A Project Report

*On*

RESTAURANT MANAGEMENT SYSTEM

*By*

BAGADIYA SOMESH PUSHPKUMAR(ROLL NO. 307014)

AGRAWAL SUDHANSHU SANJAY(ROLL NO. 307002)

*Under the guidance of*

DR. MRS. KALPANA THAKRE



**Department of Information Technology**

**Sinhgad College of Engineering**

**SAVITRIBAI PHULE PUNE UNIVERSITY**

**2019-2020**

|  |  |
| --- | --- |
| Sinhgad Technical Education Society,  Department of Information Technology  Sinhgad College of Engineering , Pune-41 |  |



Date:

**CERTIFICATE**

This is to certify that,

BAGADIYA SOMESH PUSHPKUMAR(307014)

AGRAWAL SUDHANSHU SANJAY(307002)

of class T.E IT; have successfully completed their project work on “RESTAURANT MANAGEMENT SYSTEM’’ at SINHGAD COLLEGE OF ENGINEERING in partial fulfillment of the Graduate Degree course in T.E at the Department of Information Technology, in the academic Year 2019-2020 Semester – I as prescribed by the Savitribai Phule Pune University.

DR. MRS. KALPANA THAKRE Prof.G. R. Pathak

Guide Head of the Department

(Department of Information Technology) (Department of Information Technology)

**Acknowledgement**

I feel great pleasure in expressing my deepest sense of gratitude and sincere thanks to my guide **Prof. Kalpana Thakre** for their valuable guidance during the Project work, without which it would have been very difficult task. I have no words to express my sincere thanks for valuable guidance, extreme assistance and cooperation extended to all the **Staff Members** of my Department.

This acknowledgement would be incomplete without expressing my special thanks to **Prof.G. R. Pathak** Head of the Department (Information Technology) for their support during the work.

I would also like to extend my heartfelt gratitude to my **Principal, Dr. S. D. Lokhande** who provided a lot of valuable support, mostly being behind the veils of college bureaucracy.

Last but not least I would like to thanks all the Teaching, Non- Teaching staff members of my Department, my parents and my colleagues those who helped me directly or indirectly for completing this Project successfully.

BAGADIYA SOMESH PUSHPKUMAR

AGRAWAL SUDHANSHU SANJAY

,

**Contents**

**RESTAURANT MANAGEMENT SYSTEM**

1. **ABSTRACT**
2. **INTRODUCTION**

* Problem definition

1. **SCOPE**
2. **SPECIFIC REQUIRMENTS**

* Hardware Interface
* Software Interface

1. **THEORY OF SOFTWERE USED**

* Java (JDK)
* MYSQL
* NetBeans

1. **DATABASE FORMAT**
2. **ER DIAGRAM**
3. **OUTPUT SCREEN (GUI)**
4. **SAMPLE CODE**
5. **CONCLUSION**
6. **REFERENCES**

**ABSTRACT**

**Restaurant Management System:**

"Resto Buddy" is JavaFx application with MySQL database to restaurant management. This system wake to provide service facility to restaurants. The services that are provided includes managing chain of restaurants or multiple restaurants from a single app, appoint managers , employee database(add or delete) , add menu (add or delete), place order(add or delete), bill generation , report generation. Main objective to build the system is to ease the restaurant management. With this system online, the user (the person who is currently using the app) have ease to track orders that have been completed or is being prepared , generate virtual bill and reports about the restaurant to replace traditional system where are still using paper.

**INTRODUCTION**

**PROBLEM DEFINITION**

Nowadays, there are many restaurant chains running. They need a system to help them manage their restaurants,managers,employees,menu and orders. they also need to get their report for their different restaurants. The Owner is responsible for adding new restaurants and then appointing them managers. And then this managers are responsible to manage all the employee the restaurant menu.This can be very effectively done through a desktop system compared to the pen paper based record system.

Then in a particular restaurant the manager and employees places the orders and on successful completion a order they give a bill to the customers.Then keeping the record of this orders can be a very haptic and time consuming.

**SCOPE**

This project aims at developing a Management System, aesthetic “Restaurant Management System”

In this project we are going to solve all the problem that a restaurant chain has to deal with at the time of running.This project is going to help the owner keep a record of all the restaurants and their managers.It helps the owner to add or delete restaurants. It also helps him to appoint a manager to a restaurant and to keep a record of all closed restaurants and also managers who where removed.

This will also help manager to appoint or remove employee. He can also add or remove a menu item for his restaurant. Manager can also see all the previous employee who where working in the restaurant and also the menu items that a discontinued .And the manager and employee both will be able to place the order effortlessly from this system and keep the completion status of this order.Then they can also generate the bill for orders and to print this bill.

**SPECIFIC REQUIREMENTS**

The system analysis contains a planning and design phases where a logical design of the system is developed and to work accordingly a plan is established. Also the requirements of the system are identified and the operating environment is identified.

**Hardware Requirements**

1) Operating System: Windows 7 or Above

2) RAM: 1GB or More

3) Memory Space: 200MB or More

4) Processor: Pentium Core 2 Duo or Above

**Software Requirements**

1) MySQL Version 5 or Above

2) Java 8 or Above

3) Netbeans 8 or Above

**ADDITIONAL FILES USED**

1. JDK 1.8
2. mysql-connector-java-5.1.48.jar
3. mysql-connector-java-5.1.48-bin.jar
4. rs2xml.jar

**THEORY OF SOFTWARE USED**

**SOFTWARE USED:**

1. Java 8
2. Netbeans 8.2
3. MySql 5.1

**Java**

Java is a programming language originally developed by James Gosling at Sun Microsystems (which is now a subsidiary of Oracle Corporation) and released in 1995 as a core component of Sun Microsystems Java platform. The language derives much of its syntax from C and C++ but has a simpler object model and fewer low-level facilities. Java applications are typically compiled to byte code (class file) that can run on any Java Virtual Machine (JVM) regardless of computer architecture. Java is a general-purpose, concurrent, class-based, object-oriented language that is specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere". The java is independent to platform so it’s important. Java is currently one of the most popular programming languages in use, and is widely used from application software to web applications.

James Gosling, Mike Sheridan, and Patrick Naught on initiated the Java language project in June 1991. Java was originally designed for interactive television, but it was too advanced for the digital cable television industry at the time. The language was initially called Oak after an oak tree that stood outside Gosling's office; it went by the name Green later, and was later renamed Java, from a list of random words. Gosling aimed to implement a virtual machine and a language that had a familiar C/C++ of notation.

Java is an object-oriented programming language developed by Sun Microsystems in the 1990s. Since then, Java has gained enormous popularity as a computer language. Java was chosen as the programming language for network computers. It is a universal front end for the enterprise database. Sun Microsystems states that, “Java is a simple, object-oriented, distributed, secure, architecture, robust, multi-threaded and dynamic language. The program can be written once, and run anywhere”. One of the most significant advantages of Java is that it has the ability to move easily from one computer to another. It also has the ability to run the same program on many different operating systems. With such exemplary benefits, Java is a hot favourite among techies and software professional sit allows you to create modular programs and reusable codes.

**Java Features**

**1] Simple, Small and familiar:**

Java is a simple and small language. The Syntax of java is just like C++, so it is very easy to learn. It is simple because it i) does not use header files ii) eliminated the use of pointer iii) operator overloading and virtual base classes are eliminated.

**2] Object oriented:**

Java is a pure Object oriented. Everything in java is object. All programs and data reside inside objects and classes

**3] Distributed:**

Java has networking facilities. so java can create application on network.

**4] Robust:**

Java gives importance to memory management by using the technique called Garbage Collection and Exception handling.

**5] Secure:**

Since java is used on the internet, security is an important issue. A security code is asked before a java code is interpreted on the internet.

**6] Platform independent:**

Java compiler generates a platform independent code called byte code.

**7] Portable:**

The Byte code generated by java can be used on any machine. So it can be portable.

**8] Compiled and Interpreted:**

**9] High performance:**

The use of byte code makes the performance high. the speed is also high with comparing c, c++.

**10] Multithreading and interactive:**

Multithreading means handling more than one job at a time. Java supports Multithreading.

**11] Dynamic and extensible:**

Java is a dynamic language. So it is capable of linking dynamic new classes, methods and objects. Java supports functions written in C and C++ also. These functions are called native methods. During Run-Time Native methods can be linked dynamically.

**MYSQL Database**

MySQL is the world's most popular open source database. With its proven performance, reliability and ease-of-use, MySQL has become the leading database choice for web-based applications, used by high profile web properties including Facebook, Twitter, YouTube, Yahoo! and many more.

Oracle drives MySQL innovation, delivering new capabilities to power next generation web, cloud, mobile and embedded applications.

**Ease of Use**

Nowadays, we use relational database management systems (RDBMS) to store and manage huge volume of data. This is called relational database because all the data is stored in different tables and relations are established using primary keys or other keys known as Foreign Key

**Features of MySQL**

The following list shows the most important properties of MySQL. This section is directed to the reader who already has some knowledge of relational databases. We will use some terminology from the relational database world without defining our terms exactly. On the other hand, the explanations should make it possible for database novices to understand to some extent what we are talking about.

**Relational Database System:** Like almost all other database systems on the market, MySQL is a relational database system.

**Client/Server Architecture:** MySQL is a [client/server system](https://searchdatamanagement.techtarget.com/feature/Understanding-and-comparing-six-types-of-processing-systems). There is a database server (MySQL) and arbitrarily many clients (application programs), which communicate with the server; that is, they query data, save changes, etc. The clients can run on the same computer as the server or on another computer (communication via a local network or the Internet).

Almost all of the familiar large database systems (Oracle, Microsoft SQL Server, etc.) are client/server systems. These are in contrast to the file-server systems, which include Microsoft Access, dBase and FoxPro. The decisive drawback to file-server systems is that when run over a network, they become extremely inefficient as the number of users grows.

**SQL compatibility:** MySQL supports as its database language -- as its name suggests – SQL (Structured Query Language). SQL is a standardized language for querying and updating data and for the administration of a database. There are several SQL dialects (about as many as there are database systems). MySQL adheres to the current SQL standard (at the moment SQL:2003), although with significant restrictions and a large number of extensions.

Through the configuration setting sql-mode you can make the MySQL server behave for the most part compatible with [various database systems](https://searchoracle.techtarget.com/tip/MySQL-Community-Server-vs-Oracle-MySQL-databases). Among these are IBM DB/2 and Oracle. (The setting sql-mode changes some of the syntax conventions, and performs no miracles. More details are to be had in Chapter 14.)

A readable and entertaining article on the topic of how MySQL differs from other current database systems can be found at<http://sql-info.de/mysql/gotchas.html>.

**SubSELECTs:** Since version 4.1, MySQL is capable of processing a query in the form SELECT \* FROM table1 WHERE x IN (SELECT y FROM table2) (There are also numerous syntax variants for subSELECTs.)

**Views:** Put simply, views relate to an SQL query that is viewed as a distinct database object and makes possible a particular view of the database. MySQL has supported views since version 5.0.

**Stored procedures:** Here we are dealing with SQL code that is stored in the database system.

**DATABASE FORMAT**

mysql> show tables;

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

| Tables\_in\_dbmsmini |

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

| bill |

| employee |

| employee\_history |

| manager |

| manager\_history |

| menu |

| menu\_history |

| orders |

| owner |

| restaurant |

| restaurant\_history |

| user |

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

12 rows in set (0.00 sec)

mysql> desc bill;

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

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

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

| bill\_no | int(10) | NO | PRI | NULL | auto\_increment |

| rest\_id | int(10) | YES | MUL | NULL | |

| user\_id | int(10) | YES | MUL | NULL | |

| bill\_amt | int(10) | YES | | NULL | |

| order\_type | varchar(30) | YES | | NULL | |

| bill\_date | timestamp | YES | | CURRENT\_TIMESTAMP | DEFAULT\_GENERATED |

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

6 rows in set (0.01 sec)

mysql> desc employee;

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

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

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

| emp\_id | int(10) | NO | PRI | NULL | auto\_increment |

| rest\_id | int(10) | YES | MUL | NULL | |

| emp\_name | varchar(30) | NO | | NULL | |

| emp\_password | varchar(30) | NO | | NULL | |

| emp\_contact | varchar(10) | NO | | NULL | |

| emp\_salary | int(10) | NO | | NULL | |

| emp\_designation | varchar(20) | YES | | NULL | |

| emp\_hiredate | timestamp | YES | | CURRENT\_TIMESTAMP | DEFAULT\_GENERATED |

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

8 rows in set (0.00 sec)

mysql> desc employee\_history;

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

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

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

| emp\_id | int(10) | YES | | NULL | |

| rest\_id | int(10) | YES | | NULL | |

| emp\_name | varchar(30) | YES | | NULL | |

| emp\_password | varchar(30) | YES | | NULL | |

| emp\_contact | varchar(10) | YES | | NULL | |

| emp\_salary | int(10) | YES | | NULL | |

| emp\_designation | varchar(20) | YES | | NULL | |

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

7 rows in set (0.00 sec)

mysql> desc manager;

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

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

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

| manager\_id | int(10) | NO | PRI | NULL | auto\_increment |

| rest\_id | int(10) | NO | MUL | NULL | |

| manager\_name | varchar(30) | NO | | NULL | |

| manager\_password | varchar(30) | NO | | NULL | |

| manager\_contact | varchar(10) | NO | | NULL | |

| manager\_salary | int(10) | NO | | NULL | |

| manager\_hiredate | timestamp | YES | | CURRENT\_TIMESTAMP | DEFAULT\_GENERATED |

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

7 rows in set (0.00 sec)

mysql> desc manager\_history;

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

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

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

| manager\_id | int(10) | YES | | NULL | |

| rest\_id | int(10) | YES | | NULL | |

| manager\_name | varchar(30) | YES | | NULL | |

| manager\_password | varchar(30) | YES | | NULL | |

| manager\_contact | varchar(10) | YES | | NULL | |

| manager\_salary | int(10) | YES | | NULL | |

| manager\_hiredate | timestamp | YES | | NULL | |

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

7 rows in set (0.00 sec)

mysql> desc menu;

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

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

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

| rest\_id | int(10) | NO | MUL | NULL | |

| item\_id | int(10) | NO | PRI | NULL | auto\_increment |

| item\_name | varchar(30) | NO | | NULL | |

| item\_cost | int(10) | YES | | NULL | |

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

4 rows in set (0.00 sec)

mysql> desc menu\_history;

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

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

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

| rest\_id | int(10) | YES | | NULL | |

| item\_id | int(10) | YES | | NULL | |

| item\_name | varchar(30) | YES | | NULL | |

| item\_cost | int(10) | YES | | NULL | |

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

4 rows in set (0.00 sec)

mysql> desc orders;

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

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

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

| rest\_id | int(10) | YES | MUL | NULL | |

| user\_id | int(10) | NO | MUL | NULL | |

| order\_id | int(10) | NO | PRI | NULL | auto\_increment |

| item\_id | int(10) | NO | MUL | NULL | |

| item\_qty | int(10) | NO | | NULL | |

| order\_type | varchar(20) | NO | | NULL | |

| bill\_status | varchar(30) | YES | | pending | |

| order\_status | varchar(30) | YES | | preparing | |

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

8 rows in set (0.00 sec)

mysql> desc owner;

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

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

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

| owner\_username | varchar(30) | YES | | NULL | |

| owner\_password | varchar(30) | YES | | NULL | |

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

2 rows in set (0.00 sec)

mysql> desc restaurant;

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

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

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

| rest\_id | int(10) | NO | PRI | NULL | auto\_increment |

| rest\_name | varchar(30) | NO | | NULL | |

| rest\_password | varchar(30) | NO | | NULL | |

| rest\_contact | varchar(10) | NO | | NULL | |

| rest\_location | varchar(20) | NO | | NULL | |

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

5 rows in set (0.00 sec)

mysql> desc restaurant\_history;

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

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

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

| rest\_id | int(10) | YES | | NULL | |

| rest\_name | varchar(30) | YES | | NULL | |

| rest\_password | varchar(30) | YES | | NULL | |

| rest\_contact | varchar(10) | YES | | NULL | |

| rest\_location | varchar(20) | YES | | NULL | |

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

5 rows in set (0.00 sec)

mysql> desc user;

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

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

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

| user\_id | int(10) | NO | PRI | NULL | |

| user\_name | varchar(20) | NO | | NULL | |

| user\_email | varchar(20) | NO | | NULL | |

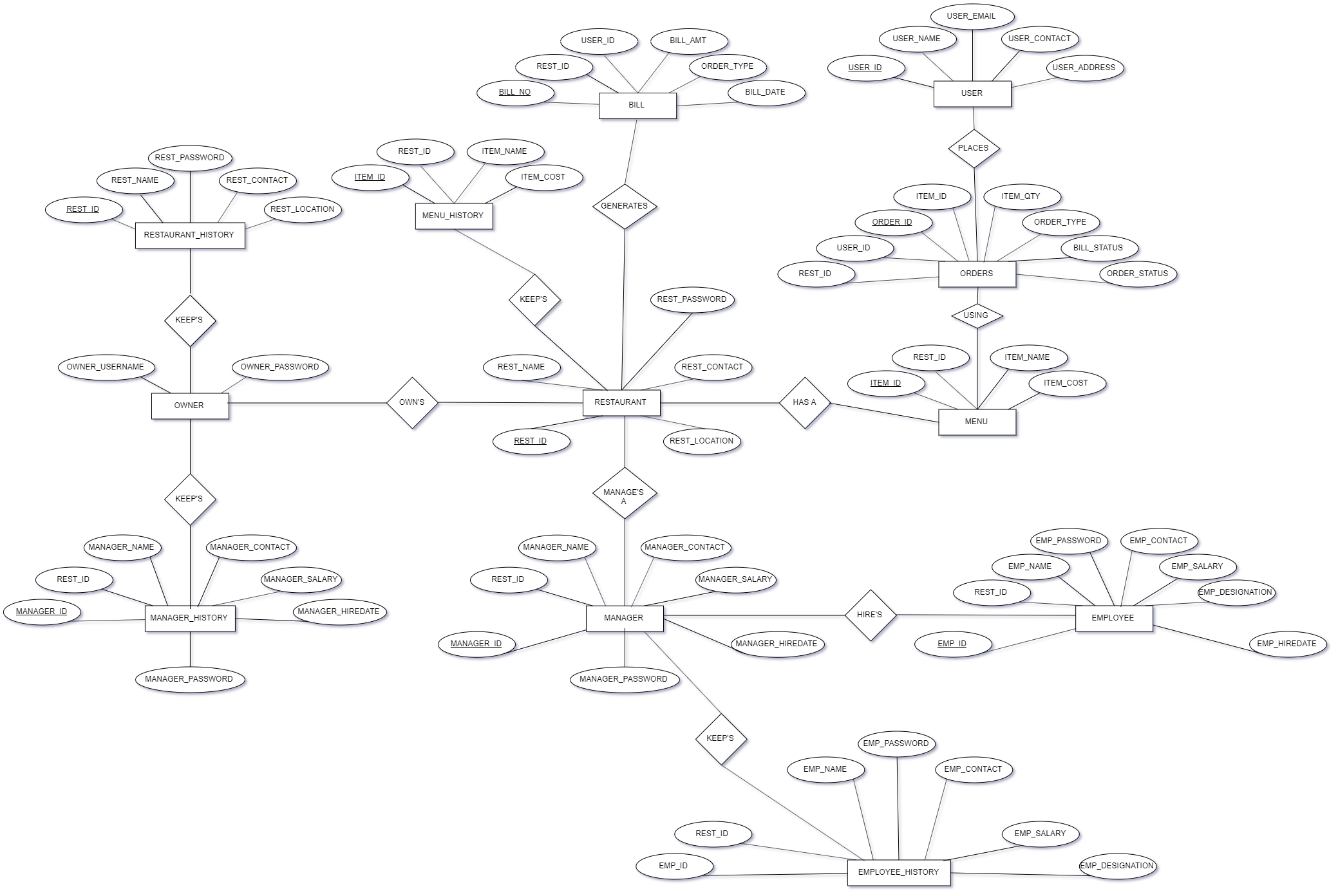
| user\_contact | varchar(20) | NO | | NULL | |

| user\_address | varchar(50) | YES | | NULL | |

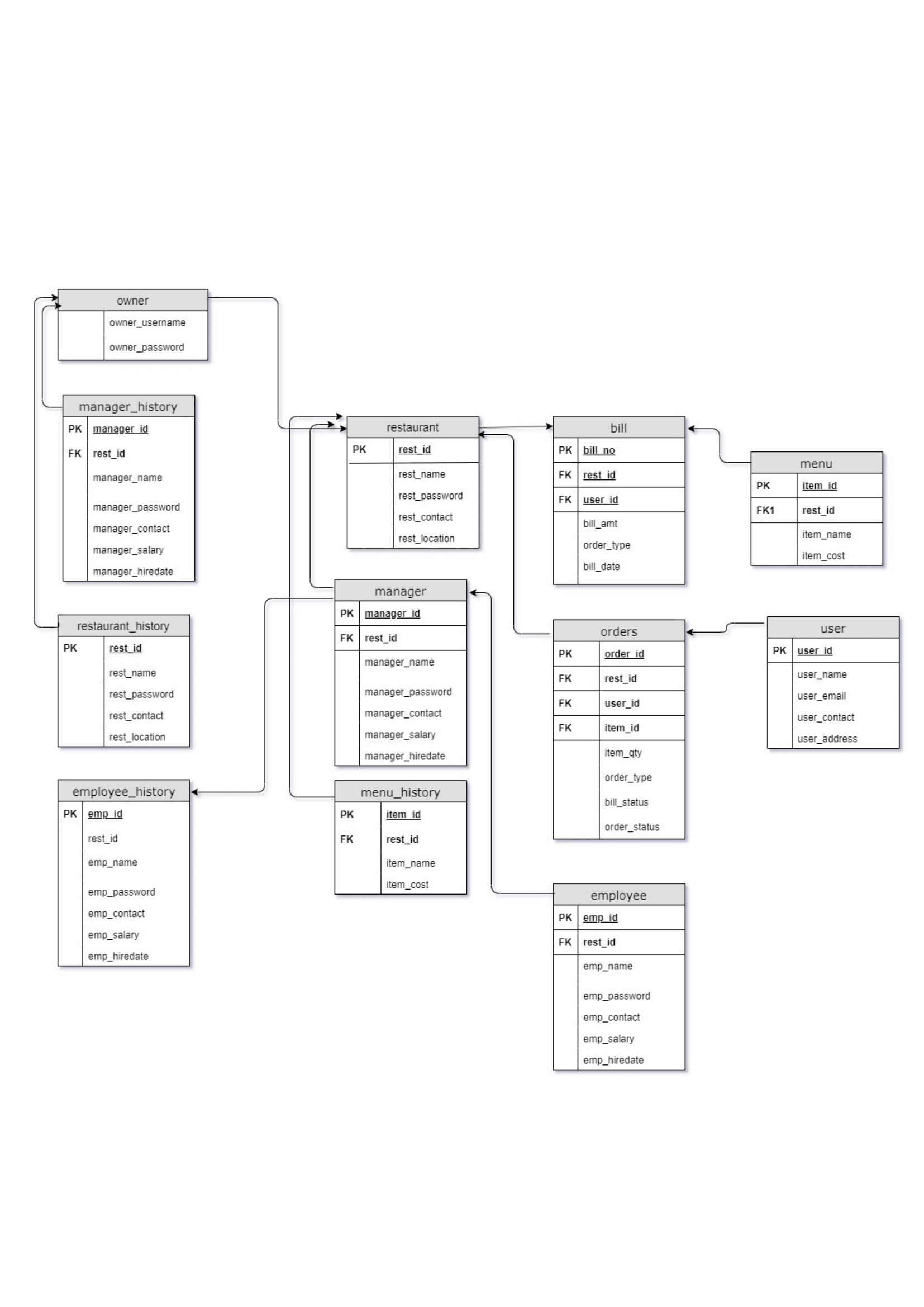
+--------------+-------------+------+-----+---------+-------+

5 rows in set (0.00 sec)

**ER DIAGRAM**

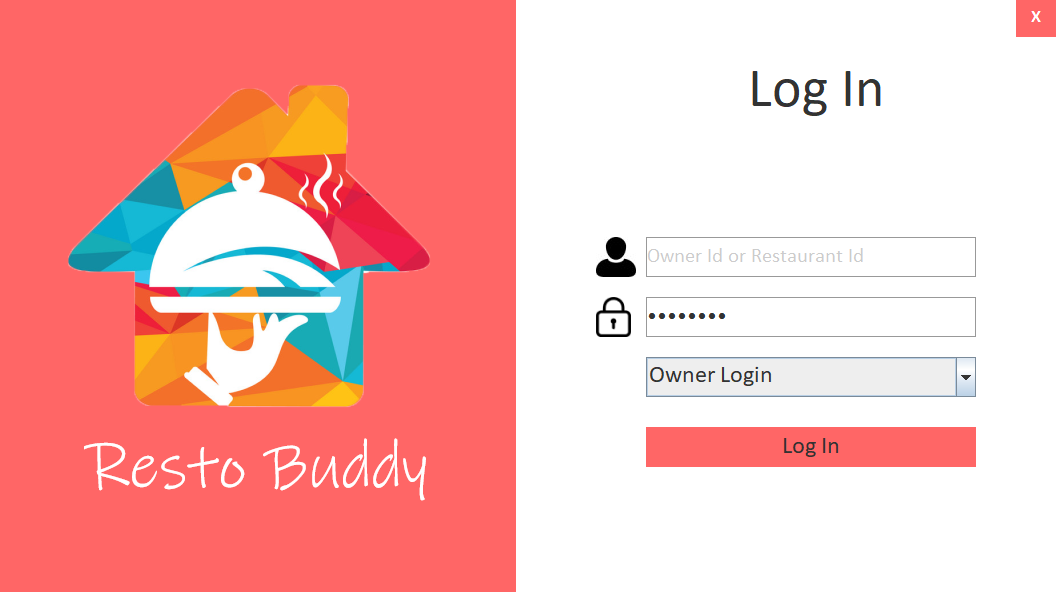
****

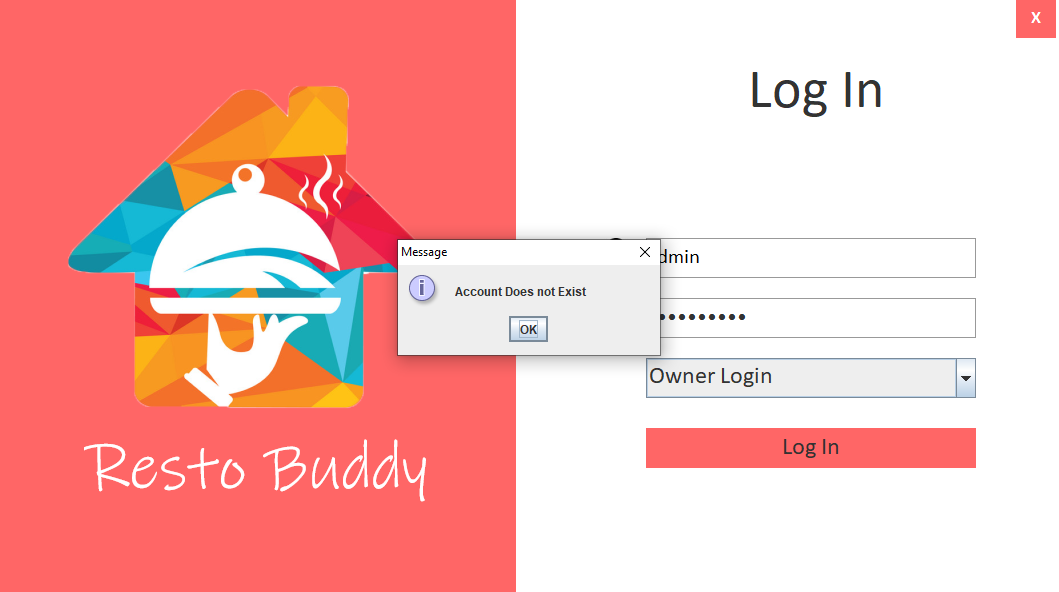
**SCHEMA DIAGRAM**

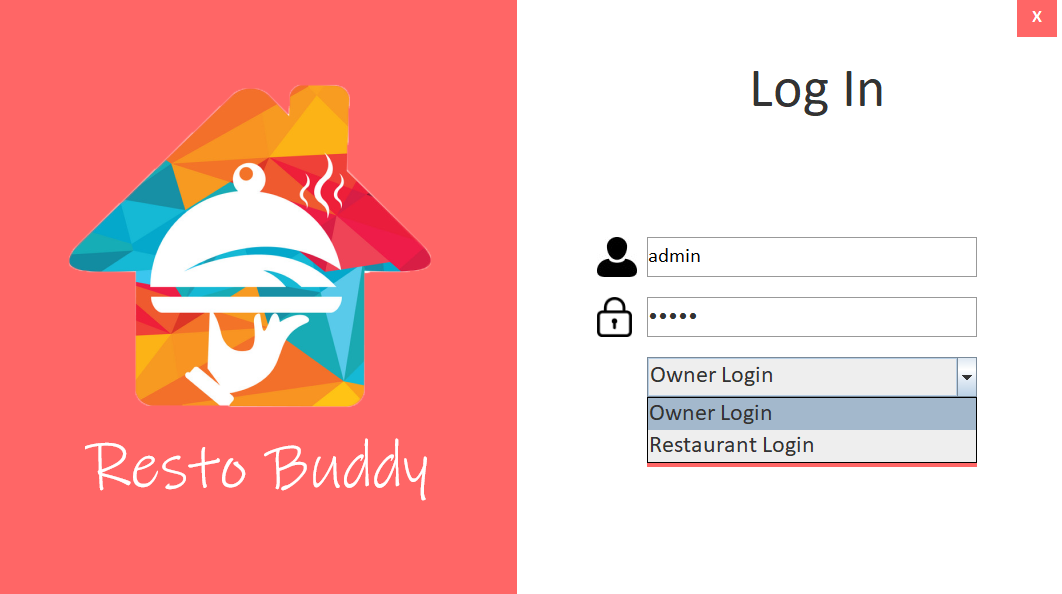


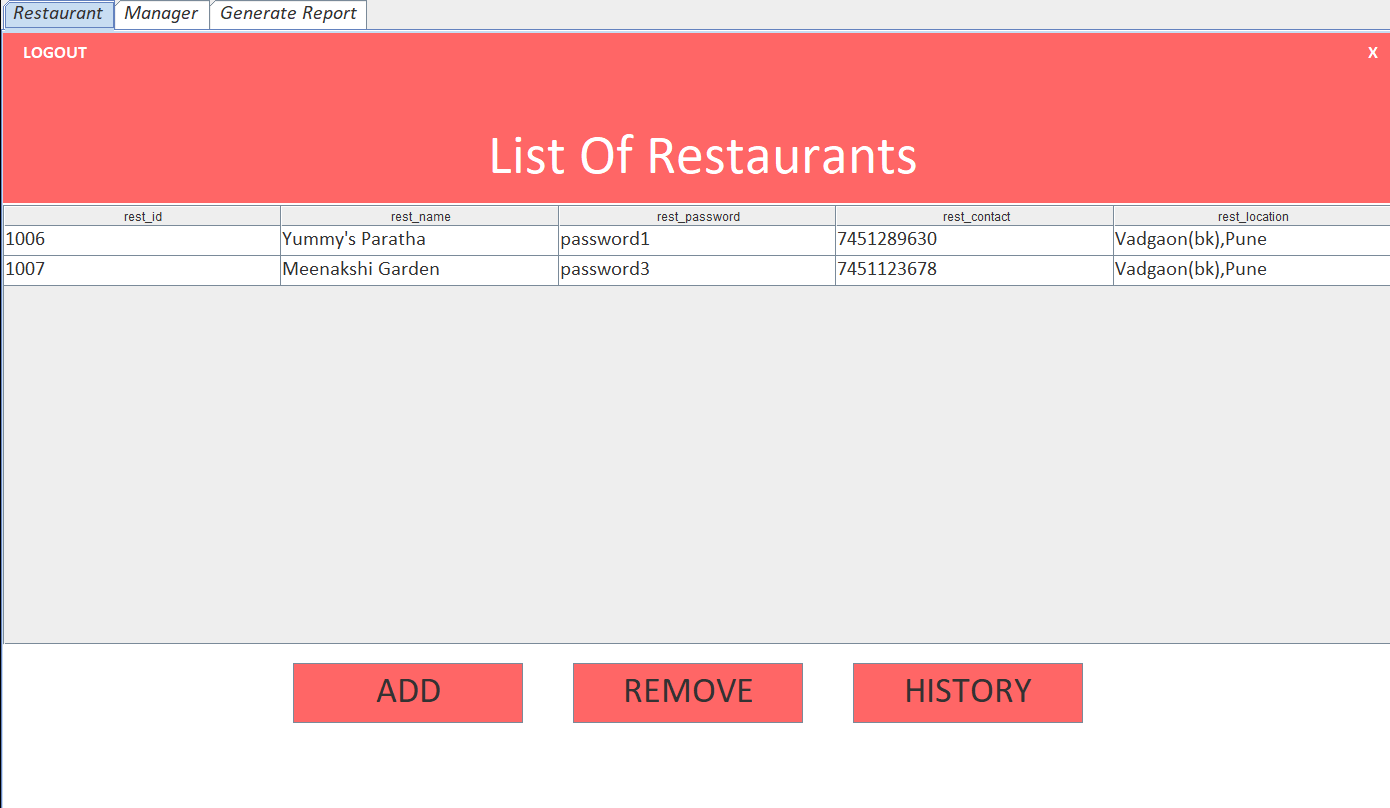
**SCHEMA DIAGRAM**

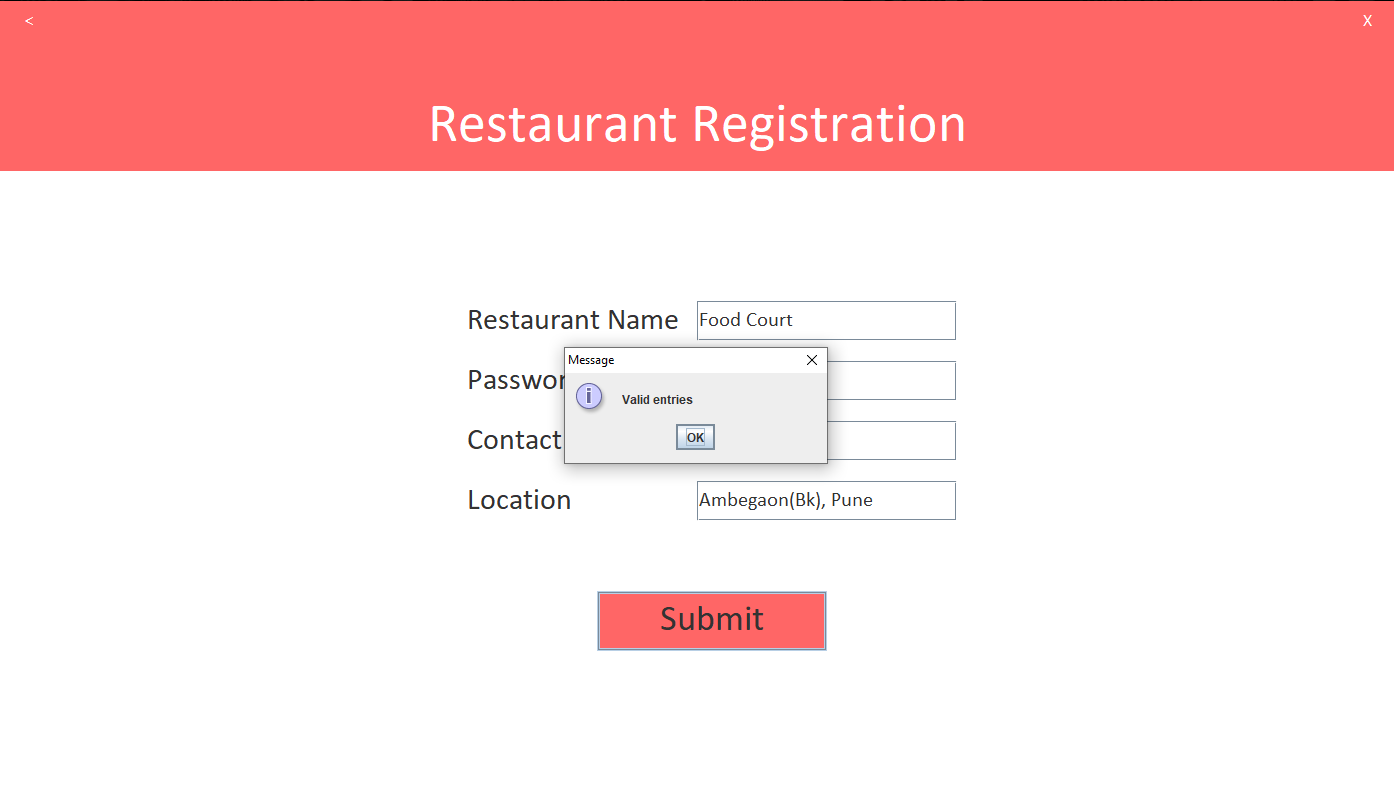
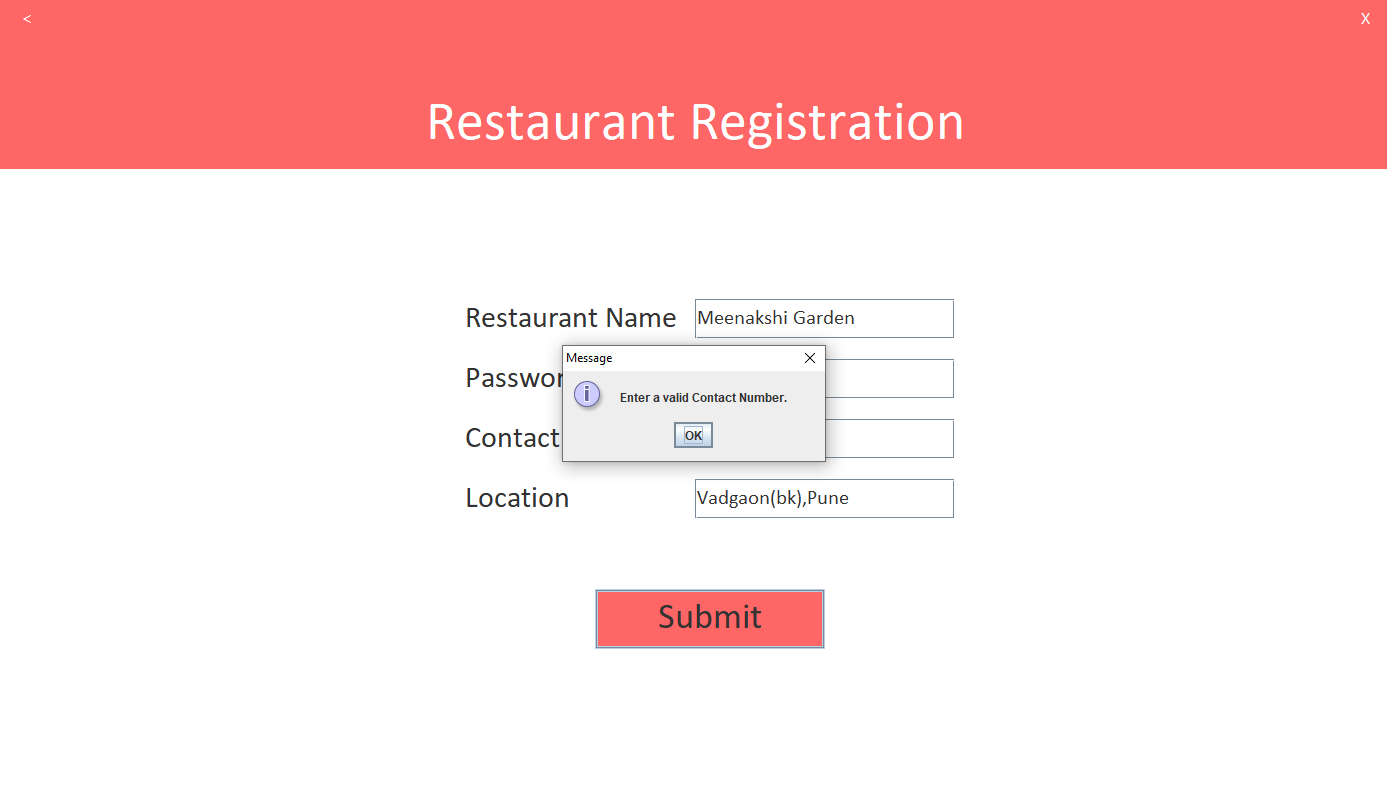
**OUTPUT SCREEN(GUI)**

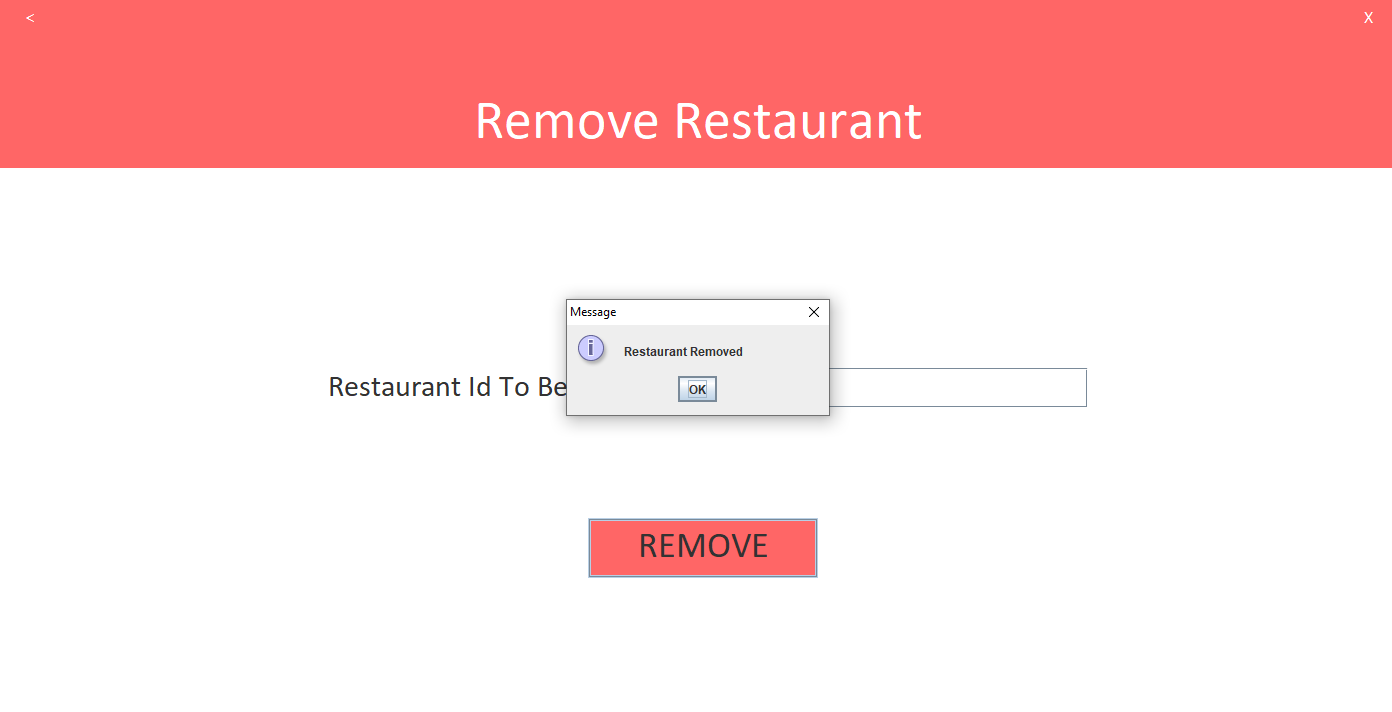
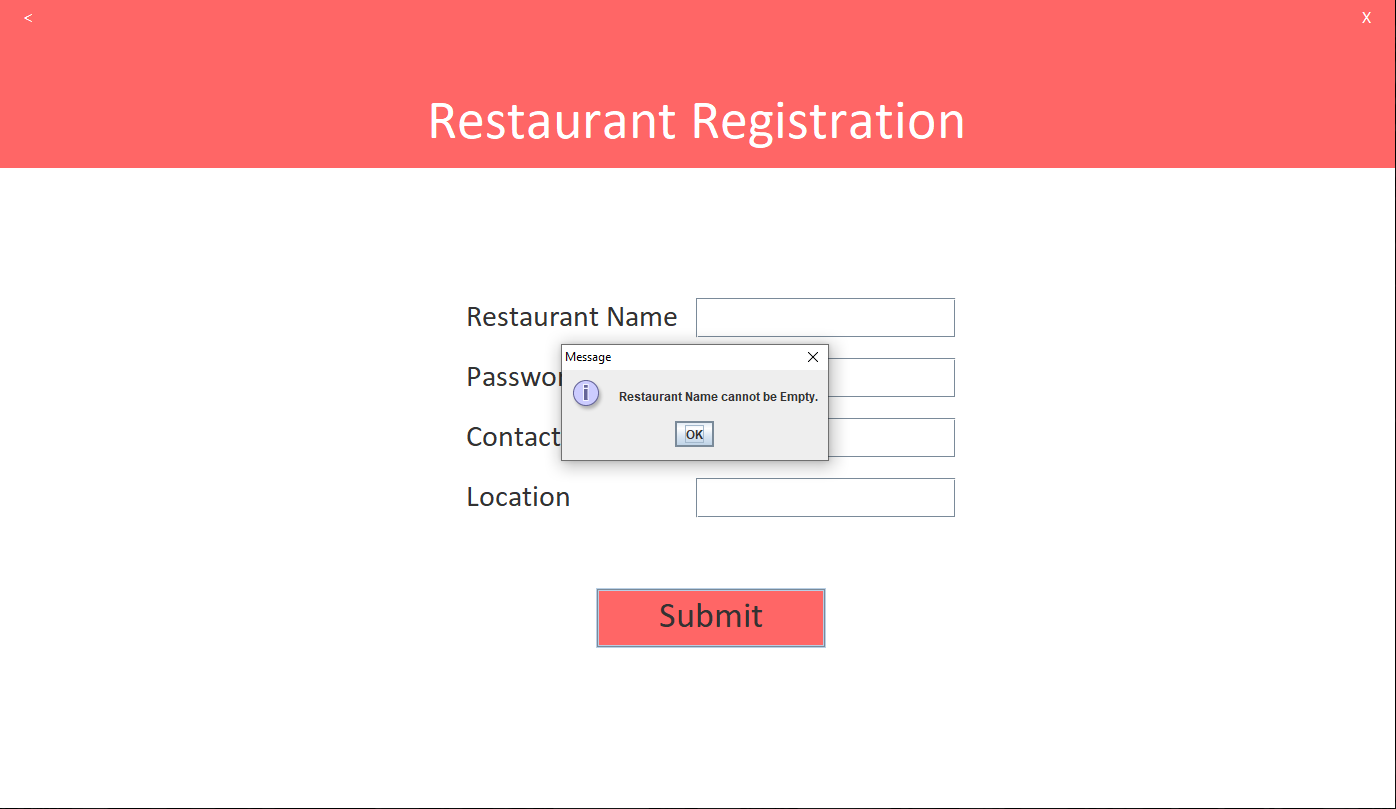
**Owner Sign In**

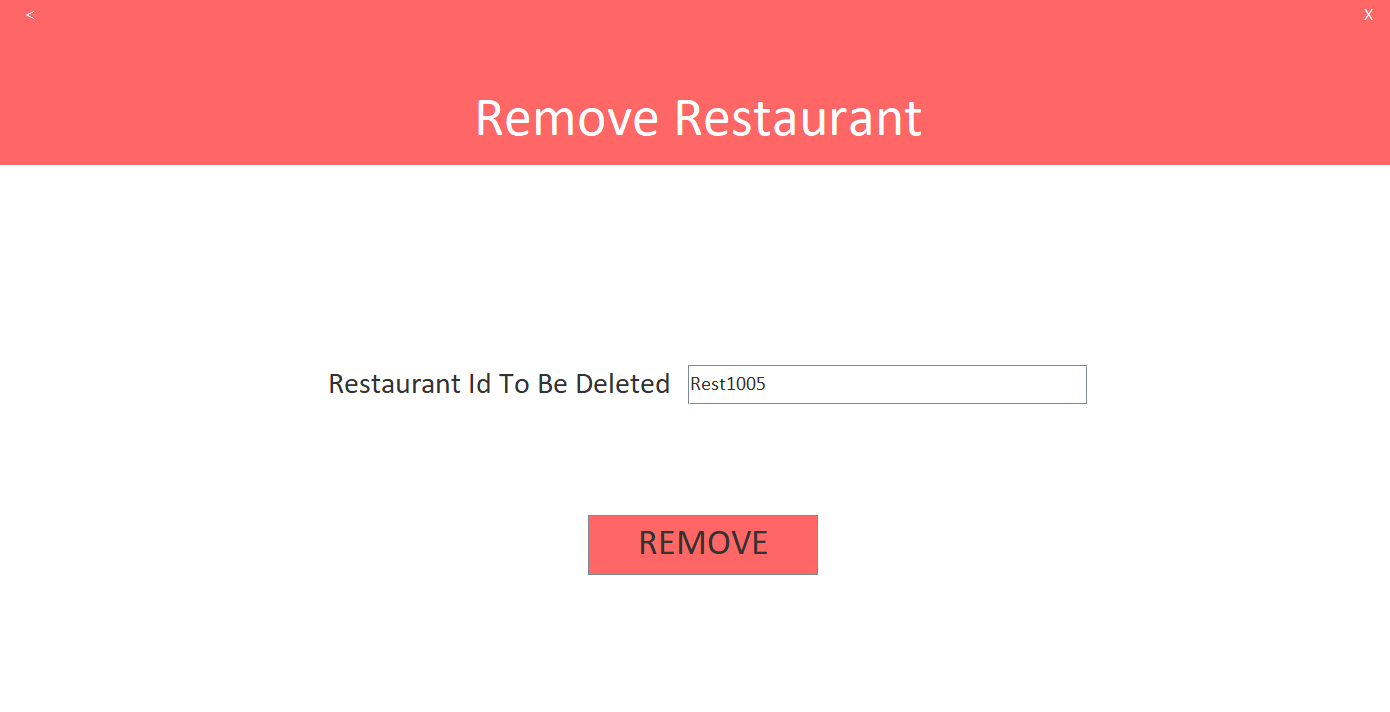


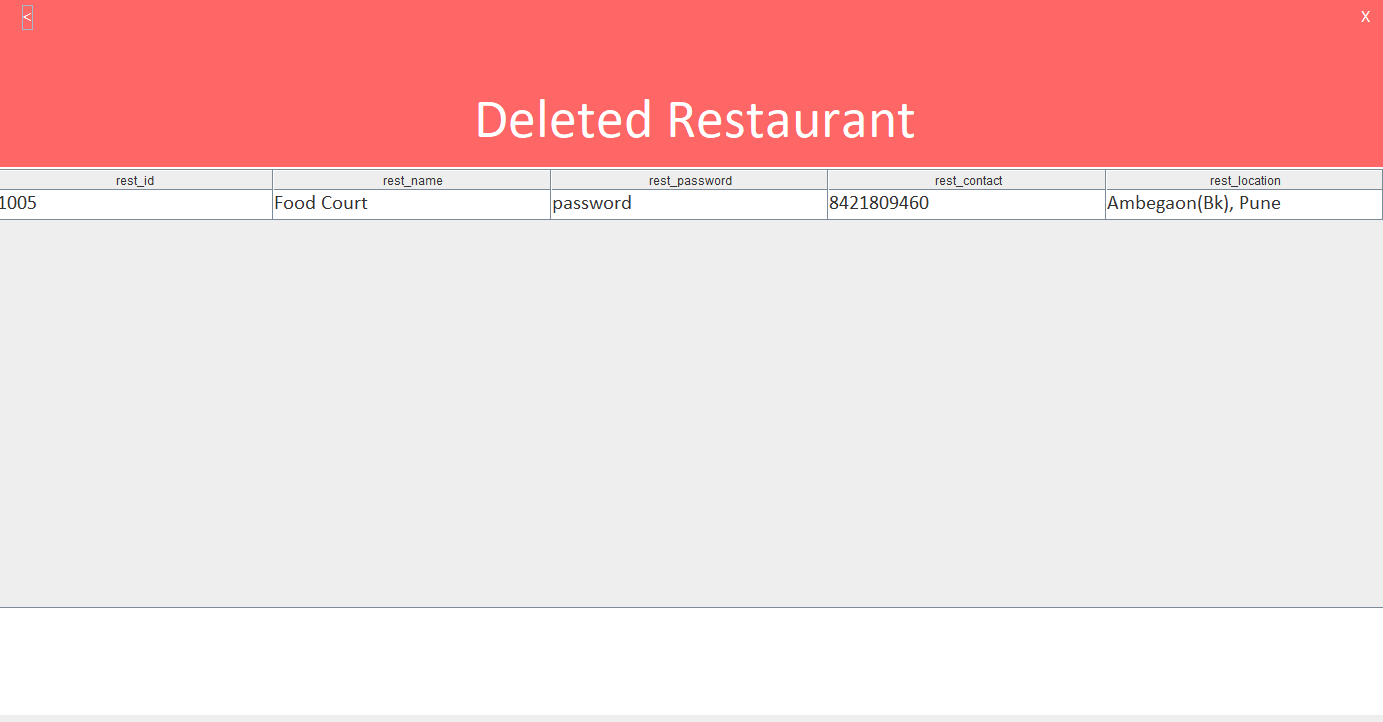


**List of Restaurants**

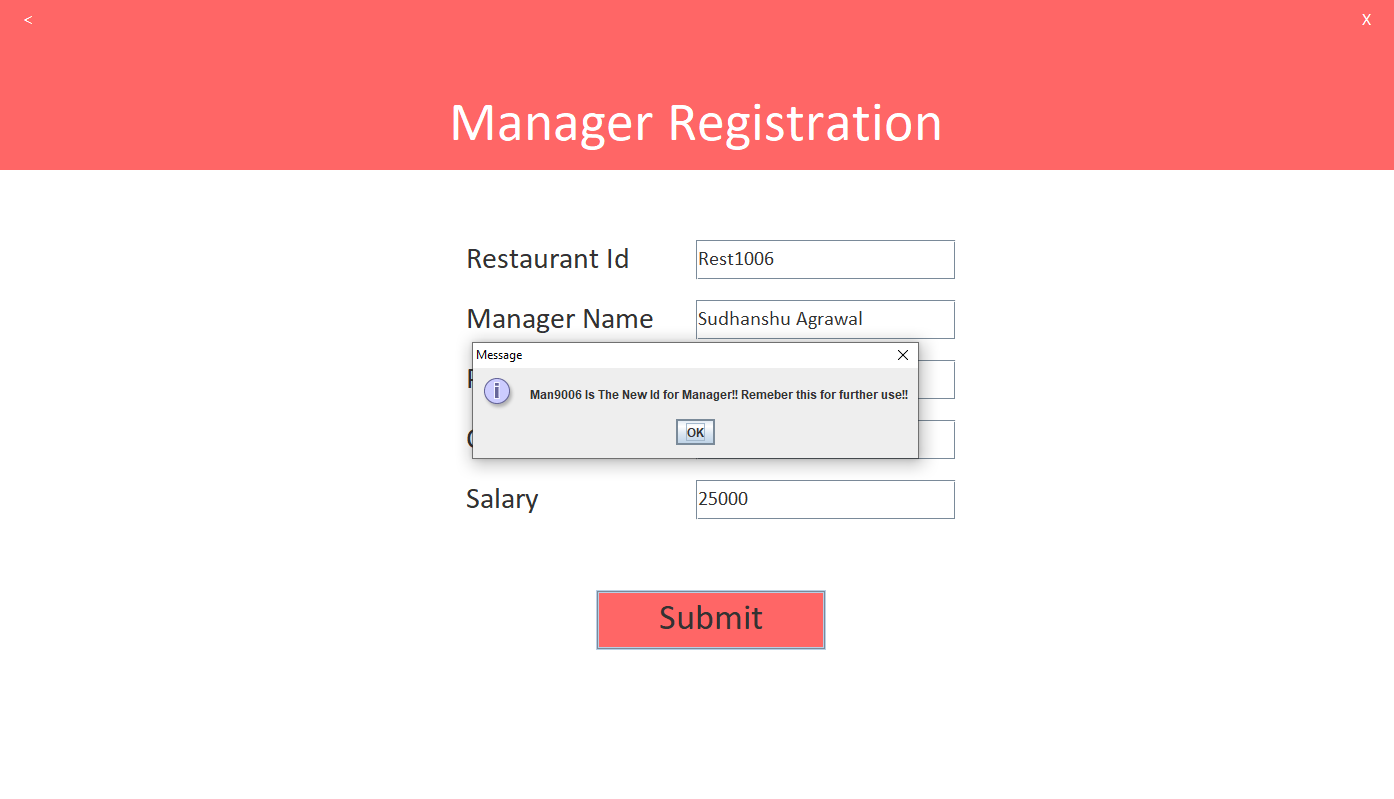
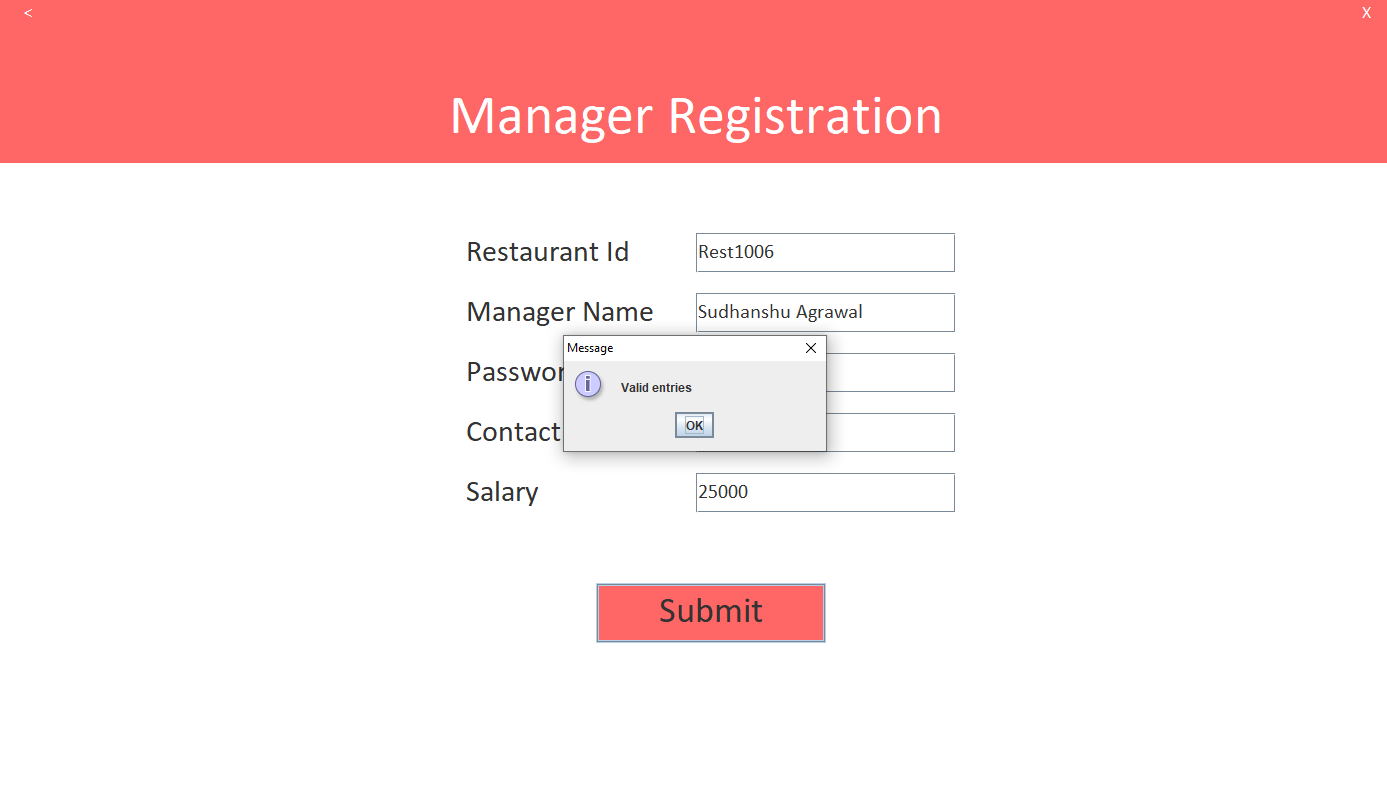
**Adding Restaurant**

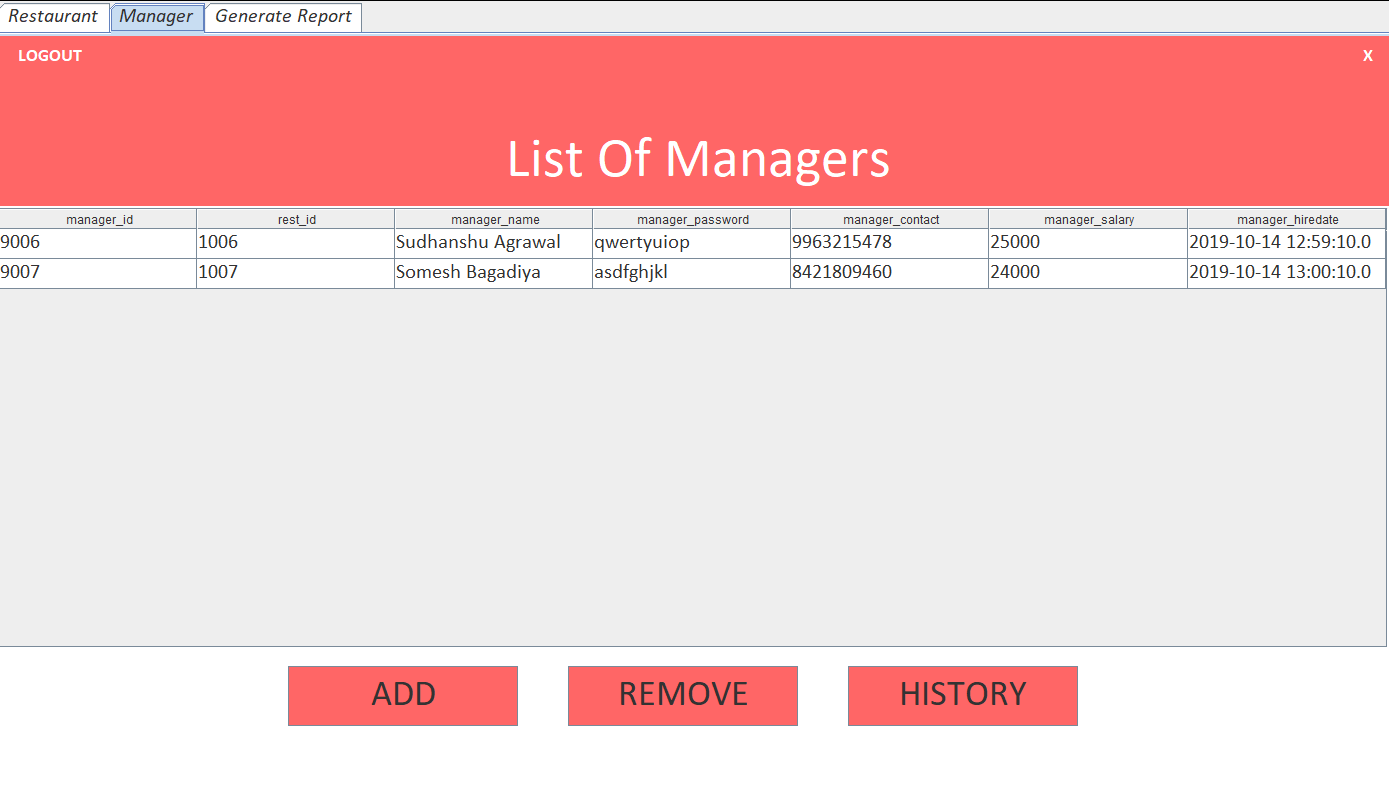
****

****

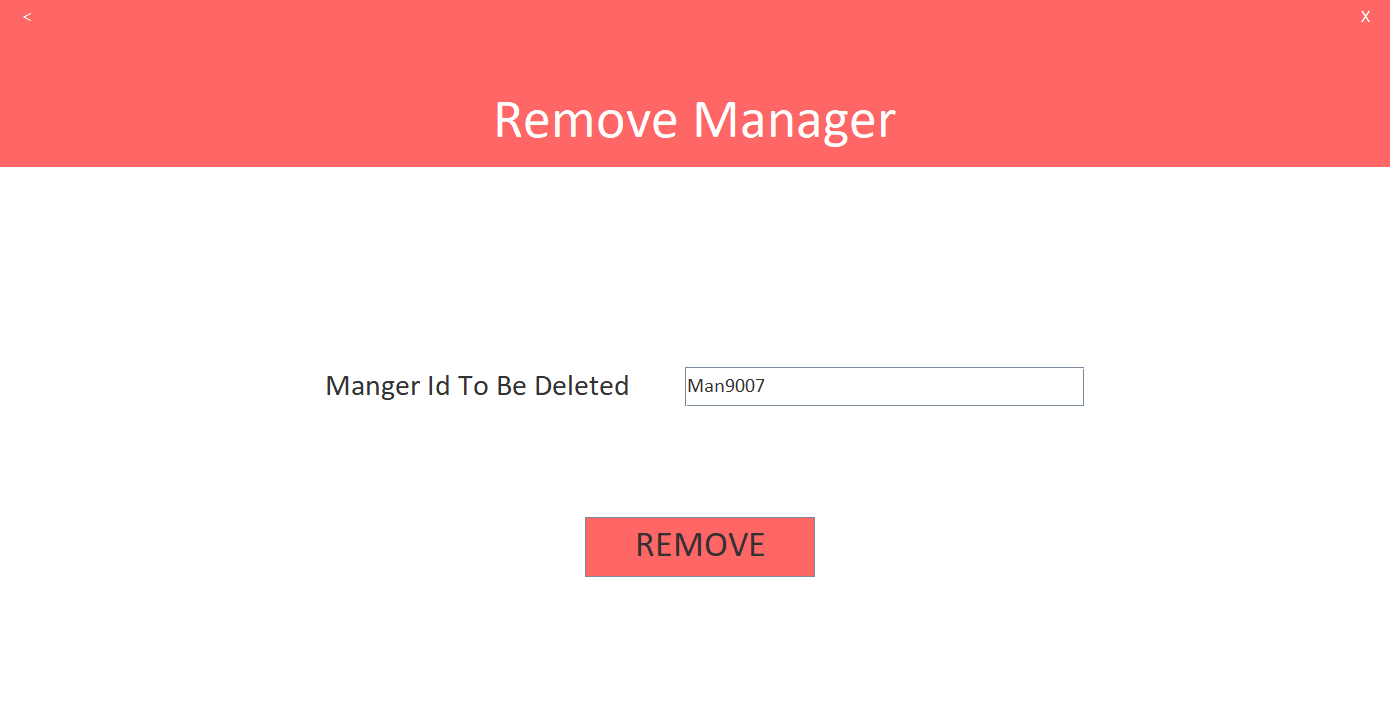
**Removed Restaurant**

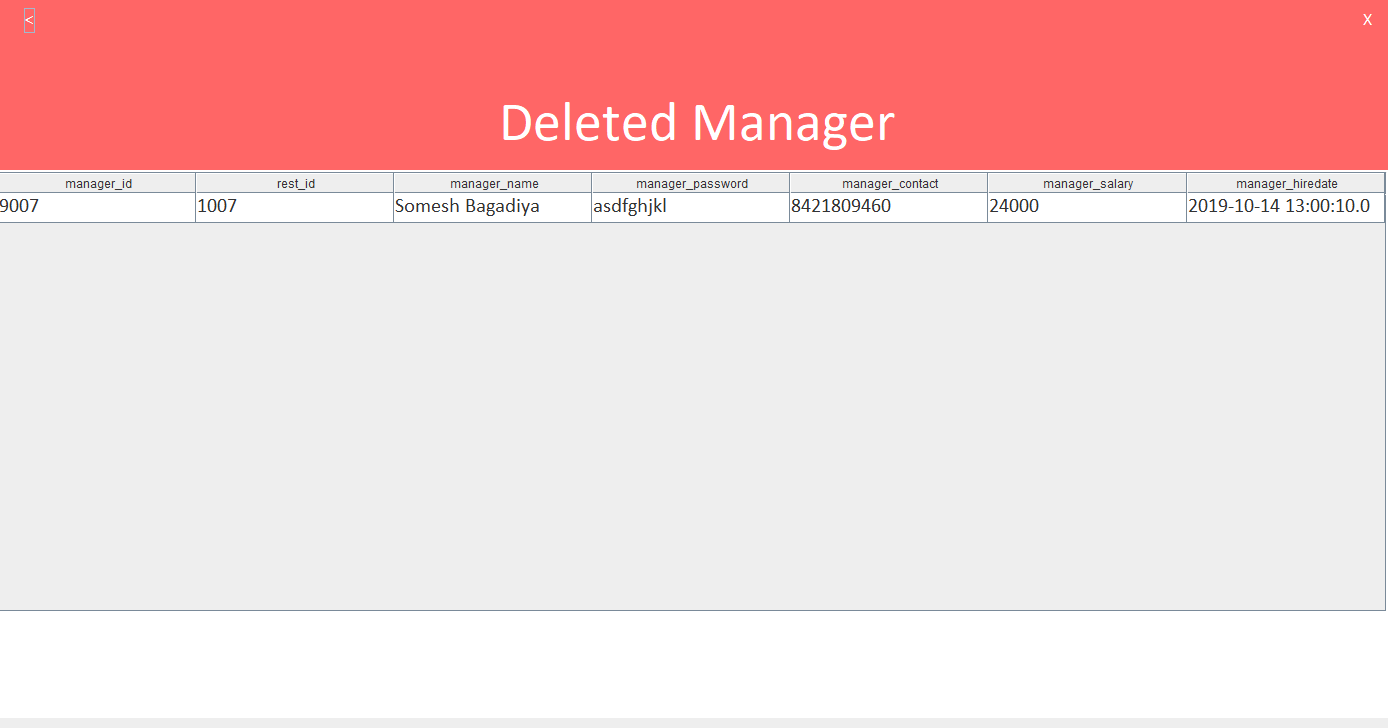
**Adding Manager**

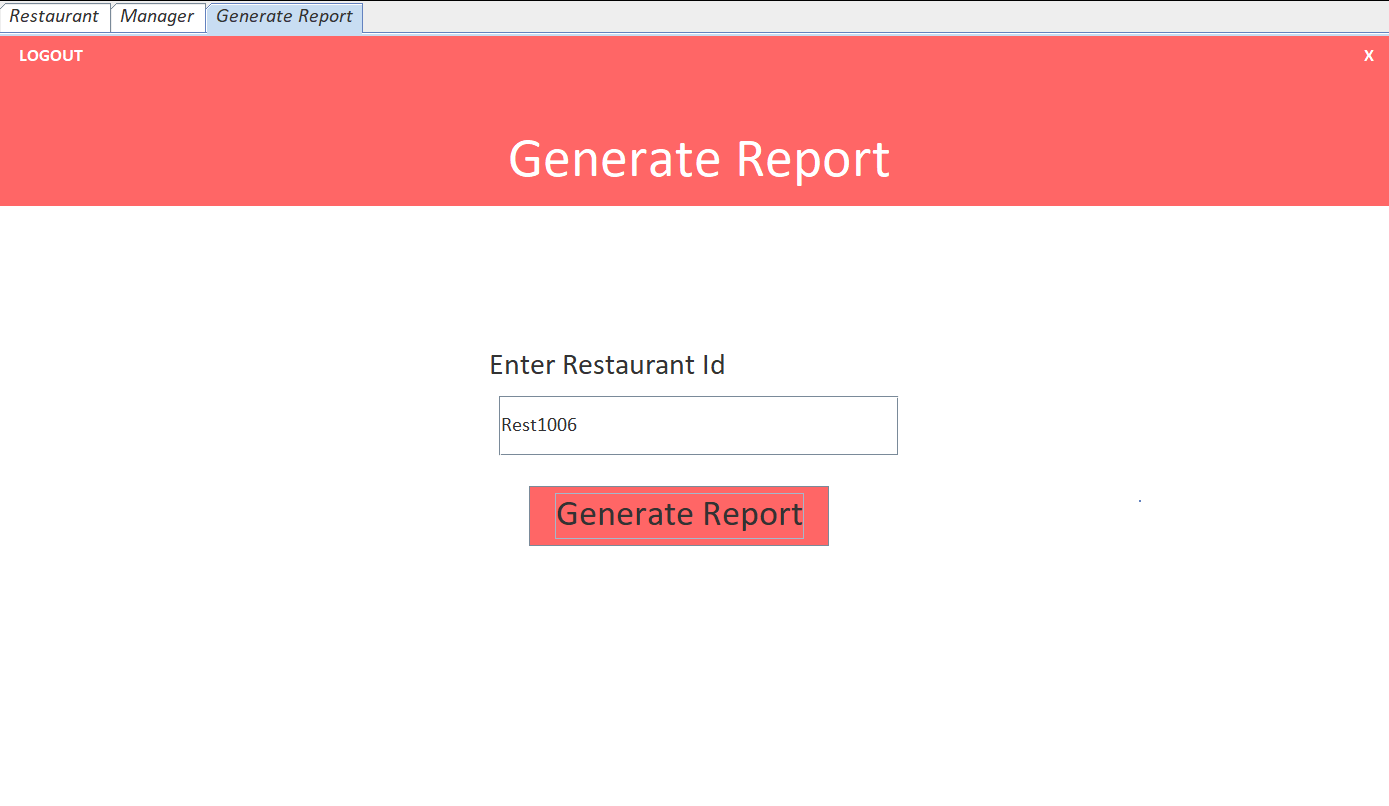
****

**List Of Managers**

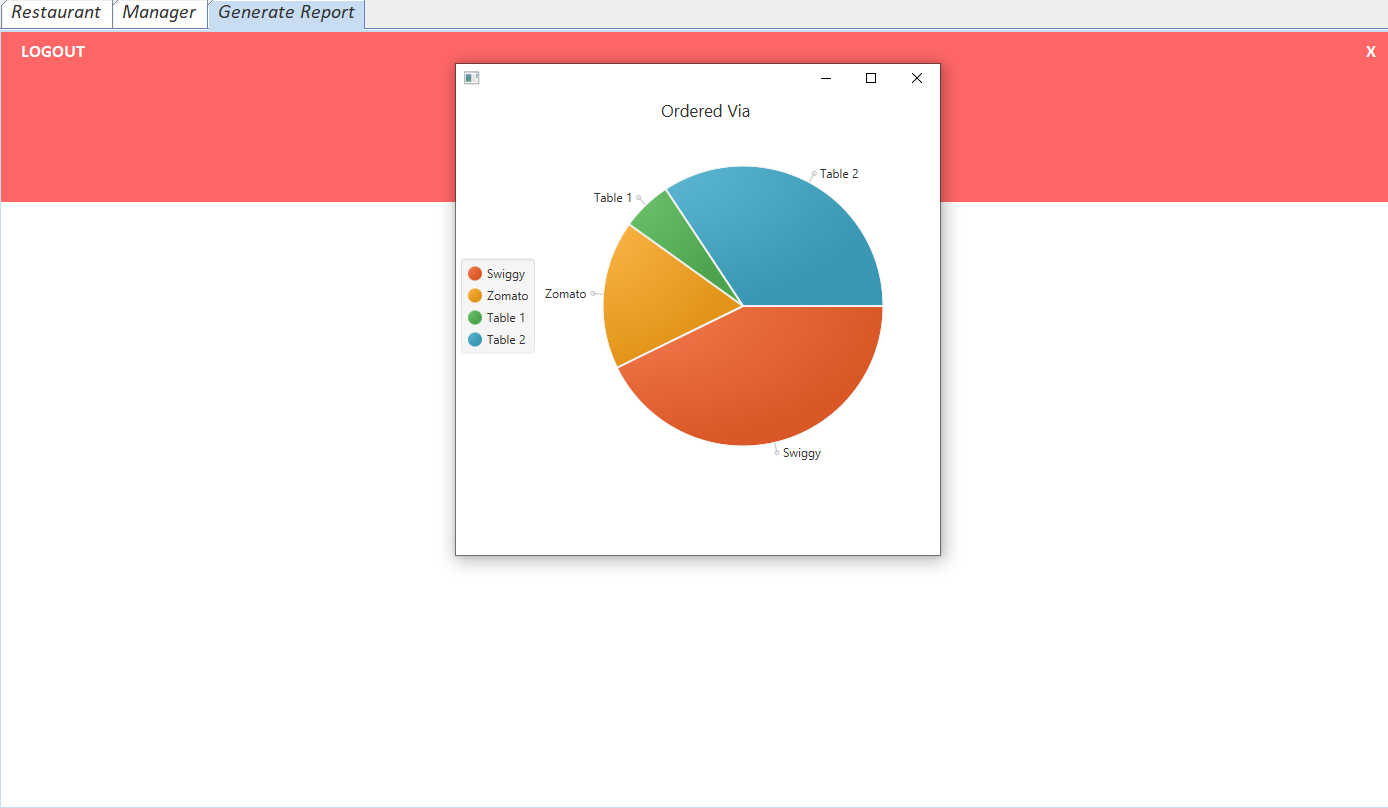
**Removing Manager**

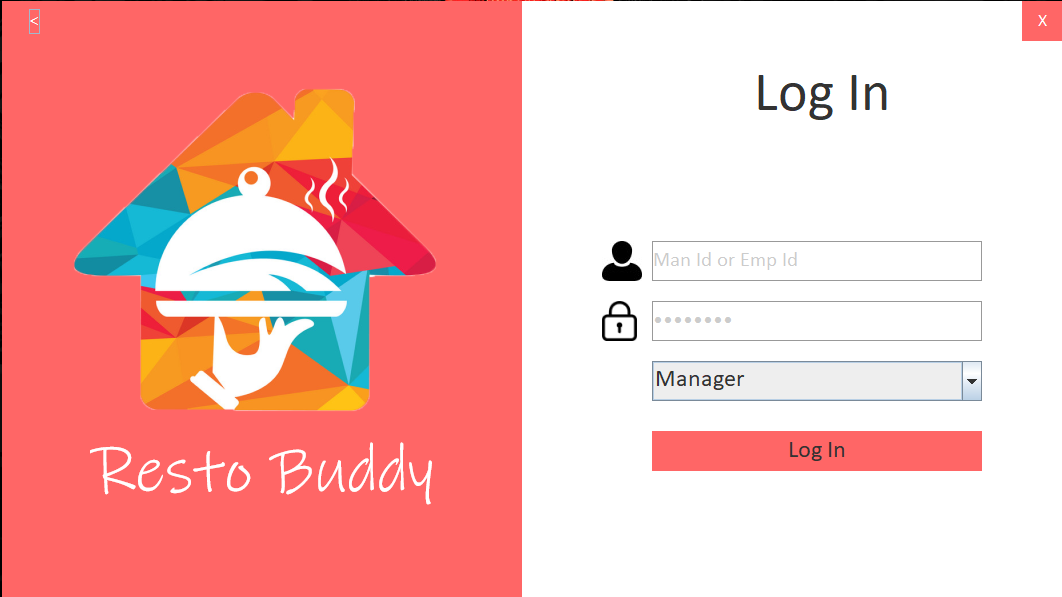
****

**Removed Manager**

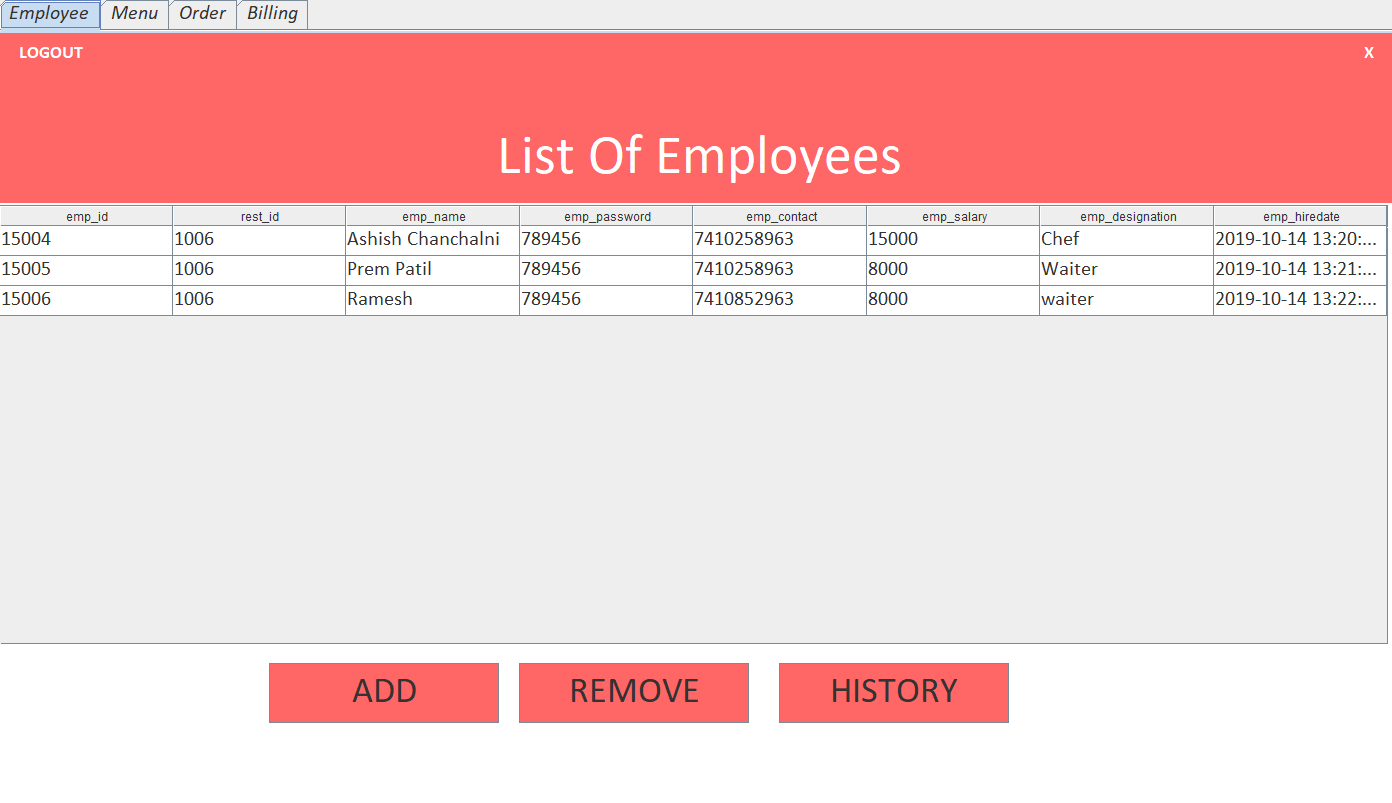
**Generate Report**

**Pie Chart**

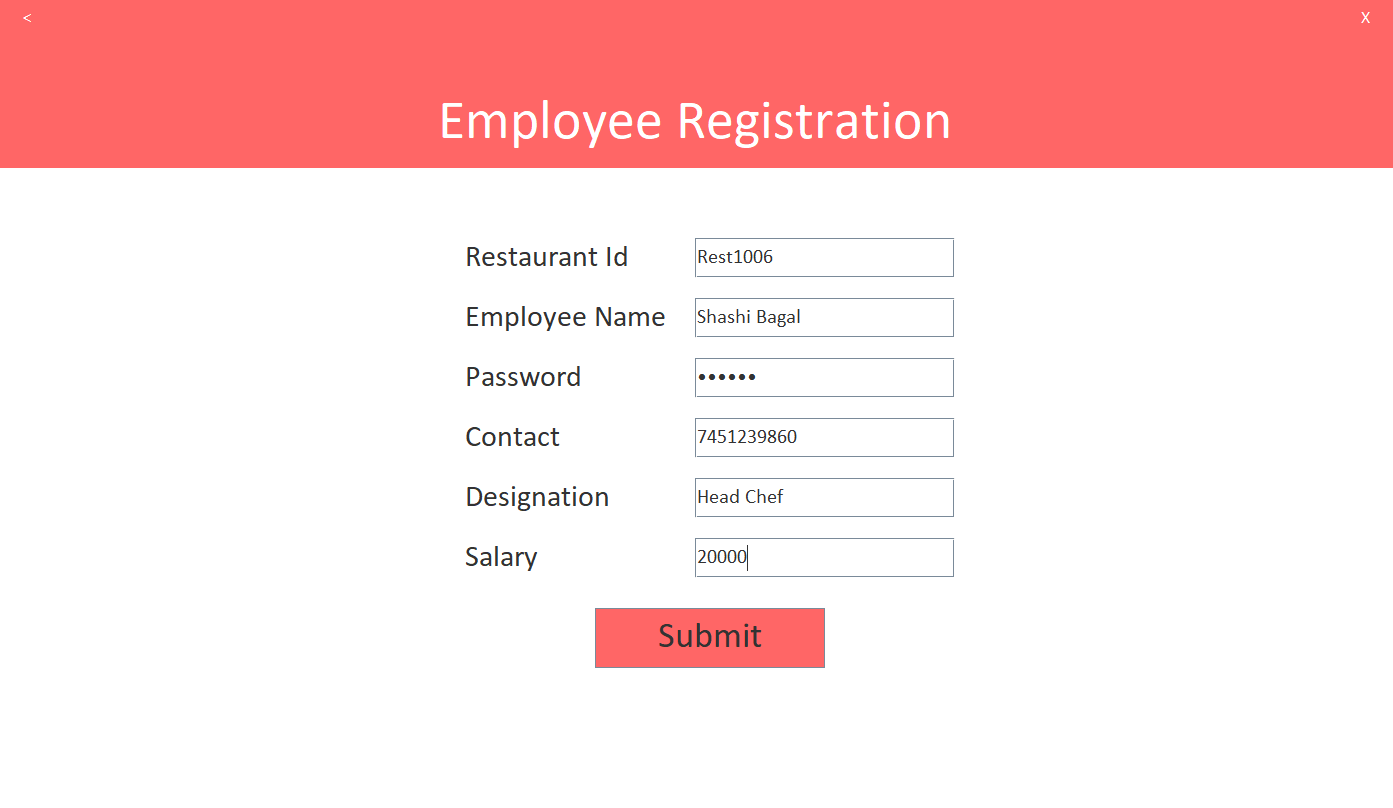
****

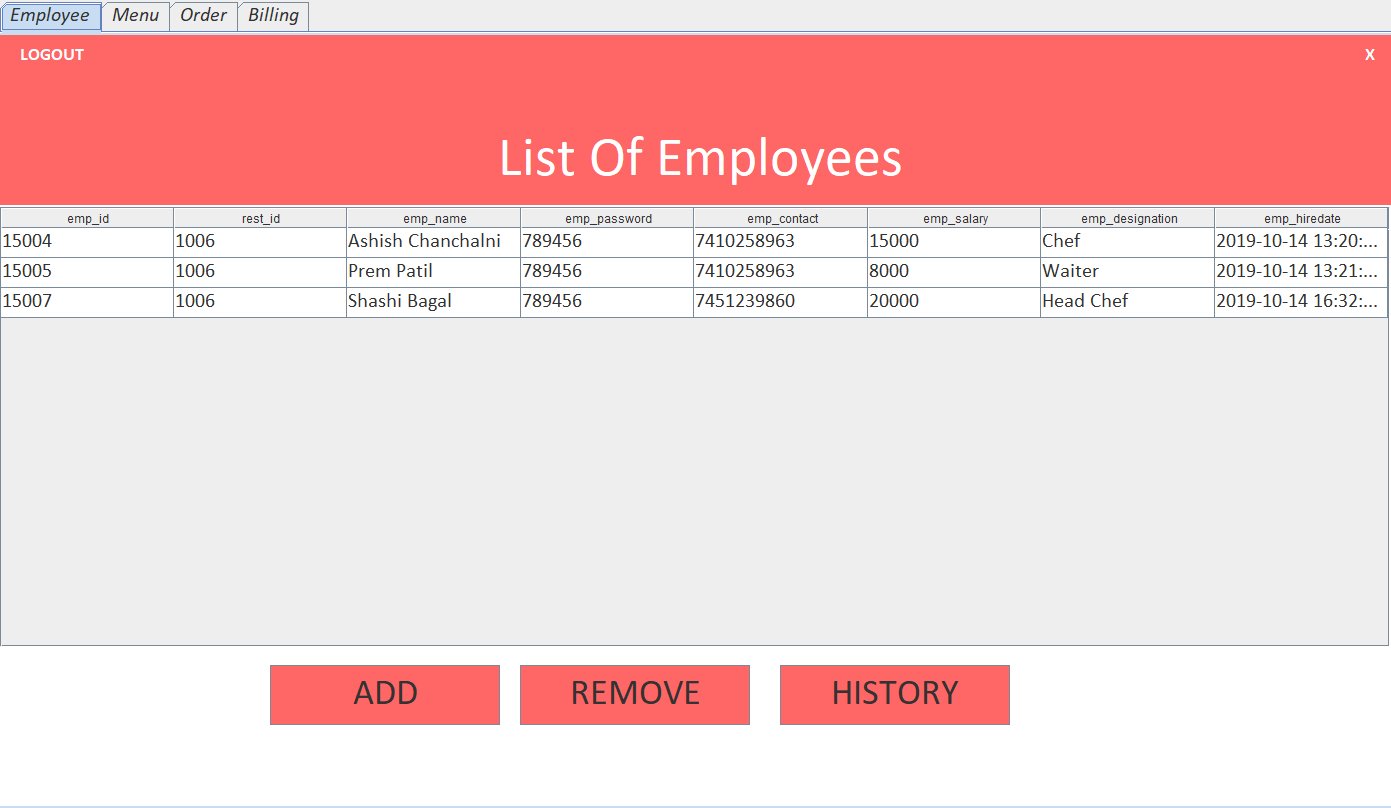
**Manager Sign In**

**List Of Employee**

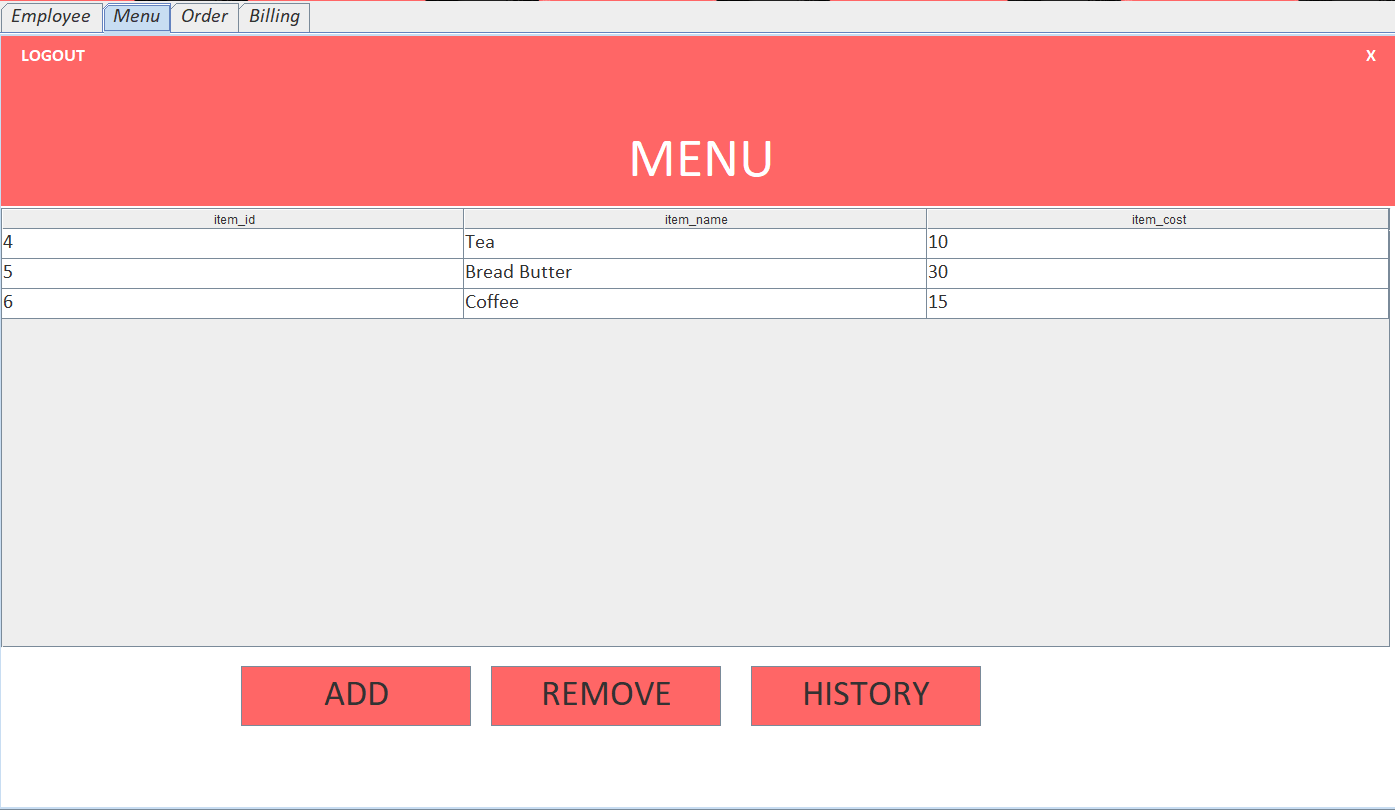
****

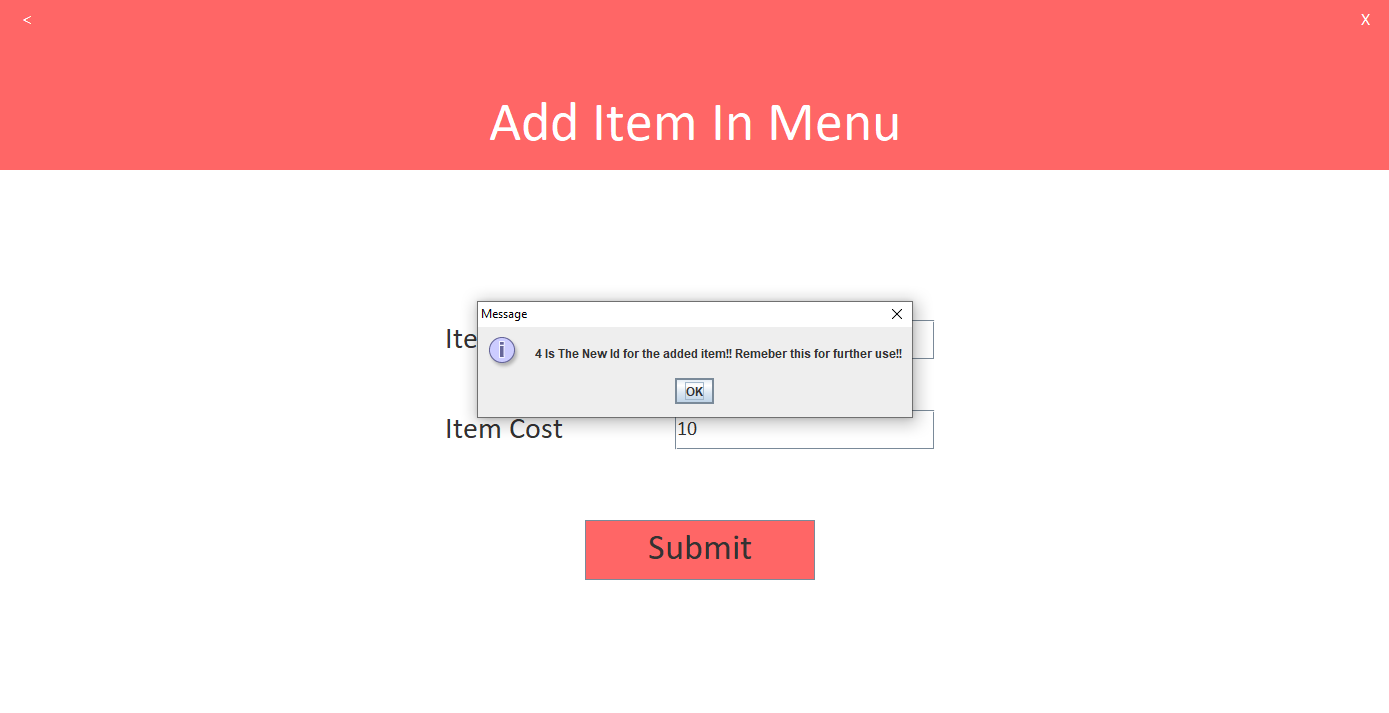
**Add Employee**

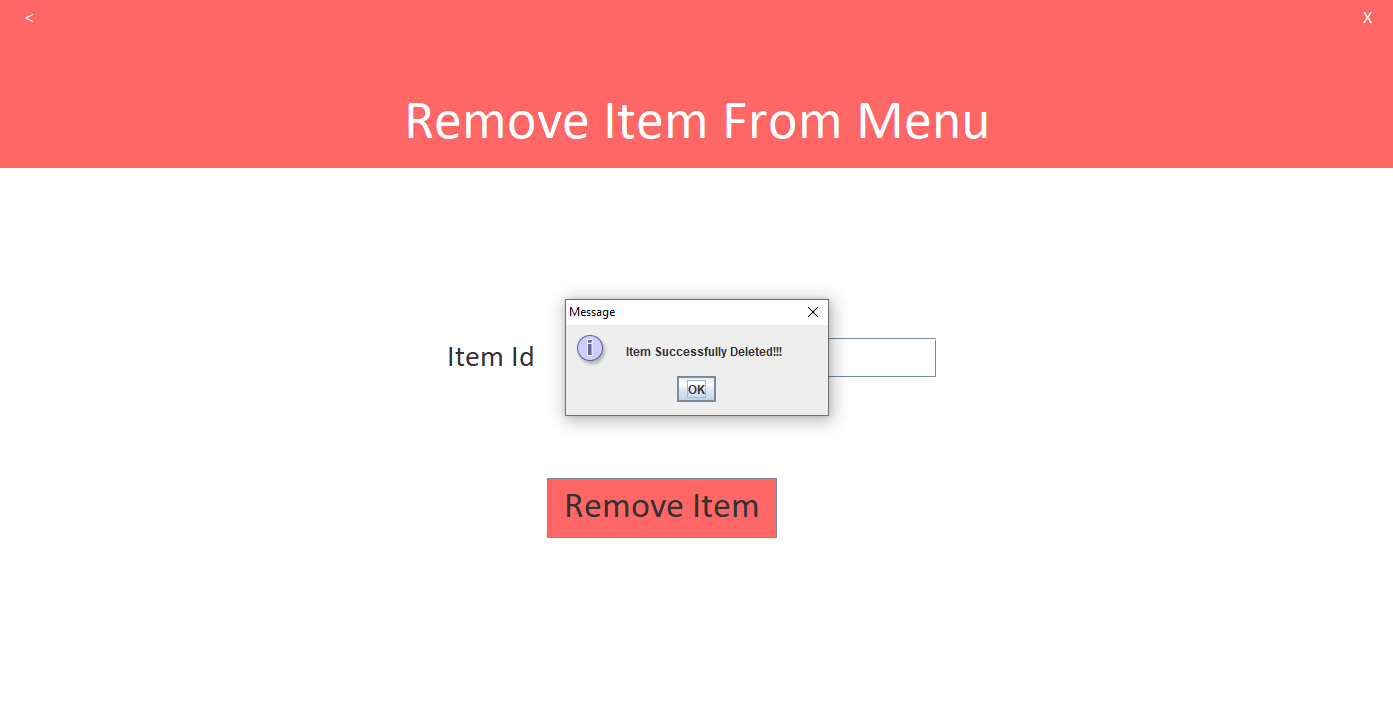
****

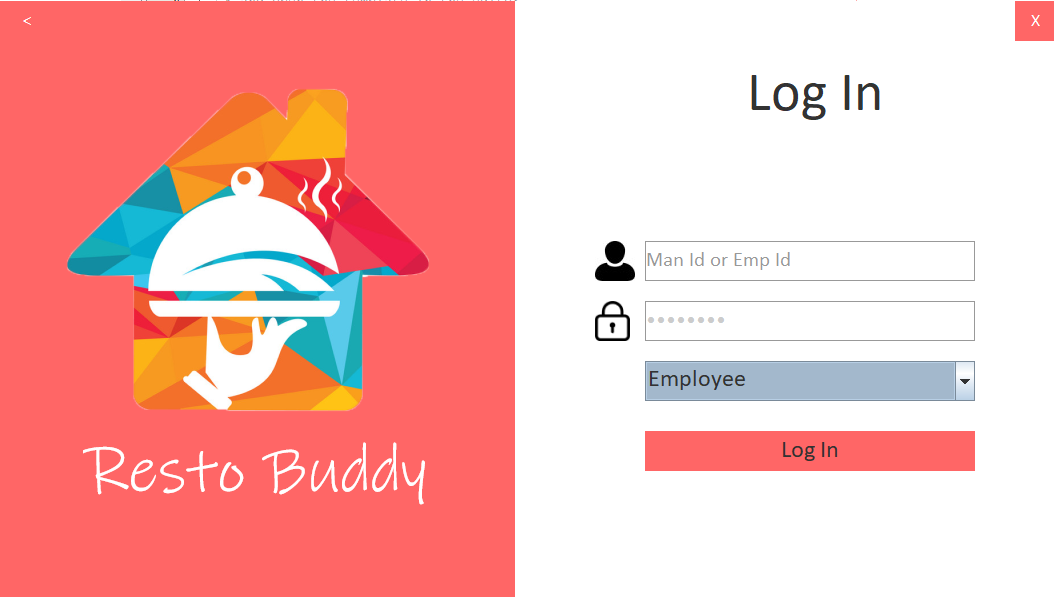
****

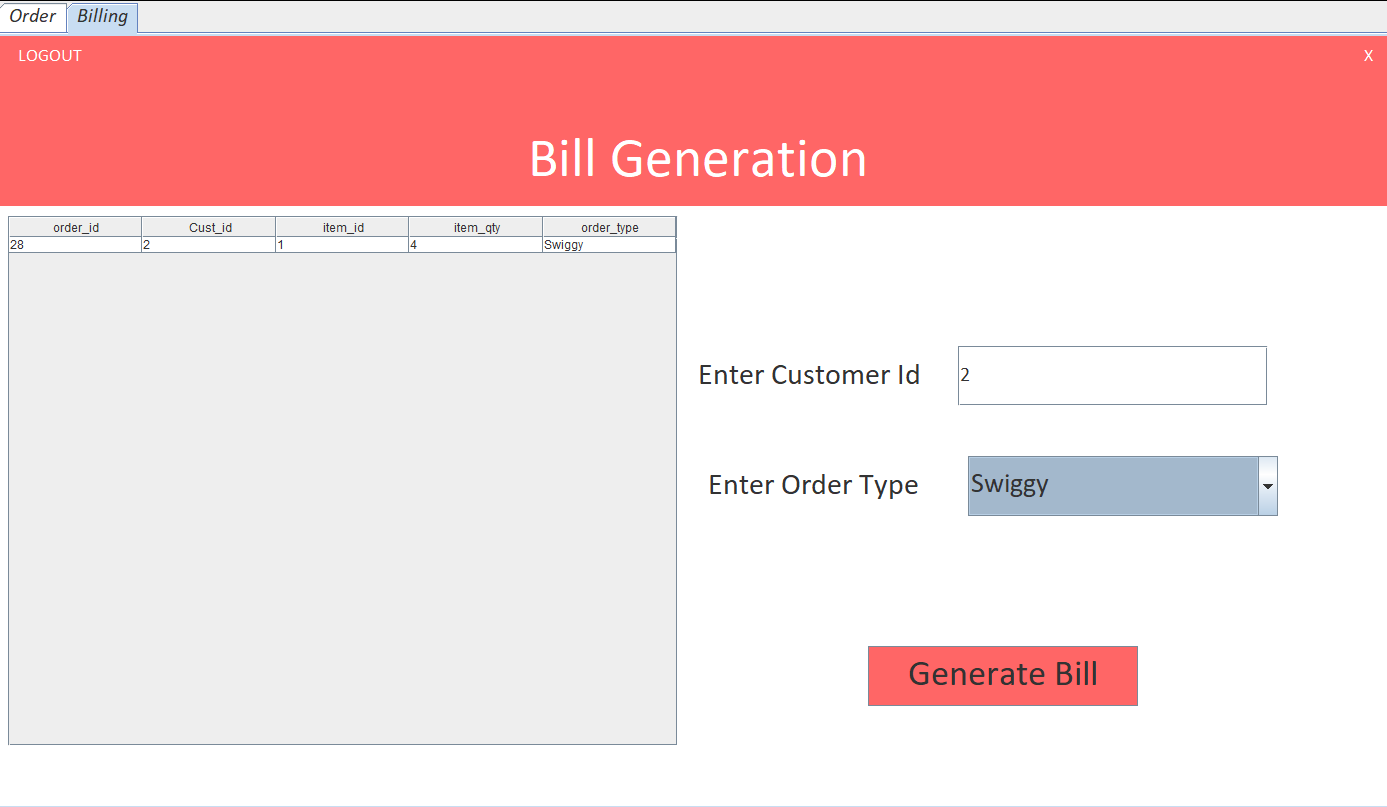
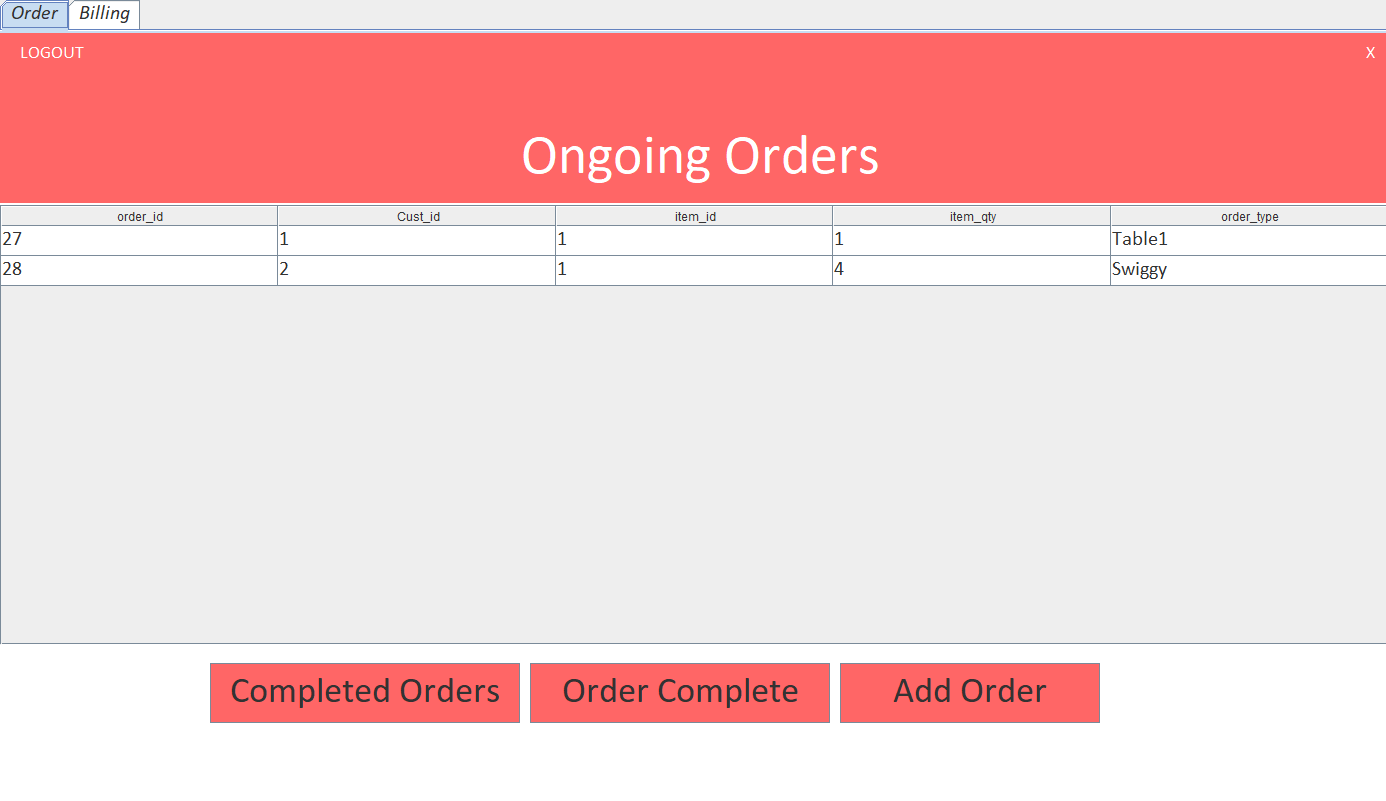
**Removed Employee**

**Menu**

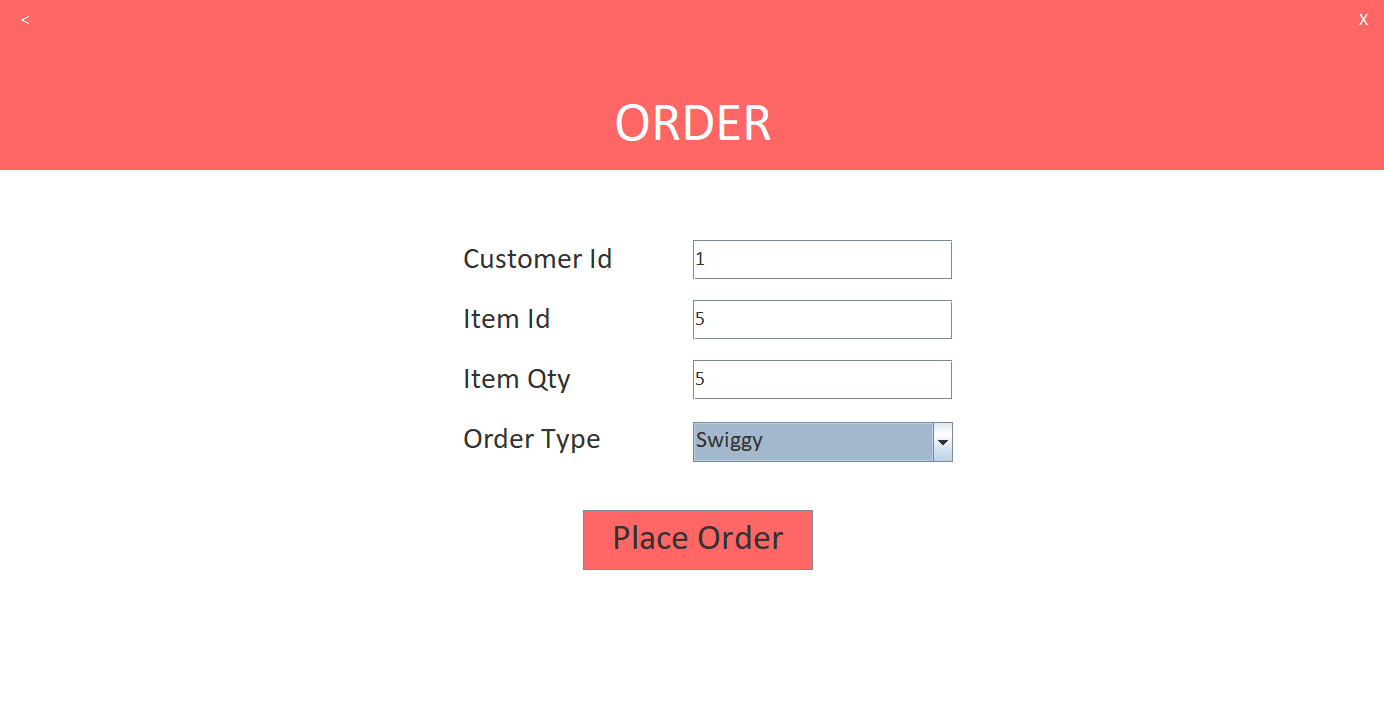
**Add into menu**

**Remove from Menu**

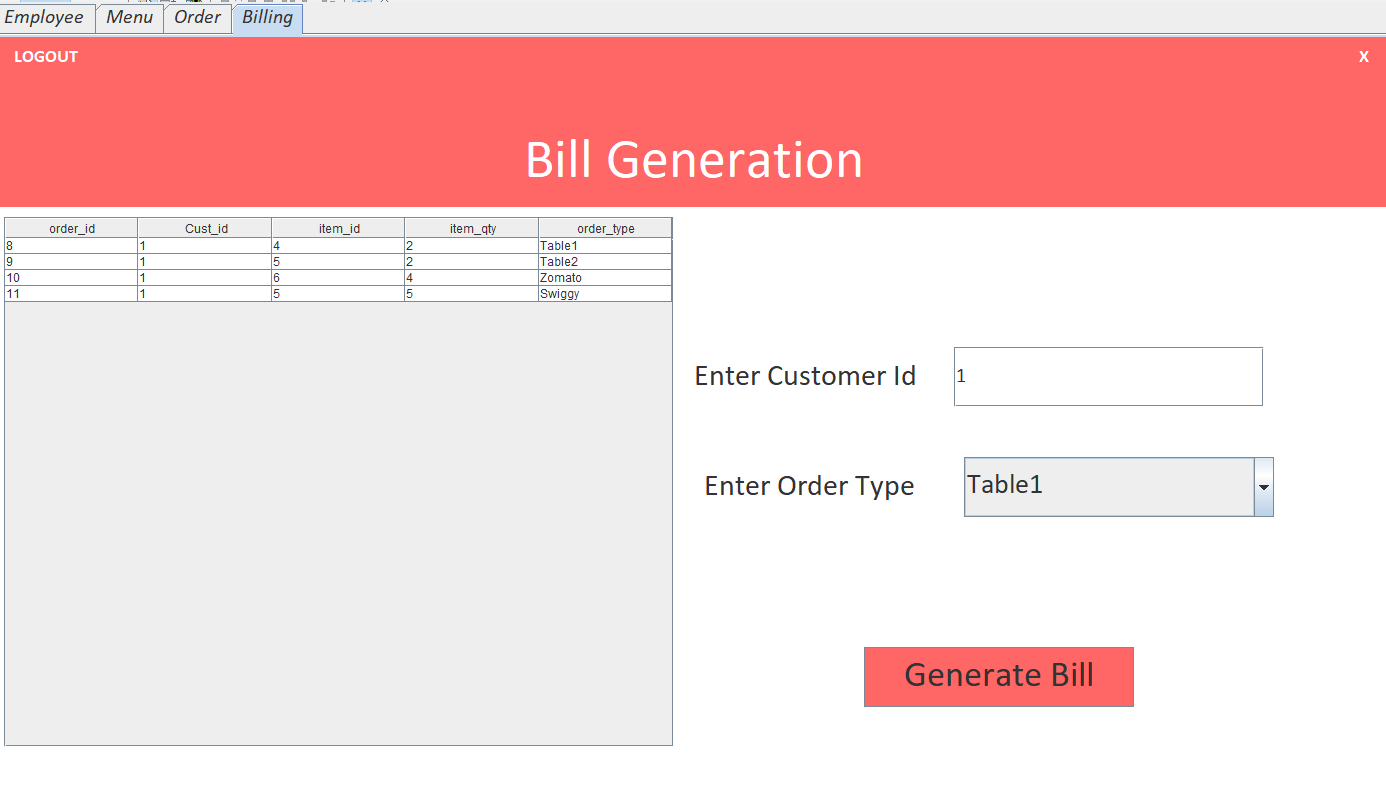
**Employee Sign In**

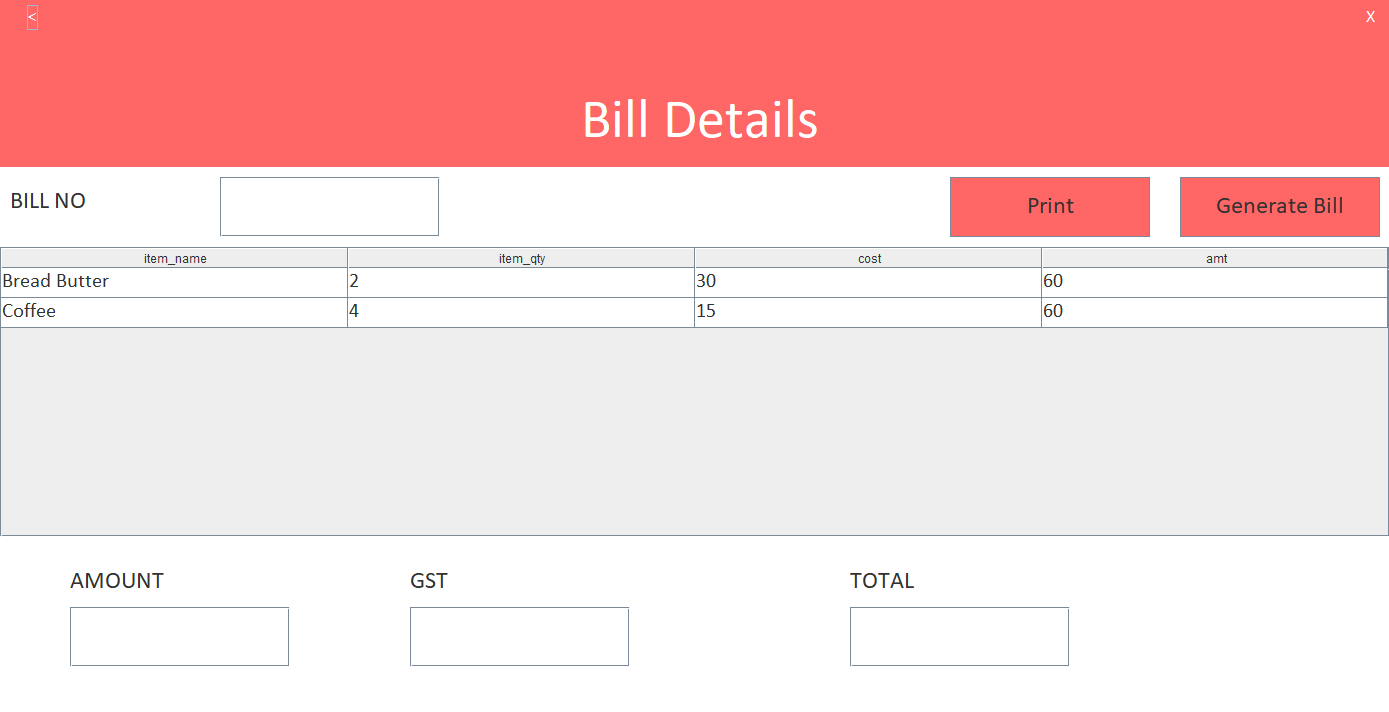
****

**List of Orders**

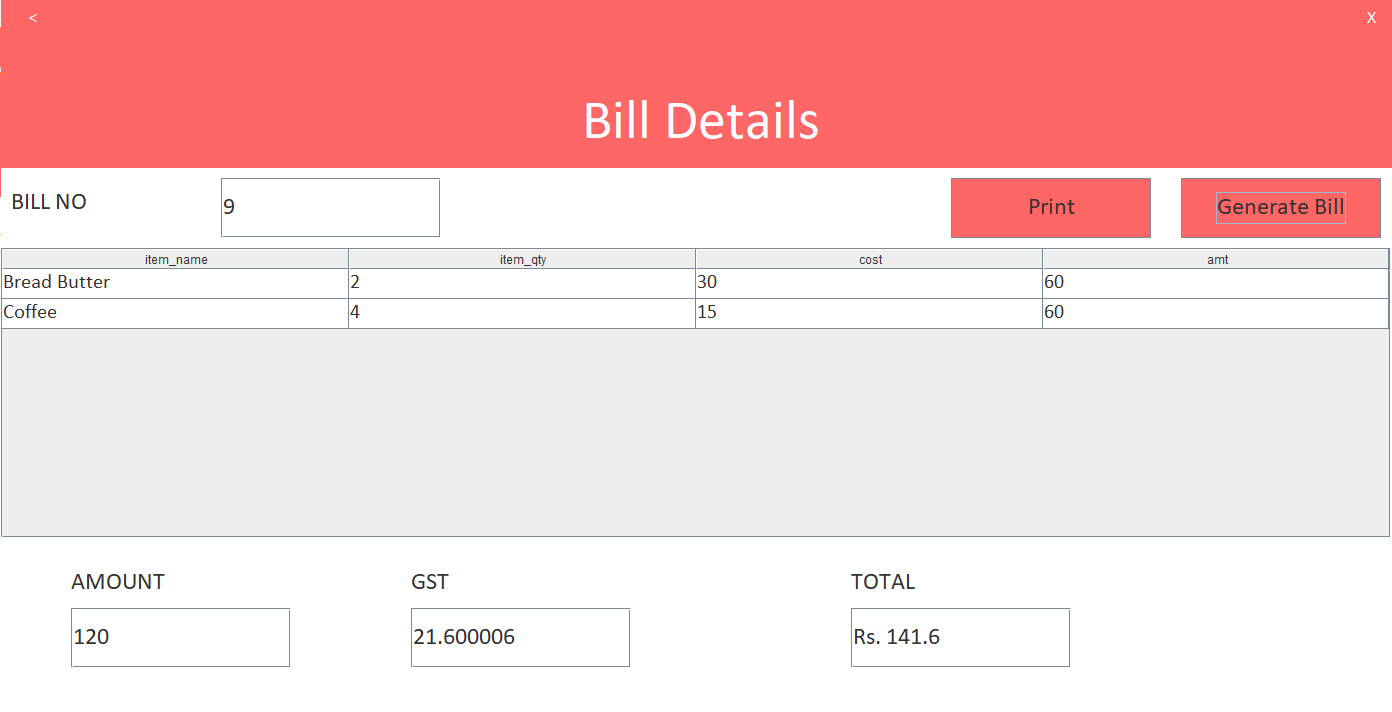
**Place Orders**

**Completed Orders**

**Billing**

**Bill Details**

**Bill**

****

**SAMPLE CODE**

**Connecting Database**

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

String id = jTextField1.getText();

id=id.substring(3);

String password = String.valueOf(jPasswordField1.getPassword());

Object signer=jComboBox1.getSelectedItem();

String signertype=signer.toString();

String rid = SignUp.Fid;

signup type=signertype;

try{

Class.forName("com.mysql.jdbc.Driver");

Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/dbmsmini","root","sudhanshu");

if("Manager".equals(signertype))

{

PreparedStatement stmt=con.prepareStatement("select \* from manager where manager\_id = ? and manager\_password = ? and rest\_id = ?");

stmt.setString(1, id);

stmt.setString(2, password);

stmt.setString(3, rid);

ResultSet rs=stmt.executeQuery();

Boolean alreadyExists = false;

while(rs.next()){

alreadyExists = true;

}

if(alreadyExists == false){

JOptionPane.showMessageDialog(null, "Account Does not Exist ");

jPasswordField1.setText("");

}else{

this.setVisible(false);

new ManagerPage1().setVisible(true);

}

}

else if("Employee".equals(signertype))

{

PreparedStatement stmt=con.prepareStatement("select \* from employee where emp\_id = ? and emp\_password = ?");

stmt.setString(1, id);

stmt.setString(2, password);

ResultSet rs=stmt.executeQuery();

boolean alreadyExists = false;

while(rs.next()){

alreadyExists = true;

}

if(alreadyExists == false){

JOptionPane.showMessageDialog(null, "Account Does not Exist ");

jPasswordField1.setText("");

}else{

this.setVisible(false);

new EmployeePage().setVisible(true);

}

}

con.close();

}catch(HeadlessException | ClassNotFoundException e){

System.out.println(e);

} catch (SQLException ex) {

Logger.getLogger(SignUp2.class.getName()).log(Level.SEVERE, null, ex);

}

}

**Restaurant Registration(Inserting in mysql database using procedure and validation)**

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

// TODO add your handling code here:

int flag=0;

String rest\_Name = jTextField1.getText().toString();

String rest\_Pass = jPasswordField1.getText().toString();

String rest\_Cont = jTextField3.getText().toString();

String rest\_Loca = jTextField4.getText().toString();

if(rest\_Name.equals(""))

JOptionPane.showMessageDialog(null,"Restaurant Name cannot be Empty.");

if(rest\_Pass.equals(""))

JOptionPane.showMessageDialog(null,"Password cannot be Empty.");

if(rest\_Cont.equals(""))

JOptionPane.showMessageDialog(null,"Conatct Details cannot be Empty.");

if(rest\_Loca.equals(""))

JOptionPane.showMessageDialog(null,"Restaurant Location cannot be Empty.");

String check = "[0-9]+";

if(!rest\_Cont.matches(check))

JOptionPane.showMessageDialog(null,"Enter a valid Contact Number.");

else

{

JOptionPane.showMessageDialog(null,"Valid entries");

flag=1;

}

if(flag==1)

{

String name = jTextField1.getText();

String password = String.valueOf(jPasswordField1.getPassword());

String contact = jTextField3.getText();

String location = jTextField4.getText();

try{

Class.forName("com.mysql.jdbc.Driver");

Connection con=DriverManager.getConnection(

"jdbc:mysql://localhost:3306/dbmsmini","root","sudhanshu");

String query = "{call resreg(?,?,?,?)}";

CallableStatement stmt = con.prepareCall(query);

stmt.setString(1, name);

stmt.setString(2, password);

stmt.setString(3, contact);

stmt.setString(4, location);

ResultSet rs = stmt.executeQuery();

String sql = "SELECT rest\_id from restaurant order by rest\_id desc limit 1";

rs = stmt.executeQuery(sql);

rs.next();

int i=rs.getInt(1);

JOptionPane.showMessageDialog(null, "Rest" + i + " Is The New Id for Registered Restaurant!! Remeber this for further use!!");

con.close();

}catch(HeadlessException | ClassNotFoundException | SQLException e){

System.out.println(e);

}

this.setVisible(false);

new OwnerPage1().setVisible(true);

}

}

**Displaying Table(show table in GUI)**

public void fetch1(){

String rid = SignUp.Fid;

try{

Class.forName("com.mysql.jdbc.Driver");

Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/dbmsmini","root","sudhanshu");

PreparedStatement stmt=con.prepareStatement("select \* from Employee where rest\_id = ?");

stmt.setString(1, rid);

ResultSet rs=stmt.executeQuery();

jTable1.setModel(DbUtils.resultSetToTableModel(rs));

}catch(Exception e){

JOptionPane.showMessageDialog(null,e);

}

}

**CONCLUSION**

In today’s ever expanding and demanding world there is always a need as well as scope to enhance the capability of individual with the help of technology. In today’s world, where computers are used extensively so as to speed up various processes, this project of ours tries to comfort problems faced in maintaining Restaurants and to manage its data through our application. Using this desktop application,owner can manage its restaurants and managers. This project of our is a systematic approach in maintaining data for Restaurant and its related data.The main aim of this system is to provide a platform with functionalities that helps the owner,manager,employees to run a restaurants effortlessly.

Due to this project, we were able to learn many things in MYSQL which is used for handling the database in backend, also in JAVA and NETBEANS. This project also gave us the experience of working in group and securing nice coordination among all of the partners.

**REFERENCES**

**Online References**

1) www.mysql.org

2) stackoverflow.com

3) www.oracle.com (Java software download).

**Books Reference**

1)Nageshwar Rao - Core Java.

2)Bert Bates, Head First Java 2nd Edition

3)Kathy Sierra - Head First Java.

4)Erich Gamma, Richard Helm - Gang of Four Design Pattern java.