EMPLOYEE MANAGEMENT SYSTEM

Submitted for partial fulfillment of the requirements for the

BACHELOR OF COMPUTER APPLICATION (BCA)

BY

Priyam Sharma

Under the guidance of

Mr. Raj Kishor Chauhan

Assistant professor

Department of BCA and MCA

IIMT GROUP OF COLLEGES

CERTIFICATE

This is to certify that the project work entitled EMPLOYEE MANAGEMENT SYSTEM is a Bonafide work carried out by Mr. PRIYAM SHARMA (210613106184) in partial fulfillment of the requirements for the award of degree of BACHELOR OF COMPUTER APPLICATION by CCS UNIVERSITY, Meerut, under our guidance and supervision.

The results embodied in this report have not been submitted to any other university or institute for the award of any degree or diploma.

Mr. Raj Kishor Chauhan

Assistant Professor

Student's Declaration

We hereby declare that the work being presented in this report entitled "EMPLOYEE MANAGEMENT SYSTEM" is an authentic record of our own work carried out under the supervision of Dr. /Mr. /Ms. RAJ KISHOR CHAUHAN .
The matter embodied in this report has not been submitted by us for the award of any other degree.

Dated : 15th Jan,2024

Signature of student
Mr. PRIYAM SHARMA
(BCA)

This is to certify that the above statement made by the candidate(s) is correct to the best of my knowledge.

Signature of HOD
Mr. Jitendra Kumar

(Bachelors of Computer Applications)
Date : 15th Jan,2024

Signature of Supervisor
Mr. Raj Kishor Chauhan

(Assistant Professor)
(BCA)

ACKNOWLEDGEMENT

I would like to express my sincerest gratitude and indebtedness to the person who gave me a moral and technical support & whose kind assistance has been instrumental in completion to this industrial training. It gives me immense pleasure to own me humble gratefulness to my faculty Mr. RAJ KISHOR CHAUHAN for this indispensable guidance and providing necessary ideas and facilities to carry out this project.

I would like to place on record my best regards and deepest sense of gratitude to Mr. JITENDRA KUMAR (Department Head), Mr. RAJ KISHOR CHAUHAN (Project Guide), of college name for their careful and precious guidance which was extremely valuable for my study both theoretically and practically.

Signature of Student
(PRIYAM SHARMA)

TABLE OF CONTENTS

S. No.	Topic	Page No.
1	Certification	2
2	Declaration	3
3	Acknowledgement	4
4	Abstract	7
5	Chapter-1: Introduction	7
6	Chapter-2: Objective	8
7	Chapter-3: Methodology	8
8	Chapter-4: Feasibility Study	10
9	Chapter-5: Implementation and Results	11-53
10	Chapter-6: Entity - Relationship Diagram	54
11	Chapter-7: Data Flow Diagram	55-56
12	Chapter-8: Testing	57
13	Chapter-9: Conclusion	58
14	Chapter-10: Future Scope	58
15	Chapter-11: References	59

TABLE OF FIGURES

Figure No.	Caption of the Figure	Page No.
1.	Front page	11
2.	Login frame	13
3.	Main Dashboard	16
4.	IT Department	19
5.	Finance Department	19
6.	Production Department	20
7.	Sales and marketing Department	20
8.	SQL Queries	23
9.	Tables in DB	23
10.	Add Employee	25
11.	Successfully Added message	25
12.	View Employee	32
13.	Update	37
14.	Successfully Updated message	37
15.	Remove Employee	44
16.	Successfully Removed message	44

Abstract

Human resource difficulties face all businesses, large and small. Because every organization has different staff management needs, we create custom employee management solutions that are tailored to your needs. This is intended to aid strategic planning and guarantee that your firm has the appropriate degree of human resources to meet your long-term objectives. This approach willhelp you to better manage your resources in the long run.

I. Introduction

Everything has been digitalized in our age of ever-increasing technology. The human workforce has grown as a result of the abundance of job options. As a result, a system that can handle the data of such a vast number of people in a company is required. Because of its user-friendly design, this project makes the process of keeping records easier. The "EMPLOYEE MANAGEMENT SYSTEM" was created to address the issues that plagued the previous manual system. This program is designed to eliminate, and in some cases, decrease, the problems that the current system has.

To eliminate data entry mistakes, the software is kept as simple as possible. When inputting incorrect data, it also displays an error notice. The user doesn't require any formal expertise to operate this system. The admin will be able to add new employees to this project. Employee data may also be seen and printed by the administrator. Admins can also remove an employee and change their details.

II. Objective

The objective of this work is to give a complete approach to personnel information management. This will be accomplished by developing and deploying an HR management system that will resultin a significant shift in the way employee data is managed.

This system's objectives include the following:

1. Design of an HR management system to meet needs such as adding and deletingemployees, viewing, and printing employee data, and updating employee information.
2. Employee data is stored in a well-designed database.
3. An easy-to-use interface that will let user interact with the system.

III. Methodology

The methodology to complete this project is as follows:

1. I explored net beans, concepts of swings and applets.

2. For further and a deeper understanding, I even referred to some articles, books, journals,websites and news articles.

Below are the important concepts on which the work has been done and with the support of theseI was able to work on my project.

NET BEANS- NetBeans is a Java-based integrated development environment (IDE). NetBeans enables the creation of applications using a set of modular software components known as modules. NetBeans is compatible with

Windows, Mac OS X, Linux, and Solaris.

It also allows other programming languages to be extended. In addition to Java programming, Third-party developers can expand NetBeans-based applications, including the NetBeans IDE.

JAVA- High-level, Object-Oriented programming language which help programmers to run theirapplications efficiently. JAVA is the programming language which comes into our minds when we talk about android application. By using JAVA as a programming language, programmer candevelop any type of android application easily.

JAVA also provides many libraries which also helps in making efficient android application. Swing is a Java GUI widget toolkit. It's part of Oracle's Java Foundation Classes (JFC), which provides an API for creating graphical-user- interfaces for Java programmers.

SWING- Swing is a Java GUI widget toolkit. It's part of Oracle's Java Foundation Classes (JFC),which provides an API for creating-graphical-user-interfaces for Java programmers. Swing was created to give a more advanced collection of graphical user interface components than the previous Abstract Window Toolkit (AWT).

Swing offers a pluggable look and feel that allows applications to have a look &feel that is unconnected to the underlying platform, as well as a look& feel that emulates the look & feel of numerous platforms.

SQL- SQL (Structured Query Language) is a computer language that is used to manage data in a
relational database management system (RDBMS) or for stream processing in a relational data stream management system (RDSMS). It's especially

beneficial for dealing with structured data,or data that has relationships between entities and variables.

IV. Feasibility Study

In order to do a feasibility study, we must consider the following:

1. Technical Feasibility

The availability of hardware & Software necessary for the creation of the system, as-well-as the compatibility and maturity of the technology planned to be used, and the availabilityof the requisite technical staff to create the system, are all factors to consider.

2. Operational Feasibility

Problems that may develop during operations are the focus of operation feasibility. There are two parts to this problem to consider:

- What are the chances that the solution provided will not be used or will not work?
- What is the inclination of the management and end users towards the solution? Though there is very least possibility of management being averse to the solution, there is a significant probability that the end users may not be interested in using the solution due to lack of training, insight etc.

3. Economic Feasibility

The concept of economic feasibility is determining whether or not the potential benefit of fixing difficulties is worthwhile. Because member needs &alternative solutions haven't been specified at this point, it is difficult to estimate the cost at this level

It is the measure of cost effectiveness of the project. The economic feasibility is nothing but judging whether the possible benefit of solving the problems is

worthwhile or not.

V. IMPLEMENTATION AND RESULT

Following are the screens of the Employee Management System where you can see all the featuresof this system in use and you can also see the GUI of the system:

1. **Front page** – This is the Entry point of our Software which will appear when we open the application. Inside this Window, the main title of the project is splashing on the top with a duration of 200ms and at the bottom , A button is placed named " Click here to Continue " with respect to its background image.

Fig 1

```
package employeemanagementsystem;

import java.awt.*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.*;


public class Splash extends JFrame  implements  ActionListener{
```

```
@Override
public void actionPerformed(ActionEvent e) {
    setVisible(false);
    new Login();
}

Splash(){
    getContentPane().setBackground(Color.WHITE);
    setLayout(null);


    JLabel heading =new JLabel("EMPLOYEE MANAGEMENT  SYSTEM");
    heading.setBounds(80, 30, 1200,60);
    heading.setFont(new  Font("serif",Font.PLAIN,60));
    heading.setForeground(Color.RED);
    add(heading);

    ImageIcon i1=new ImageIcon(ClassLoader.getSystemResource("icons/front.jpg"));
    Image i2=i1.getImage().getScaledInstance(1100, 700, Image.SCALE_DEFAULT);
    ImageIcon i3= new ImageIcon(i2);
    JLabel image=new JLabel(i3);
    image.setBounds(50, 100, 1080, 500);
    add(image);

    JButton clickHere=new JButton("Click Here To Continue");
    clickHere.setBounds(400, 400, 300, 70);
    clickHere.setBackground(Color.BLACK);
    clickHere.setForeground(Color.WHITE);
    clickHere.addActionListener(this);
    image.add(clickHere);

    setVisible(true);
    setSize(1170, 650);
```

```
        setLocation(200,50);

        while(true){
            heading.setVisible(false);
            try {
                Thread.sleep(400);
            } catch (Exception e) {

            }

            heading.setVisible(true);
            try {
                Thread.sleep(400);
            } catch (Exception e) {

            }
        }
    }

    public static void main(String[] args) {
        new Splash();
    }
}
```

2.Login frame – This is the login frame of this system where user have to enter the required Credentials to have access for the main dashboard.

The default Username is “admin” and Password is “123456”.

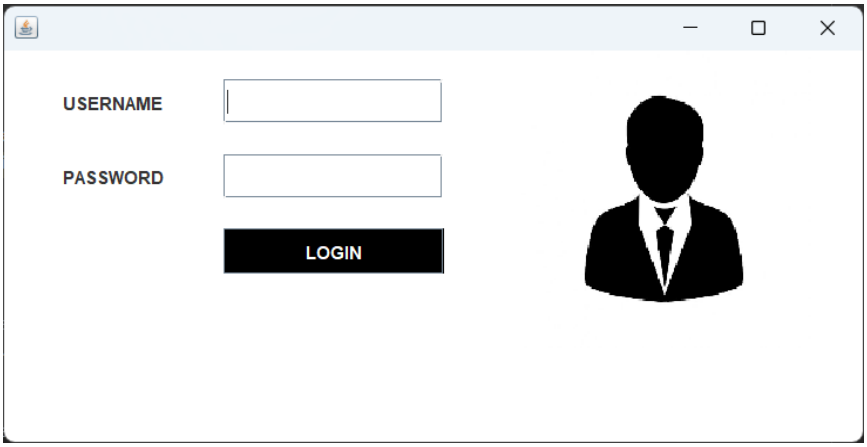


Fig 2

```
package employeemanagementsystem;
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.*;

public class Login extends JFrame implements ActionListener {

    @Override
    public void actionPerformed(ActionEvent ae) {
        try {
            String username=userText.getText();
            String password=passText.getText();
            DaoLayer dao = new DaoLayer();
            ResultSet rs =dao.LoginAuthentication(username, password);
            if(rs.next()){
                setVisible(false);
                new Home1();
            }
            else{
                JOptionPane.showMessageDialog(null, "invalid Username and Password");
            }
```

```
    } catch (Exception e) {
        e.printStackTrace();
    }
}
JTextField userText;
JTextField passText;

Login(){
    getContentPane().setBackground(Color.WHITE);
    setLayout(null);

    JLabel username=new JLabel("USERNAME ");
    username.setBounds(40,20, 100, 30);
    add(username);

     userText = new JTextField();
    userText.setBounds(150,20,150,30);
    add(userText);

    JLabel password=new JLabel("PASSWORD ");
    password.setBounds(40,70, 100, 30);
    add(password);

      passText = new JTextField();
    passText.setBounds(150,70,150,30);
    add(passText);

    JButton login=new JButton("LOGIN");
    login.setBounds(150, 120, 150, 30);
    login.setBackground(Color.BLACK);
    login.setForeground(Color.WHITE);
    login.addActionListener(this);
    add(login);
```

```
        ImageIcon i1=new ImageIcon(ClassLoader.getSystemResource("icons/second.jpg"));
        Image i2=i1.getImage().getScaledInstance(200, 200, Image.SCALE_DEFAULT);
        ImageIcon i3= new ImageIcon(i2);
        JLabel image=new JLabel(i3);
        image.setBounds(350, 0, 200, 200);
        add(image);

        setSize(600, 300);
        setLocation(450 ,250);
        setVisible(true);
    }
    public static void main(String[] args) {
        new Login();
    }
}
```

3.Main Dashboard – After login in, user is directed to the main dashboard of this system where you can select the department

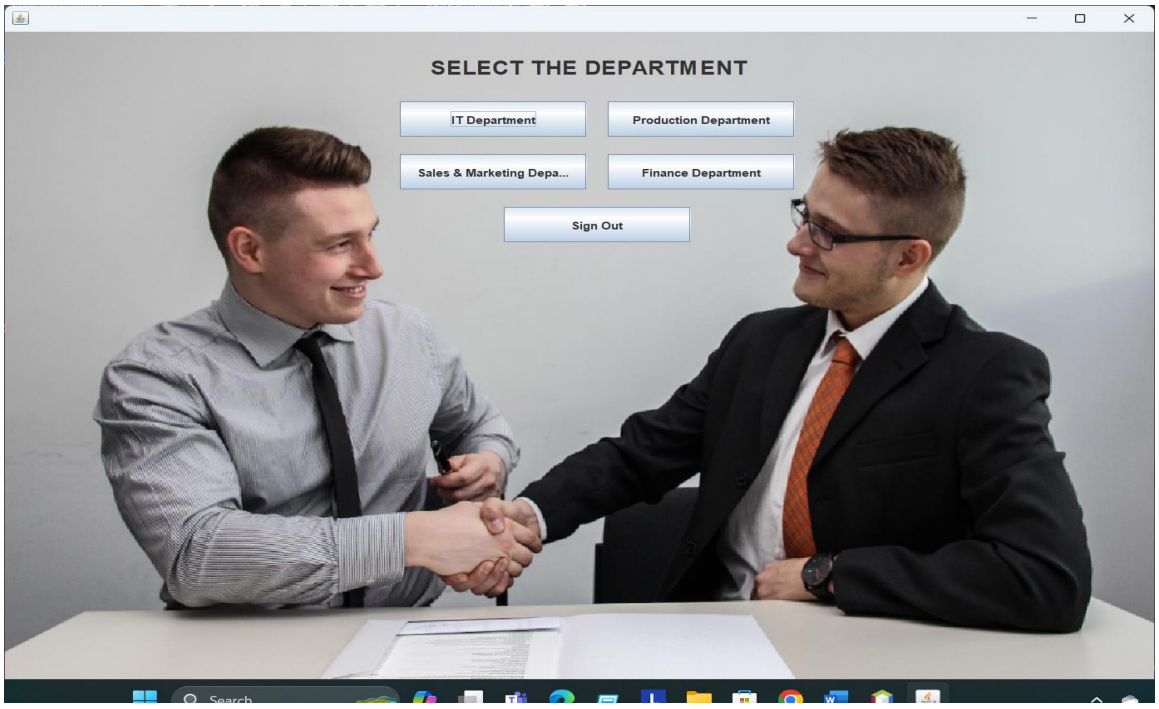


Fig 3

```
package employeemanagementsystem;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class Home1 extends JFrame  implements ActionListener  {
    JButton IT,prod,sales,finance,exit;
    @Override
    public void actionPerformed(ActionEvent ae) {
        if(ae.getSource() == IT){
            setVisible(false);
            String imgUrl ="icons/ITDept.jpg";
            String department = "IT Department";
            new Home2(imgUrl,department);
        }
        else if(ae.getSource() == prod){
            setVisible(false);
            String imgUrl ="icons/prod.jpg";
            String department = "Production Department";
            new Home2(imgUrl,department);
        }
        else if(ae.getSource() == sales){
             setVisible(false);
             String imgUrl ="icons/salesDept.jpg";
            String department = "Sales and Marketing Department";
            new Home2(imgUrl,department);
        }
        else if(ae.getSource()==exit){
            System.exit(0);
        }
        else{
            setVisible(false);
            String imgUrl ="icons/financeDept.jpg";
            String department = "Finance Department";
```

```
            new Home2(imgUrl,department);
        }
    }
    public  Home1(){
        setLayout(null);

         ImageIcon i1=new ImageIcon(ClassLoader.getSystemResource("icons/dept.jpg"));
        Image i2=i1.getImage().getScaledInstance(1120, 800, Image.SCALE_DEFAULT);
        ImageIcon i3= new ImageIcon(i2);
        JLabel image=new JLabel(i3);
        image.setBounds(0, 0, 1120, 800);
        add(image);

        JLabel heading =new JLabel("SELECT THE DEPARTMENT");
        heading.setBounds(410, 20, 400, 40);
        heading.setFont(new Font("segou UI",Font.BOLD,22));
        image.add(heading);

         IT= new JButton("IT Department");
        IT.setBounds(380, 80 ,180, 40);
        IT.addActionListener(this);
        image.add(IT);

          prod= new JButton("Production Department");
        prod.setBounds(580, 80 ,180, 40);
         prod.addActionListener(this);
        image.add(prod);

          sales= new JButton("Sales & Marketing Department");
        sales.setBounds(380, 140 ,180, 40);
         sales.addActionListener(this);
        image.add(sales);

          finance= new JButton("Finance Department");
```

```
        finance.setBounds(580, 140 ,180, 40);
         finance.addActionListener(this);
        image.add(finance);

         exit= new JButton("Sign Out");
        exit.setBounds(480, 200 ,180, 40);
         exit.addActionListener(this);
        image.add(exit);

        setSize(1120, 800);
        setLocation(250, 50);
        setVisible(true);

    }
    public static void main(String[] args) {
        new Home1();
    }
}
```

4. **Interfaces of different departments:** You can add, view, remove or update employee details or can go back to Main menu.

i) IT Department-

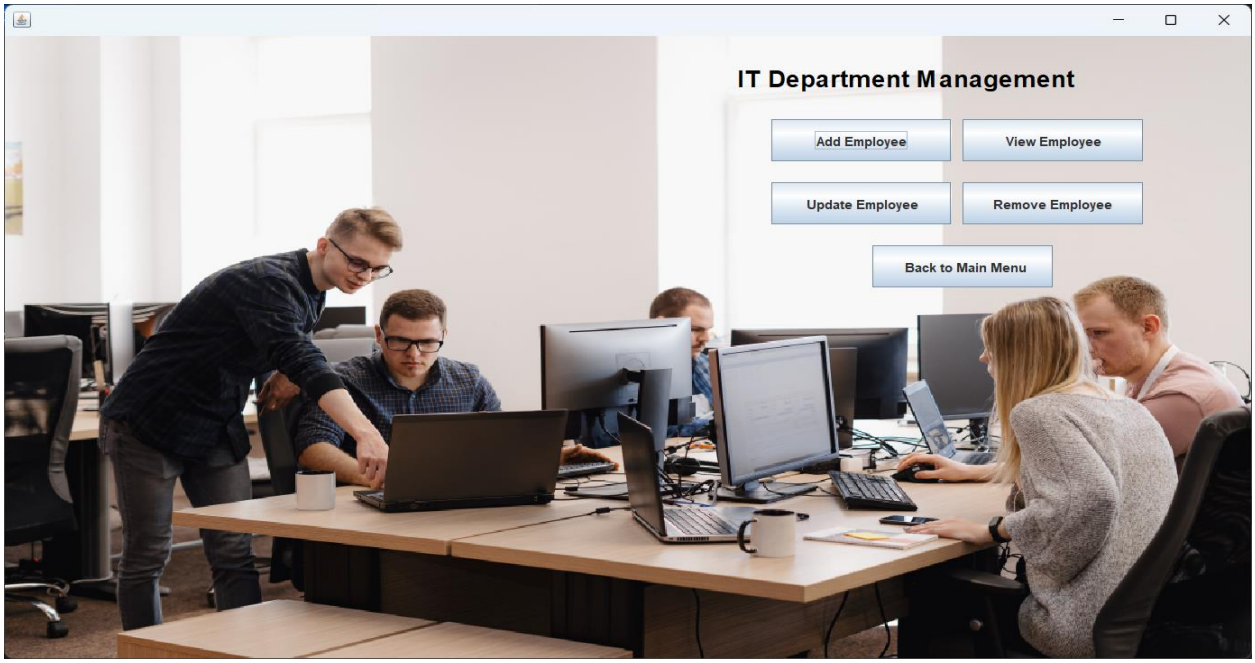


Fig 4

ii) Finance Department-

Fig 5

iii) Production Department-

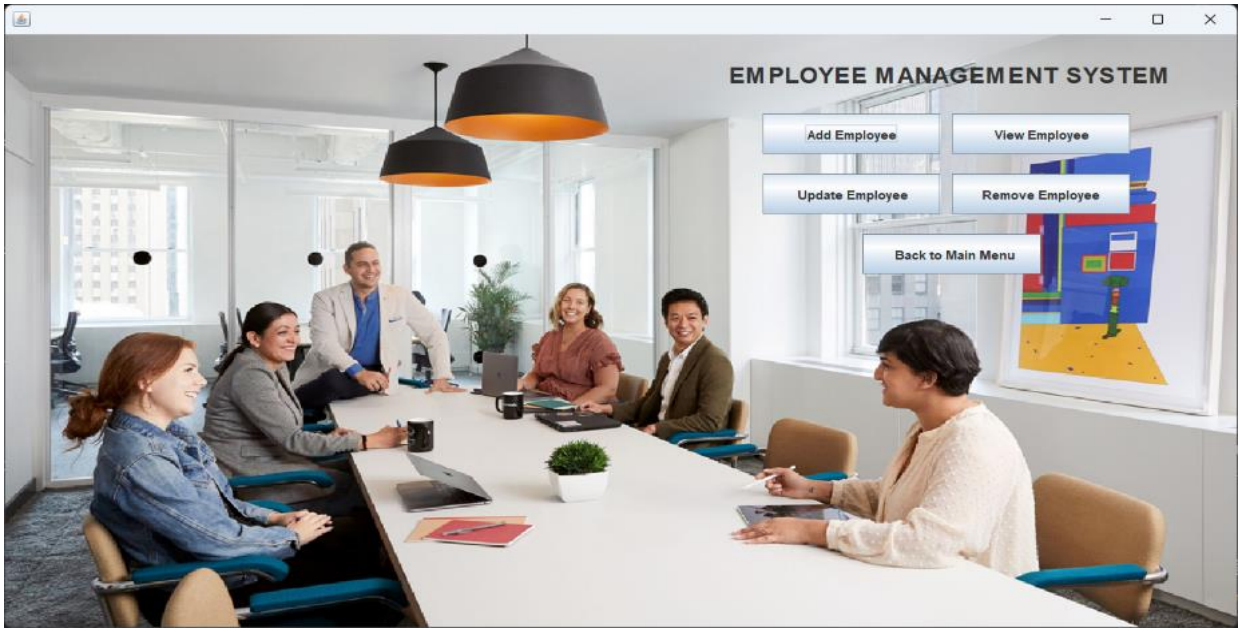


Fig 6

iv) Sales & marketing Department-

Fig 7

```
package employeemanagementsystem;

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class Home2 extends JFrame  implements ActionListener  {
    JButton add,view,update,remove,back;

    String imgUrl,department;

    public Home2(String imgUrl,String department){

        this.imgUrl=imgUrl;
        this.department=department;

        setLayout(null);

         ImageIcon i1=new ImageIcon(ClassLoader.getSystemResource(imgUrl));
        Image i2=i1.getImage().getScaledInstance(1120, 630, Image.SCALE_SMOOTH);
        ImageIcon i3= new ImageIcon(i2);
        JLabel image=new JLabel(i3);
        image.setBounds(0, 0, 1120, 630);
        add(image);

        JLabel heading =new JLabel(department+" Management");
        heading.setBounds(650, 20, 400, 40);
        heading.setFont(new Font("segou UI",Font.BOLD,22));
        heading.setForeground(Color.BLACK);
        image.add(heading);

         add= new JButton("Add Employee");
        add.setBounds(680, 80 ,160, 40);
```

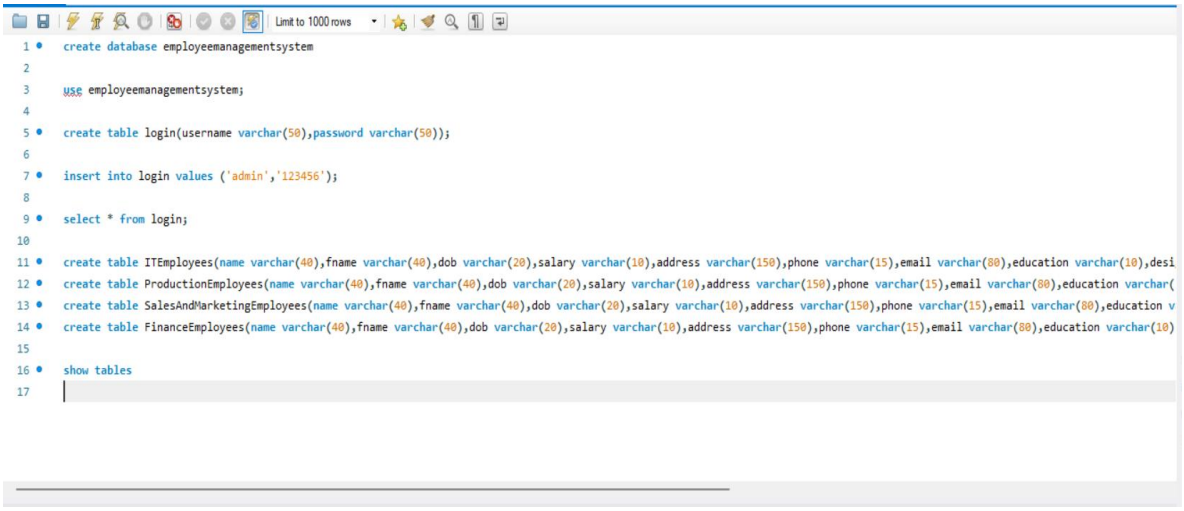
```
    add.addActionListener(this);
    image.add(add);

      view= new JButton("View Employee");
    view.setBounds(850, 80 ,160, 40);
     view.addActionListener(this);
    image.add(view);

      update= new JButton("Update Employee");
    update.setBounds(680, 140 ,160, 40);
     update.addActionListener(this);
    image.add(update);

      remove= new JButton("Remove Employee");
    remove.setBounds(850, 140 ,160, 40);
     remove.addActionListener(this);
    image.add(remove);

     back= new JButton(" Back to Main Menu");
    back.setBounds(770, 200 ,160, 40);
     back.addActionListener(this);
    image.add(back);

    setSize(1120, 630);
    setLocation(250, 100);
    setVisible(true);

}

    @Override
public void actionPerformed(ActionEvent ae) {
    if(ae.getSource() == add){
        setVisible(false);
        new AddEmployee(imgUrl, department);
```

```
        }
        else if(ae.getSource() == view){
            setVisible(false);
            new ViewEmployee(imgUrl, department);
        }
        else if(ae.getSource() == update){
             setVisible(false);
            new ViewEmployee(imgUrl, department);
        }
        else if (ae.getSource()==remove){
            setVisible(false);
            new RemoveEmployee(imgUrl, department);
        }
        else{
            setVisible(false);
            new Home1();
        }
    }

    public static void main(String[] args) {
            // new Home2();
    }
}
```

5. Operations-

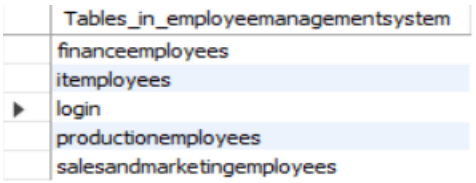
- **Connecting Database:** MySQL Database is used in backend for storing the data.

Below are the SQL queries used to create the database named "employeemanagementsystem" using MySQL Workbench.



```
create database employeemanagementsystem

use employeemanagementsystem;

create table login(username varchar(50),password varchar(50));

insert into login values ('admin','123456');

select * from login;

create table ITEmployees(name varchar(40),fname varchar(40),dob varchar(20),salary varchar(10),address varchar(150),phone varchar(15),email varchar(80),education varchar(10),desi
create table ProductionEmployees(name varchar(40),fname varchar(40),dob varchar(20),salary varchar(10),address varchar(150),phone varchar(15),email varchar(80),education varchar(
create table SalesAndMarketingEmployees(name varchar(40),fname varchar(40),dob varchar(20),salary varchar(10),address varchar(150),phone varchar(15),email varchar(80),education v
create table FinanceEmployees(name varchar(40),fname varchar(40),dob varchar(20),salary varchar(10),address varchar(150),phone varchar(15),email varchar(80),education varchar(10)

show tables
```

Fig. 8



Tables_in_employeemanagementsystem
financeemployees
itemployees
login
productionemployees
salesandmarketingemployees

Fig.9

Now, the code used to connect with the database is given below:

```
package employeemanagementsystem;

import java.sql.*;

public class Conn {

    public Connection c;

    public Statement s;


   public Conn(){

        try {

c=DriverManager.getConnection("jdbc:mysql://localhost:3306/employeemanagementsystem","root","root");
```

```
            s=c.createStatement();
            System.out.println("Connection Established");

        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

- **Data Access Object Layer (DAO Layer):** In this layer, All the functions are defined related to Create, Read, Update And Delete Operations .Each function Have a specific task which acts as an intermediary to communicate with the database .

```
package employeemanagementsystem;

import com.mysql.cj.util.StringUtils;
import java.sql.ResultSet;
import javax.swing.JOptionPane;

public class DaoLayer {

    public ResultSet LoginAuthentication(String username, String password) {
        try {
            Conn c = new Conn();
            String query = "select * from login where username = '" + username + "'and  password ='" + password +
"'";
            ResultSet rs = c.s.executeQuery(query);
            return rs;

        } catch (Exception e) {
            e.printStackTrace();
            return null;
        }
```

```
    }


    public void AddEmployee(String name, String fname, String dob, String salary, String address, String phone,
String email, String education, String designation, String aadhaar, String empid, String department) {

        String table =department.toLowerCase().replaceAll("\\s","") ;

        try {

            Conn c = new Conn();

            String query = "insert into "+table +" values ('" + name + "' , '" + fname + "' , '" + dob + "','" + salary + "','" +
address + "','" + phone + "','" + email + "','" + education + "','" + designation + "','" + aadhaar + "','" + empid + "')";

            c.s.executeUpdate(query);

            JOptionPane.showMessageDialog(null, "Details Added Succcessfully");

        } catch (Exception e) {

            e.printStackTrace();

        }

    }


    public ResultSet FetchEmployee(String department) {

        String table =department.toLowerCase().replaceAll("\\s","") ;

        try {

            Conn c = new Conn();

            ResultSet rs = c.s.executeQuery("select * from "+table);

            return rs;

        } catch (Exception e) {

            e.printStackTrace();

            return null;

        }

    }


    public ResultSet FetchEmployee(String empid,String department) {

        String table =department.toLowerCase().replaceAll("\\s","") ;

        try {

            Conn c = new Conn();

            ResultSet rs = c.s.executeQuery("select * from "+table+" where empid =" + empid);

            return rs;

        } catch (Exception e) {
```

```
            e.printStackTrace();

            return null;

        }

    }


    public void UpdateEmployee(String fname, String salary, String address, String phone, String email, String education, String designation, String empid,String department) {

        String table =department.toLowerCase().replaceAll("\\s","") ;

        try {

            Conn c = new Conn();

            String query = "update "+table +" set fname = '" + fname + "', salary = '" + salary + "', address = '" + address + "', phone = '" + phone + "', email =  '" + email + "', education = '" + education + "', designation = '" + designation + "' where empId = '" + empid + "'";

            c.s.executeUpdate(query);

            JOptionPane.showMessageDialog(null, "Details Updated Succcessfully");

        } catch (Exception e) {

            e.printStackTrace();

        }

    }


    public void RemoveEmployee(String empid,String department){

        String table =department.toLowerCase().replaceAll("\\s","") ;

        try {

                Conn c = new Conn();

                String query = "delete from "+table +" where empid = '" +empid + "'";

                c.s.executeUpdate(query);

                JOptionPane.showMessageDialog(null, "Employee Removed Successfully ");

        } catch (Exception e) {

            e.printStackTrace();

        }

    }

}
```

1. **Add Employee Details:** By clicking on “Add Employee” Button on previous menu, you will redirect to another window where you can add new employees by filling details of the employee and after clicking on “Add Employee” below message will appear.**10**

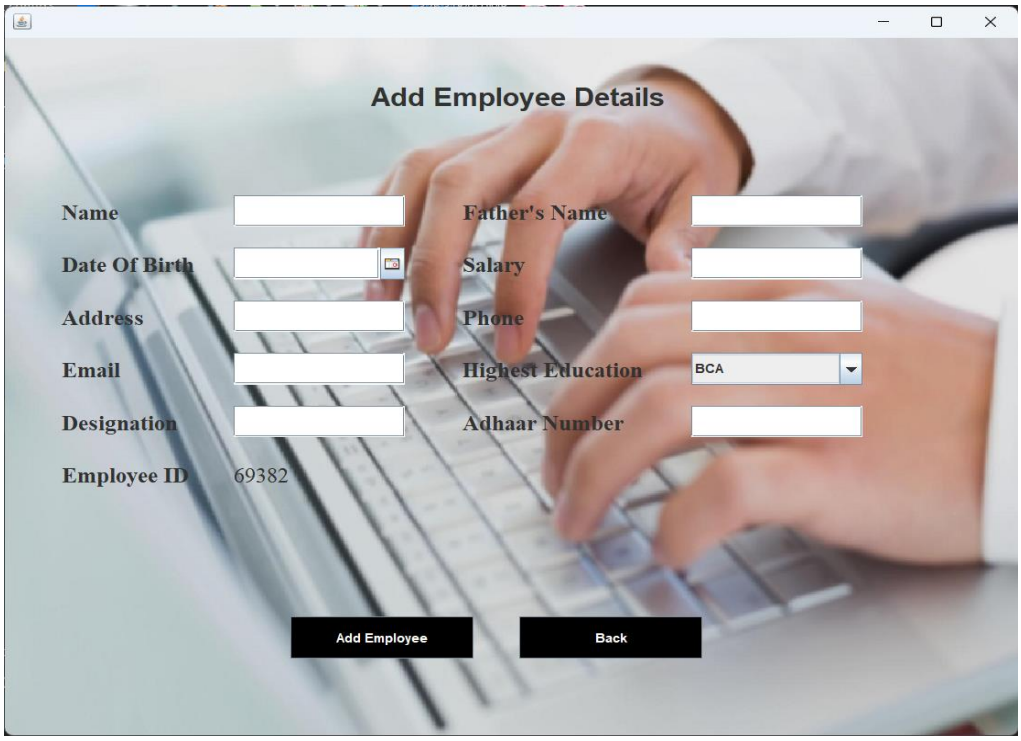


Fig 10

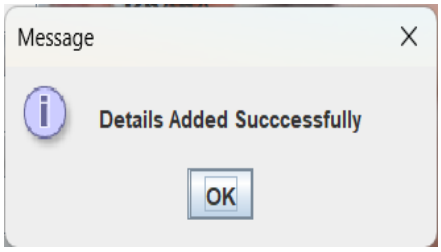


Fig 11

```
package employeemanagementsystem;

import java.awt.*;

import javax.swing.*;

import com.toedter.calendar.JDateChooser;

import java.awt.event.*;
```

```
import java.util.Random;


public class AddEmployee extends JFrame implements ActionListener {
    String imgUrl,department;


    JTextField tname, tfname, tsalary, taddress, tphone, temail, tdesignation, taadhaar;
    JLabel tempid;
    JDateChooser tdob;
    JComboBox teducation;
    JButton add, back;


    @Override
    public void actionPerformed(ActionEvent ae) {
        if (ae.getSource() == add) {
            String name = tname.getText();
            String fname = tfname.getText();
            String dob = ((JTextField) tdob.getDateEditor().getUiComponent()).getText();
            String salary = tsalary.getText();
            String address = taddress.getText();
            String phone = tphone.getText();
            String email = temail.getText();
            String education = (String) teducation.getSelectedItem();
            String designation = tdesignation.getText();
            String aadhaar = taadhaar.getText();
            String empId = tempid.getText();


            DaoLayer dao= new  DaoLayer();

dao.AddEmployee(name,fname,dob,salary,address,phone,email,education,designation,aadhaar,empId,departme
nt);
            setVisible(false);
            new Home2(imgUrl,department);


        } else {
            setVisible(false);
```

```
            new Home2(imgUrl,department);
        }
    }

    AddEmployee(String imgUrl,String department) {
        this.imgUrl=imgUrl;
        this.department=department;

        Random n = new Random();
        int number = n.nextInt(999999);

        getContentPane().setBackground(Color.white);
        setLayout(null);

        //heading
        JLabel heading = new JLabel("Add Employee Details");
        heading.setBounds(320, 30, 500, 50);
        heading.setFont(new Font("SAN_SERIF", Font.BOLD, 25));
        add(heading);

        //name
        JLabel name = new JLabel("Name");
        name.setBounds(50, 150, 150, 30);
        name.setFont(new Font("SERIF", Font.BOLD, 20));
        add(name);

        tname = new JTextField();
        tname.setBounds(200, 150, 150, 30);
        add(tname);

        //father name
        JLabel fname = new JLabel("Father's Name");
        fname.setBounds(400, 150, 150, 30);
        fname.setFont(new Font("SERIF", Font.BOLD, 20));
```

```
add(fname);

tfname = new JTextField();
tfname.setBounds(600, 150, 150, 30);
add(tfname);

//dob
JLabel dob = new JLabel("Date Of Birth");
dob.setBounds(50, 200, 150, 30);
dob.setFont(new Font("SERIF", Font.BOLD, 20));
add(dob);

tdob = new JDateChooser();
tdob.setBounds(200, 200, 150, 30);
add(tdob);

//salary
JLabel salary = new JLabel("Salary");
salary.setBounds(400, 200, 150, 30);
salary.setFont(new Font("SERIF", Font.BOLD, 20));
add(salary);

tsalary = new JTextField();
tsalary.setBounds(600, 200, 150, 30);
add(tsalary);

//address
JLabel address = new JLabel("Address");
address.setBounds(50, 250, 150, 30);
address.setFont(new Font("SERIF", Font.BOLD, 20));
add(address);

taddress = new JTextField();
taddress.setBounds(200, 250, 150, 30);
```

```
        add(taddress);


        //phone
        JLabel phone = new JLabel("Phone");
        phone.setBounds(400, 250, 150, 30);
        phone.setFont(new Font("SERIF", Font.BOLD, 20));
        add(phone);


        tphone = new JTextField();
        tphone.setBounds(600, 250, 150, 30);
        add(tphone);


//        email
        JLabel email = new JLabel("Email");
        email.setBounds(50, 300, 150, 30);
        email.setFont(new Font("SERIF", Font.BOLD, 20));
        add(email);


        temail = new JTextField();
        temail.setBounds(200, 300, 150, 30);
        add(temail);


        //education
        JLabel education = new JLabel("Highest Education");
        education.setBounds(400, 300, 170, 30);
        education.setFont(new Font("SERIF", Font.BOLD, 20));
        add(education);


        String courses[] = {"BCA", "B.Tech", "BBA", "BA", "B.Com", "B.Sc", "BJMC", "B.Ed", "MBA", "M.Tech",
"M.Sc", "PhD",};


        teducation = new JComboBox(courses);
        teducation.setBounds(600, 300, 150, 30);
        add(teducation);
```

```
//designation
JLabel designation = new JLabel("Designation");
designation.setBounds(50, 350, 150, 30);
designation.setFont(new Font("SERIF", Font.BOLD, 20));
add(designation);

tdesignation = new JTextField();
tdesignation.setBounds(200, 350, 150, 30);
add(tdesignation);

//aadhar
JLabel aadhaar = new JLabel("Adhaar Number");
aadhaar.setBounds(400, 350, 150, 30);
aadhaar.setFont(new Font("SERIF", Font.BOLD, 20));
add(aadhaar);

taadhaar = new JTextField();
taadhaar.setBounds(600, 350, 150, 30);
add(taadhaar);

//employee id
JLabel empid = new JLabel("Employee ID");
empid.setBounds(50, 400, 150, 30);
empid.setFont(new Font("SERIF", Font.BOLD, 20));
add(empid);

tempid = new JLabel("" + number);
tempid.setBounds(200, 400, 150, 30);
tempid.setFont(new Font("SERIF", Font.PLAIN, 20));
add(tempid);

add = new JButton("Add Employee");
add.setBounds(250, 550, 160, 40);
add.setBackground(Color.BLACK);
```

```
        add.setForeground(Color.WHITE);
        add.addActionListener(this);
        add(add);

        back = new JButton("Back");
        back.setBounds(450, 550, 160, 40);
        back.setBackground(Color.BLACK);
        back.setForeground(Color.WHITE);
        back.addActionListener(this);
        add(back);

         ImageIcon i1=new ImageIcon(ClassLoader.getSystemResource("icons/add_employee.jpg"));
        Image i2=i1.getImage().getScaledInstance(900, 700, Image.SCALE_SMOOTH);
        ImageIcon i3= new ImageIcon(i2);
        JLabel image=new JLabel(i3);
        image.setBounds(0, 0, 900, 700);
        add(image);

        setSize(900, 700);
        setLocation(300, 50);
        setVisible(true);
    }

    public static void main(String[] args) {
//        new AddEmployee();
    }
}
```

i) **View Employee Details:** You can view the list of employees of department selected previously in tabular format.

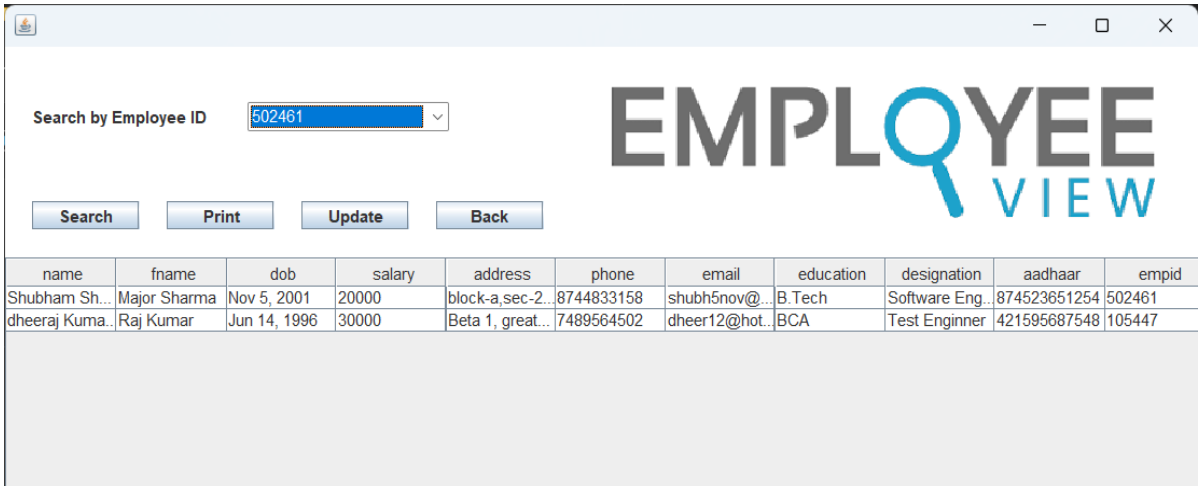


Fig 13

- **Search Employee Details:** You can Search an Employee's details by their unique Employee ID.

- **Print Employee Details:** You can Print employee details in PDF format by clicking Print using your Default Windows Printer.

- **Update Employee Details:** You can also redirect to the Update operation by selecting the Employee ID in the above drop down menu and then click on "Update" button

- **Back:** This button redirects you to previous menu.

```
package employeemanagementsystem;


import java.awt.*;
import java.awt.event.*;
import java.awt.print.PrinterException;
import javax.swing.*;
import net.proteanit.sql.DbUtils;
import java.sql.*;
import java.util.logging.Level;
import java.util.logging.Logger;
```

```
public class ViewEmployee extends JFrame implements ActionListener {

    String imgUrl,department;

    JTable table;
    Choice employeeid;
    JButton Search, Print, Update, Back;
    DaoLayer dao= new DaoLayer();

    ViewEmployee(String imgUrl,String department) {
        this.imgUrl=imgUrl;
        this.department=department;

        getContentPane().setBackground(Color.WHITE);
        setLayout(null);

        JLabel search = new JLabel("Search by Employee ID");
        search.setBounds(20, 40, 150, 20);
        add(search);

        employeeid = new Choice();
        employeeid.setBounds(180, 40, 150, 20);
        add(employeeid);

        try {
            ResultSet rs = dao.FetchEmployee(department);
            while (rs.next()) {
                employeeid.add(rs.getString("empid"));
            }

        } catch (Exception e) {
            e.printStackTrace();
        }
```

```
table = new JTable();
try {
    ResultSet rs = dao.FetchEmployee(department);
    table.setModel(DbUtils.resultSetToTableModel(rs));

} catch (Exception e) {
    e.printStackTrace();
}

JScrollPane jsp = new JScrollPane(table);
jsp.setBounds(0, 150, 900, 600);
add(jsp);

Search = new JButton("Search");
Search.setBounds(20, 110, 80, 20);
Search.addActionListener(this);
add(Search);

Print = new JButton("Print");
Print.setBounds(120, 110, 80, 20);
Print.addActionListener(this);
add(Print);

Update = new JButton("Update");
Update.setBounds(220, 110, 80, 20);
Update.addActionListener(this);
add(Update);

Back = new JButton("Back");
Back.setBounds(320, 110, 80, 20);
Back.addActionListener(this);
add(Back);
```

```
    ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("icons/viewemp.png"));
    Image i2 = i1.getImage().getScaledInstance(404, 125, Image.SCALE_DEFAULT);
    ImageIcon i3 = new ImageIcon(i2);
    JLabel image = new JLabel(i3);
    image.setBounds(450, 10, 404, 125);
    add(image);

    setSize(900, 700);
    setLocation(300, 100);
    setVisible(true);
}

@Override
public void actionPerformed(ActionEvent ae) {
    if (ae.getSource() == Search) {
        try {
            ResultSet rs = dao.FetchEmployee(employeeid.getSelectedItem(),department);
            table.setModel(DbUtils.resultSetToTableModel(rs));
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    else if (ae.getSource() == Print) {
        try {
            table.print();
        } catch (PrinterException ex) {
            Logger.getLogger(ViewEmployee.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

    else if (ae.getSource() == Update) {
        setVisible(false);
        new UpdateEmployee(employeeid.getSelectedItem(),imgUrl,department);
```

```
        } else {
            setVisible(false);
            new Home2(imgUrl,department);
        }
    }

    public static void main(String[] args) {
//        new ViewEmployee();
    }
}
```

ii) **Update Employee Details:** You can make updates in employee details using employee id by Simply clicking on “Update Employee” button . and then you will redirect to “View Employee page ” from where you can search by Employee ID and click on “Update” button .This will show you the window which has Existing details of an Employee and Now you can Edit the Selected Employee Details.

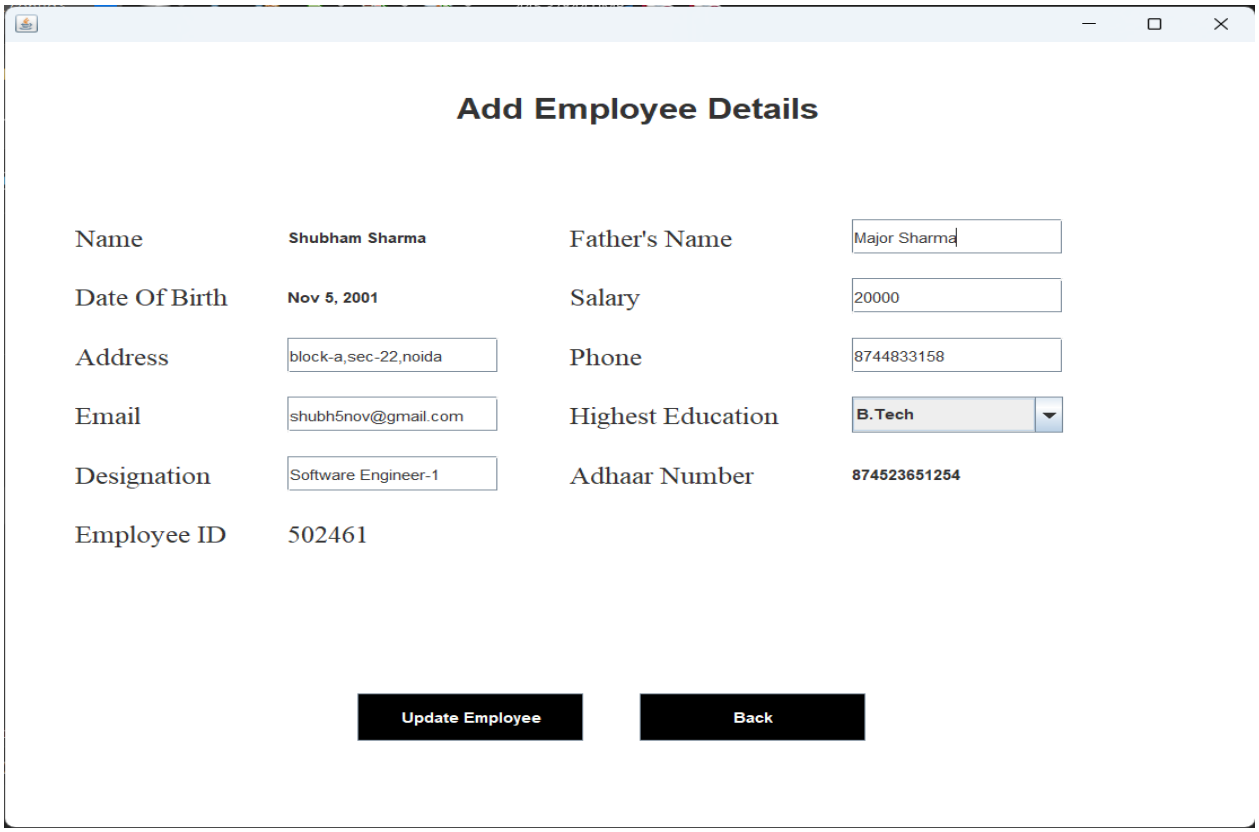


Fig 14

After Clicking on “Update Employee” button , this message will pop up.

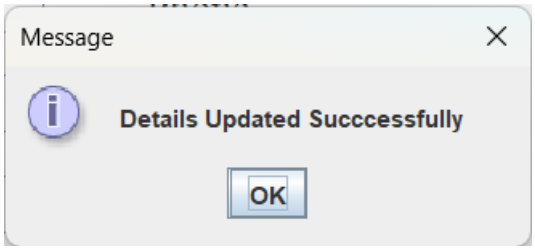


Fig 15

```
package employeemanagementsystem;

import java.awt.*;

import javax.swing.*;

import java.awt.event.*;

import java.sql.*;


public class UpdateEmployee extends JFrame implements ActionListener {

    String imgUrl,department;


    JTextField tfname, tsalary, taddress, tphone, temail, tdesignation; // this variables are declared as they are editable

    JLabel tname, tdob, taadhaar, tempid;  // these are not editable

    JComboBox teducation;  // it is also an editable

    JButton update, back;

    String empid;


    DaoLayer dao=new DaoLayer();

    UpdateEmployee(String empid, String imgUrl, String department) {

        this.imgUrl =imgUrl;

        this.department =department;

        this.empid = empid;

        getContentPane().setBackground(Color.white);

        setLayout(null);


        //heading

        JLabel heading = new JLabel("Add Employee Details");

        heading.setBounds(320, 30, 500, 50);

        heading.setFont(new Font("SAN_SERIF", Font.BOLD, 25));

        add(heading);
```

```
//name
JLabel name = new JLabel("Name");
name.setBounds(50, 150, 150, 30);
name.setFont(new Font("SERIF", Font.PLAIN, 20));
add(name);

tname = new JLabel();
tname.setBounds(200, 150, 150, 30);
add(tname);

//father name
JLabel fname = new JLabel("Father's Name");
fname.setBounds(400, 150, 150, 30);
fname.setFont(new Font("SERIF", Font.PLAIN, 20));
add(fname);

tfname = new JTextField();
tfname.setBounds(600, 150, 150, 30);
add(tfname);

//dob
JLabel dob = new JLabel("Date Of Birth");
dob.setBounds(50, 200, 150, 30);
dob.setFont(new Font("SERIF", Font.PLAIN, 20));
add(dob);

tdob = new JLabel();
tdob.setBounds(200, 200, 150, 30);
add(tdob);

//salary
JLabel salary = new JLabel("Salary");
salary.setBounds(400, 200, 150, 30);
```

```
        salary.setFont(new Font("SERIF", Font.PLAIN, 20));
        add(salary);

        tsalary = new JTextField();
        tsalary.setBounds(600, 200, 150, 30);
        add(tsalary);

        //address
        JLabel address = new JLabel("Address");
        address.setBounds(50, 250, 150, 30);
        address.setFont(new Font("SERIF", Font.PLAIN, 20));
        add(address);

        taddress = new JTextField();
        taddress.setBounds(200, 250, 150, 30);
        add(taddress);

        //phone
        JLabel phone = new JLabel("Phone");
        phone.setBounds(400, 250, 150, 30);
        phone.setFont(new Font("SERIF", Font.PLAIN, 20));
        add(phone);

        tphone = new JTextField();
        tphone.setBounds(600, 250, 150, 30);
        add(tphone);

//      email
        JLabel email = new JLabel("Email");
        email.setBounds(50, 300, 150, 30);
        email.setFont(new Font("SERIF", Font.PLAIN, 20));
        add(email);

        temail = new JTextField();
```

```
        temail.setBounds(200, 300, 150, 30);
        add(temail);

        //education
        JLabel education = new JLabel("Highest Education");
        education.setBounds(400, 300, 150, 30);
        education.setFont(new Font("SERIF", Font.PLAIN, 20));
        add(education);

        String courses[] = {"BCA", "B.Tech", "BBA", "BA", "B.Com", "B.Sc", "BJMC", "B.Ed", "MBA", "M.Tech",
"M.Sc", "PhD",};

        teducation = new JComboBox(courses);
        teducation.setBounds(600, 300, 150, 30);
        add(teducation);

        //designation
        JLabel designation = new JLabel("Designation");
        designation.setBounds(50, 350, 150, 30);
        designation.setFont(new Font("SERIF", Font.PLAIN, 20));
        add(designation);

        tdesignation = new JTextField();
        tdesignation.setBounds(200, 350, 150, 30);
        add(tdesignation);

        //aadhar
        JLabel aadhaar = new JLabel("Adhaar Number");
        aadhaar.setBounds(400, 350, 150, 30);
        aadhaar.setFont(new Font("SERIF", Font.PLAIN, 20));
        add(aadhaar);

        taadhaar = new JLabel();
        taadhaar.setBounds(600, 350, 150, 30);
        add(taadhaar);
```

```
//employee id
JLabel empidL = new JLabel("Employee ID");
empidL.setBounds(50, 400, 150, 30);
empidL.setFont(new Font("SERIF", Font.PLAIN, 20));
add(empidL);

tempid = new JLabel();
tempid.setBounds(200, 400, 150, 30);
tempid.setFont(new Font("SERIF", Font.PLAIN, 20));
add(tempid);

try {
    //this try catch is used to parse data from database
    ResultSet rs = dao.FetchEmployee(empid,department);
    System.out.println(rs);

    while (rs.next()) {
        // setting the data from database to these labels and textfields
        tname.setText(rs.getString("name"));
        tfname.setText(rs.getString("fname"));
        tdob.setText(rs.getString("dob"));
        tsalary.setText(rs.getString("salary"));
        taddress.setText(rs.getString("address"));
        tphone.setText(rs.getString("phone"));
        temail.setText(rs.getString("email"));
        tdesignation.setText(rs.getString("designation"));
        teducation.setSelectedItem(rs.getString("education"));
        taadhaar.setText(rs.getString("aadhaar"));
        tempid.setText(rs.getString("empid"));

    }
} catch (Exception e) {
    e.printStackTrace();
```

```
    }

    update = new JButton("Update Employee");
    update.setBounds(250, 550, 160, 40);
    update.setBackground(Color.BLACK);
    update.setForeground(Color.WHITE);
    update.addActionListener(this);
    add(update);

    back = new JButton("Back");
    back.setBounds(450, 550, 160, 40);
    back.setBackground(Color.BLACK);
    back.setForeground(Color.WHITE);
    back.addActionListener(this);
    add(back);

    setSize(900, 700);
    setLocation(300, 50);
    setVisible(true);
}

@Override
public void actionPerformed(ActionEvent ae) {
    if (ae.getSource() == update) {
        String fname = tfname.getText();
        String salary = tsalary.getText();
        String address = taddress.getText();
        String phone = tphone.getText();
        String email = temail.getText();
        String education = (String) teducation.getSelectedItem();
        String designation = tdesignation.getText();

        try {
            // this used to update the new details into database
```

```
            dao.UpdateEmployee(fname, salary, address, phone, email, education, designation,
empid,department);
            setVisible(false);
            new Home2(imgUrl,department);

        } catch (Exception e) {
            e.printStackTrace();
        }

    } else {
        setVisible(false);
        new Home2(imgUrl,department);
    }
  }

  public static void main(String[] args) {
//      new UpdateEmployee("");
  }
}
```

iii) **Remove Employee Details:** You can remove employee by Searching for a particular Employee ID , and their details will pop up below .

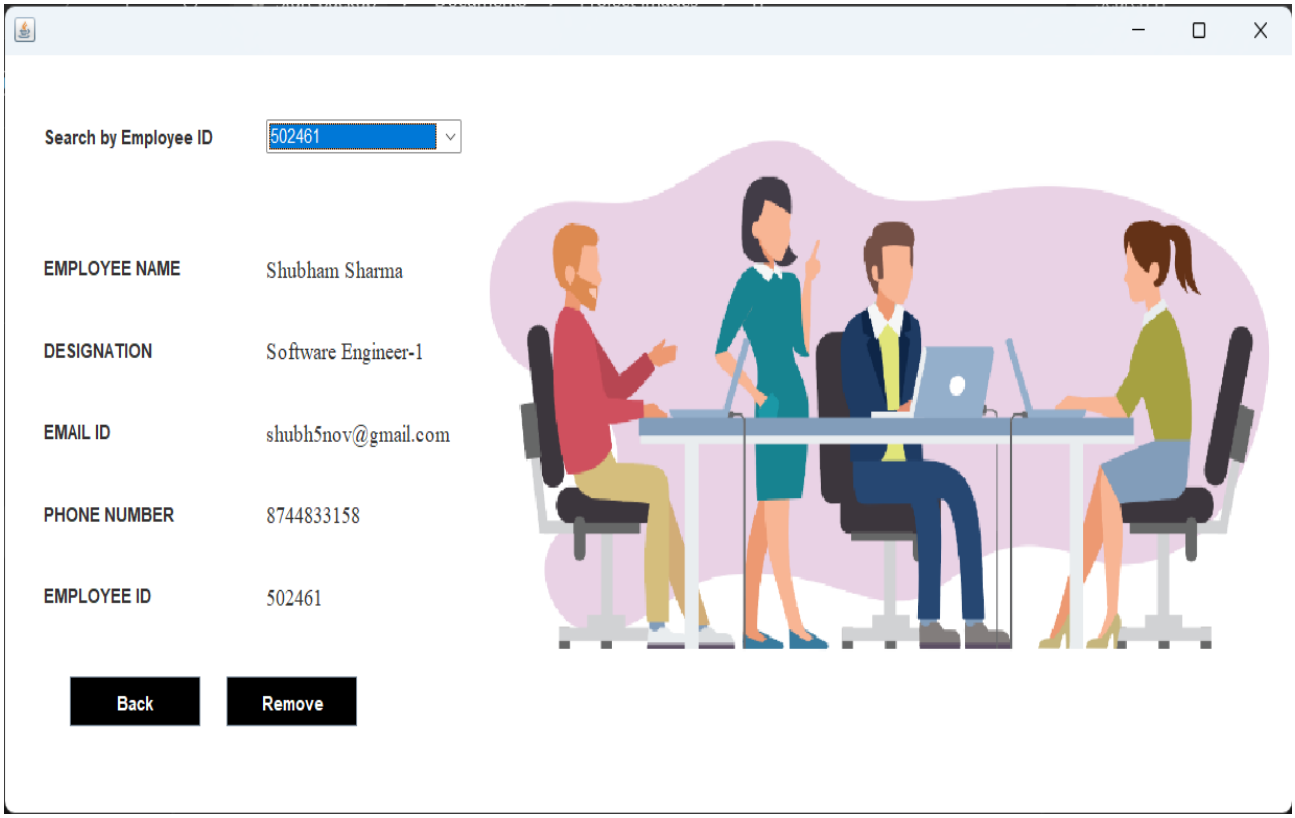


Fig 16

After clicking on “Remove ” Button , This message will pop up ,

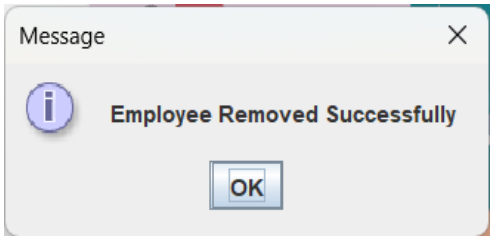


Fig 17

```
package employeemanagementsystem;


import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.sql.*;


public class RemoveEmployee extends JFrame implements ActionListener {
    String imgUrl,department;
```

```
    DaoLayer dao= new DaoLayer();

    Choice cempid;
    JButton back, remove;

    RemoveEmployee(String imgUrl,String department) {
        this.imgUrl=imgUrl;
        this.department=department;

        getContentPane().setBackground(Color.WHITE);
        setLayout(null);

        JLabel heading = new JLabel("Search by Employee ID");
        heading.setBounds(30, 40, 150, 20);
        add(heading);

        cempid = new Choice();
        cempid.setBounds(200, 40, 150, 20);
        add(cempid);

        try {
            ResultSet rs = dao.FetchEmployee(department);
            while (rs.next()) {
                cempid.add(rs.getString("empid"));
            }
        } catch (Exception e) {
            e.printStackTrace();
        }

        JLabel name = new JLabel("EMPLOYEE NAME ");
        name.setBounds(30, 120, 200, 20);
        add(name);

        JLabel tname = new JLabel();
```

```
        tname.setBounds(200, 120, 200, 20);
        tname.setFont(new Font("serif", Font.PLAIN, 15));
        add(tname);

        JLabel designation = new JLabel("DESIGNATION ");
        designation.setBounds(30, 170, 200, 20);
        add(designation);

        JLabel tdesignation = new JLabel();
        tdesignation.setBounds(200, 170, 200, 20);
        tdesignation.setFont(new Font("serif", Font.PLAIN, 15));
        add(tdesignation);

        JLabel email = new JLabel("EMAIL ID ");
        email.setBounds(30, 220, 200, 20);
        add(email);

        JLabel temail = new JLabel();
        temail.setBounds(200, 220, 200, 20);
        temail.setFont(new Font("serif", Font.PLAIN, 15));
        add(temail);

        JLabel phone = new JLabel("PHONE NUMBER ");
        phone.setBounds(30, 270, 200, 20);
        add(phone);

        JLabel tphone = new JLabel();
        tphone.setBounds(200, 270, 200, 20);
        tphone.setFont(new Font("serif", Font.PLAIN, 15));
        add(tphone);

        JLabel empid = new JLabel("EMPLOYEE ID ");
        empid.setBounds(30, 320, 200, 20);
        add(empid);
```

```
        JLabel tempid = new JLabel();
        tempid.setBounds(200, 320, 200, 20);
        tempid.setFont(new Font("serif", Font.PLAIN, 15));
        add(tempid);

        back = new JButton("Back");
        back.setBounds(50, 380, 100, 30);
        back.setBackground(Color.BLACK);
        back.setForeground(Color.WHITE);
        back.addActionListener(this);
        add(back);

        remove = new JButton("Remove");
        remove.setBounds(170, 380, 100, 30);
        remove.setBackground(Color.BLACK);
        remove.setForeground(Color.WHITE);
        remove.addActionListener(this);
        add(remove);

        try {
            ResultSet rs =dao.FetchEmployee(cempid.getSelectedItem(),department);
            while (rs.next()) {
                tname.setText(rs.getString("name"));
                tdesignation.setText(rs.getString("designation"));
                tphone.setText(rs.getString("phone"));
                temail.setText(rs.getString("email"));
                tempid.setText(rs.getString("empid"));
            }

        } catch (Exception e) {
            e.printStackTrace();
        }
```

```
        cempid.addItemListener(new ItemListener() {
            @Override
            public void itemStateChanged(ItemEvent ae) {
                try {
                    ResultSet rs = dao.FetchEmployee(cempid.getSelectedItem(),department);
                    while (rs.next()) {
                        tname.setText(rs.getString("name"));
                        tdesignation.setText(rs.getString("designation"));
                        tphone.setText(rs.getString("phone"));
                        temail.setText(rs.getString("email"));
                        tempid.setText(rs.getString("empid"));
                    }

                } catch (Exception e) {
                    e.printStackTrace();
                }
            }

        });

        ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("icons/delete.png"));
        Image i2 = i1.getImage().getScaledInstance(600, 400, Image.SCALE_DEFAULT);
        ImageIcon i3 = new ImageIcon(i2);
        JLabel image = new JLabel(i3);
        image.setBounds(370, 0, 600, 400);
        add(image);

        setSize(1000, 500);
        setLocation(300, 150);
        setVisible(true);
    }

    @Override
    public void actionPerformed(ActionEvent ae) {
```

```
        if (ae.getSource() == remove) {
            try {
                dao.RemoveEmployee(cempid.getSelectedItem(),department);
                setVisible(false);
                new Home2(imgUrl,department);
            } catch (Exception e) {
                e.printStackTrace();
            }
        } else {
            setVisible(false);
            new Home2(imgUrl,department);
        }
    }

    public static void main(String[] args) {
//        new RemoveEmployee();
    }
}
```

User Interface Design:

User Interface Design is concerned with the dialogue between a user and the computer. It is concerned with everything from starting the system or logging into the system to the eventually presentation of desired inputs and outputs.

The overall flow of screens and messages is called a dialogue.
The following steps are various guidelines for User Interface Design:

• The system user should always be aware of what to do next.

• The screen should be formatted so that various types of information, instructions and messages always appear in the same general display area.

• Message, instructions or information should be displayed long enough to allow the system user to read them.

• Use display attributes sparingly.

• Default values for fields and answers to be entered by the user should

be specified.

• A user should not be allowed to proceed without correcting an error.

• The system user should never get an operating system message or fatal error

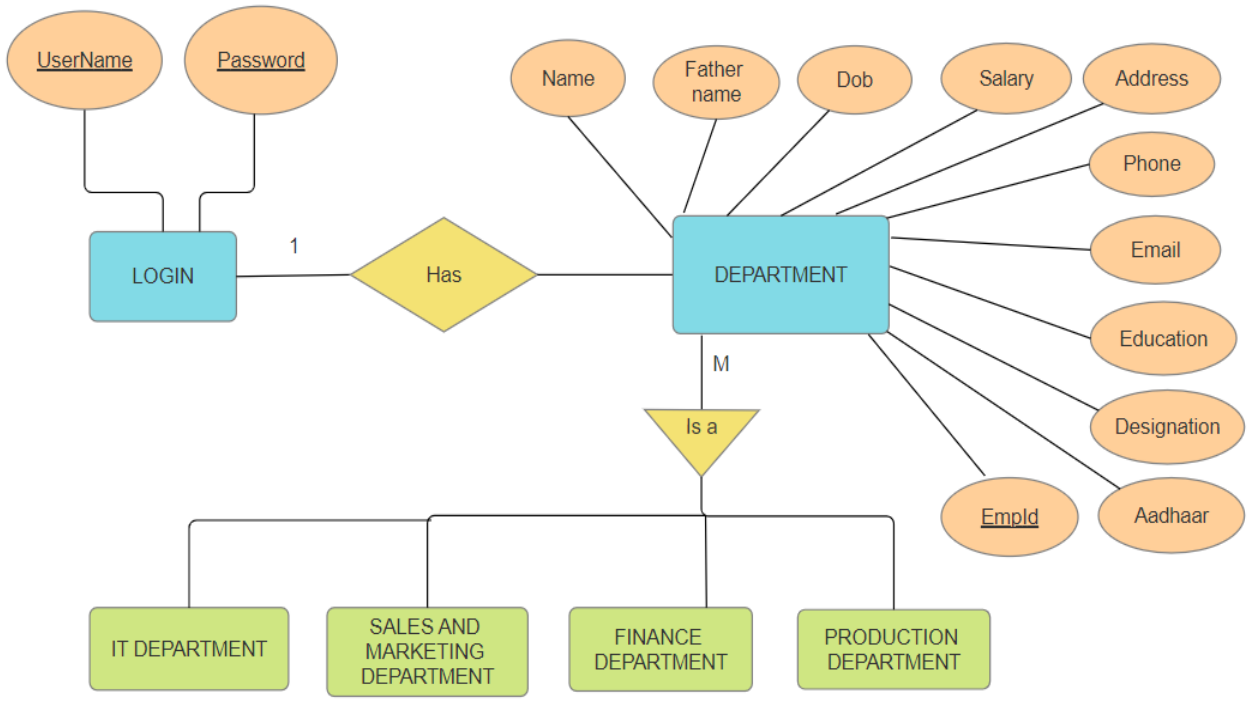
ER Diagram:

Entities are represented by rectangle.
Relation represented by diamond.
"Is a" relation represent the Specialization.
Ellipses symbolizes Attributes

It has following entities-

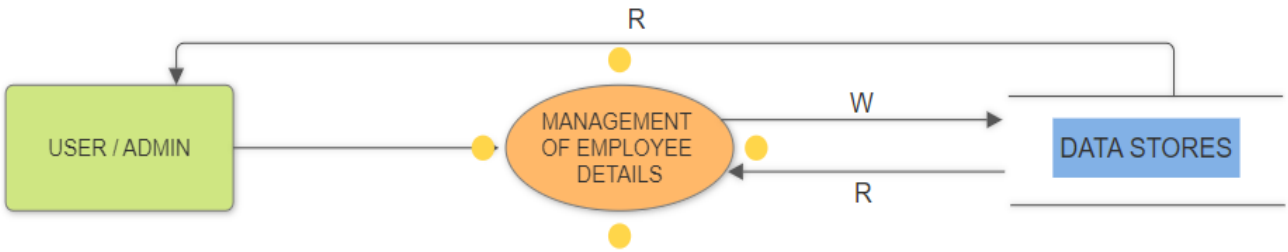
- Login
- Department
- IT Department
- Sales and Marketing Department
- Finance Department
- Production Department

Note:- " we use the concept of Specialization to divide the Departments into 4 sub-division."

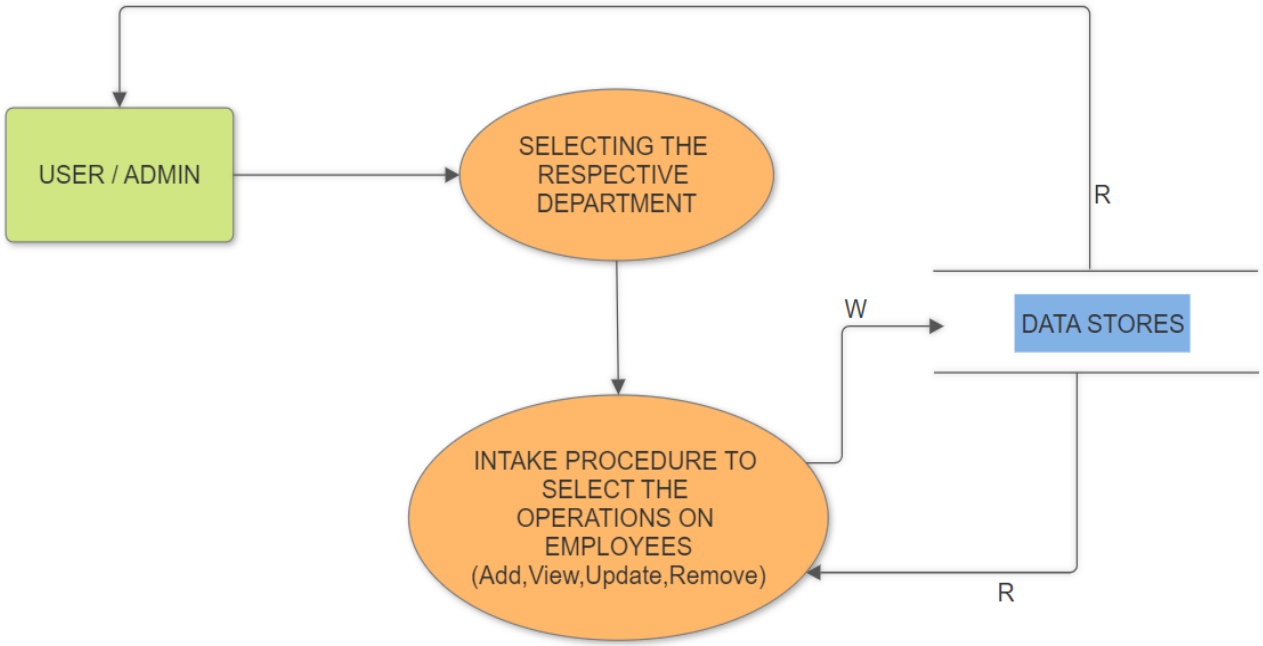


DATA FLOW DIAGRAM

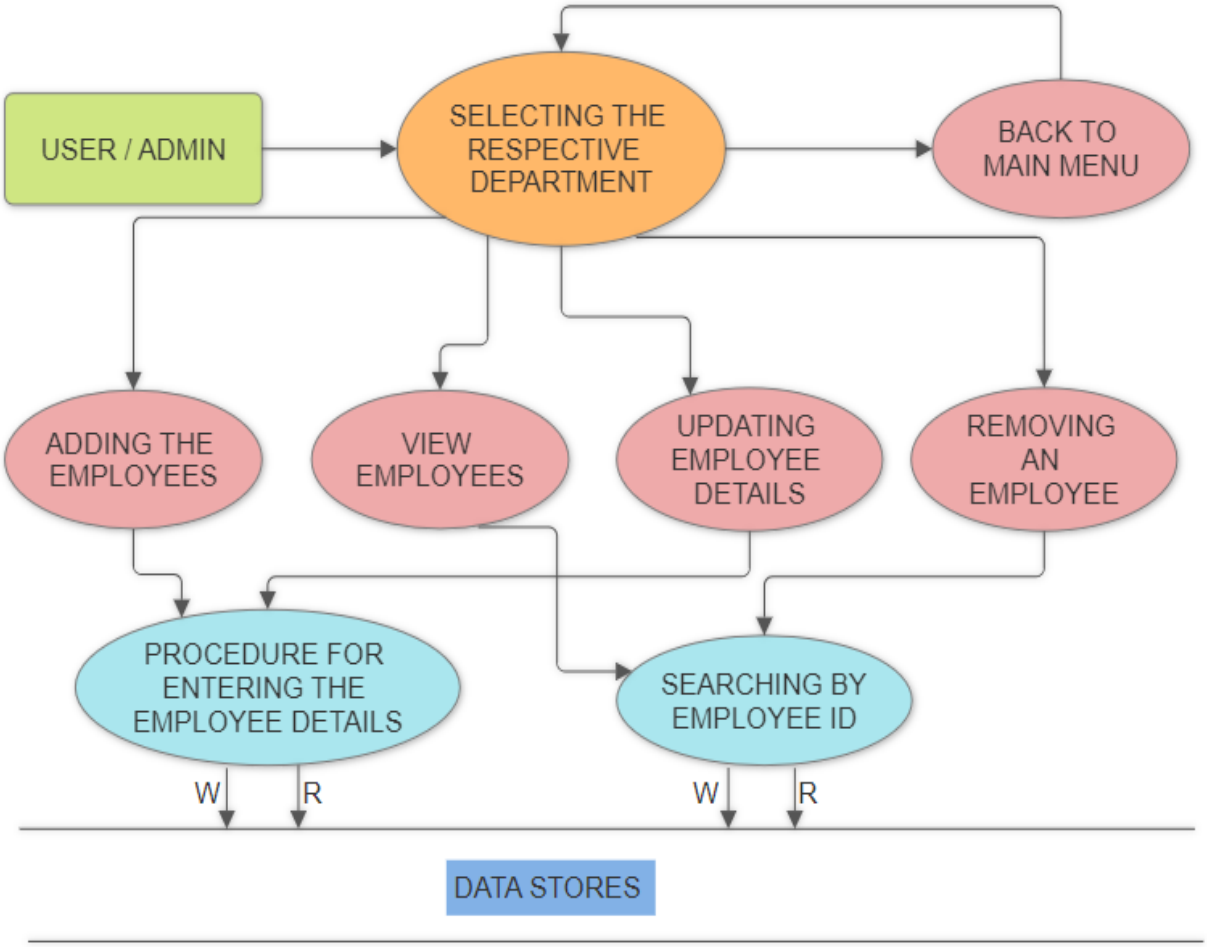
DFD AT ZERO LEVEL



DFD AT ONE LEVEL



DFD AT TWO LEVEL



I. Testing

Test ID	Test-Purpose	Test-Condition	Expected-Output	Output	Remark
TC1	Check Username & Password	If user details are not correct, display error message	Grant access to main dashboard.	Access granted tomain dashboard	Test successful
TC2	To add new user to the system	If user already exists, error message should bedisplayed.	New user should beadded.	New user added successfully	Test Successful
TC3	To view existing employee information	If employee exists, theninformation should be displayed, else error message should be displayed.	Employee information should be displayed.	Employee information displayed.	Test Successful
TC4	To remove an employee	If employee exists, thenemployee should be removed else error message should be displayed.	Employee should be removed.	Employee removed successfully.	Test Successful
TC5	Update employee information	If employee exists, theninformation should be updated.	Employee information should be updated.	Employee informationupdated successfully	Test Successful

Conclusion

The goal of the initiative is to digitalize personnel databases in businesses and provide administrator's access to computers. Employees and administrators use software as an information system. The user can store his or her database safe and secure for an indefinite amount of time here. Adding, deleting, accessing, and changing employee information is simple and easy using the Employee Management System.

Future Scope of Employee Management System

- It will help in storing and retrieval of employee information working in any organization.
- In a very short time the collection will be obvious, simple and sensible. It will reduce the cost of collecting the management and collection procedure.
- Our project based at Business Process automation
- In computer system, it is not necessary to create the manifest but we can directly print it, which saves our time.
- To utilize resources in an efficient manner by increasing their productivity through automation
- The System generates information which can be used for various purposes
- It satisfies the user requirement
- Be easy to understand by the user and operator
- In future, there will be a better Graphical User Interface and there will be more features added to this system. If Graphical User Interface is improved then this system will be more user friendly and more features added will make this system a lot better and HR will be able to perform more operations.

XVII. References

1. Renae Broderick, John W. Boudreau, “Human resource management, information technology, and the competitive edge”, Academy of Management Executive, 1992Vol. 6 No. 2

2. Julie Bulmash, “Human Resource Management and Technology”, Chapter 3.

3. Ian Sommerville, “Software Engineering”, 9th Edition, Addison-Wesley, 2011.\

4. Juan Manuel Munoz Palacio, Information systems development methodologies forData-driven Decision Support Systems, 2010.

5. Deitel, PJ & Deitel, HM, 2008, Internet & World Wide Web How to Program, Dorling Kindersley, India.

6. Google for problem solving.

7. http://www.javapoint.com/java-tutorial