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.

1.1Overview 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.

1.2History

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 Siebel.

1.3Applications 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 is 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 DB.

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

For Organizations with huge data centers having a lot of servers hosting numerous applications, it is always a major problem to monitor if each of the servers is up and functional all the time. The problem is more acute during late night shifts when the usual number of network/systems engineers working is less.

So, it becomes very important for the organizations to know if a server is down or non-functional and take corrective action immediately. Unfortunately, for some less time critical applications, it is usually the client who informs that there is a problem with the server when he/she tries to logon to the application Organizations would be very interested in knowing about these server failures immediately and take corrective action before the client starts complaining.

1.5 Objectives of the project

- This describes how management system maintains database.
- Creating database of organization, employee, servers and notification team.
- List of all servers which have problems.
- Notify it to the engineer who can resolve problem through admin.

1.6 Organization of the report

This section deals with the Introduction and organization of the project report. Chapter 2 discusses the Specific to the problem-Requirement Analysis-Design. Chapter 3 discusses the Design and Implementation Chapter 4 gives information about the snapshot and results Chapter 5 include conclusion and future scope. Chapter 6 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 describe high level system design, software requirements etc.

2.1 Basic Definition

Server Notification Management System is an application or set of applications that lets network engineers to manage networks independent component inside a bigger network management.

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 Organization, Employees etc.,
- This will help the engineers about the server which is not working by providing them the details of the fault Server.

2.3 Requirement Analysis

SOFTWARE

Windows 10

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

MySQL

MySQL is an open-source relational database system (RDBMS).Its name is a combination of "My", the name of co-founder Michael Widenius's daughter My and "SQL", the abbreviation for Structured Query Language. A relational database organizes data into one or more data tables in which data may be related to each other; these relations help structure the data. SQL is a language programmer use to create, modify and extract data from the relational database, as well as control user access to the database. In addition to relational databases and SQL, an RDBMS like MySQL works with an operating system to implement a relational database in a computer's storage system, manages users, allows for network access and facilitates testing database integrity and creation of backups.

FRONT END

Java Swings

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 10/11.
- Backend: MySQL
- Front end: Java Swings
- Platform: JDBC jar

Hardware components

- Processor: Intel core i5
- Processor speed: 1.19 GHz
- Hard disk: Minimum 20GB.
- RAM: 1 GB or more.

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

ORGANIZATION: It contains the attributes Org_ID, Org_Name, Address, Contact and Status.

EMPLOYEE: It contains the attributes Emp_ID, Org_ID, FName, LName, Email, Phn_no, Type_Emp, SID, Status and Password.

ADMIN: It contains the attributes ServerFailedDate, Emp_ID, SID, Severity and Reason.

SERVER: It contains the attributes SID, IPAddress and Status.

NOTIFICATION TEAM: It contains the Notification_ID, Password and Org_ID.

3.1 ER DIAGRAM

Figure 3.1 gives schema diagram. This ER diagram includes Employee, Organization, Server, Admin, NotificationTeam as Entities and it has respective attributes and relations as shown below.

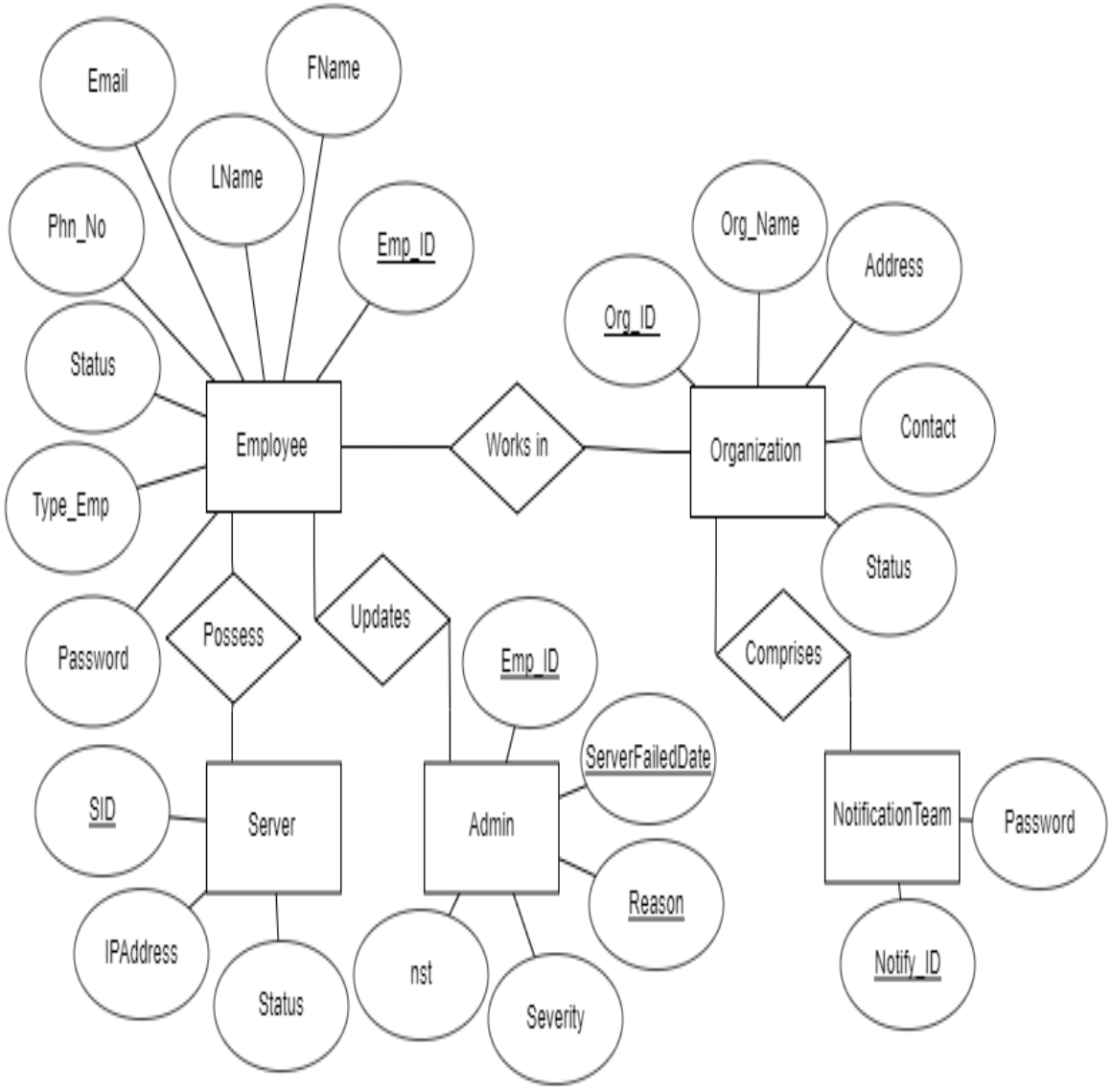

Figure 3.1 ER Diagram

3.2 SCHEMA DIAGRAM

Figure 3.2 gives schema diagram. In this Schema, it includes Employee, Organization, Server, Admin, NotificationTeam tables. In Employee table, Emp_ID is PK and Org_ID, SID are FK. In Admin table ServerFailedDate, Reason, Emp_ID are PK and EMP_ID is FK. In Organization table Org_ID is PK. In Notification team , Notify_ID is PK, Org_ID is FK. In Server table, SID is a PK.

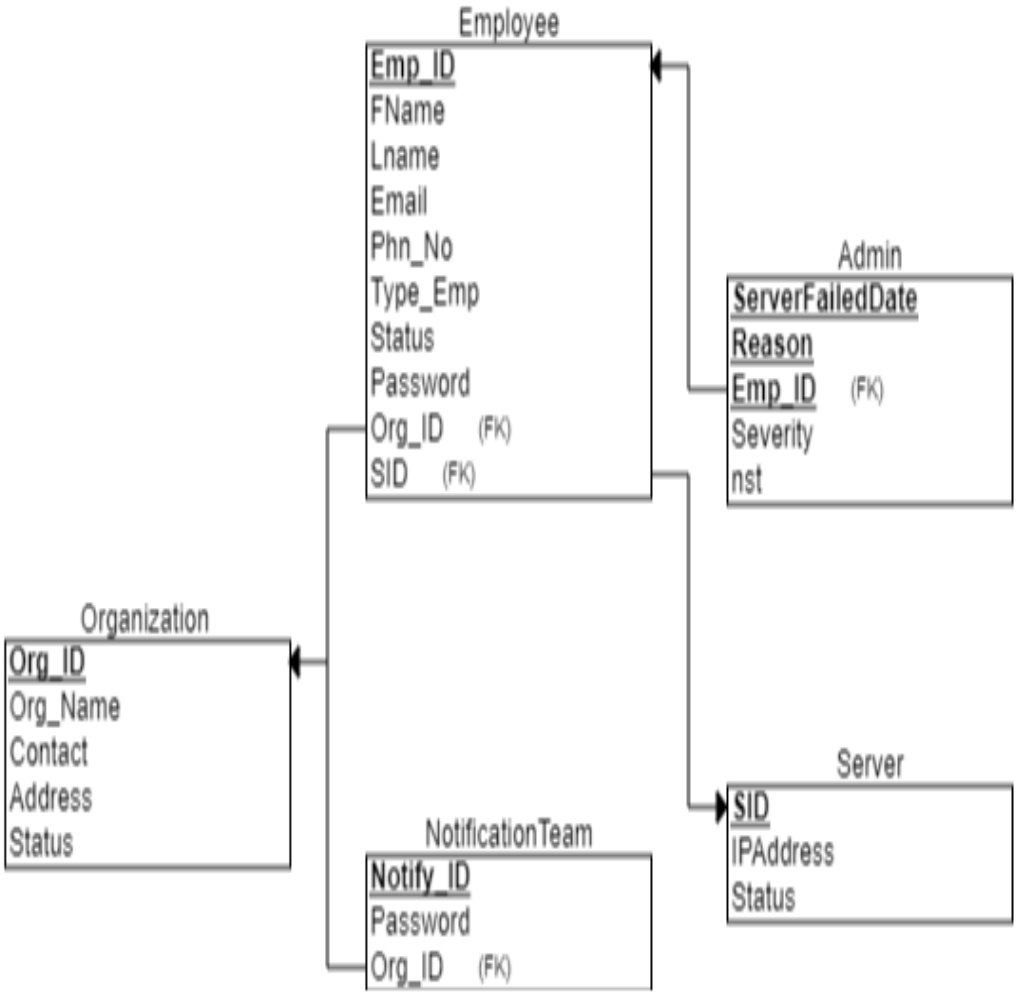

Figure 3.1 Schema diagram

3.3 PSEUDOCODE

CONNECT:

To connect backend with front end

```
public void Connect() {
 try {
        Class.forName("com.mysql.cj.jdbc.Driver");
     try {
         con = DriverManager.getConnection("jdbc:mysql://localhost:3306/sms",
"root","oracle");
            }
       catch (SQLException ex) {
              ex.printStackTrace();
       }
    }
   catch (ClassNotFoundException ex) {
          Logger.getLogger(orgde.class.getName()).log(Level.SEVERE, null, ex);
       }
   }
```

INSERT:

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

This query is used in 4 pages.

```
private void insertActionPerformed(java.awt.event.ActionEvent evt) {
       jPanel2.setVisible(true);
       try {
 String Org_ID,FName,LName,Email,Phn_No,SID,Status,password;
 Org_ID=e2.getSelectedItem().toString();
 FName=e3.getText();
 LName=e4.getText();
 Email=e5.getText();
 Phn_No=e6.getText();
```

```
 SID=e7.getSelectedItem().toString();
 Status=e8.getSelectedItem().toString();
 password=e9.getText();
 pst=con.prepareStatement("insert into `employee`
(Emp_ID,Org_ID,FName,LName,Email,Phn_no,Type_Emp,SID,Status,password)val
ues (?,?,?,?,?,?,?,?,?,?)");
    pst.setInt(1, Integer.parseInt(e1.getText()));
    pst.setString(2,Org_ID);
    pst.setString(3,FName);
    pst.setString(4,LName);
    pst.setString(5,Email);
    pst.setString(6,Phn_No);
    pst.setString(7,Type_Emp);
    pst.setString(8,SID);
    pst.setString(9,Status);
    pst.setString(10,password);

    pst.executeUpdate();
    JOptionPane.showMessageDialog(this,"record saved");
    table_update();
      e2.setSelectedItem("");
    e3.setText("");
    e4.setText("");
    e5.setText("");
    e6.setText("");
    if(admin.isSelected())
    {
      Type_Emp="Admin";
    }
    if(employ.isSelected())
    {
      Type_Emp="Employee";
    }
```

```
    e7.setSelectedItem("");
    e8.setSelectedItem("");
    e9.setText("");
    e1.requestFocus();
    } catch (SQLException ex) {

     ex.printStackTrace();
     }
}
```

DELETE:

To delete data from the table , following query is used .

This query is used in 4 pages.

```
private void deleteActionPerformed(java.awt.event.ActionEvent evt) {
try
 {
  DefaultTableModel DFT=(DefaultTableModel) empde.getModel();
  int selectedIndex=empde.getSelectedRow();
  int Emp_ID=Integer.parseInt(DFT.getValueAt(selectedIndex,1).toString());
  pst=con.prepareStatement("delete from employee where Emp_ID=?");
  pst.setInt(1,Integer.parseInt(e1.getText()));
  pst.executeUpdate();
  JOptionPane.showMessageDialog(this,"record deleted");
  table_update();

    e1.setText("");
    e2.setSelectedItem("");
    e3.setText("");
    e4.setText("");
    e5.setText("");
    e6.setText("");
    e7.setSelectedItem("");
```

```
    e8.setSelectedItem("");
    e9.setText("");
    e1.requestFocus();
        } catch (Exception ex) {
            ex.printStackTrace();
        } }
```

UPDATE:

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

This query is used in 3 pages.

```
private void editActionPerformed(java.awt.event.ActionEvent evt) {
 try{
 String Org_ID ,FName,LName,Email,Phn_no,Type_Emp = null,SID ,Status
,password ;         int row=empde.getSelectedRow();
 String value=(empde.getModel().getValueAt(row,0).toString());
 String query="update `employee` set
Org_ID=?,FName=?,LName=?,Email=?,Phn_no=?,Type_Emp=?,SID=?,Status=?,pas
sword=? where Emp_ID=?";
 pst=con.prepareStatement(query);
 Org_ID=e2.getSelectedItem().toString();
 FName=e3.getText();
 LName=e4.getText();
 Email=e5.getText();
 Phn_no=e6.getText();
 SID=e7.getSelectedItem().toString();
 Status=e8.getSelectedItem().toString();
 password=e9.getText();
 pst=con.prepareStatement("update `employee` set
Org_ID=?,FName=?,LName=?,Email=?,Phn_no=?,Type_Emp=?,SID=?,Status=?,pas
sword=? where Emp_ID=?");

 pst.setInt(10, Integer.parseInt(e1.getText()));
 pst.setString(1, Org_ID);
```

```
pst.setString(2,FName);
pst.setString(3,LName);
pst.setString(4,Email);
pst.setString(5,Phn_no);
if(admin.isSelected()){
    Type_Emp="Admin";
        }
        if(employ.isSelected()){
        Type_Emp="Employee";
        }
   pst.setString(6,Type_Emp);
   pst.setString(7,SID);
   pst.setString(8,Status);
   pst.setString(9,password);
   pst.executeUpdate();
  table_update();
   JOptionPane.showMessageDialog(this,"record updated");
   e1.setText("");
  e2.setSelectedItem("");
  e3.setText("");
  e4.setText("");
   e5.setText("");
         e6.setText("");
         e7.setSelectedItem("");
         e8.setSelectedItem("");
         e9.setText("");
         e1.requestFocus();
         } catch (SQLException ex) {
        ex.printStackTrace();
        JOptionPane.showMessageDialog(null,ex);
     }
  }
```

VIEW:

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

This is used in all pages .

```
private void table_update() {
      int CC;
      try {
         pst = con.prepareStatement("SELECT * FROM organization");
         ResultSet Rs = pst.executeQuery();

         ResultSetMetaData RSMD = Rs.getMetaData();
         CC = RSMD.getColumnCount();
         DefaultTableModel DFT = (DefaultTableModel) orgde.getModel();
         DFT.setRowCount(0);
         while (Rs.next()) {
            Vector v2 = new Vector();
            for (int i = 1; i <= CC; i++) {
               v2.add(Rs.getInt("Org_ID"));
               v2.add(Rs.getString("Org_Name"));
               v2.add(Rs.getString("Address"));
               v2.add(Rs.getString("Contact"));
               v2.add(Rs.getString("Status"));
            }
            DFT.addRow(v2);
         }
      } catch (Exception e) {
      }
   }
```

SEARCH:

To search the particular row from the table by name , the following query is used .

This is used in a page .

```
private void searchActionPerformed(java.awt.event.ActionEvent evt) {
try{
         pst=con.prepareStatement("select * from organization where Org_ID=?");
```

```
            int Org_ID=Integer.parseInt(O7.getText());
            pst.setInt(1,Org_ID);
            ResultSet rs=pst.executeQuery();
            if(rs.next()==false)
            { JOptionPane.showMessageDialog(this,"Sorry record not found");
               O1.setText("");
               O2.setText("");
               O3.setText("");
               O4.setText("");
               O2.requestFocus();
            }
            else{
               O1.setText(rs.getString("Org_ID"));
               O2.setText(rs.getString("Org_Name"));
               O3.setText(rs.getString("Address"));
               O4.setText(rs.getString("Contact"));
               O5.setSelectedItem(rs.getString("Status"));
               pst.setString(5, O5.getSelectedItem().toString());
            }
         }catch(Exception e){
             System.out.println("exception2"+e);
         }
          jPanel2.setVisible(true);
            table_search();
    }
```

TRIGGER:

```
BEGIN
  IF (NEW.ServerFailedDate >CURDATE()) THEN
     SIGNAL SQLSTATE '02000' SET MESSAGE_TEXT='Notification Date cannot
exceed current Date';
  END IF;
END
```

CHAPTER 4

RESULT AND ANALYSIS

This chapter includes result and snapshots of the implementation.

Figure 4.1 HOME PAGE

Figure 4.1 describes the home page and it includes Organization, User, Admin, Notify Team Buttons.

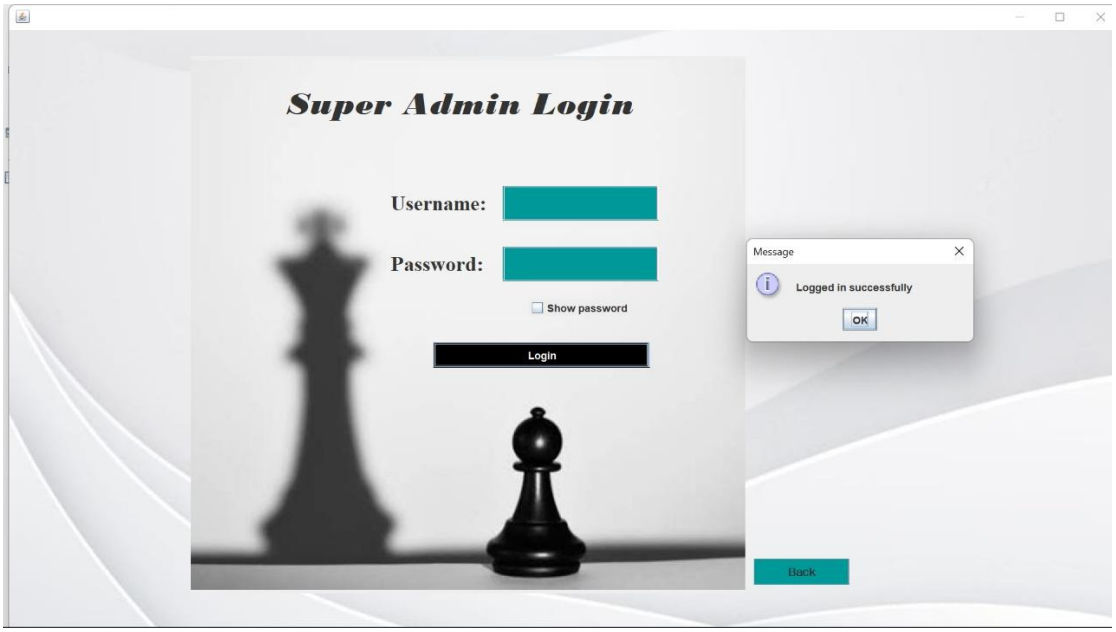

Figure 4.2 SUPER ADMIN LOGIN

Figure 4.2 describes the login information about Super admin and this page contains a Username and Secret Password for login.

Figure 4.3: ORGANIZATION DETAILS

Through this page (Figure 4.3) the Organization can fill the details about its ID, Name, Address, Contact and its status and it can be deleted, updated, viewed and can be searched.

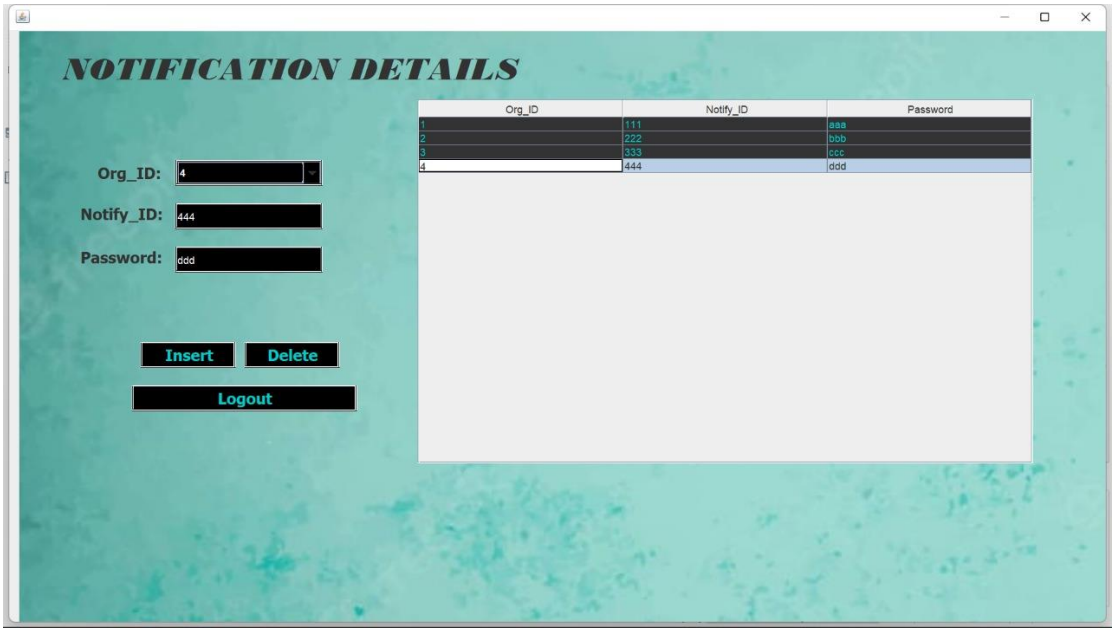

Figure 4.4 NOTIFICATION TEAM DETAILS

Figure 4.4 describes by clicking the notification details in figure 4.3 we can enter to this page (figure 4.4) and can fill details about notification team and it includes Org_ID, Notify_ID and its password.

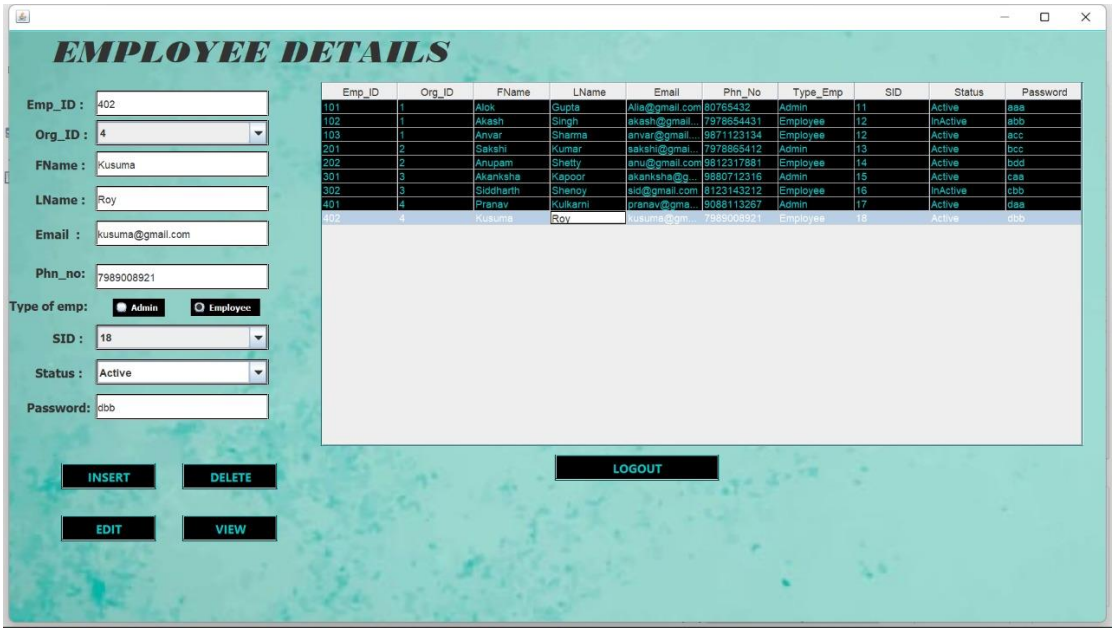

Figure 4.5 EMPLOYEE DETAILS

Figure 4.5 describes by clicking the notification details in figure 4.3 we can enter to this page (figure 4.5) and we can fill details about Employees and it includes Id, Name, Email, Phono, SID, Status and Password

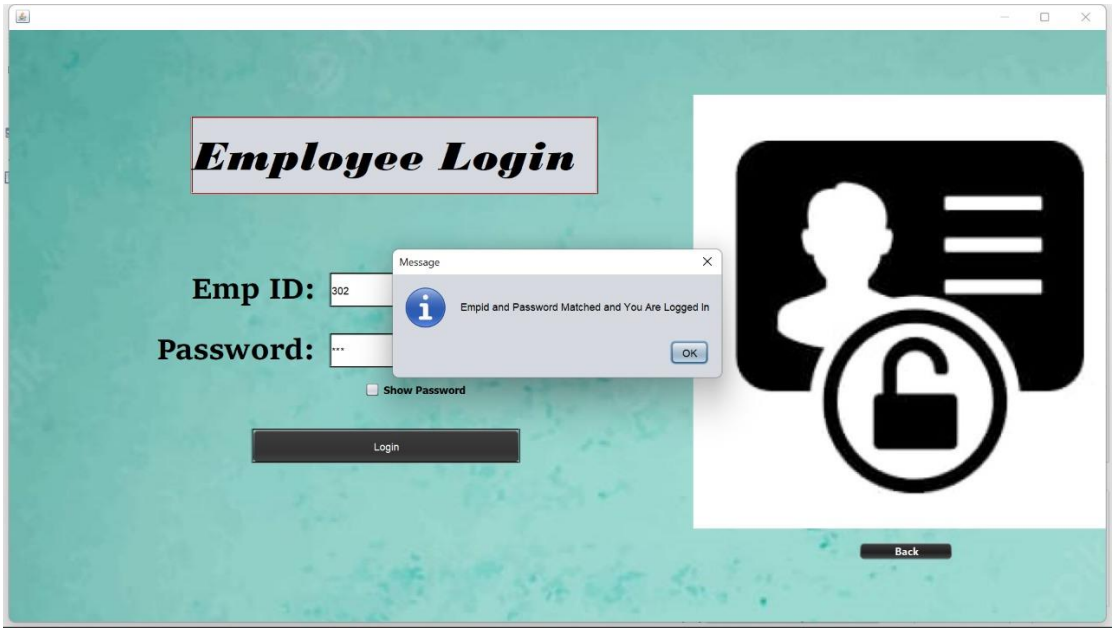

Figure 4.6 EMPLOYEE LOGIN

Figure 4.6 describes that by clicking the user button in figure 4.1 the employee can login. To login successfully employee has to enter their Emp_ID and respective Password.

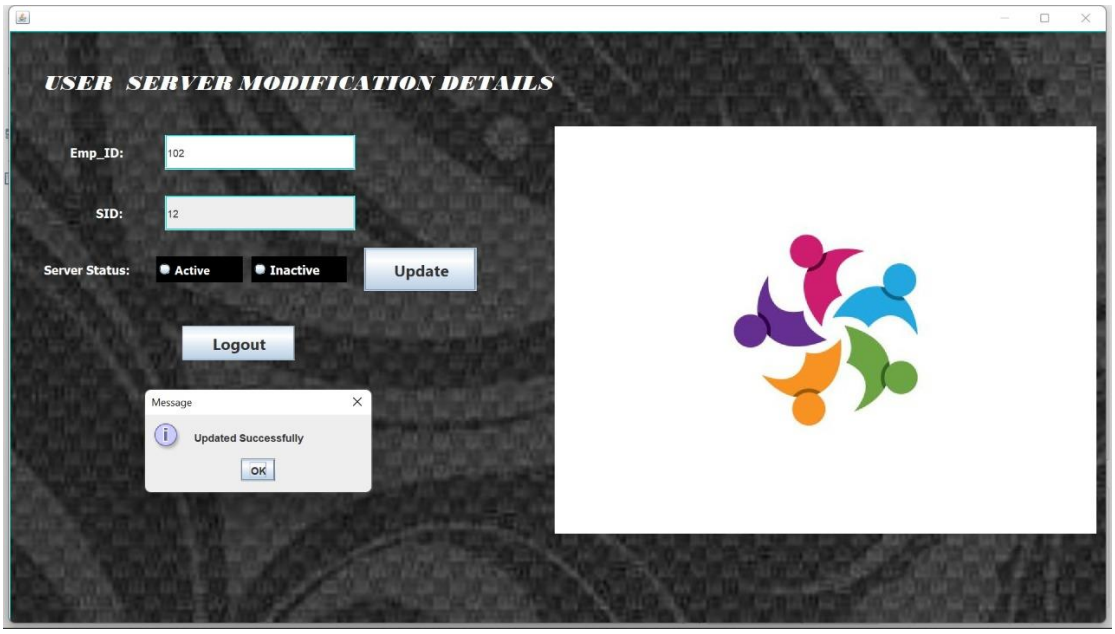

Figure 4.7 USER SERVER MODIFICATION DETAILS

Through this page (figure 4.7) employee can update his server status. Employee updates about its server status Active or Inactive.

Figure 4.8 ADMIN LOGIN

Figure 4.8 describes that by clicking the Admin login button in figure 4.1 the Admin can login. To Login Successfully Admin should enter proper Org_ID, Emp_ID and respective Password.

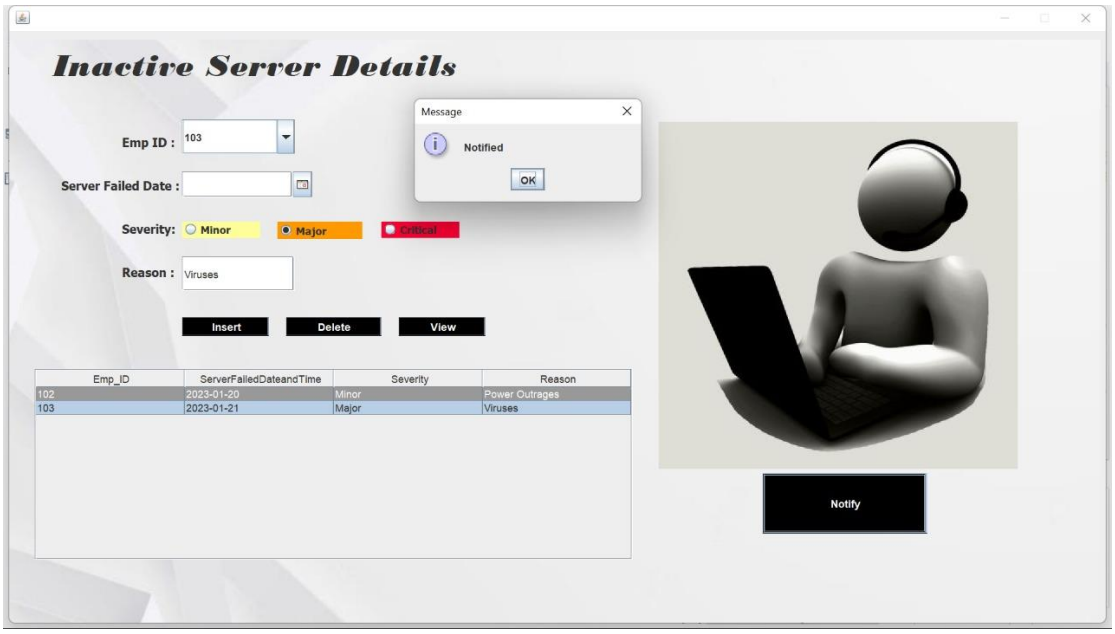

Figure 4.9 INACTIVE SERVER DETAILS

Through this page (figure 4.9) the admin will fill the details of employee about the server status. It includes ServerFailedDate, severity and reason.

Figure 4.10 NOTIFY LOGIN

Figure 4.10 describes that by clicking Notify button in figure 4.1, the notify team can login. To login successfully Notification team has to enter its ID and Password properly.

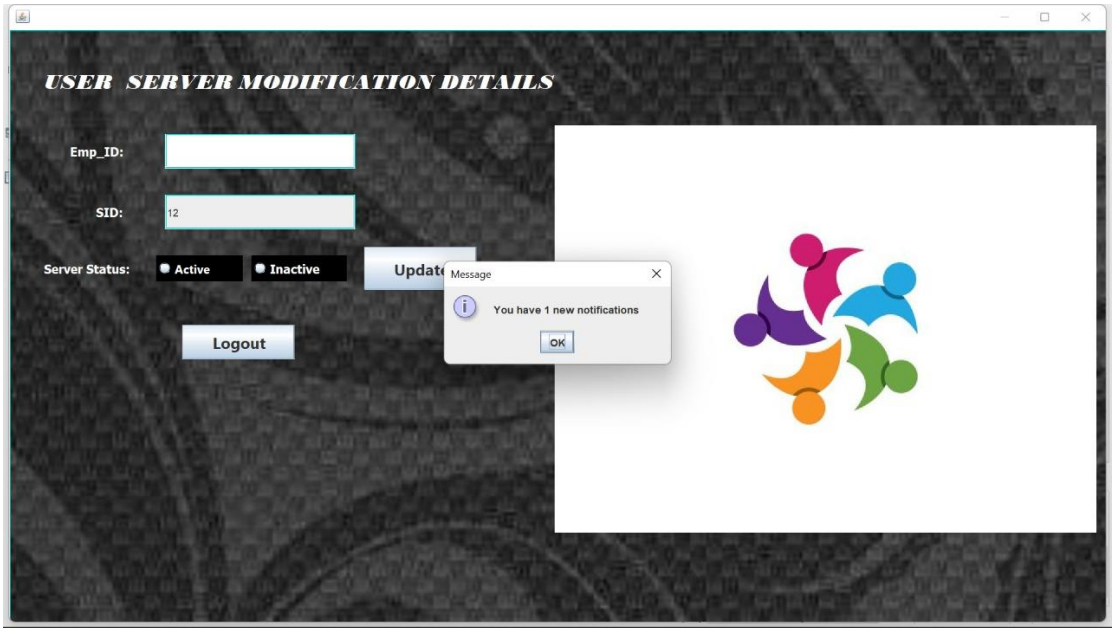

Figure 4.11 POP UP MESSAGE

Figure 4.11 describes number of notifications that an organization is having about the inactive server through a Popup Message.

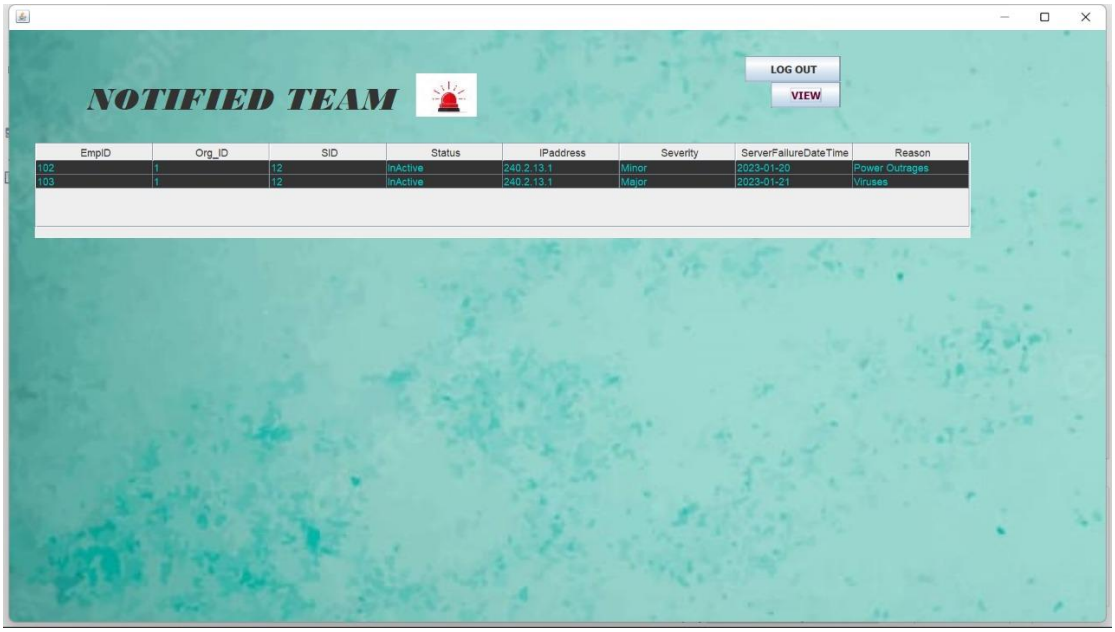

Figure 4.12 NOTIFIED TEAM

Figure 4.12 describes that the notified team will get the details of employee and server about the Inactive Server and its reason.

BACKEND

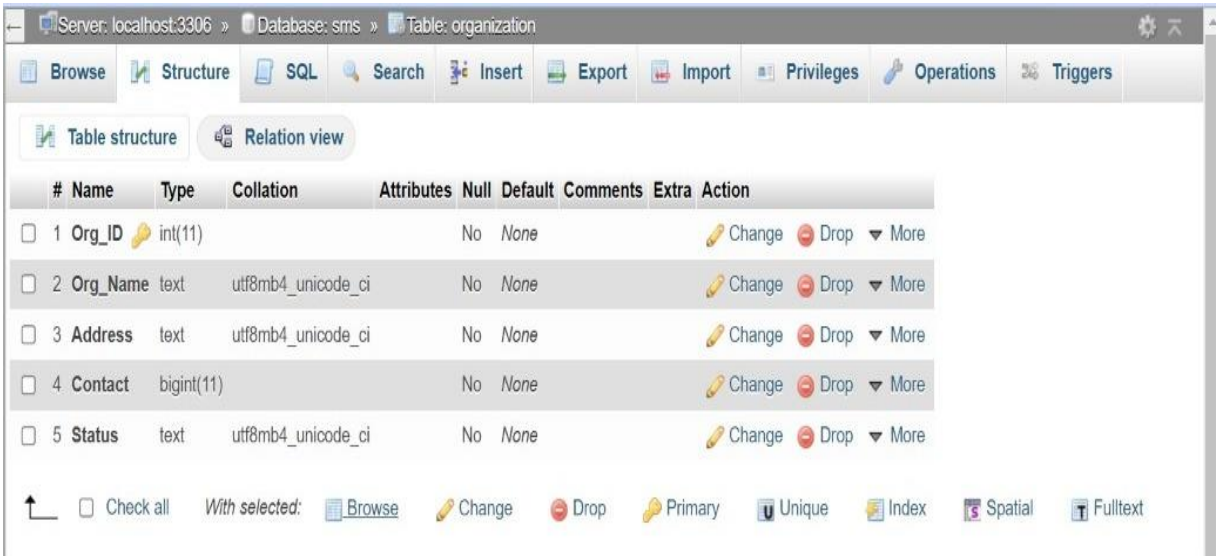

Figure 4.13 ORGANIZATION

Describes attributes, keys and attributes domain present in Organization table.

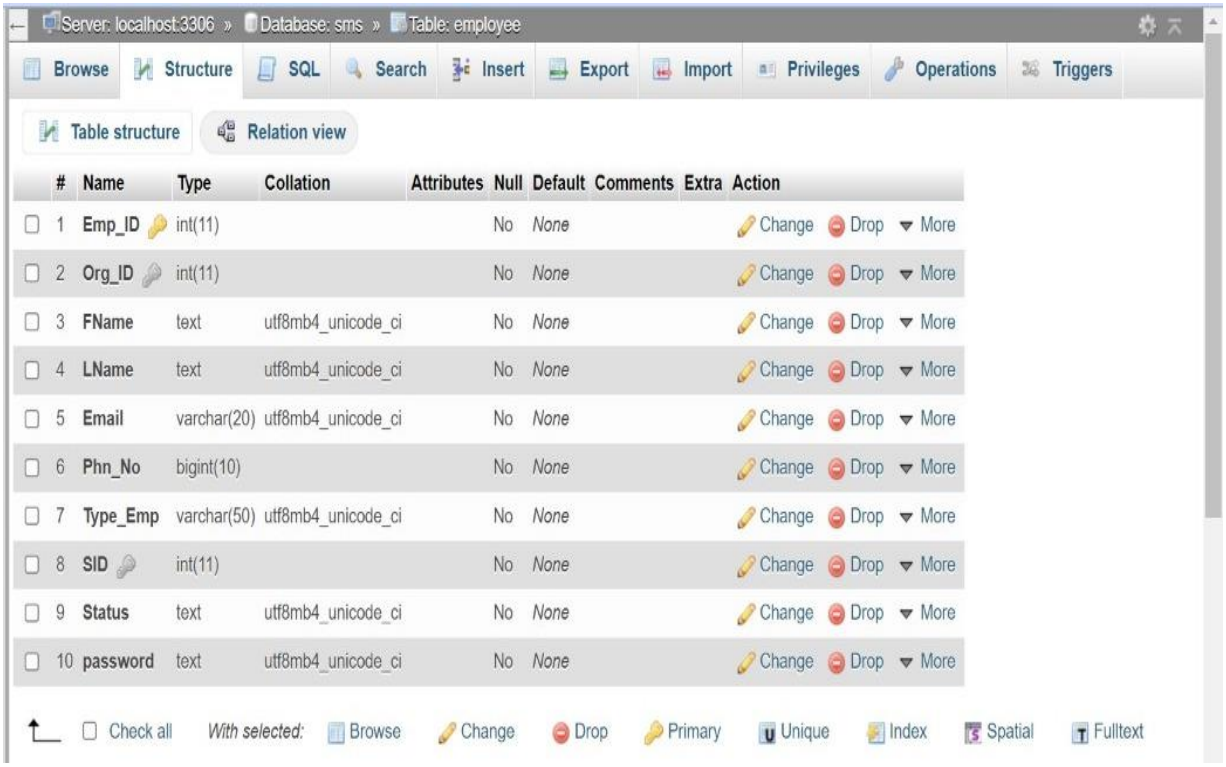

Figure 4.14 EMPLOYEE

Describes attributes, Keys and attributes domain present in Employee table.

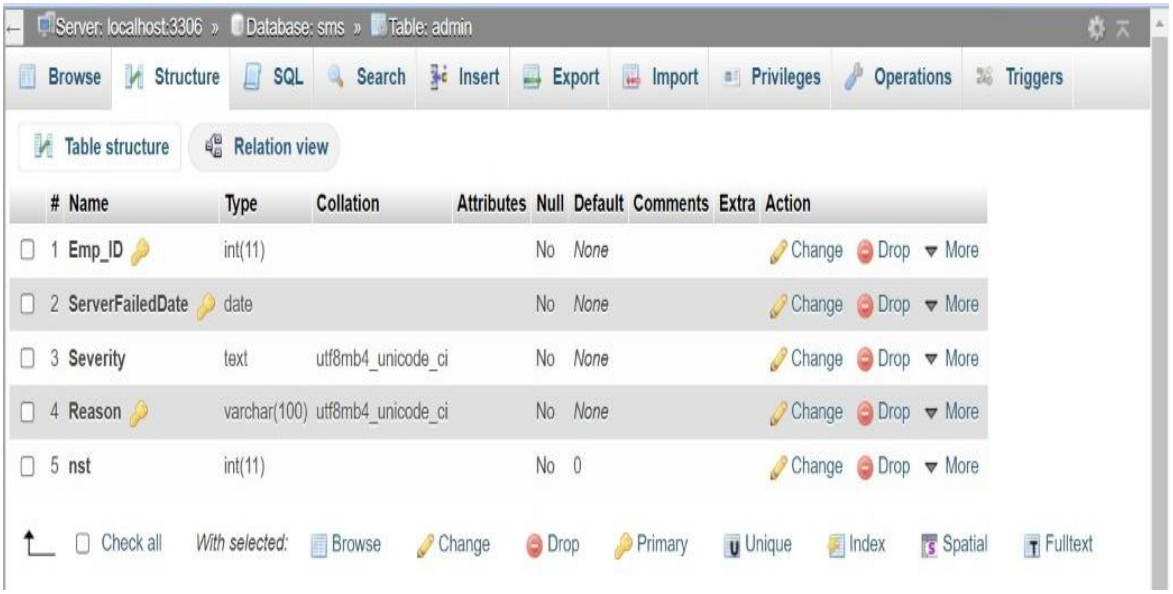

Figure 4.15 ADMIN

Describes attributes, Keys and attributes domain present in Admin table.

Figure 4.16 SERVER

Describes attributes, Keys and attributes domain present in Server table.

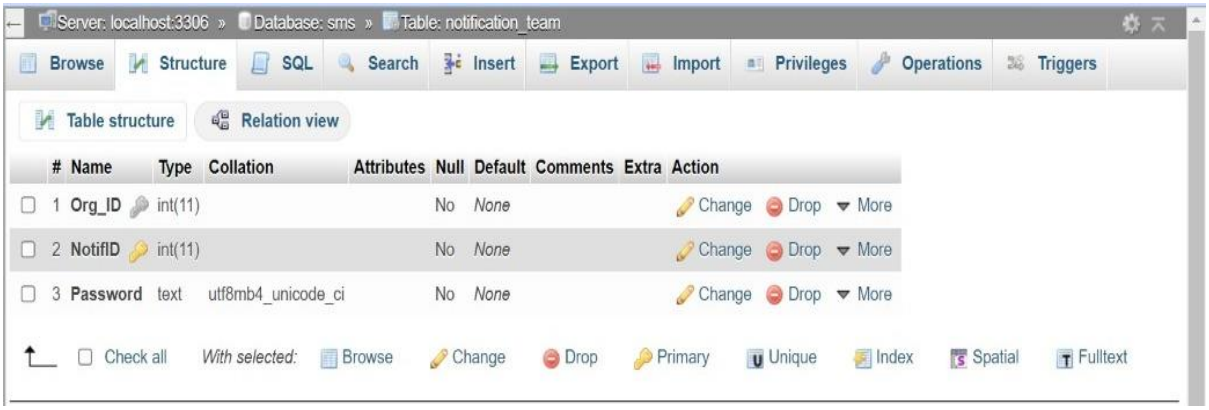

Figure 4.17 NOTIFICATION TEAM

Describes attributes, Keys and attributes domain present in Notification team table.

FRONTEND

INSERT

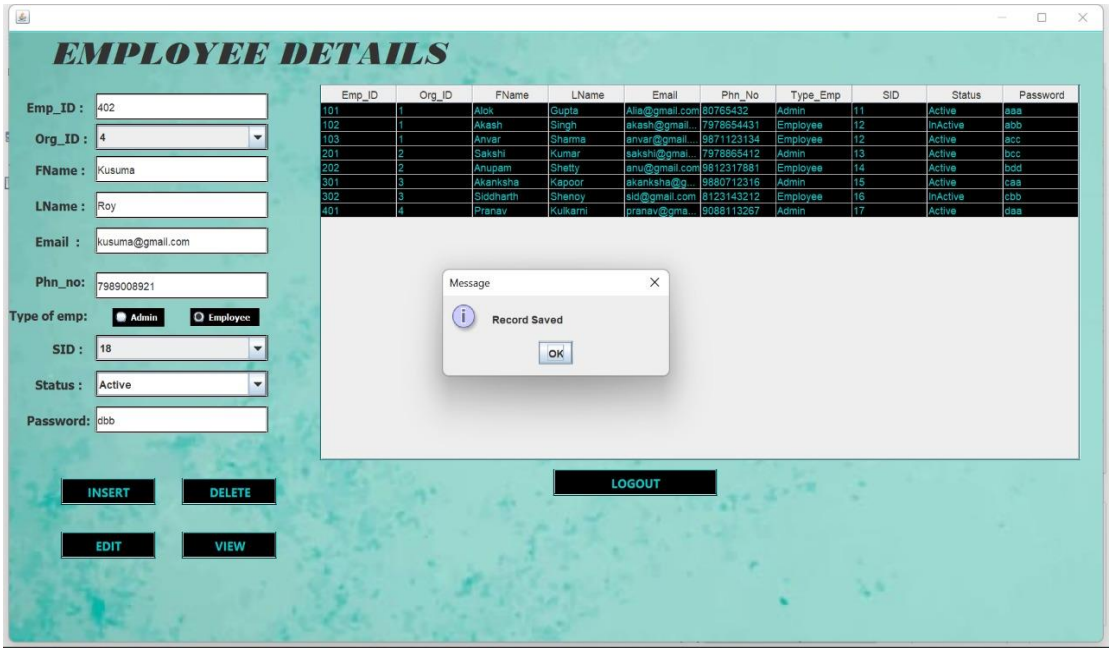

Figure 4.18 INSERTION

Describes that Record has been successfully inserted in the table.

DELETE

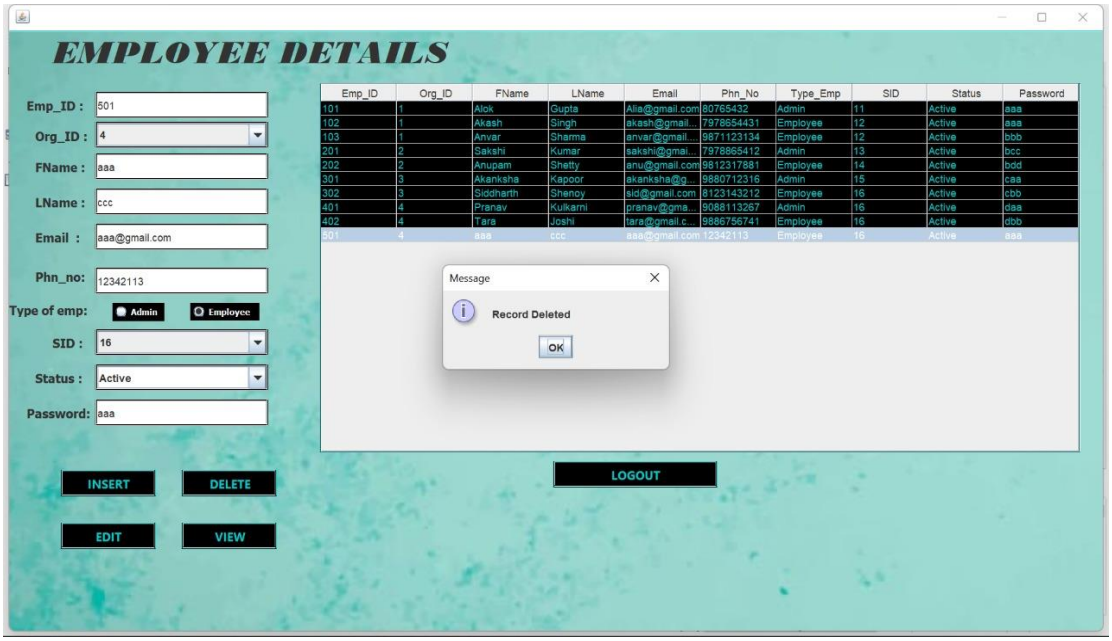

Figure 4.19 DELETION

Describes that Record has been deleted successfully from the table.

UPDATE

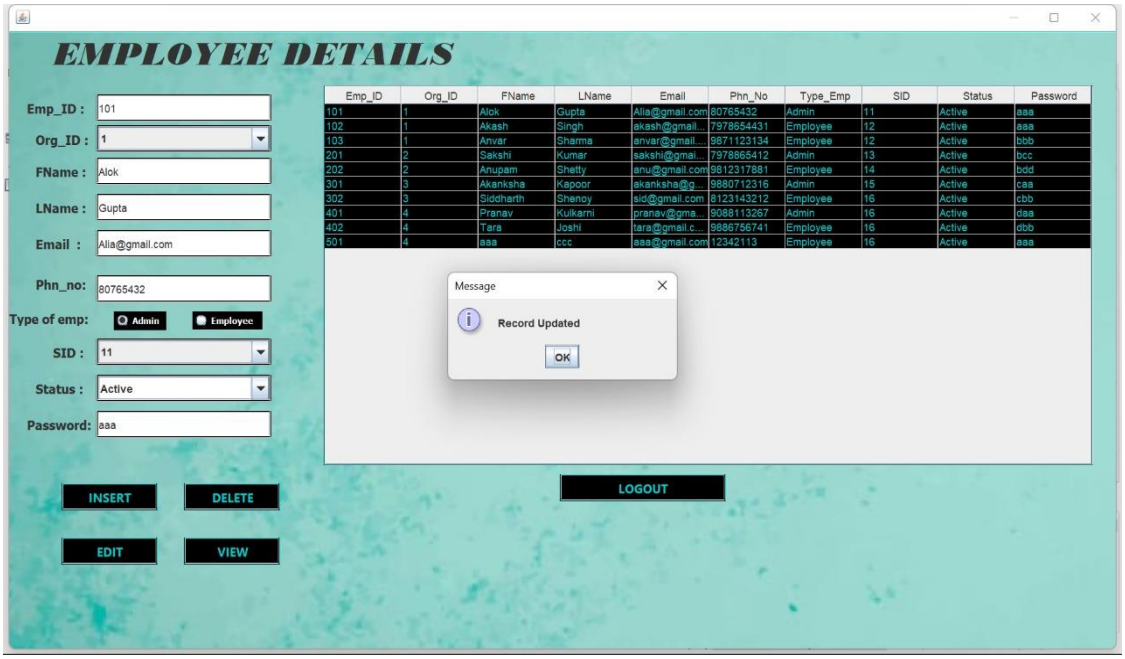

Figure 4.20 UPDATE

Describes that Record has been successfully updated.

SEARCH

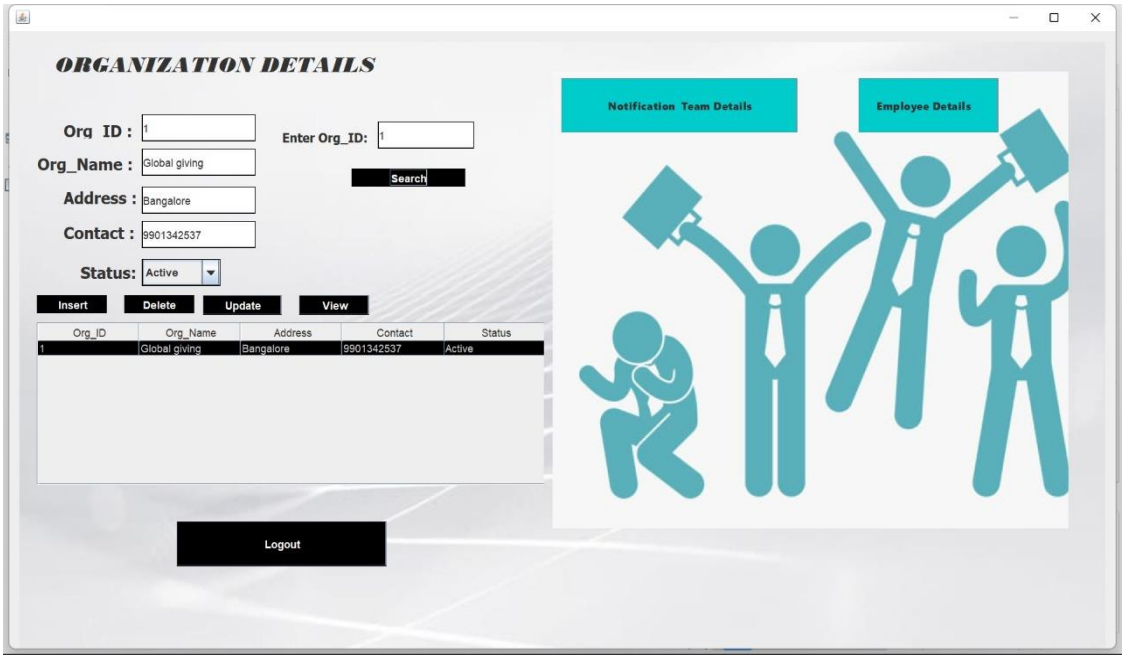

Figure 4.21 SEARCH

Describes that Records can be searched using search button.

TRIGGER

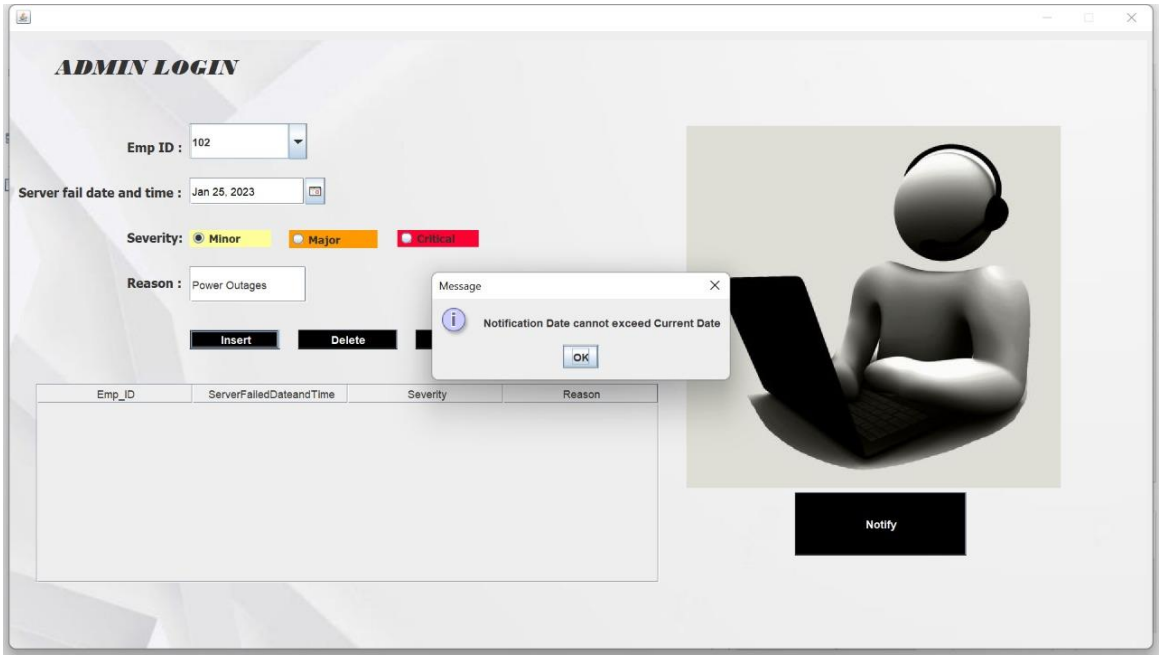

Figure 4.22 TRIGGER

Describes that while inserting date shouldn't exceed the current date, if it exceeds then warning message will popup.

CONCLUSION

It has been a great pleasure for us to work on this exciting and challenging project. This project proved good for us, and it provided practical knowledge of not only programming in java swings and some of applet-based applications and know some extent windows application and SQL server, and about handling procedure related with “SERVER NOTIFICATION MANAGEMENT SYSTEM”.

This also provides knowledge about the latest technology used in developing web enabled applications and client server technology that will be great demand in future. Database monitoring is vital activity for the maintenance of the performance and health of your database management system. It refers to the task associated with examining the operational status of your database.

Today management is one of the most essential features of all form. Management provides sophistication to perform any kind of task in particular way. This is Server Notification Management System, Manages the performance helps to maintain standards of organization.

Future Scope

The future scope of this project Server Notification Management System is very wide. The future version of system there is some point that we may implement on them:

Server Management Plays an important role in Organizations. As there is a steep increase in dependency of server , the need for server speed and high availability of server, there is a great scope for a network Engineers Professionals in India.

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.