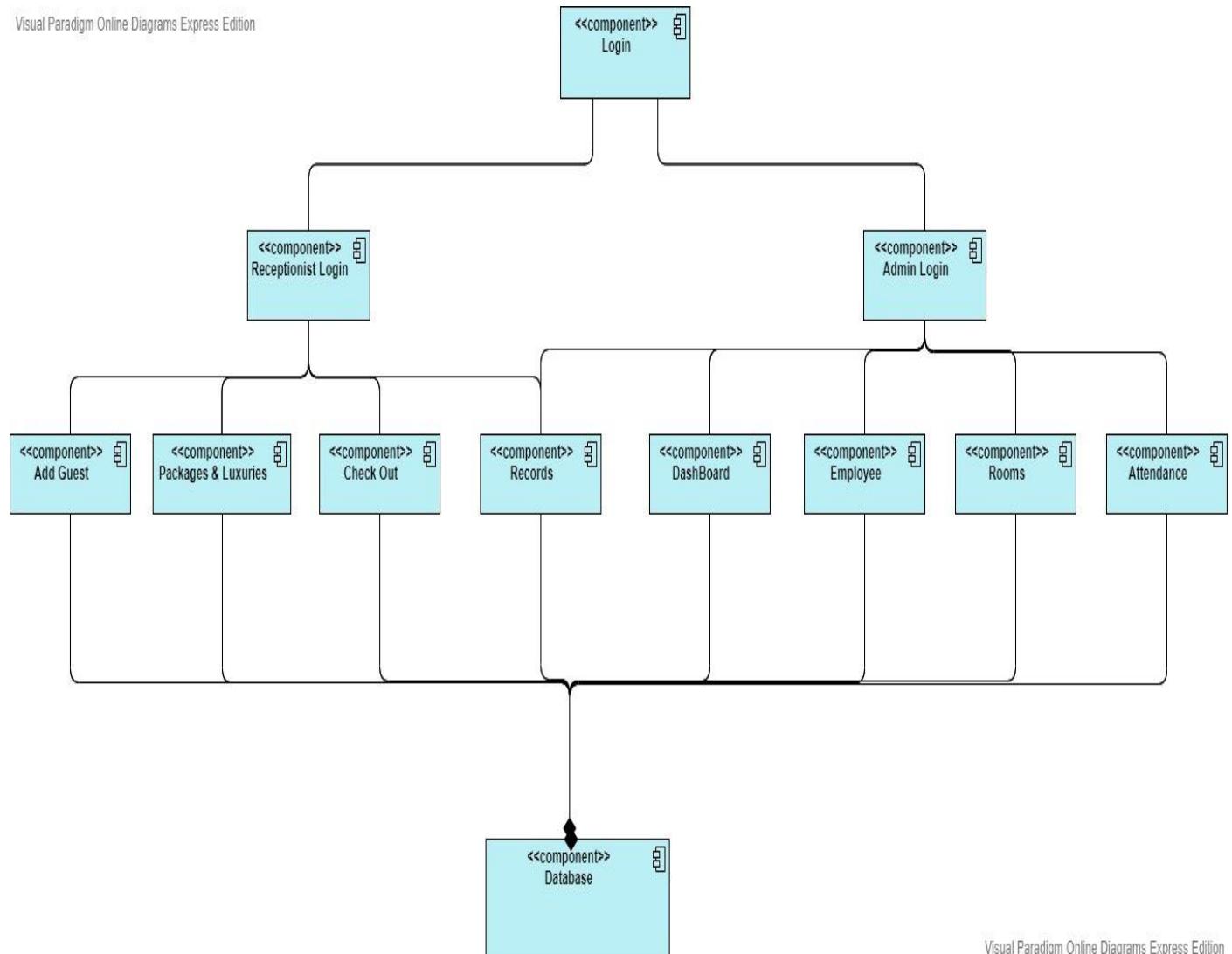


2.9 STATE DIAGRAM

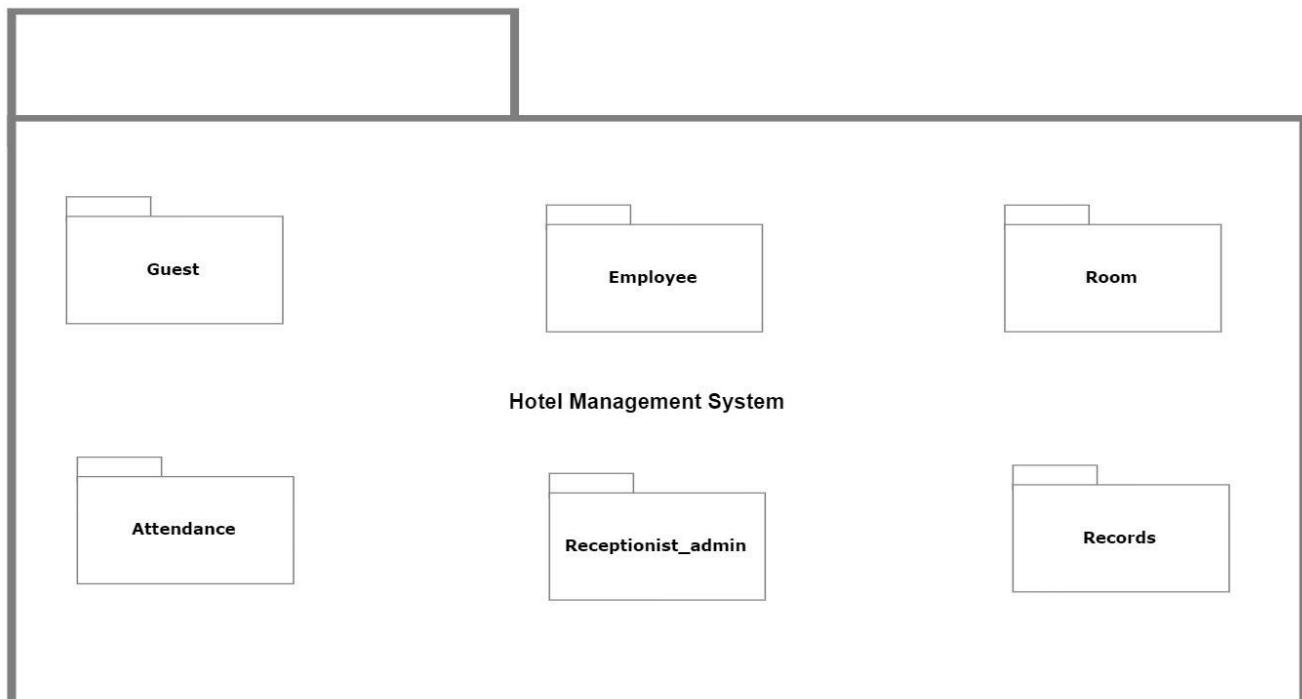


3. SYSTEM DESIGN

3.1 COMPONENT DIAGRAM



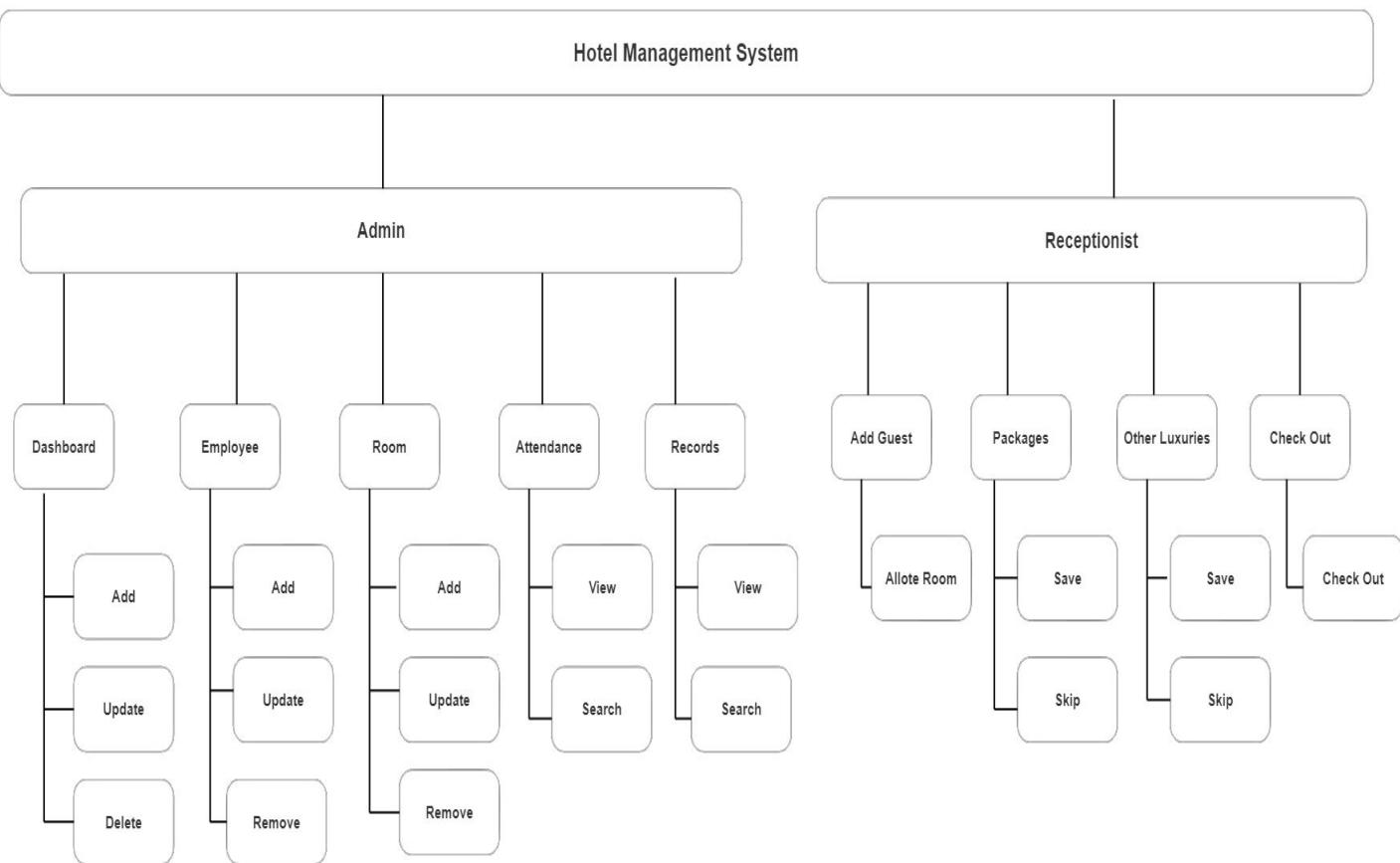
3.2 PACKAGE DIAGRAM



4.

SYSTEM CODING

4.1 MENU TREE



4.2 LIST OF TABLES WITH ATTRIBUTES AND CONSTRAINTS

Table Name: Receptionist_admin (references to Employee)

Sr no.	Attribute name	Type	Constraints	Description
1	username (pk)	varchar(20)	Primary Key	Username to login
2	password	varchar(20)	Not Null	Password to login
3	emp_id (fk)	varchar(15)	Foreign Key	ID of Employee
4	emp_designation	varchar(30)	Not Null	Designation of Employee

Table Name: Employee

Sr no.	Attribute name	Type	Constraints	Description
1	emp_id (pk)	varchar(15)	Primary Key	ID of Employee
2	emp_name	varchar(50)	Not Null	Name of Employee
3	emp_contact_no	varchar(13)	Not Null	Contact number of Employee
4	emp_gender	varchar(10)	Not Null	Gender of Employee
5	emp_dept	varchar(30)	Not Null	Department of Employee
6	emp_designation	varchar(30)	Not Null	Designation of Employee
7	emp_salary	int	Not Null	Salary of Employee

Table Name: Room

Sr no.	Attribute name	Type	Constraints	Description
1	room_id (pk)	varchar(15)	Primary Key	ID of Room
2	room_no	varchar(10)	Not Null	Room number
3	room_type	varchar(20)	Not Null	Type of Room
4	room_bed	varchar(20)	Not Null	Bed available in Room
5	room_price	int	Not Null	Price of Room
6	booked	varchar(5)	Not Null	Is Room Booked

Table Name: Guest

Sr no.	Attribute name	Type	Constraints	Description
1	guest_id (pk)	varchar(15)	Primary Key	ID of Guest
2	guest_name	varchar(50)	Not Null	Name of Guest
3	guest_contact_no	varchar(13)	Not Null	Contact number of Guest
4	guest_nationality	varchar(25)	Not Null	Nationality of Guest
5	guest_idproof_type	varchar(25)	Not Null	Type of ID Proof

6	guest_idproof_no	varchar(40)	Not Null	ID Proof number of Guest
7	no_of_adults	int	Not Null	Number of adults
8	no_of_children	int	Not Null	Number of Children

Table Name: Attendance (references to employee)

Sr no.	Attribute name	Type	Constraints	Description
1	emp_id (fk)	varchar(15)	Foreign Key	ID of Employee
2	at_date	date	Not Null	Present Date
3	check_in	datetime	Not Null	Time of check in
4	check_out	datetime	Null	Time of check out
5	total_hours	varchar(20)	Null	Total hours

Table Name: Records (references to room, guest)

Sr no.	Attribute name	Type	Constraints	Description
1	record_id (pk)	int	Primary Key	ID of Records
2	guest_id (fk)	varchar(15)	Foreign Key	ID of Guest
3	guest_name	varchar(50)	Not Null	Name of Guest
4	guest_contact_no	varchar(13)	Not Null	Contact number of Guest
5	no_of_adults	int	Not Null	Number of adults
6	no_of_children	int	Not Null	Number of Children
7	room_id (fk)	varchar(15)	Not Null	ID of Room
8	room_no	varchar(10)	Not Null	Room number
9	no_of_days	int	Not Null	Number of days Room booked
10	total_price	double	Not Null	Total Price including room and other luxuries
11	check_in	datetime	Not Null	Time of check in
12	check_out	datetime	Null	Time of check out

13	check_out_yn	varchar(15)	Not Null	Does Guest check out or not
----	--------------	-------------	----------	-----------------------------

4.3 Program Description with Naming Conventions.

Login Code:

```

username = txtusername.Text;
SqlConnection con = new SqlConnection();
con.ConnectionString = "Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=D:\\Projects\\Hotel
management System\\Hotel Management Software\\Database1.mdf;Integrated Security=True";
SqlCommand cmd = new SqlCommand();
cmd.Connection = con;
String query = "select emp_designation from Receptionist_admin where username='" + txtusername.Text + "'"
and password=''" + txtpassword.Text + "'";
con.Open();
cmd.CommandText = query;

using (SqlDataReader rd = cmd.ExecuteReader())
{
    if (rd.HasRows)
    {
        while(rd.Read())
        {
            String designation = rd.GetString(0);
            if(designation=="admin")
            {
                Amin_login ad = new Amin_login();
                this.Hide();
                ad.Show();
            }
            if(designation=="receptionist")
            {
                Receptionist_login rl = new Receptionist_login();
                this.Hide();
                rl.Show();
            }
            if(designation=="timekeeper")
            {
                Timekeeper_login tl = new Timekeeper_login();
                this.Hide();
                tl.Show();
            }
        }
    }
    else
    {
        MessageBox.Show("Login Unsuccessful!!! \nPlease try again");
        txtusername.Text = "";
        txtpassword.Text = "";
    }
}

```

function.cs file

class function

```
{  
    public SqlConnection getconnection()  
    {  
        SqlConnection con = new SqlConnection();  
        con.ConnectionString = "Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=D:\\Projects\\Hotel  
management System\\Hotel Management Software\\Database1.mdf;Integrated Security=True";  
        return con;  
    }  
  
    public DataSet getData(String query)  
    {  
        SqlConnection con = getconnection();  
        SqlCommand cmd = new SqlCommand();  
        cmd.Connection = con;  
        cmd.CommandText = query;  
        SqlDataAdapter adp = new SqlDataAdapter(cmd);  
        DataSet ds = new DataSet();  
        adp.Fill(ds);  
        return ds;  
    }  
  
    public int setData(String query)  
    {  
        SqlConnection con = getconnection();  
        SqlCommand cmd = new SqlCommand();  
        cmd.Connection = con;  
        con.Open();  
        cmd.CommandText = query;  
        int i=cmd.ExecuteNonQuery();  
        con.Close();  
        return i;  
    }  
  
    public SqlDataReader getForCombo(String query)  
    {  
        SqlConnection con = getconnection();  
        SqlCommand cmd = new SqlCommand();  
        con.Open();  
        cmd.Connection = con;  
        cmd = new SqlCommand(query, con);  
        SqlDataReader sdr = cmd.ExecuteReader();  
        return sdr;  
    }  
}
```

Add Code: -

```
    id = txtempid.Text;  
    name = txtempname.Text;  
    cnum = txtcontactno.Text;  
    desig = txtdesignation.Text;  
    String sal1 = txtsalary.Text;  
    if (id == "" && name == "" && cnum == "" && desig == "" && sal1 == "" &&  
        cmbgender.SelectedIndex.Equals(-1) && cmbdepartment.SelectedIndex.Equals(-1))  
    {
```

```
        MessageBox.Show("Please Fill all the textboxes");
    }
    else
    {
        sal = Int32.Parse(txtsalary.Text);
        gen = cmbgender.SelectedItem.ToString();
        dept = cmbdepartment.SelectedItem.ToString();
        SqlConnection con1 = new SqlConnection();           //checking the employee
        con1.ConnectionString = "Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=D:\Projects\Hotel management
System\Hotel Management Software\Database1.mdf;Integrated Security=True";
        SqlCommand cmd1 = new SqlCommand();
        cmd1.Connection = con1;
        String query2 = "select * from Employee where emp_id=" + id + " and emp_name=" + name + ";";
        con1.Open();
        cmd1.CommandText = query2;
        using (SqlDataReader rd1 = cmd1.ExecuteReader())
        {
            if (rd1.HasRows)
            {
                MessageBox.Show("Employee ID already exist !! \nPlease Click on Update");
                return;
            }
            else
            {
                query = "insert into Employee values(" + id + "," + name + "," + cnum + "," + gen
                + "," + dept + "," + desig + "," + sal + ")";
                int i= fn.setData(query);
                if(i!=0)
                {
                    ad_employee_Load(this, null);
                }
                else
                {
                    MessageBox.Show("Record Insertion Failed!!!");
                }
            }
        }
    }
    clearAll();
}
```

Update Code: -

```
id = txttempid.Text;
name = txttempname.Text;
cnum = txtcontactno.Text;
desig = txtdesignation.Text;
String sal1 = txtsalary.Text;
if(id == "" && name == "" && cnum == "" && desig == "" && sal1 == "" &&
cmbgender.SelectedIndex.Equals(-1) && cmbdepartment.SelectedIndex.Equals(-1))
{
    MessageBox.Show("Please Fill all the textboxes");
}
else
{
    sal = Int32.Parse(txtsalary.Text);
    gen = cmbgender.SelectedItem.ToString();
    dept = cmbdepartment.SelectedItem.ToString();
```

```
SqlConnection con1 = new SqlConnection();      //checking the employee
con1.ConnectionString = "Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=D:\Projects\Hotel management
System\Hotel Management Software\Database1.mdf;Integrated Security=True";
SqlCommand cmd1 = new SqlCommand();
cmd1.Connection = con1;
String query2 = "select * from Employee where emp_id=" + id + " and emp_name=" + name + ";";
con1.Open();
cmd1.CommandText = query2;
using (SqlDataReader rd1 = cmd1.ExecuteReader())
{
    if (rd1.HasRows)
    {
        query = "UPDATE Employee SET emp_contact_no=" + cnum + ",emp_dept=" +
dept + ",emp_designation=" + desig + ",emp_salary=" + sal + " WHERE emp_id=" + id + "
and emp_name=" + name + ";";
        int i = fn.setData(query);
        if (i != 0)
        {
            ad_employee_Load(this, null);
        }
        else
        {
            MessageBox.Show("Record Updation Failed!!!");
        }
    }
    else
    {
        MessageBox.Show("Employee ID does not exist !! \nPlease Click on Add");
        return;
    }
}
clearAll();
```

Delete Code: -

```
id = txttempid.Text;
name = txttempname.Text;
cnum = txtcontactno.Text;
desig = txtdesignation.Text;
String sal1 = txtsalary.Text;
if (id == "" && name == "" && cnum == "" && desig == "" && sal1 == "" &&
cmbgender.SelectedIndex.Equals(-1) && cmbdepartment.SelectedIndex.Equals(-1))
{
    MessageBox.Show("Please Fill all the textboxes");
}
else
{
    SqlConnection con1 = new SqlConnection();      //checking the employee
    con1.ConnectionString = "Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=D:\Projects\Hotel management
System\Hotel Management Software\Database1.mdf;Integrated Security=True";
    SqlCommand cmd1 = new SqlCommand();
    cmd1.Connection = con1;
    String query2 = "select * from Employee where emp_id=" + id + " and emp_name=" + name + ";";
    con1.Open();
    cmd1.CommandText = query2;
```

```
using (SqlDataReader rd1 = cmd1.ExecuteReader())
{
    if (rd1.HasRows)
    {
        query = "delete from Employee where emp_id=" + id + " and emp_name=" + name
        + ";";
        int i = fn.setData(query);
        if (i != 0)
        {
            ad_employee_Load(this, null);
        }
        else
        {
            MessageBox.Show("Record Deletion Failed!!!");
        }
    }
    else
    {
        MessageBox.Show("Employee ID does not exist !! \nPlease Click on Add");
        return;
    }
}
clearAll();
```

Search Code: -

```
String search = txtSearch.Text;
String query2 = "select * from Employee where emp_name like '" + search + "%'";
DataSet ds = fn.getData(query2);
dataGridView1.DataSource = ds.Tables[0];
```

clearAll code: -

```
txtempid.Text = "";
txtempname.Text = "";
txtSearch.Text = "";
txtsalary.Text = "";
txtdesignation.Text = "";
cmbgender.SelectedIndex = -1;
cmbdepartment.SelectedIndex = -1;
```

Attendance Code: -

```
query="select * from attendance where at_date BETWEEN CONVERT(date,'" + dpfrom.Value.ToString("dd-MM-yyyy") + "',105) AND CONVERT(date,'" + dpto.Value.ToString("dd-MM-yyyy") + "',105);";
DataSet ds = fn.getData(query);
dataGridView1.DataSource = ds.Tables[0];
```

Check-in code: -

```
id = txtempid.Text;
SqlConnection con = new SqlConnection();
con.ConnectionString = "Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=D:\Projects\Hotel management System\Hotel Management Software\Database1.mdf;Integrated Security=True";
SqlCommand cmd = new SqlCommand();
cmd.Connection = con;
String query1 = "select * from Employee where emp_id ='" + id + "'";
con.Open();
cmd.CommandText = query1;
using (SqlDataReader rd = cmd.ExecuteReader())
```

```
{  
    if (rd.HasRows)  
    {  
        con.Close();  
        String date1 = DateTime.Now.ToString("MM-dd-yyyy");  
        String checkin = DateTime.Now.ToString("MM-dd-yyyy HH:mm:ss");  
        SqlConnection con1 = new SqlConnection();  
        con1.ConnectionString = "Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=D:\\Projects\\Hotel  
management System\\Hotel Management Software\\Database1.mdf;Integrated Security=True";  
        SqlCommand cmd1 = new SqlCommand();  
        cmd1.Connection = con1;  
        String query2 = "select * from attendance where emp_id ='" + id + "' and at_date= '"+date1+"';";  
        con1.Open();  
        cmd1.CommandText = query2;  
        using(SqlDataReader rd1=cmd1.ExecuteReader())  
        {  
            if(rd1.HasRows)  
            {  
                MessageBox.Show("Employee's check in already present!!");  
                con1.Close();  
            }  
        }  
    }  
    else  
    {  
        con1.Close();  
        query = "insert into attendance values('" + id + "','" + date1 + "','" + checkin + "','-','-');";  
        int i = fn.setData(query);  
        if (i != 0)  
        {  
            //MessageBox.Show("Records insertion successful!!");  
            Timekeeper_login_Load(this, null);  
        }  
        else  
        {  
            MessageBox.Show("Records insertion failed!!");  
        }  
    }  
}  
else  
{  
    MessageBox.Show("Employee ID does not exist!!");  
    con.Close();  
}
```

Checkout code: -

```
String checkout = DateTime.Now.ToString("MM-dd-yyyy HH:mm:ss");  
DateTime chout = DateTime.Now;  
id = txtempid.Text;  
if(id !="" && at_date!="")  
{  
    at_date = DateTime.Now.ToString("MM-dd-yyyy");  
    SqlConnection con = new SqlConnection();  
    con.ConnectionString = "Data  
Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=D:\\Projects\\Hotel management  
System\\Hotel Management Software\\Database1.mdf;Integrated Security=True";  
    SqlCommand cmd = new SqlCommand();  
    cmd.Connection = con;
```

```

String query1 = "select check_in from attendance where emp_id=" + id + " and at_date=" + at_date
+ "";
con.Open();
cmd.CommandText = query1;
using (SqlDataReader rd = cmd.ExecuteReader())
{
    if (rd.HasRows)
    {
        while (rd.Read())
        {
            checkin = rd.GetString(0);
            DateTime chin = DateTime.Parse(checkin);
            TimeSpan th = chout - chin;
            String thr = th.Hours.ToString() + ":" + th.Minutes.ToString() + ":" +
            th.Seconds.ToString();
            query = "UPDATE attendance SET check_out=" + checkout +
            ",total_hours=" + thr + " WHERE emp_id=" + id + " AND at_date=" +
            at_date + ";";
            int i = fn.setData(query);
            if (i != 0)
            {
                Timekeeper_login_Load(this, null);
            }
            else
            {
                MessageBox.Show("Updation failed !!");
            }
        }
    }
    else
    {
        MessageBox.Show("ID and date selected does not match");
    }
}
con.Close();
}
else
{
    MessageBox.Show("Please Fill the ID or select any row from table");
}

```

4.4 VALIDATIONS

Receptionist_admin Table

Attribute name	Valid Values	Invalid Values
username (pk)	Vaibhav1827	-
password	vaibhav	-
emp_id (fk)	E01	1,2
emp_designation	admin	Adm1234

Employee Table

Attribute name	Valid Values	Invalid Values
emp_id (pk)	E01	1,2
emp_name	Vaibhav Jadhav	Vai54554
emp_contact_no	7977142979	vaibhav
emp_gender	Male	1,2,A
emp_dept	administrative	1,2,A
emp_designation	admin	Admin1425
emp_salary	100000	A,b,c

Room Table

Attribute name	Valid Values	Invalid Values
room_id (pk)	R01001	1,2,@
room_no	101	A,B,@
room_type	deluxe	1,2,A
room_bed	single	1,2,A
room_price	4000	400.26562,A,@
booked	NO	1,2,@

Guest Table

Attribute name	Valid Values	Invalid Values
guest_id (pk)	G0001	G0001
guest_name	Pranav Narvekar	A124
guest_contact_no	9870313412	A,b,@
guest_nationality	india	1,2,A
guest_idproof_type	Aadhar_card	1,2,A
guest_idproof_no	5454-5454-5454	No whitespace
no_of_adults	1	1.5, A
no_of_children	0	1.5, A

Attendance Table

Attribute name	Valid Values	Invalid Values
emp_id (fk)	E01	1,2
at_date	17-09-2020	35.5, AB
check_in	2020-09-17 11:00:15	35.5, AB
check_out	2020-09-17 19:00:20	35.5, AB
total_hours	8	ABC

Records Table

Attribute name	Valid Values	Invalid Values
record_id (pk)	RC001	1,5, A
guest_id (fk)	G0001	1,3, A
guest_name	Pranav Narvekar	1541,2.5
guest_contact_no	9870313412	A, B
no_of_adults	1	1.5, A
no_of_children	0	1.5, A
room_id (fk)	R01001	1,2,@
room_no	101	A, B
no_of_days	2	A, b
total_price	8000	A,B
check_in	2020-09-17 11:00:15	16.5, AB
check_out	2020-09-19 10:15:19	16.5, AB
check_out_yn	Y	1,2,A

4.5 TEST CASES, TEST DATA AND TEST RESULTS

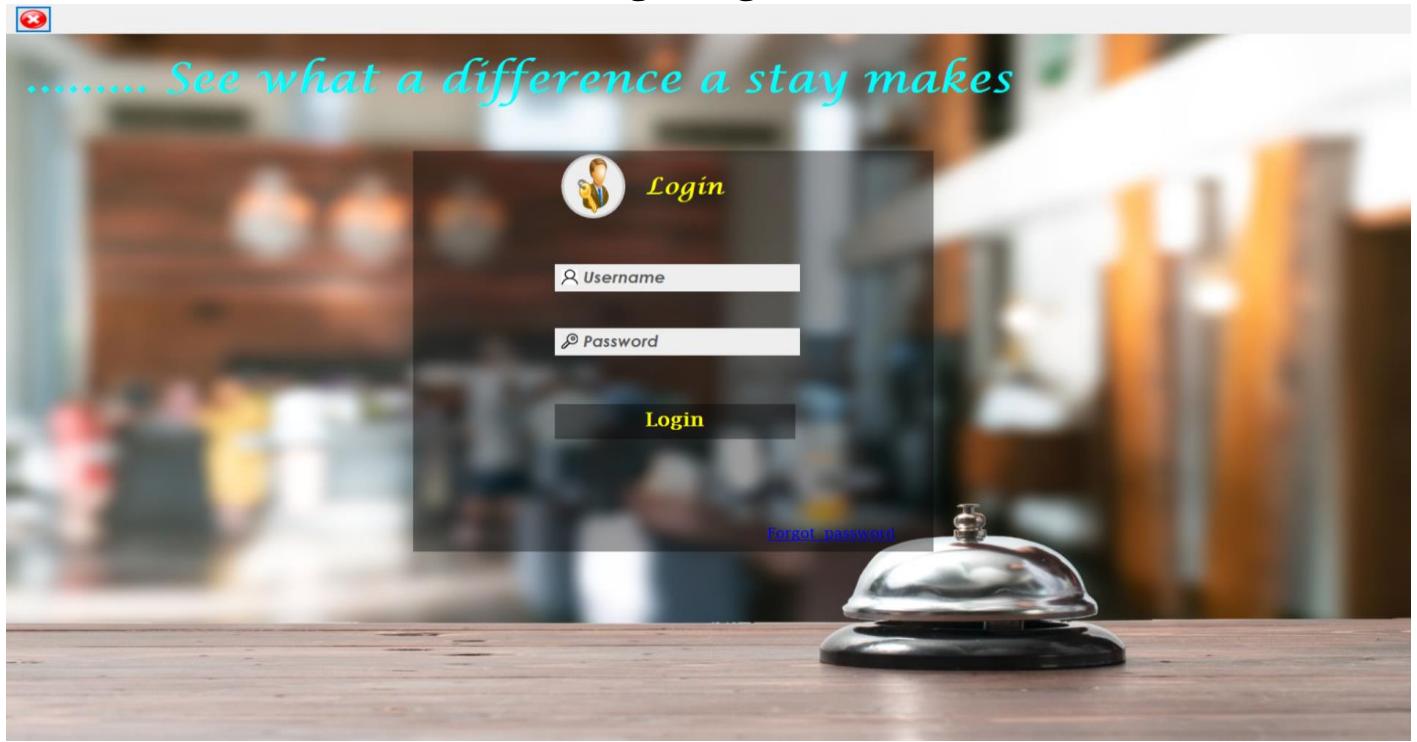
Test Case Id :	Module (Unit to be tested)	Why?	Desired Output	Observed Output	Pass or Fail
1	Authentication & Login	Providing login into the system only for authentic users.	Access should be provided only after proper check of username and password.	Access was provided only after proper check of username and password.	Pass
2	Adding Employee	Admin can add new Employee.	Adding feature should be provided to the Admin.	Adding feature was provided to the Admin.	Pass
3	Deleting Employee Details	Only Admin can delete details.	Deleting feature should be provided to the Admin.	Deleting feature was provided to the Admin.	Pass
4	Updating Employee Details	Only Admin of can update or modify details.	Updating feature should be provided to the Admin to update their Employee details.	Updating feature was provided to the Admin.	Pass
5	Viewing Employee Details	Only Admin can view details of Employee.	Viewing Employee details should be provided to the Admin.	Viewing an Employee details was provided to the Admin details.	Pass
6	Adding Room	Only Admin can add new Rooms	Adding feature should be provided to the admin.	Adding feature was provided to the Admin.	Pass
7	Updating Room Details	Only Admin can update Room details.	Updating feature should be provided to the admin to update their Room details.	Updating feature was provided to the admin.	Pass
8	Removing Room	Only Admin can Remove Room.	Removing feature should be provided to the Admin.	Removing feature was provided to the Admin.	Pass

9	Guest Registration	Only Receptionist can Register Guest	Registering feature should be provided to the Receptionist.	Registration feature was provided to the Receptionist	Pass
10	Viewing Records	Only Receptionist and Admin can View Records	Viewing records should be only provided to Admin and Receptionist only	Viewing records feature was provided to the Admin and Receptionist only	Pass
11	Viewing Attendance	Only Admin can View Attendance	Viewing Attendance should be only provided to Admin	Viewing Attendance feature was provided to the Admin only.	Pass
12	Checking out of Guest	Only Receptionist can check out guest	Checking out Guest feature should be provided to the Receptionist	Checking out Guest feature was provided to the Receptionist	Pass
13	Marking Attendance	Only timekeeper can mark attendance	Marking attendance feature should be provided to timekeeper only	Marking attendance feature was provided to timekeeper only	Pass
14	Giving Packages	Only Receptionist can give packages	Giving Packages feature should be provided to the Receptionist	Giving Packages feature was provided to the Receptionist	Pass
15	Giving Other Luxuries	Only Receptionist can give other luxuries	Giving Other Luxuries feature should be provided to the Receptionist	Giving Other Luxuries feature was provided to the Receptionist	Pass
16	Assigning username and password	Only Admin can give assign username and password	Assigning username and password feature should be provided to the Admin	Assigning username and password feature was provided to the Admin	Pass

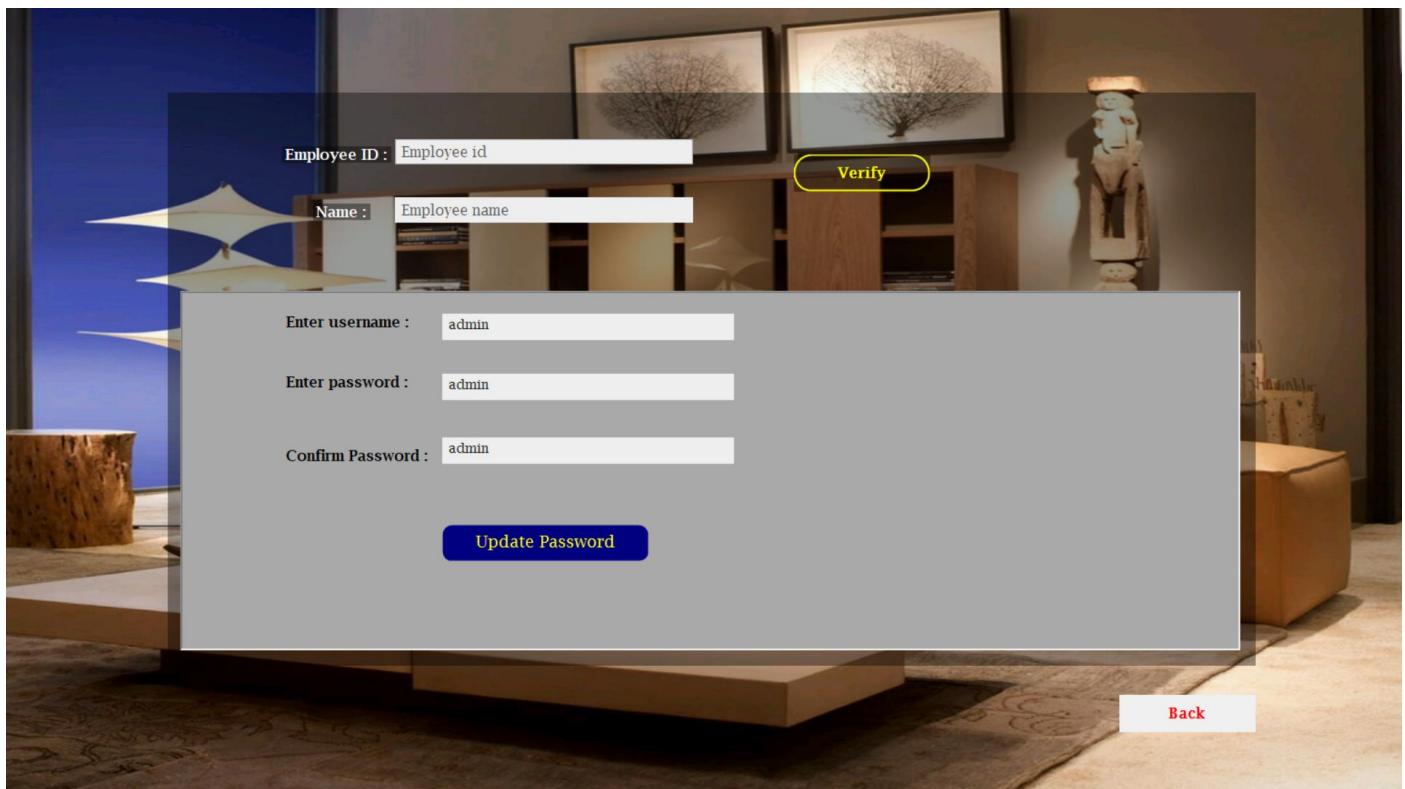
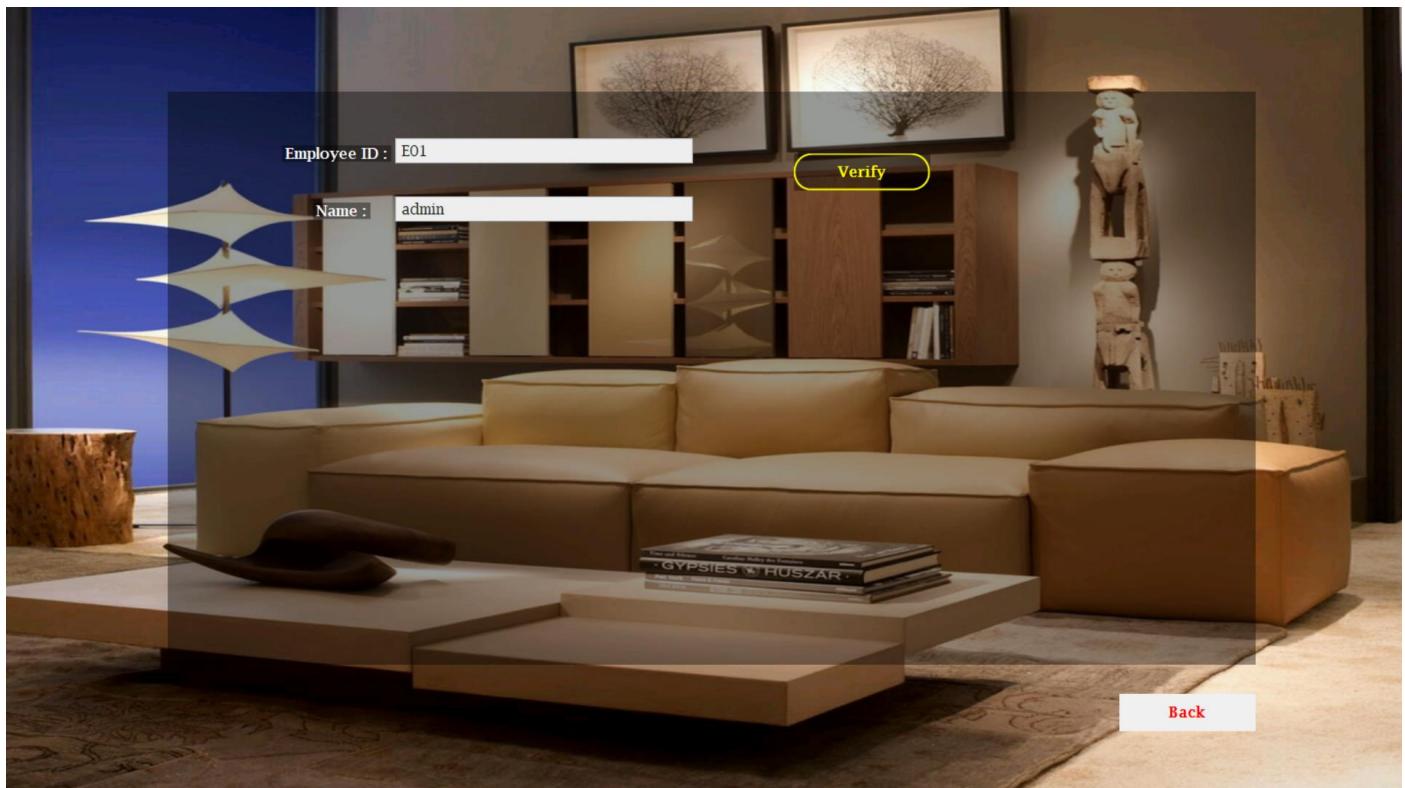
17	Two Username cannot be same	Usernames are used as ID to login	Two or more username feature should give and error	Two or more username feature gives error	Pass
----	-----------------------------	-----------------------------------	----------------------------------------------------	------------------------------------------	------

4.6 Screen Layouts

Login Page



Recovery Form for Password



Admin Dashboard

Date : 02-12-2020
Time : 11:36:03


Welcome Admin

Dashboard

Employee

Rooms

Records

Attendance

Logout

Username

Employee ID

Password

Desgination

	username	password	emp_id	emp_designation
▶	admin	admin	E01	admin
	receptionist	receptionist	E02	receptionist
	timekeeper	timekeeper	E03	timekeeper

Add
Update
Remove

Admin Employee

Date : 02-12-2020
Time : 11:37:25


Welcome Admin

Dashboard

Employee

Rooms

Records

Attendance

Logout

	emp_id	emp_name	emp_contact_no	emp_gender	emp_dept	emp_designation	emp_salary
▶	E01	admin	9875632110	Male	Administrative	admin	50000
	E02	receptionist	9875412350	Male	Front Office	receptionist	20000
	E03	timekeeper	8745251364	Male	Attendance	timekeeper	20000

Add
Update
Remove

Employee ID :

Contact No. :

Designation :

Employee Name :

Department :

Salary :

Gender :

36 | Page

Vaibhav Jadhav

Admin Rooms

Date : 03-12-2020
Time : 11:25:38

Welcome Admin

The screenshot shows the Admin Rooms section of the Hotel Management System. At the top, there's a yellow header bar with the date and time. Below it is a navigation sidebar with icons and labels: Dashboard, Employee, Rooms, Records, Attendance, and Logout. The Logout button is highlighted with a red box. The main area features a table of room information and several input fields for managing rooms.

room_id	room_no	room_type	room_bed	room_price	booked
R01001	101	Deluxe	Single	2000	YES
R01002	102	Deluxe	Single	2000	YES
R01003	103	Deluxe	Double	3000	YES
R01004	104	Deluxe	Double	3000	YES
R01005	105	Deluxe	Triple	4500	YES
R01006	106	Deluxe	Double	3000	NO
R02001	201	Twin	Double	4500	NO

Room Number:

Add Room Update Room Remove Room

Room ID : R01002 Bed : Single

Room Number : 102 Price : 2000

Type : Deluxe Booked : YES NO

Admin Records

Date : 03-12-2020
Time : 11:25:45

Welcome Admin

The screenshot shows the Admin Records section of the Hotel Management System. It has a similar layout to the Admin Rooms page, with a sidebar and a main content area featuring a table of guest records and a search interface.

record_id	guest_id	guest_name	guest_contact_no	no_of_adults	no_of_children	room_id	room_no	no_
RC001	G0001	Pranav	7854125230	1	0	R01001	101	
RC002	G0002	Jayesh	9854126301	2	0	R01002	102	
RC003	G0003	Nisha	7454142635	1	1	R01004	104	
RC004	G0004	Vedika	9874512354	1	0	R01005	105	
RC005	G0005	Rutuja	7854682310	1	0	R01003	103	

Search Type :

Admin Attendance

Date : 03-12-2020
Time : 11:26:14



Welcome Admin

Date from : Date To :

	emp_id	at_date	check_in	check_out	total_hours
▶	E01	02-12-2020	12-02-2020 10:43:40	12-02-2020 10:44:05	0:25
	E02	02-12-2020	12-02-2020 10:43:47	12-02-2020 10:44:20	0:33
	E03	02-12-2020	12-02-2020 10:43:53	12-02-2020 10:44:12	0:19

[Dashboard](#)
[Employee](#)
[Rooms](#)
[Records](#)
[Attendance](#)
[Logout](#)

Receptionist Login

Date : 03-12-2020
Time : 11:27:38



Welcome Receptionist

Name :

Contact number :

Nationality :

ID Proof Type :

ID Proof Number :

Number of Adults :

Number of Children :

Room Type :

Room Bed :

Room Number :

Number of Days :

Price :

Total Price :

[Add Guest](#)
[Packages](#)
[Other Luxuries](#)
[Check Out](#)
[Records](#)
[Log Out](#)

Packages Form

Date : 03-12-2020
Time : 11:30:41

Welcome Receptionist

Add Guest

Packages

Other Luxuries

Check Out

Records

Log Out

Guest Name : vedika

Guest ID : G0004

Package Type : Seasonal Package

Price per package : 1000

Quantity : 2

Total : 1800

Save

Seasonal Package

This Package will include :

- 1.Breakfast
- 2.Lunch
- 3.Dinner
- 4.Seasonal items

[Note: This package will be applicable to the number of days the guest stays]

Thank You !!!

Other Luxuries

Date : 03-12-2020
Time : 11:31:44

Welcome Receptionist

Add Guest

Packages

Other Luxuries

Check Out

Records

Log Out

Guest Name : vedika

Guest Name : G0004

Suit / Dress Type : Dress

Price : 150

Quantity : 3

Total : 450

Add to List

Save

Type	Price	Quantity	Total Price
Saree	200	2	400
Dress	150	3	450

Grand Total : 850

Check Out

Date : 03-12-2020
Time : 11:32:27

Welcome Receptionist

Guest name:

guest_id	guest_name	guest_contact_no	guest_nationality	guest_idproof_type	guest_idproof_no
G0001	Pranav	7854125230	India	Aadhaar card	5454-2525-3525
G0002	Jayesh	9854126301	India	Aadhar Card	5656-1423-1425
G0003	Nisha	7454142635	India	Aadhar Card	6584-3525-7474
G0004	Vedika	9874512354	India	Aadhar Card	6747-2235-1212
G0005	Rutuja	7854682310	India	Aadhaar Card	5454-2535-6525

< >

Guest ID : Guest Name Check Out

Records

Date : 03-12-2020
Time : 11:32:38

Welcome Receptionist

Search Type : InHotel Guest Details

record_id	guest_id	guest_name	guest_contact_no	no_of_adults	no_of_children	room_id	room_no	no.
RC003	G0003	Nisha	7454142635	1	1	R01004	104	
RC004	G0004	Vedika	9874512354	1	0	R01005	105	
RC005	G0005	Rutuja	7854682310	1	0	R01003	103	

< >

Timekeeper Login

Date : 03-12-2020 Time : 12:40:05

[Logout](#)

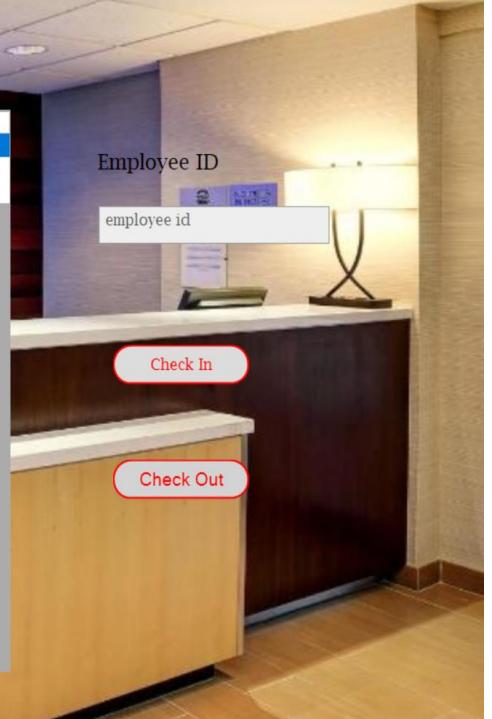
 Welcome

emp_id	at_date	check_in	check_out	total_hours
E01	02-12-2020	12-02-2020 10:43:40	12-02-2020 10:44:05	0:02:5
E02	02-12-2020	12-02-2020 10:43:47	12-02-2020 10:44:20	0:03:3
E03	02-12-2020	12-02-2020 10:43:53	12-02-2020 10:44:12	0:01:9

Employee ID

[Check In](#)

[Check Out](#)



5. FUTURE ENHANCEMENT

- The current System is a Desktop- Based application, Website for Hotel promotion can be made in future
- Website for Online Booking of Hotel Rooms can be also done in future using pre-allocation
- Online Application for admin and managers can be made in future so that they can monitor the activities of the Hotel
- E-mail module can be added so that when a Guest books his Room, an email notification can be sent which can contain all the necessary information of the booking.
- If there is any rectification or enhancement proposed by the user, the software will be modified and the desired module will be implemented into the current system
- If there is any change in the business process, the current system will be made to make provisions for such changes under the administrator power

6. Conclusion

- This system is user friendly since it has been developed in Visual Studio 2019 a successful GUI environment.
- Since the connection can be extended to any database. The control will be more powerful.
- Connecting it to any type of database extends the development control. Any suggestions for future development of the system are welcome.
- This desktop application can be used in hotel for keeping records, marking and maintaining attendance, adding rooms, recruiting and updating employee details
- Also, it can keep records of Employee's salary.
- By using this desktop application, a lot of burden of keeping records would be lesser as we can have everything saved onto our databases.

7. REFERENCES AND BIBILOGRAPHY

- www.google.com
- www.youtube.com
- www.stackoverflow.com
- www.tutorialspoint.com
- www.w3school.com