CHAPTER 1

INTRODUCTION

Database management system or DBMS is a software designed to assist in managing and utilizing large collection in data, and the need of such system as well as their use is growing rapidly. The Alternative to using a DBMS is used to hoc approaches that do not carry over from one application to another.

1.1 Overview of DBMS

A database management system stores data in such a way that it becomes easier to retrieve, manipulate, and produce information. Database is a collection of related data and data is a collection of facts and figures that can be processed to produce information. The area of the Database Management system is microcosm of computer science in general. The issues addressed and the technique used to span a wide spectrum, including languages, object orientation and other programming paradigm ,compilation, operating system, concurrent programming ,data structures, algorithms, theory, parallel and distributed systems user interface, expert systems and artificial intelligence, statistical techniques, and dynamic programming.

1.2 History

From the earliest days of computers, storing and manipulating data have been a major application focus. The first general purpose DBMS was designed by Charles Bachman at General Electric in the early 1960s was called The Integrated Data Store. It formed the basis for the network data model, which was standardized by the Conferences on Data Systems Languages(CODASYL) and strongly influenced database systems through the 1960s.Bachman was the first recipient of ACM's Turing Award(the computer science equivalent of a Nobel prize) for work in the database Area; he receives the award in 1973.

In the late 1960's IBM developed the Information Management System (IMS) DBMS, used even today in many major installations. IMS form, the basis for an alternative data representation framework called the hierarchical data model. The SABRE system for making airlines reservation was jointly developed by American

Airlines and IBM around the same time, and it allowed several people to access the same data through.

An interesting phenomenon is the emergence of several enterprise resource planning (ERP) and management resources planning (MRP) packages, which add a substantial layer of application –oriented features on top of a DBMS widely used packages include systems from Bann, Oracle, peopleSoft, SAP and Siebal.

1.3 Applications of DBMS

Nowadays DBMS are used in almost all the areas ranges from science, engineering, medicine, business, industry, government, art, entertainment, education and training.

DBMS in the field of Library Management System

There are thousands of books in the library so it is very difficult to keep records of all the books in a copy or register. DBMS is used to maintain all the information related to book issue date, name of book, author and availability of book.

DBMS in the field of Banking

Another major application is the banks. Thousands of transactions through daily can do this without going bank. To manage such huge transactions is just because of DBMS that manages bank transactions.

DBMS in the field of Universities and colleges

Examinations are done online today and universities and colleges maintain all these records through DBMS. Student's registrations details, results, courses and grades all the information are stored in database.

DBMS in the field of Telecommunications

Any telecommunication company cannot even think about their business without DBMS.DBMS is required for these companies to store the call details and monthly postpaid bills.

DBMS in the field of Online Shopping

Online shopping has become a big trend of these days. No one wants to go to shops and to waste his time. Everyone wants to shop from home. So all these products are added and sold only with the help of DBMS. Purchase information, invoice and payment all these are done with the help of database.

DBMS in the field of Military

Military keeps records of millions of soldiers and it has millions of files that should be kept secure and safe. As DBMS provides a big security assurance to military information so it is widely used in militaries. One can easily search for all the information about anyone within seconds with the help of DBMS.

1.4 Problem Statement

Restaurant management database system includes the management of front office, HR, finance, inventory, material management, quality management, security, housekeeping,booking and reservation, security etc. At all these situations, the restaurant management system database project can be used which helps in maintain all these information just in one stretch. This project defines scope of Restaurant Management System (RMS), Development schedule, Development process, techniques, tools and platform with reasoning. RMS is the system that provides an easy management and maintenance of relevant activities regarding a restaurant. It also maintains record of staff,order, inventory and stock management. It also deals with the financial management of RMS. It also provides the online room and table reservation.

1.5 Objectives of the project:

The main objectives are,

- To show how restaurant management system is developed to manage the database of any restaurant.
- Creating a database of various available service in restaurant.
- Evaluate the need for a management perspective to better serve customers and improve outcomes to the development ofthe business.
- List all the services available in the restaurant.

1.6 Organization of the report

Chapter 1:

It deals with the Introduction and organization of the project report.

Chapter 2:

It discusses the Specific to the problem-Requirement Analysis-Design.

Chapter 3:

It discusses the Design and Implementation.

Chapter 4:

It gives information about the snapshot and results.

Chapter 5:

It includes conclusion and future scope.

Chapter 6:

It gives the references of the project.

Chapter 2

REQUIREMENT ANALYSIS AND DESIGN

This chapter includes the requirements for the development of the project. These requirements describes high level system design, software requirements etc.

2.1 Basic Definition

Restaurant Management System is mainly designed to maintain details of the restaurant and also employee information record.

2.2 Advantages

- This system decreases the chance of error.
- This system requires less time for completion of any work.
- It is used to maintain the information such as menu and employee information.
- Work load and man power is very fast.

2.3 Requirement Analysis

SOFTWARE

Windows10 is a personal computer operating system developed and released by Microsoft as part of the windows NT family of operating systems.it was released on July 29,2015. It is the first version of windows that receives ongoing feature updates. Devices in enterprise environments can receive these updates at a slower pace, or use long-term support milestones that only receive critical updates, such as security patches etc.

BACK END

Interactive enterprise manager screens display details about a SQL statement. This includes the SQL text, Top activity by various dimensions, CPU and wait activity over time, key SQL statistics ,and execution plans, SQL profiles and SQL plan

baselines will be displayed if the exist, and a monitored execution is displayed, if available.

FRONT END

Java Swings – Netbeans IDE is an application programming interface (API) for the programming language java, which defines how a client may access a database. It is java based data access technology and uses JDBC for java database connectivity .It is Part of the java standard edition platform, form oracle corporation. It provides methods to query and update data in a database, and is oriented towards relational databases. A JDBC bridge enables connections to any accessible data source in the java virtual machine (JVM) host environment.

Software requirements

- Operating system - Windows 10
- Backend - MySQL
- Front end - Java swings
- Platform - JDBC jar

Hardware components

- Processor - Intel core i3
- Processor speed - 2.1 Ghz
- Ram - 4 GB
- Hard disk - 1 TB

Chapter 3:

Design and Implementation

This chapter describes entities, the attributes and from that how the design has been achieved to provide the ER diagram. It also covers how the schema diagram is evolved.

SCHEMA DESCRIPTION

EMPLOYEE: EID, Name, DOB, Salary, Contact, Gender

MENU: FName, Type, Price

TABL: num, No_of_seats

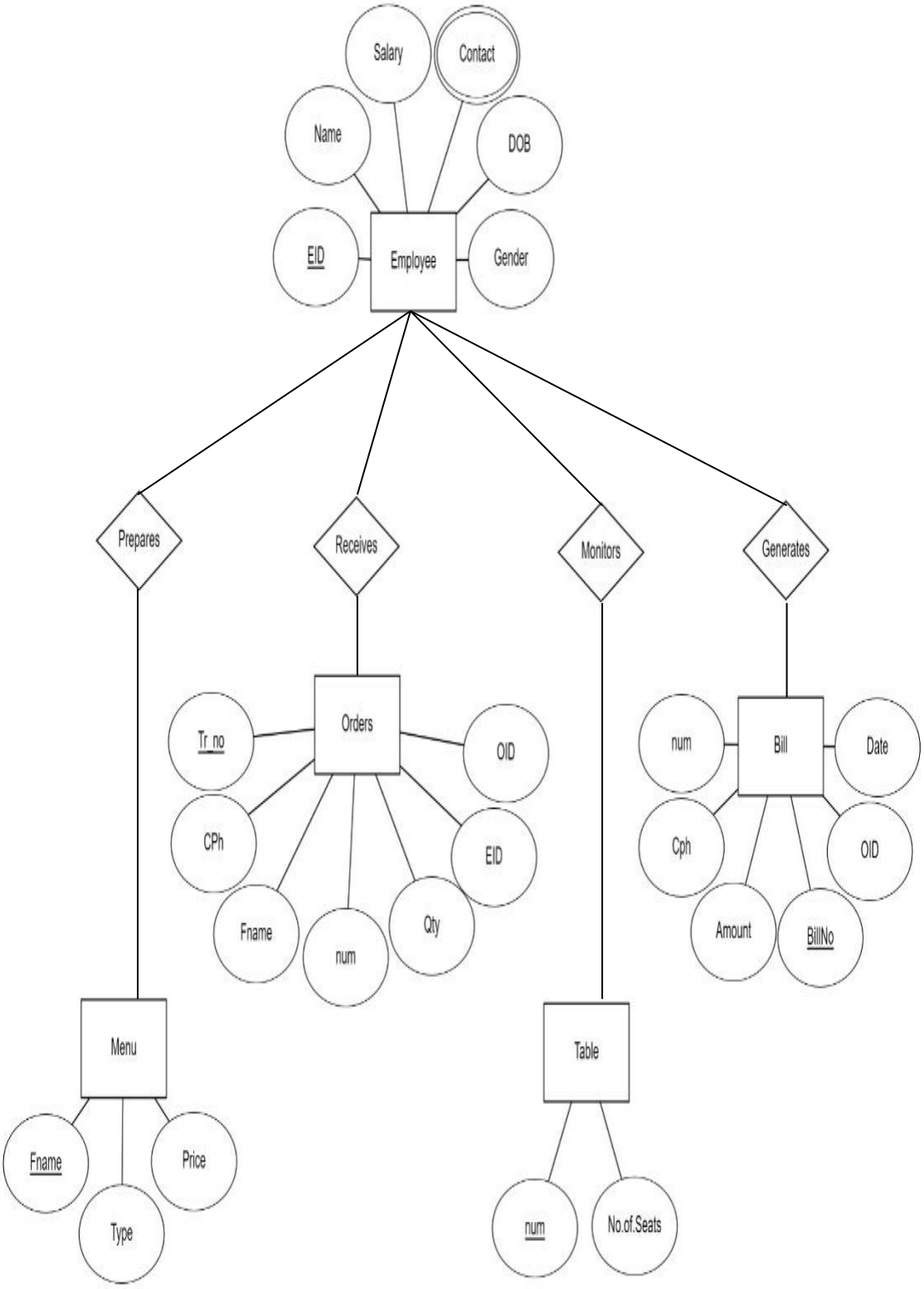
ORD: Tr_No, FName, Qty, CPh, EID, OID, num

BILL: Bill_No, CPh, Amount, Date, OID, num

3.1 ER DIAGRAM

Figure 3.1 gives ER diagram that describes the entities, attributes, relationship between entity and attributes. In this ER diagram we have entities such as employee, orders, menu, table and bill. Relationship is like this:

- Employee is related to menu by the relationship prepares with cardinality ratio M:N
- Employee is related to orders by the relationship receives with cardinality ratio M:N
- Employee is related to bill by the relationship generates with cardinality ratio M:N
- Employee is related to tables by the relationship monitors with cardinality ratio M:N



3.1 ER Diagram

3.2 SCHEMA DIAGRAM

Figure 3.2 gives schema diagram. **Include description like PK, FK etc**

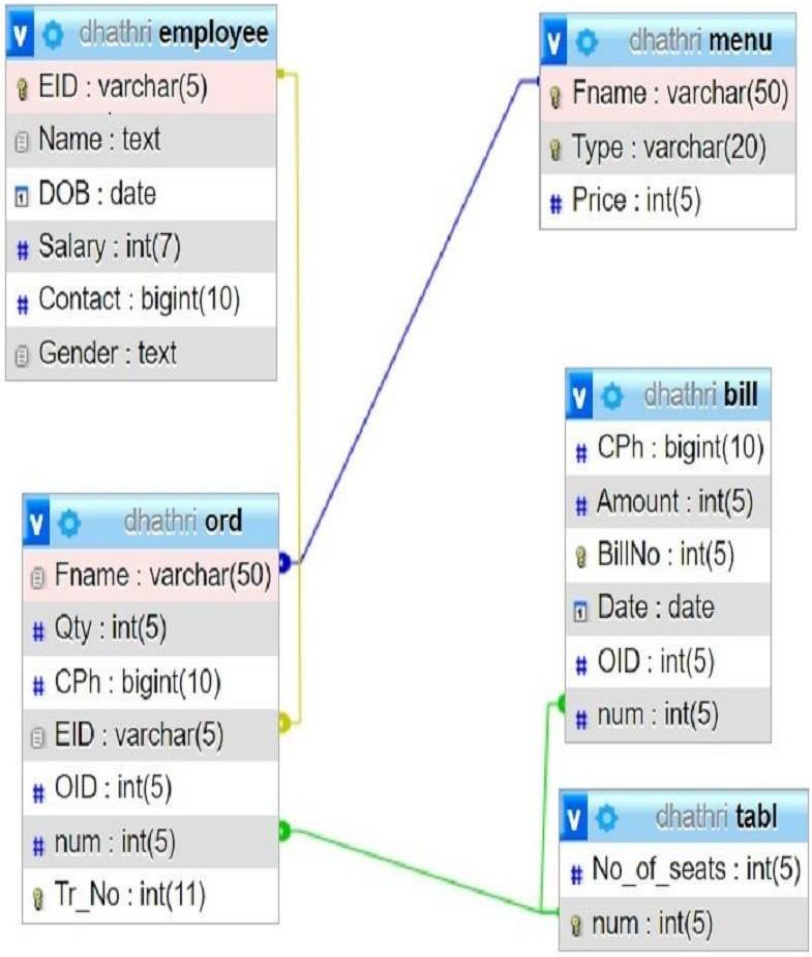


Figure 3.2 gives schema diagram

3.3 PSEUDOCODE

TO CONNECT :

To connect database into the table , following query is used .

This query is used in all pages.

```
public void Connect ()
{   try{
Class.forName("com.mysql.cj.jdbc.Driver");

con=DriverManager.getConnection("jdbc:mysql://localhost:3
306/dhathri","root","oracle");
             } catch (Exception e) {}
    }
```

INESRT:

To insert data into the table , following query is used .

This query is used in all 5 pages.

```
private void
jButton1ActionPerformed(java.awt.event.ActionEventevt) {
try{
        String eid1,name1,salary1,dob1,contact1;
        eid1=eid.getText();
        name1=nametxt.getText();
        salary1=salary.getText();
        dob1=dob.getText();
        contact1=contact.getText();

pst=con.prepareStatement("insert into
employee(eid,name,dob,salary,contact)values(?,?,?,?,?)");
pst.setString(1,eid1);
pst.setString(2,name1);
pst.setString(3,dob1);
pst.setString(4,salary1);
```

```
pst.setString(5,contact1);
pst.executeUpdate();
JOptionPane.showMessageDialog(null, "RecordSaved");
tableUpdate();
eid.setText("");
nametxt.setText("");
salary.setText("");
dob.setText("");
contact.setText("");
eid.requestFocus();
       }
catch(HeadlessException | SQLException e){}
```

DELETE:

To deletedata from the table , following query is used .

This query is used in all 5 pages.

```
private void
jButton3ActionPerformed(java.awt.event.ActionEventevt) {
try{
DefaultTableModel m1=(DefaultTableModel)
jTable1.getModel();
             int si=jTable1.getSelectedRow();
            String eid1=eid.getText();
pst=con.prepareStatement("delete from employee where
eid=?");
pst.setString(1,eid1);
pst.executeUpdate();
JOptionPane.showMessageDialog(null, "Record Deleted");
tableUpdate();
eid.setText("");
nametxt.setText("");
salary.setText("");
dob.setText("");
```

```
contact.setText("");
eid.requestFocus();
        }
catch(Exception e)
        {

        }
```

UPDATE:

To update the values of the tables , the following query is used .

This query is used in all 5 pages.

```
try
        {
DefaultTableModel m1=(DefaultTableModel)
jTable1.getModel();
        int si=jTable1.getSelectedRow();
        String eid1,name1,salary1,dob1,contact1;
        eid1=eid.getText();
        name1=nametxt.getText();
        salary1=salary.getText();
        dob1=dob.getText();
        contact1=contact.getText();
pst=con.prepareStatement("update employee set
name=?,dob=?,salary=?,contact=? where eid=?");
pst.setString(1,name1);
pst.setString(2,dob1);
pst.setString(3,salary1);
pst.setString(4,contact1);
pst.setString(5,eid1);
pst.executeUpdate();
JOptionPane.showMessageDialog(null, "Record updated");
tableUpdate();
eid.setText("");
nametxt.setText("");
```

```
salary.setText("");
dob.setText("");
contact.setText("");
eid.requestFocus();
        }
catch(Exception e)
        {
        }

    }
```

VIEW:

To view contents of the table , the following query is used .

This is used in all pages .

```
    private void tableUpdate()
    {
        int cc;
try{
pst=con.prepareStatement("SELECT * FROM employee");
ResultSetrs=pst.executeQuery();
ResultSetMetaDatarsmd=rs.getMetaData();
            cc=rsmd.getColumnCount();
DefaultTableModeldft=(DefaultTableModel)
jTable1.getModel();
dft.setRowCount(0);
            while(rs.next())
            {
                Vector v2=new Vector();
for(int i=1;i<cc;i++)
                {
                    v2.add(rs.getString("eid"));
                    v2.add(rs.getString("name"));
                    v2.add(rs.getString("dob"));
                    v2.add(rs.getString("salary"));
```

```
                    v2.add(rs.getString("contact"));
                }
dft.addRow(v2);
            }
        }
catch(Exception e)
        {
        } }
```

Chapter 4

RESULT AND ANALYSIS

This chapter includes result and snapshots of the implementation. .Figure4.1 describes the login page.

FRONTEND DESIGN:

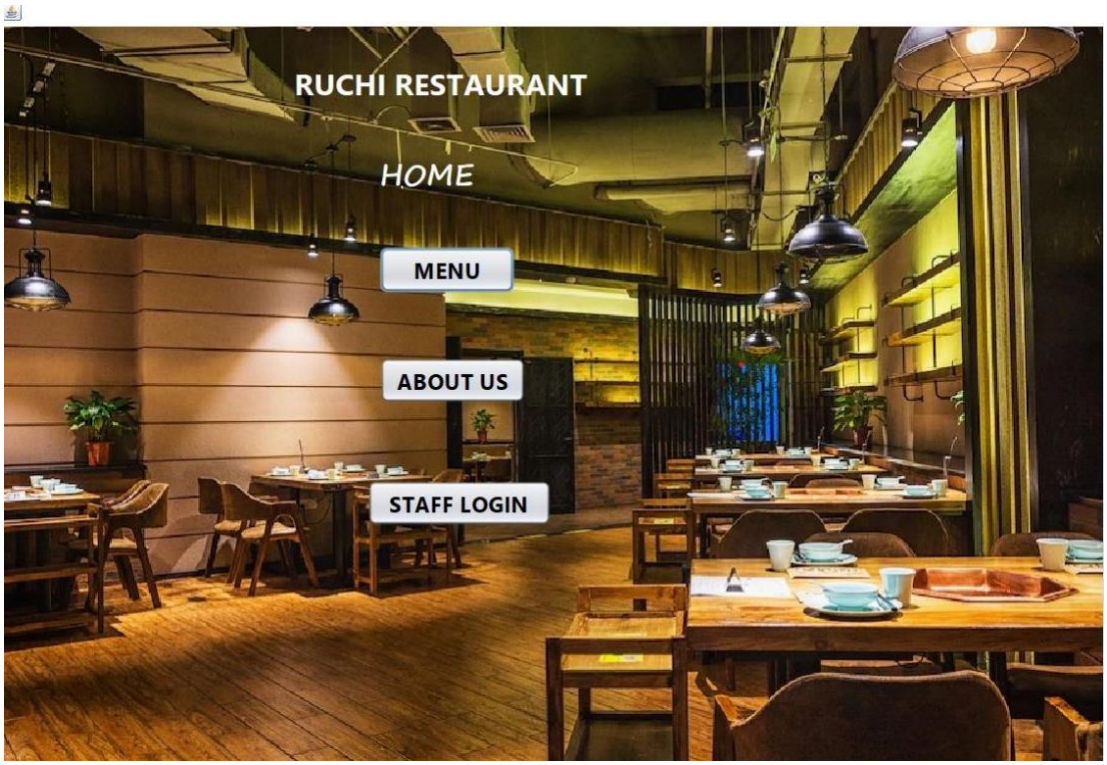


Fig: 4.1 HOME PAGE

Figure 4.1 describes the home page using which a user can view the menu, employee can login himself before using the application.

Fig 4.2: MENU PAGE

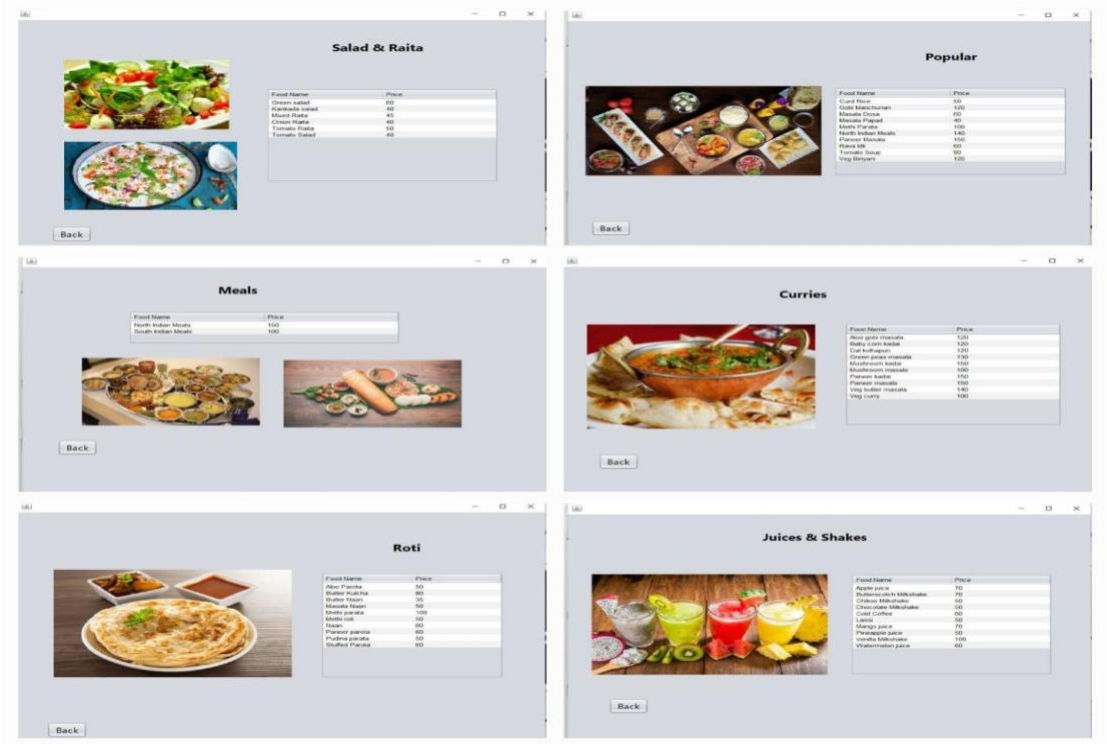


Fig 4.3: DETAILED PAGE

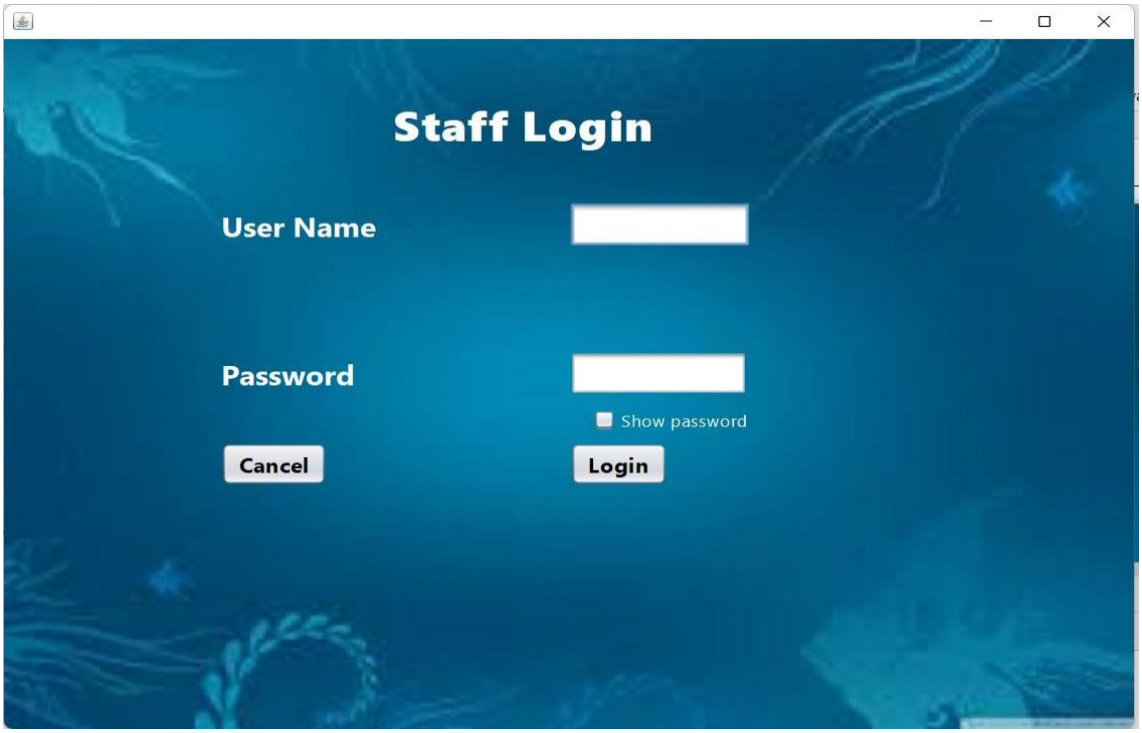


Fig 4.4: LOGIN PAGE

Implementation of Login page is given in Figure 4.4

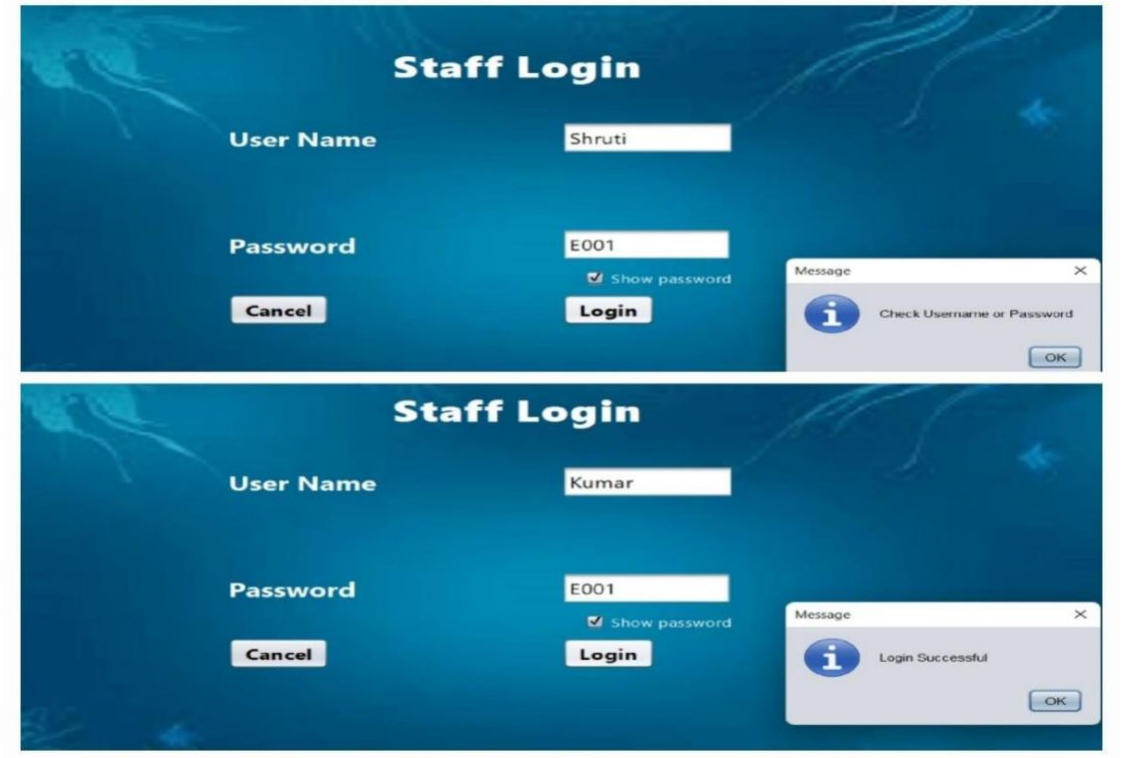


Fig 4.5: LOGIN PAGE

Authenticated users are allowed to login as shown in Figure 4.5

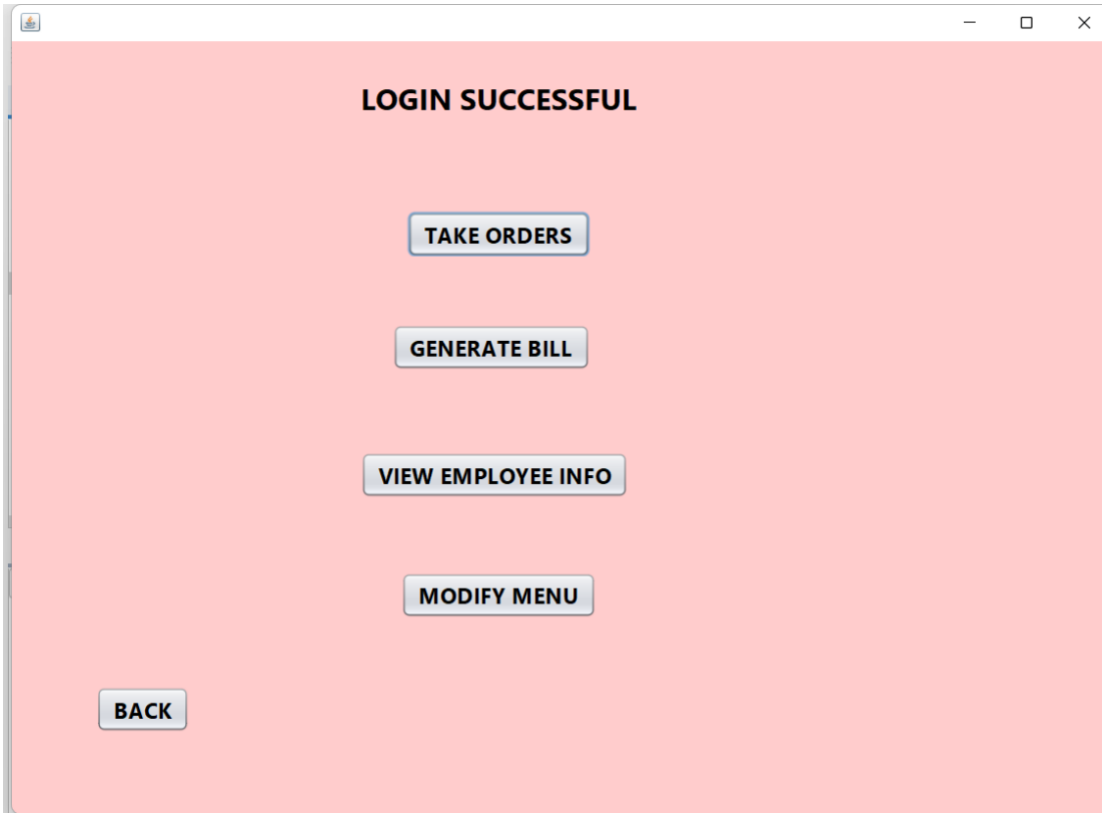


Fig 4.6 OPTIONS

Fig 4.6 shows different options for employees and managers.

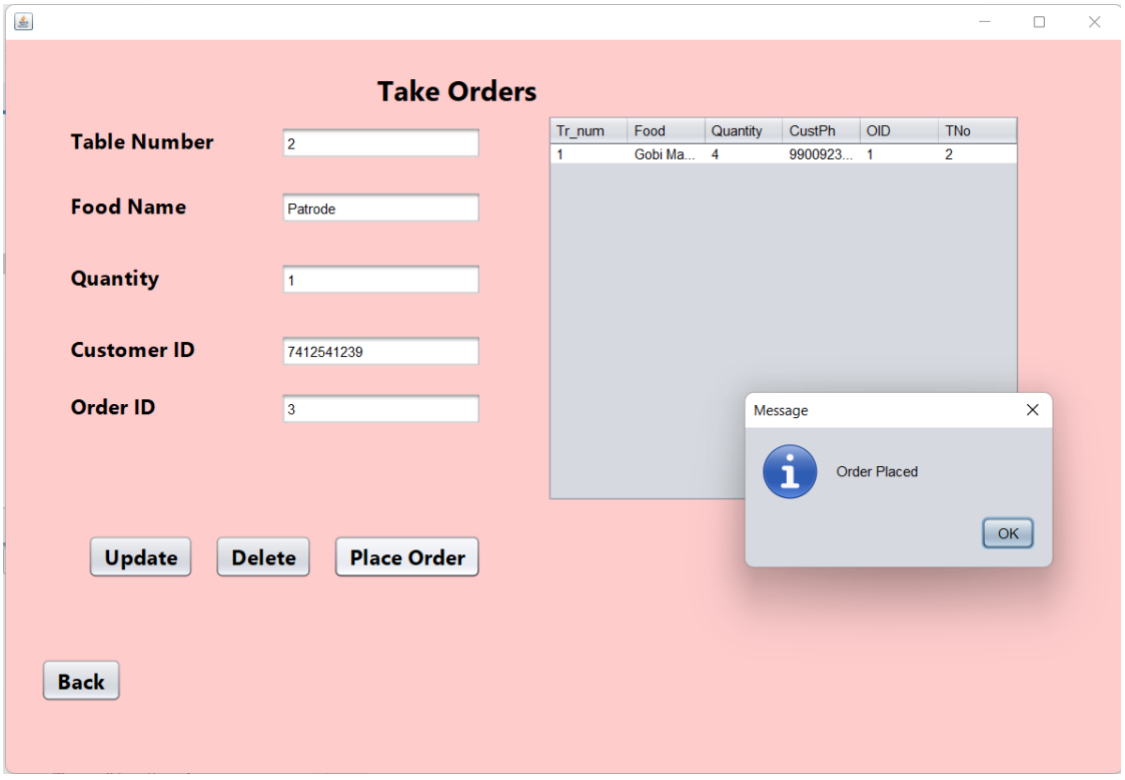


Fig-4.7 ORDER TAKING

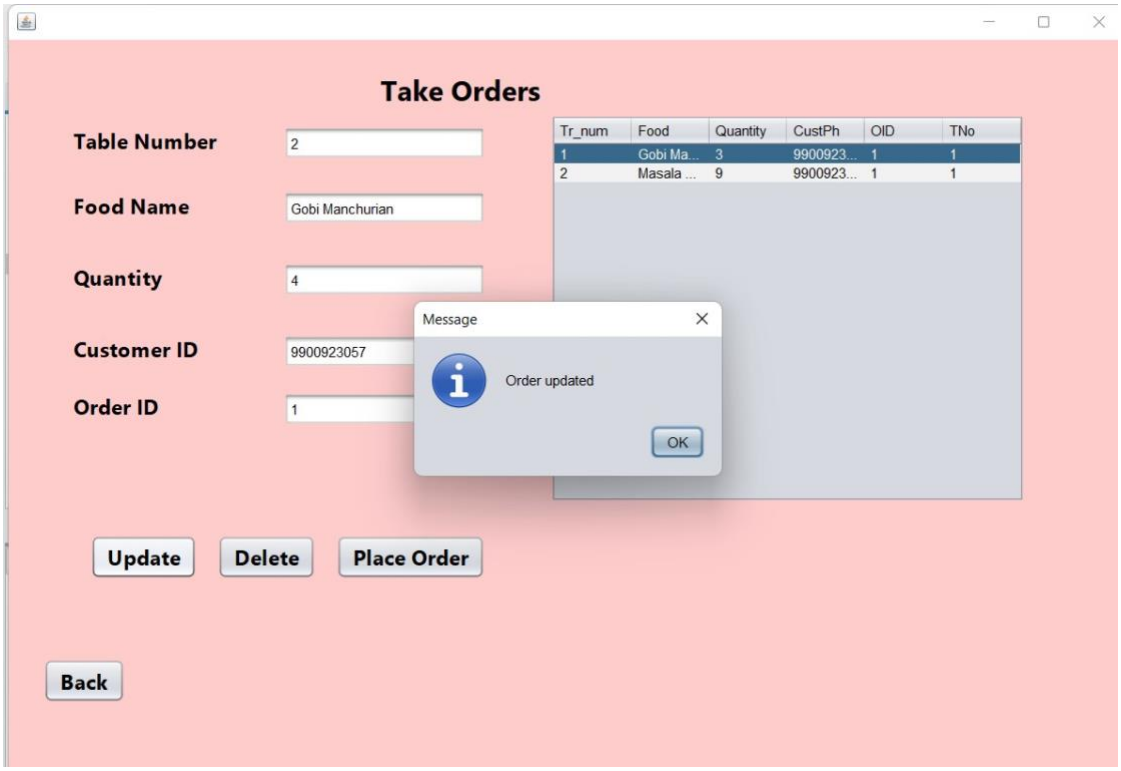


Fig: 4.8 ORDER UPDATING

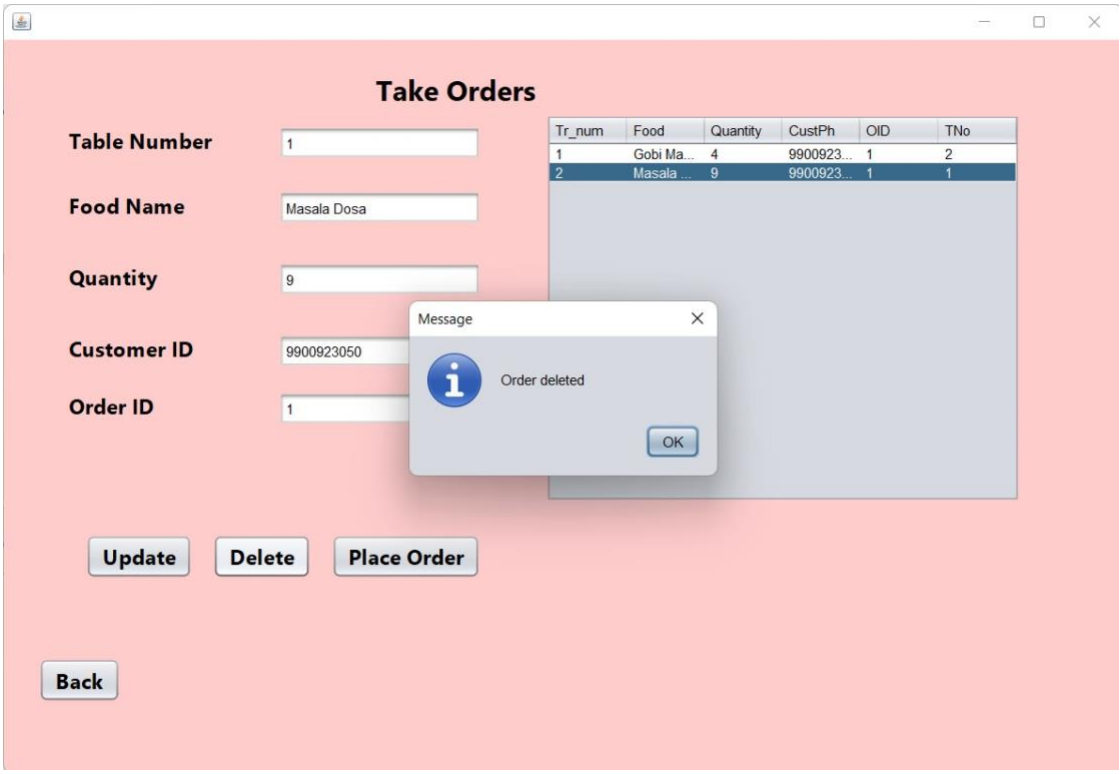


Fig: 4.9 ORDER DELETION

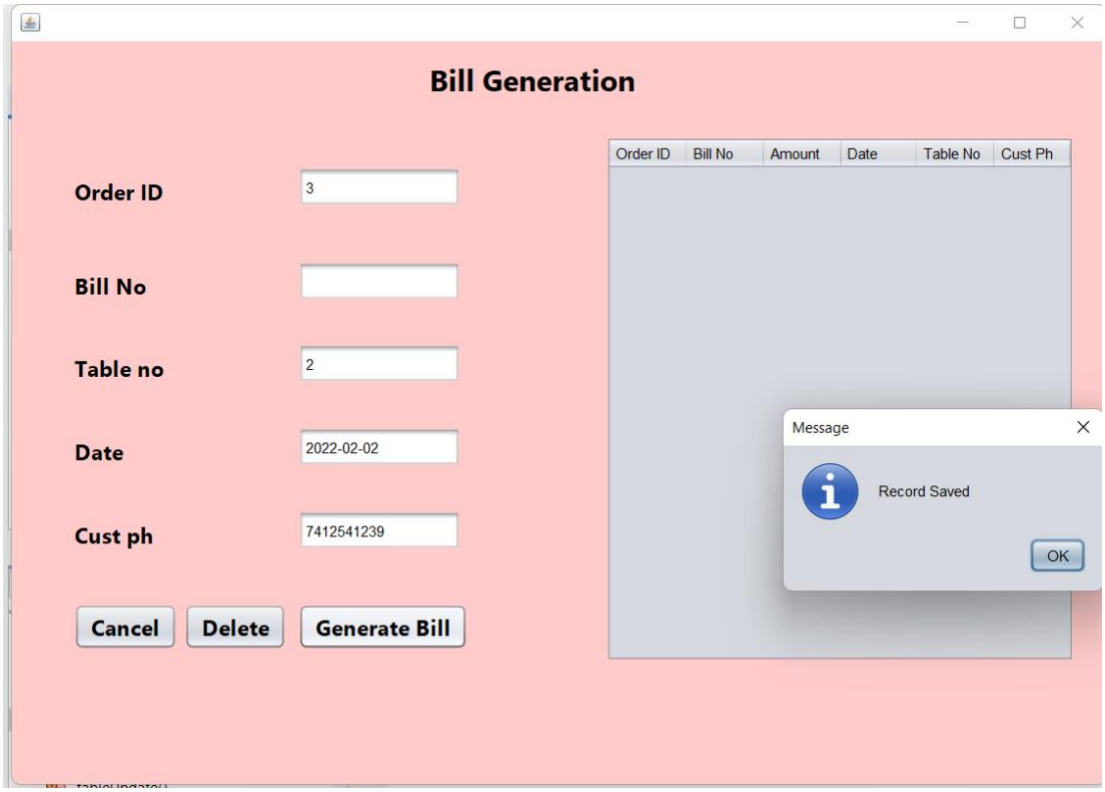


Fig: 4.10 BILL GENERATION

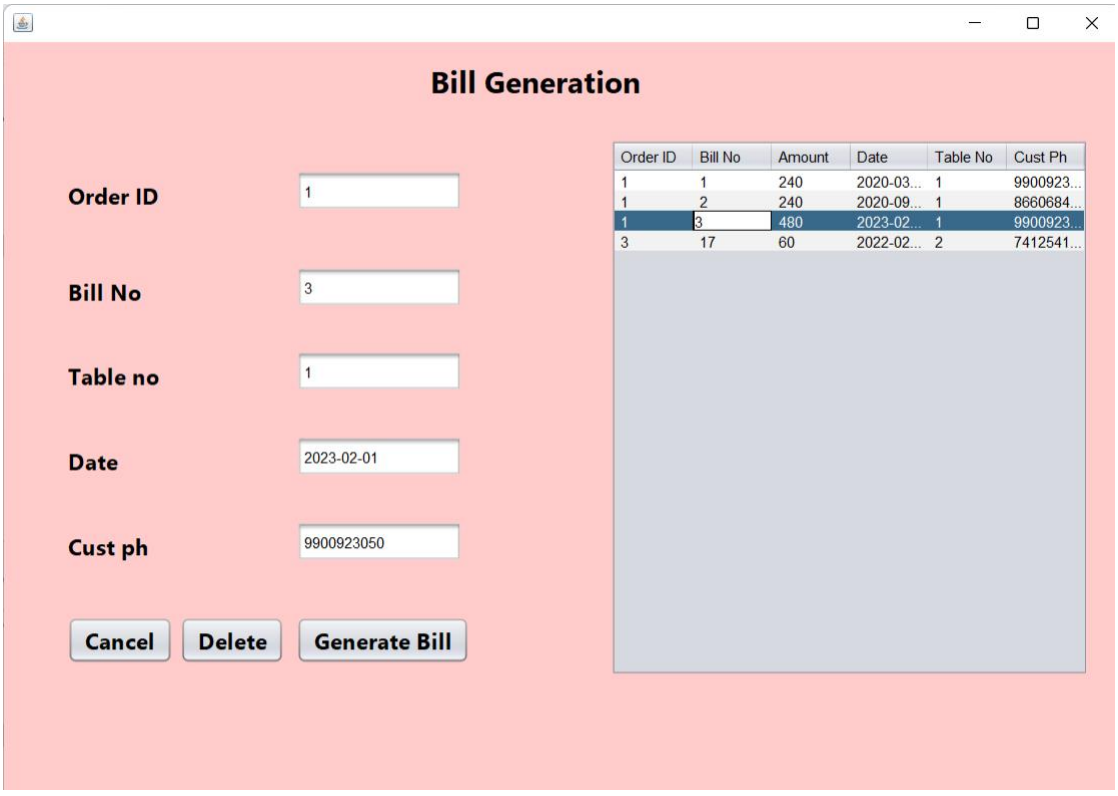


Fig: 4.11 NEW BILL GENERATED

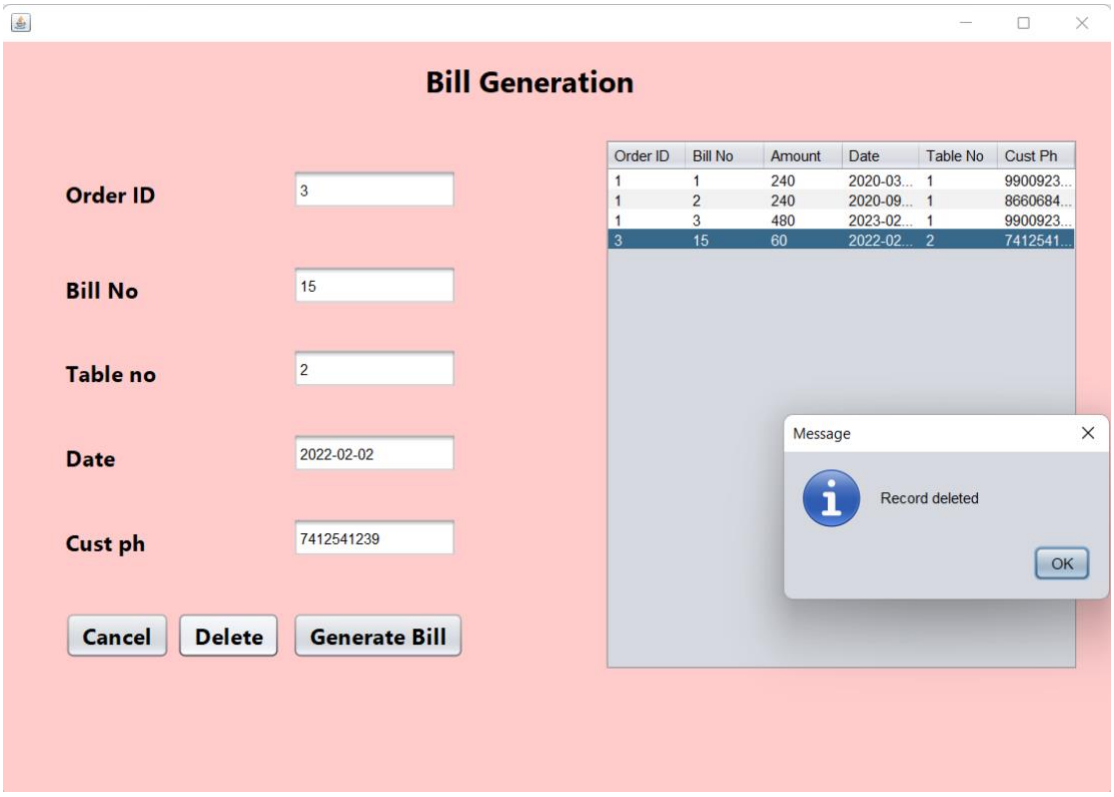


Fig: 4.12 BILL DELETION

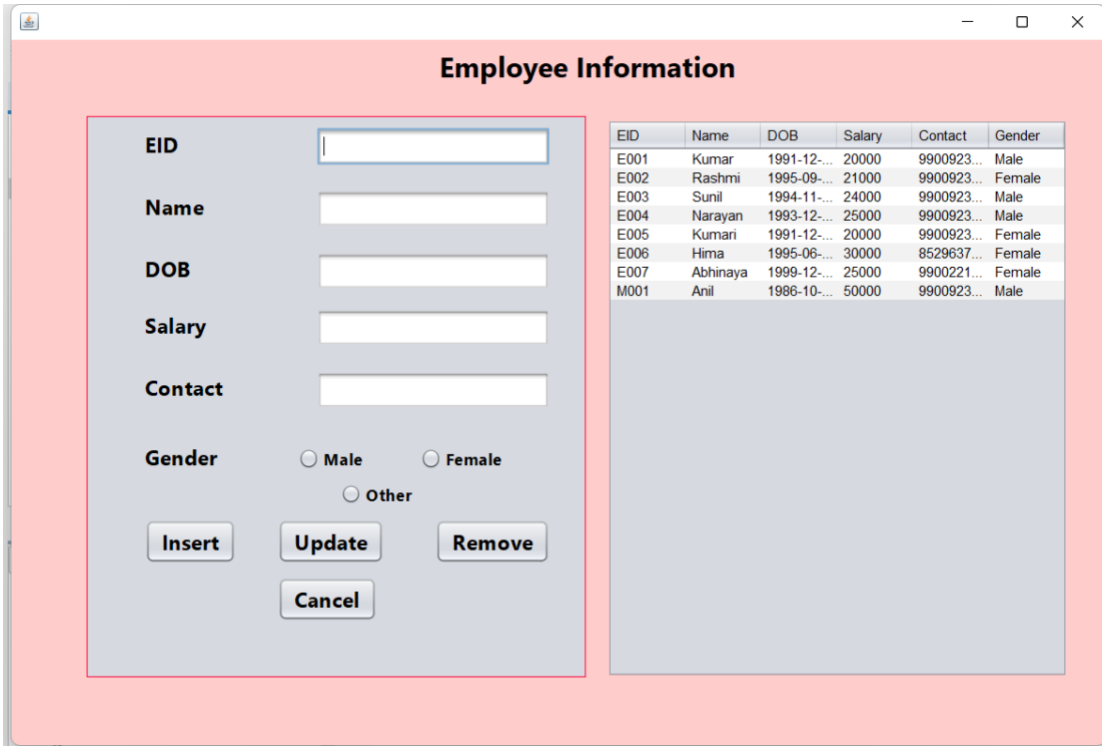


Fig 4.13 VIEW EMPLOYEE INFORMATION

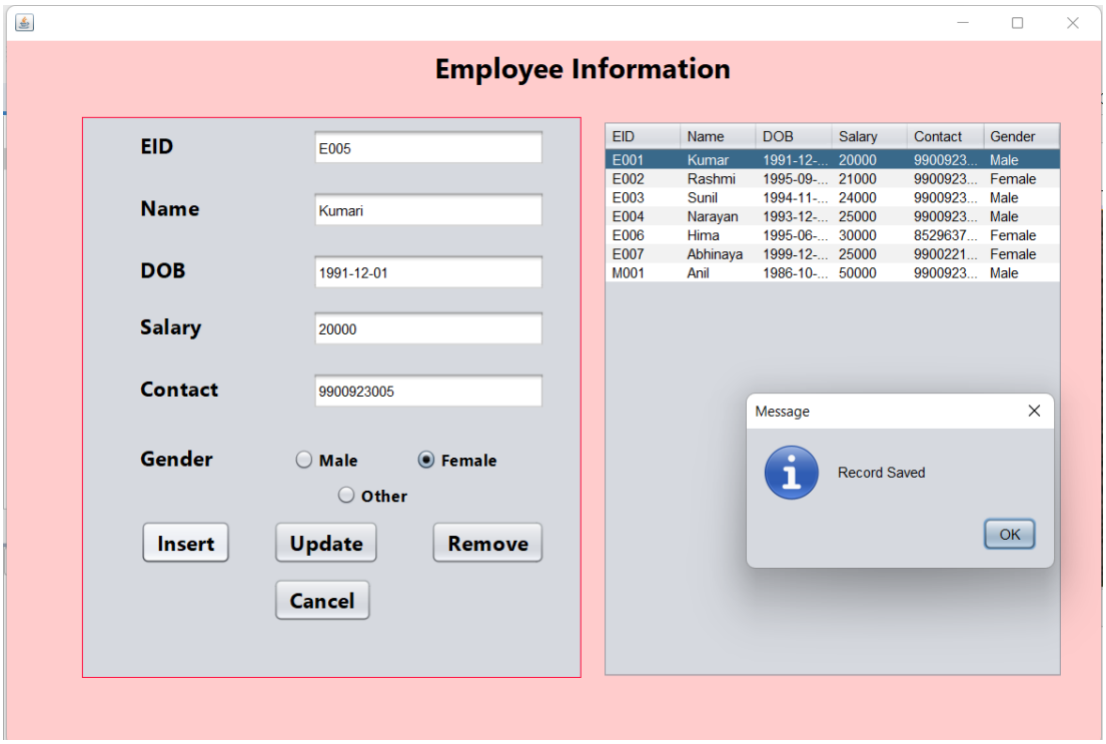


Fig:4.14 INSERTING NEW EMPLOYEE

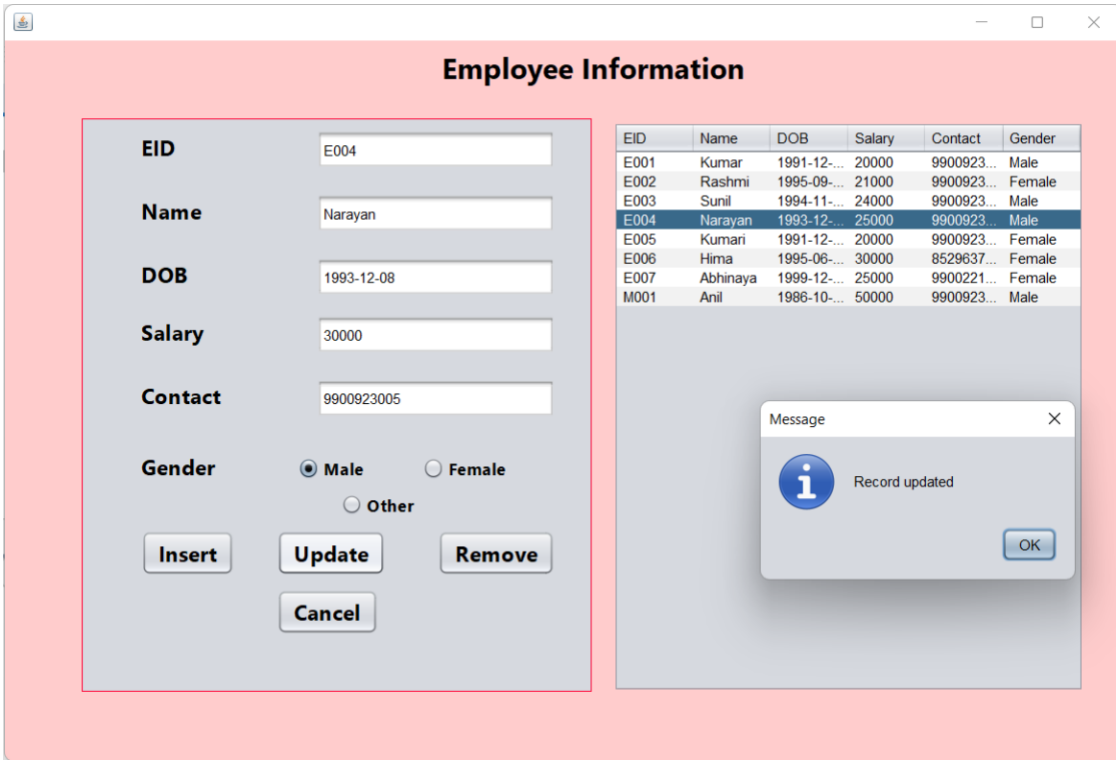


Fig:4.15 UPDATING EMPLOYEE INFORMATION

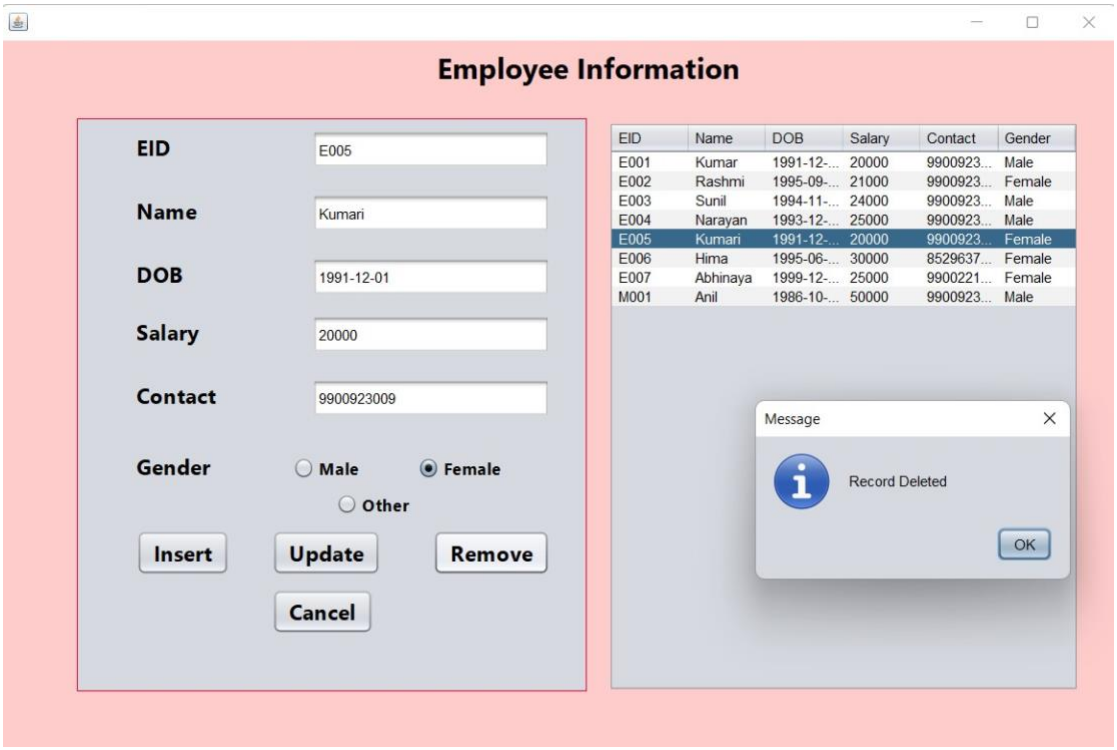


Fig: 4.16 DELETING EMPLOYEE INFORMATION

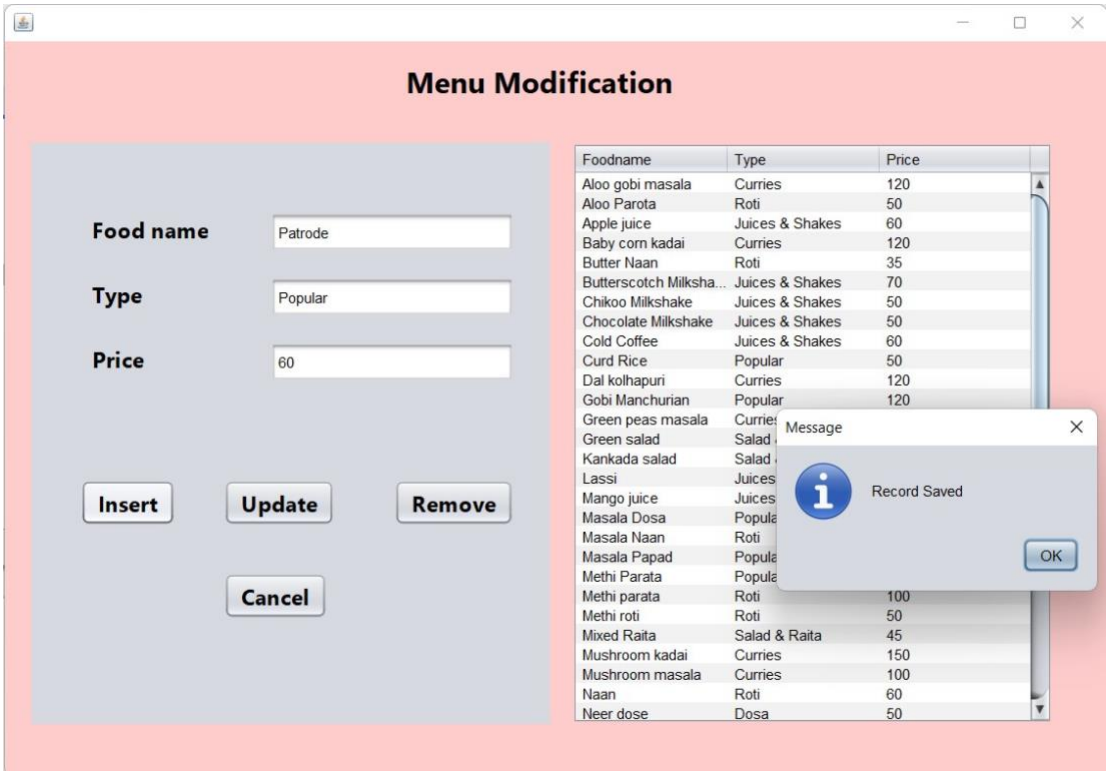


Fig: 4.17 INSERTING NEW FOOD ITEM

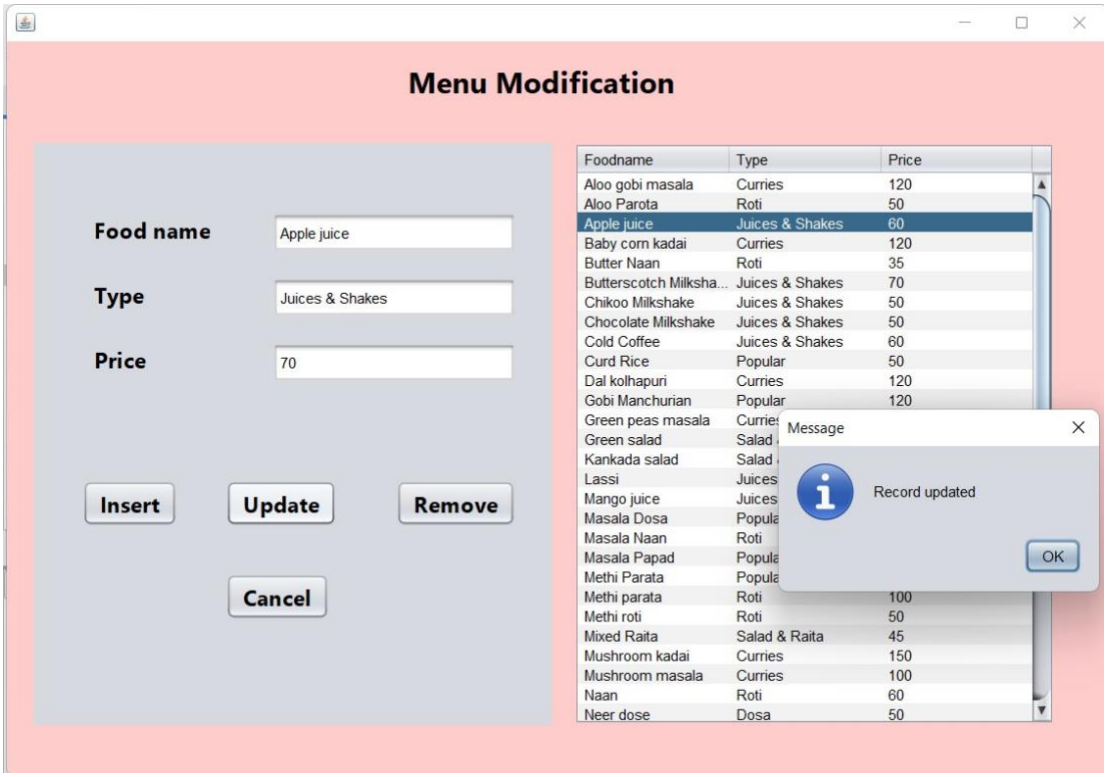


Fig: 4.18 UPDATING MENU

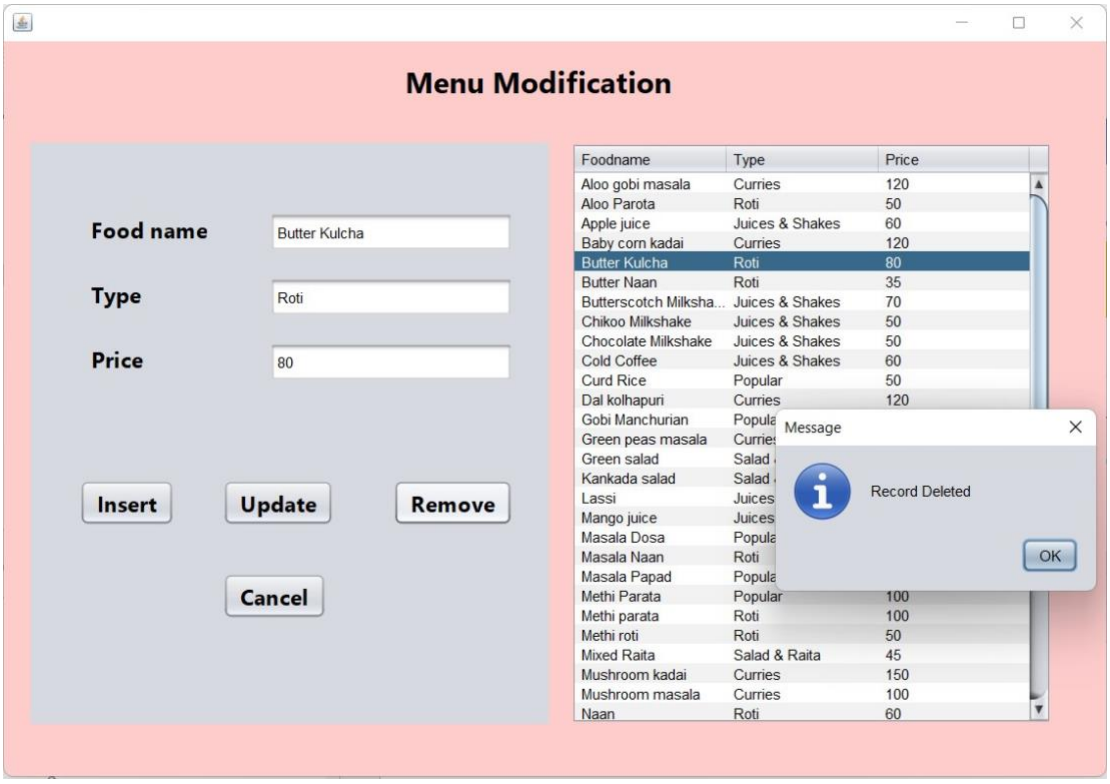


Fig: 4.19 DETEING FOOD ITEM

BACKEND DATABASE:

Tables in Backend

#	Name	Type	Collation	Attributes	Null
1	EID	varchar(5)	utf8mb4_unicode_ci		No
2	Name	text	utf8mb4_unicode_ci		No
3	DOB	date			No
4	Salary	int(7)			No
5	Contact	bigint(10)			No
6	Gender	text	utf8mb4_unicode_ci		No

Fig 4.20 Employee

#	Name	Type	Collation	Attributes	Null	Default
1	Fname	varchar(50)	utf8mb4_unicode_ci		No	*None*
2	Type	varchar(20)	utf8mb4_unicode_ci		No	*None*
3	Price	int(5)			No	*None*

Fig 4.21 Menu

#	Name	Type	Collation	Attributes	Null	Default
1	No_of_seats	int(5)			No	*None*
2	num	int(5)			No	*None*

Fig 4.22 Table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	CPh	bigint(10)			No	None		
2	Amount	int(5)			No	None		
3	BillNo	int(5)			No	None		AUTO_INCREMENT
4	Date	date			No	None		
5	OID	int(5)			No	None		
6	num	int(5)			No	None		

Fig 4.23 Bill

#	Name	Type	Collation	Attributes	Null	Default
1	Fname	varchar(50)	utf8mb4_unicode_ci		No	None
2	Qty	int(5)			No	None
3	CPh	bigint(10)			No	None
4	EID	varchar(5)	utf8mb4_unicode_ci		No	None
5	OID	int(5)			No	None
6	num	int(5)			No	None
7	Tr_No	int(11)			No	None

Fig 4.24 Order

CONCLUSION

Today management is one of the most essential features of all form. Management provides sophistication to perform any kind of task in a particular form. This is Restaurant management system; it is used to manage most restaurant related activities in the restaurant. The primary aim of is to improve accuracy and enhance safety and efficiency in the restaurant.

We may also conclude that by using restaurant software, processing both new orders and refills can be done quickly and simply with just a few keystrokes or mouse clicks with new, easy to learn and use Graphical User Interface (GUI) restaurant management solution. That due to automation where the restaurant manager does his or her work much faster, a switch from product oriented to service oriented which is one of the most important keys in Restaurant. In other words that the manager will have more time in counseling his/her customers, where the goal of servicing customer is one of the important solutions to avoid any error.

This project is to design an efficient maintain database system and it will make easy to access the records based on requirement.

It will also improve decision making by reducing the processing time as well as reducing the communication gap between admin and customer. It will also reduce error in records.

Future Scope

The future scope of this project **RESTAURANT MANAGEMENT SYSTEM** is very wide. There are many additional features, which are planned to be incorporated during the future enhancements of this project. The Future version of System there is some point that we may implement on them:

- Developing an application for android devices that works on the same database which is the mini of MySql.

REFERENCES:

[1]. Ramez Elmasri and Shamkant B. Navathe, "Database systems Models, Languages, Design and Application Programming", Pearson, 7th Edition, 2017.

[2]. Raghu Ramakrishnan, and Gehrke, "Database management systems", McGraw Hill, 3rd Edition, 2014.

[3]. Herbert Schildt: JAVA the Complete Reference, 7th/9th Edition, Tata McGraw Hill, 2007.

[4]. Jim Keogh: J2EE-TheCompleteReference, McGraw Hill, 2007.