

PCS AUTOMATION

Developed By

Name: K Naveenraj

Reg. No: S200040300164

NIIT

PCS AUTOMATION

Batch Code : S210167

Start Date : 12.11.2020

End Date : 12.12.2020

Name of the Coordinator : LOPAMUDRA BERA

Name of the Developer : K NAVEENRAJ

Date of Submission : 12.12.2020



CERTIFICATE

This is to certify that this report, titled **PCS AUTOMATION** embodies the original work done by **Mr.K Naveenraj**, in partial fulfillment of his course requirement at NIIT.

Coordinator : LOPAMUDRA BERA

ACKNOWLEDGEMENT

I would like to express my special thanks of gratitude to my teacher **LOPAMUDRA BERA** as well as to the education institute **NIIT** who gave me the golden opportunity to do this wonderful project on the topic **PROFESSIONET CONSULTANCY SERVICES**, which also helped me in doing a lot of Research and i came to know about so many new things I am really thankful to them.

Secondly i would also like to thank my parents and friends who helped me a lot in finalizing this project within the limited time frame.

ABSTRACT

This project manages the entire process of allocating project to existing employees as per his/her skillset working in that company

Its also provides single window system to Employee,HR and Project Managers of a company to cater the skill specific requirements emerging in Projects.

CONFIGURATION

Hardware:

- Laptop -i7
 - 16 Ram,2Tb Hard Disk

Operating System:Windows 10 x64

Software:JRE,Eclipse,MySql

TABLE OF CONTENTS

Chapter 1- Introduction

1.0 Aim

1.1 Objectives

Chapter 2- Project Requirement Specification

2.0 Literature Research

2.1 Statement of Requirements

Chapter 3- Project Analysis

3.0 Project Plan

3.1 System Architecture

3.2 Business Process Model

3.3 Software Requirement Specification

3.4 High Level Use Case Diagrams

Chapter 4- Project Design

4.0 Low Level Use Case Diagrams

4.1 User Interface Design

4.2 Systems Input and Output Design

4.3 Database Structure

4.4 Sample Code

Challenges

Observations

References

Appendix

AIM

To create an mapping application that operates via online recruiting website.

OBJECTIVES

Following are the objectives to be achieved through Skill Mapping Application

- User registration
- Skill map
- Job Postings
- Profile Validation
- Recruitment

This objectives are to be automated which were previously done manually.

LITERATURE RESEARCH

Introduction

To automate the existing recruitment procedures, which would help the HR Consultants to map the Job seekers with appropriate skills and offer workforce solutions to business needs.

Background Information

The recruitment and skill mapping were previously done manually by HR which would take weeks to sort the required user profile.

STATEMENT OF REQUIREMENTS

PROJECT OBJECTIVES

Title	PCS Automation
Subtitle	Employee Management System
Author(s)	K Naveenraj
Author's E-mail	naveenrajkrishnan@gmail.com
Author's Phone	9942543332
Description	Automated Consultancy Service
Version	1.01

About Your Company

Professionet Consultancy Services(PCS) is a business consultancy which provides a wide range of business services to clients.

Need for Process Automation

- Adds consistency and quality to recruitment
- Improves the productivity of HR team
- Saves time by easing the workload
- Enables organizations to find the right talent

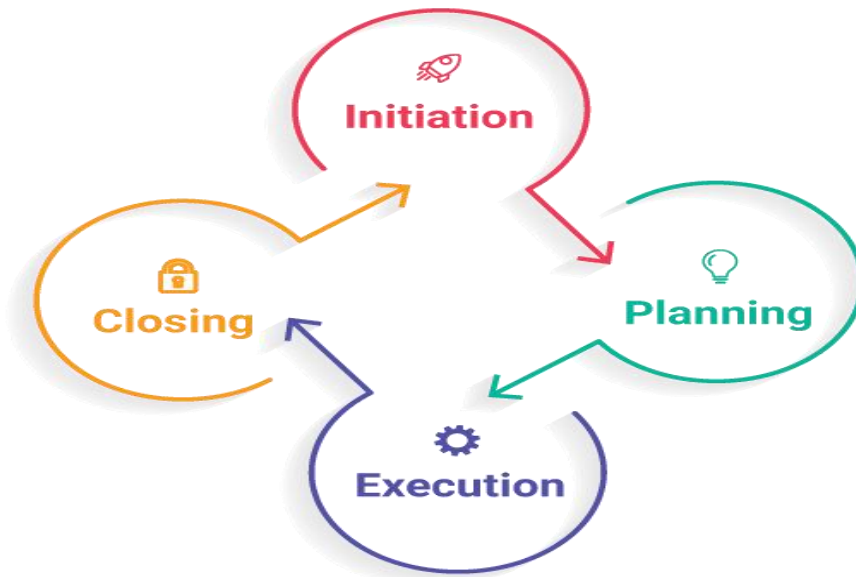
Software Requirements

- JRE
- ECLIPSE
- MYSQL

Benefits

- Improves the Recruitment Qualities.
- Pre-Screening of Candidates.

PROJECT LIFE CYCLE MODEL



The Initiation Phase: The initiation phase aims to define and authorize the project.

Vision: Automate the Selection Process

The Planning Phase: The purpose of this phase is to lay down a detailed strategy of how the project has to be performed and how to make it a success.

Strategic Planning- overall approach to the project

Implementation Planning- ways to apply the decisions

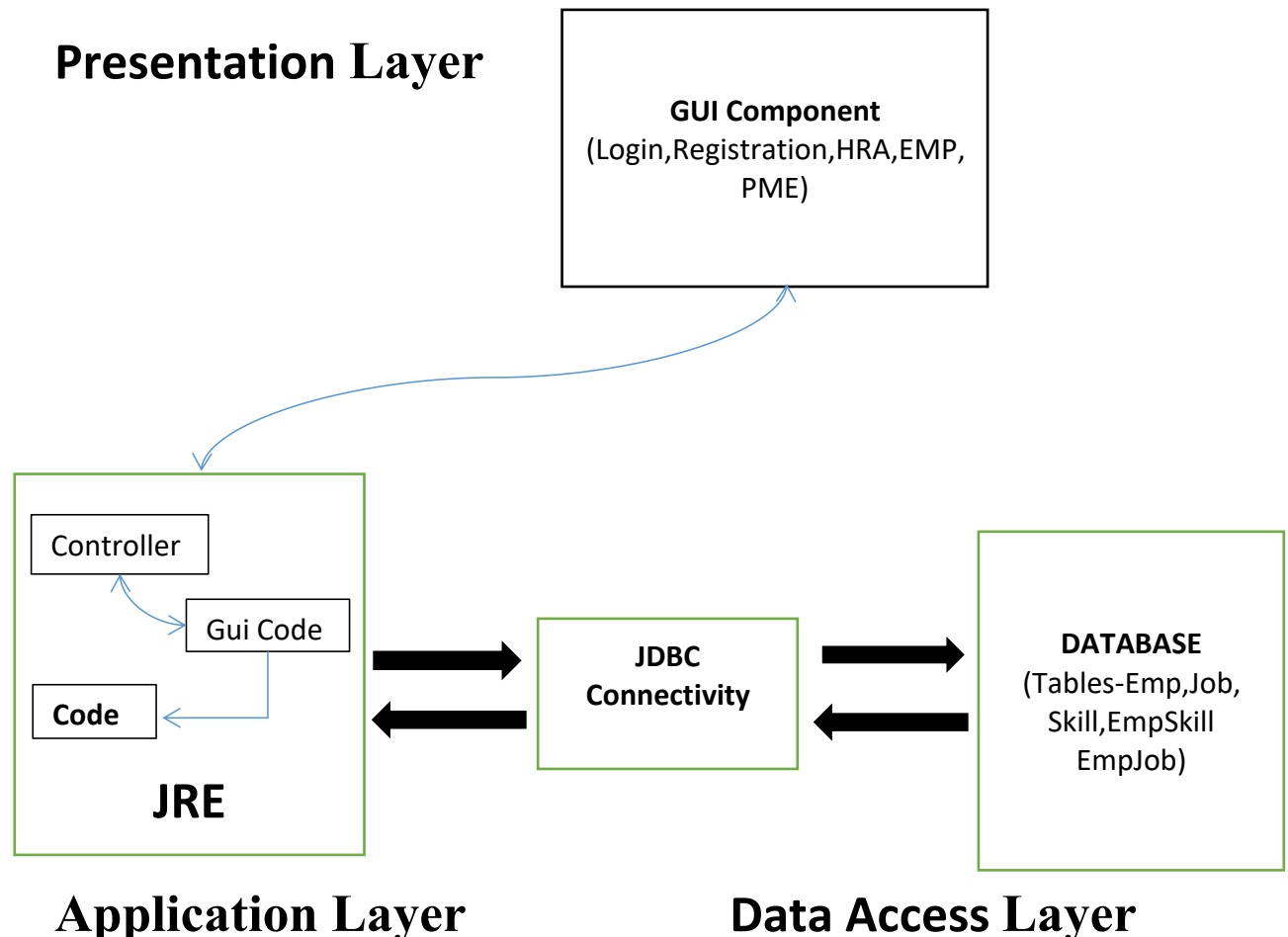
The Execution Phase: In this phase, the decisions and activities defined during the planning phase are implemented.

The Termination Phase: This is the last phase of any project, and it marks the official closure of the project.

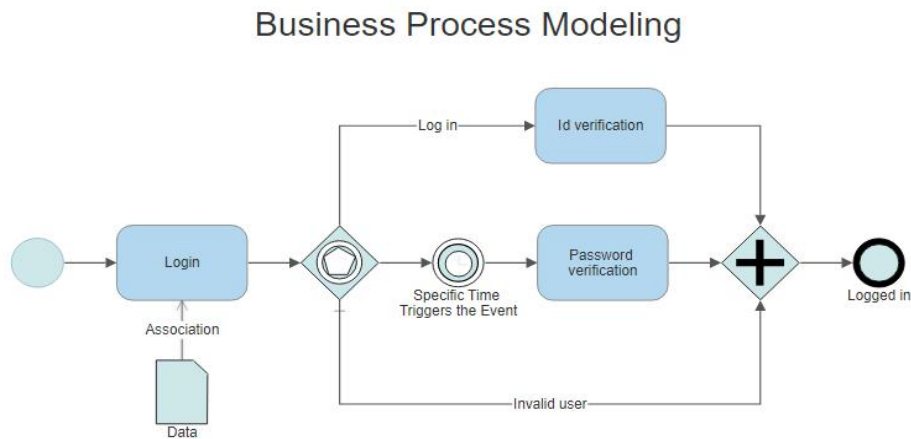
SYSTEM ARCHITECTURE

3-Tier Architecture

1. A **Presentation Layer** that sends content to browsers in the form of HTML/JS/CSS. This might leverage frameworks like React, Angular, Ember, Aurora, etc.
2. An **Application Layer** that uses an application server and processes the business logic for the application. This might be written in C#, Java, C++, Python, Ruby, etc.
3. A **Data Layer** which is a database management system that provides access to application data. This could be MSSQL, MySQL, Oracle, or PostgreSQL, Mongo, etc.



BUSINESS PROCESS MODEL



Business process modeling (or) process modeling, is the analytical representation or put simply an illustration of an organization's business processes. Modeling processes is a critical component for effective business process management.

Benefits of business process modeling:

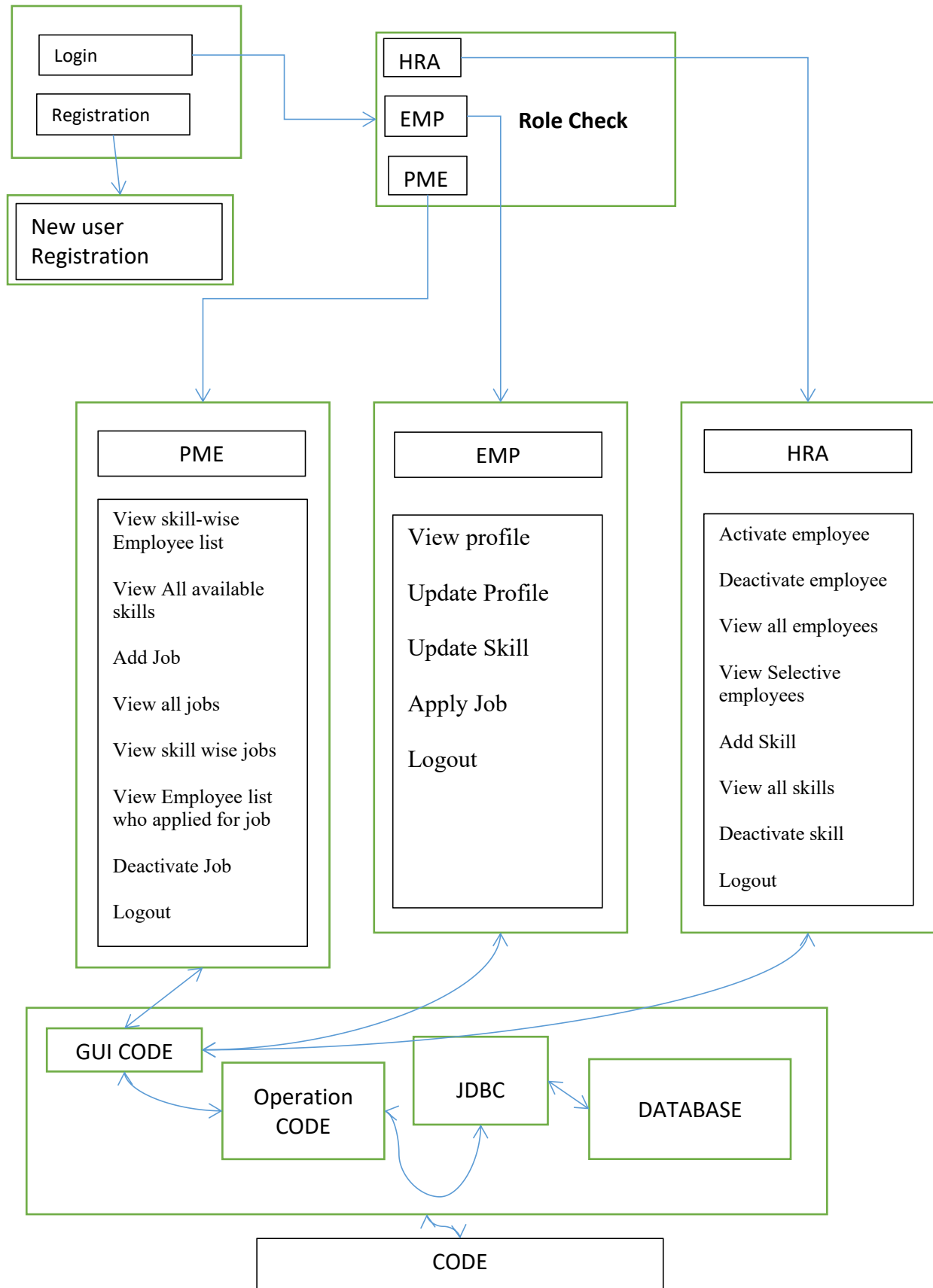
- Gives everyone a clear understanding of how the process works
- Provides consistency and controls the process
- Identifies and eliminates redundancies and inefficiencies
- Sets a clear starting and ending to the process

SOFTWARE REQUIREMENTS SPECIFICATION

Table of Contents

1. Introduction
 - 1.1 Purpose
 - 1.2 Scope
 - 1.3 Definitions, acronyms and abbreviations
 - 1.4 References
 - 1.5 Overview
2. General Description
 - 2.1 Product perspective
 - 2.2 Product functions
 - 2.3 User characteristics
 - 2.4 Constraints
 - 2.5 Assumption and dependencies
3. Specific Requirements
 - 3.1 Functional requirements
 - 3.2 Non-functional requirements
 - 3.3 External interface requirements
 - 3.4 Performance requirements
 - 3.5 Design constraints
 - 3.6 Attributes
 - 3.7 Other requirements
4. Appendices
5. Index

HIGH LEVEL USER CASE DIAGRAMS

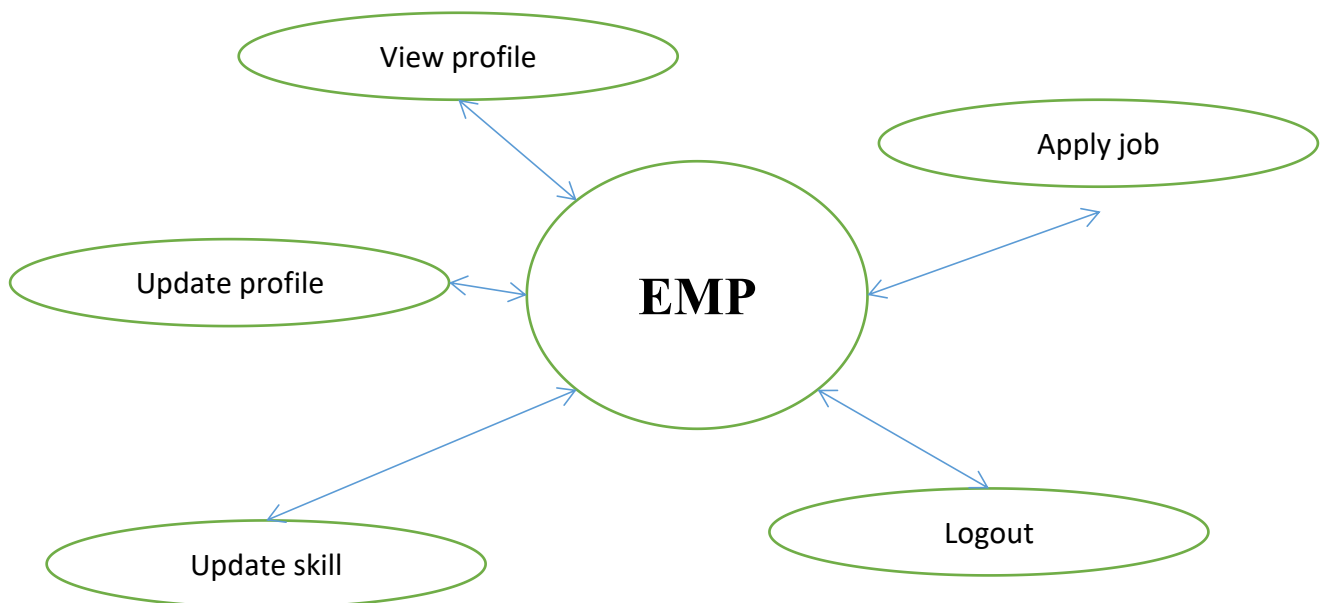
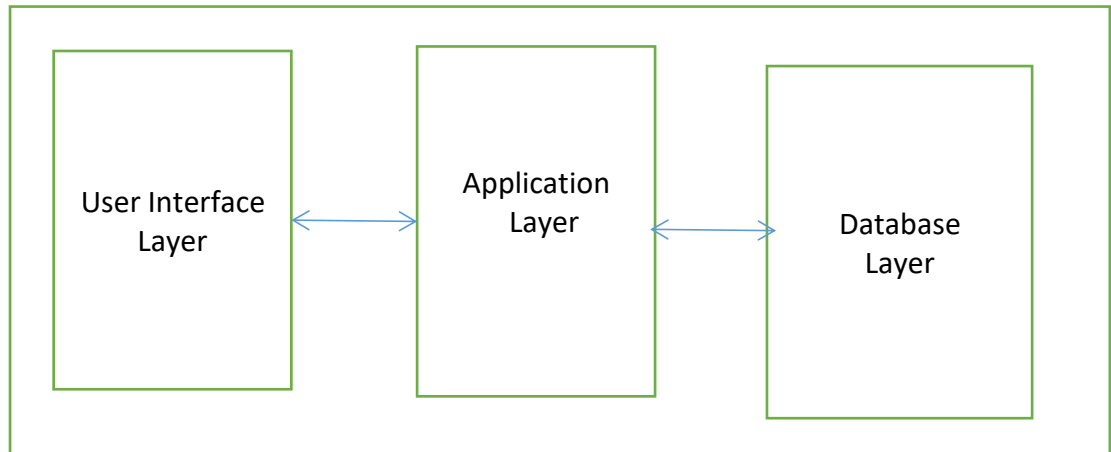


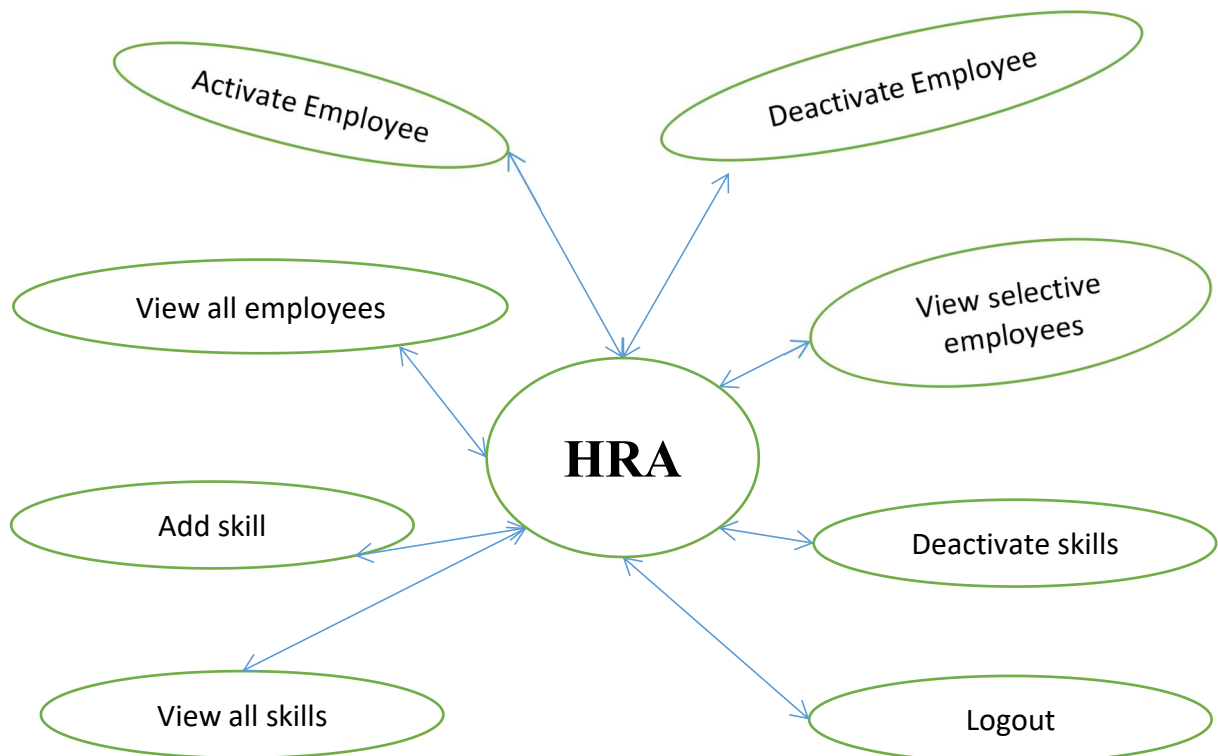
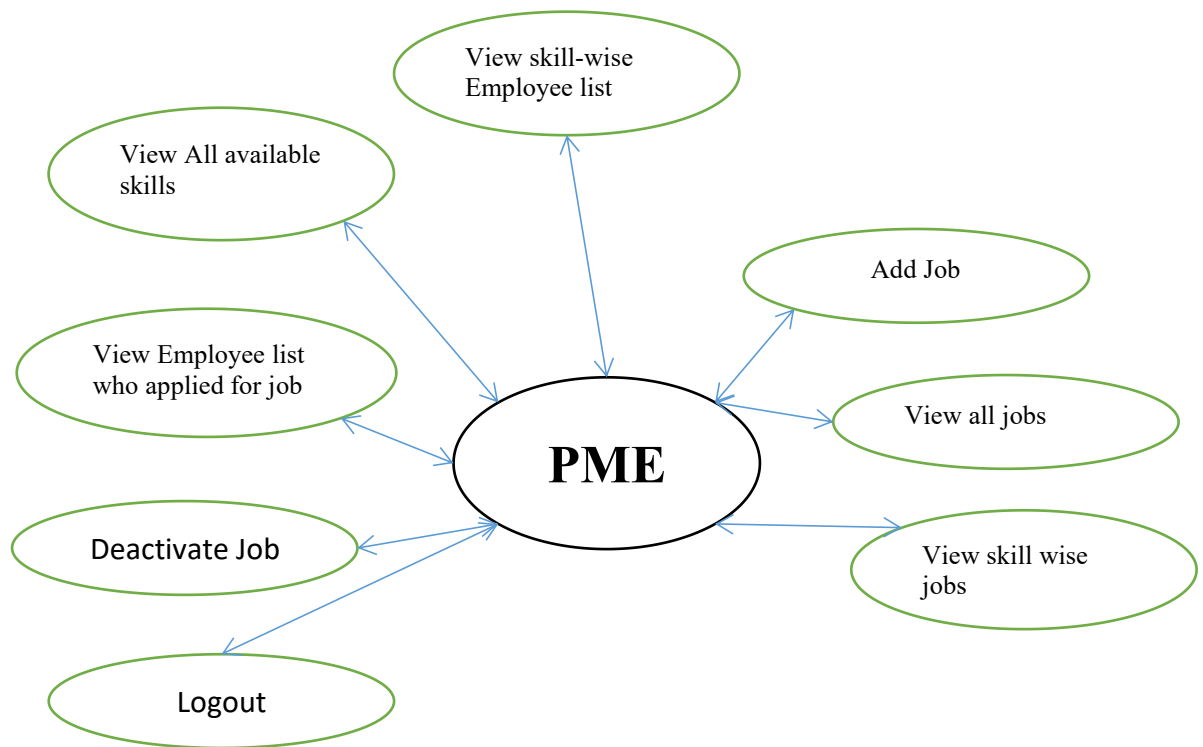
LOW LEVEL USER CASE DIAGRAMS

EMP

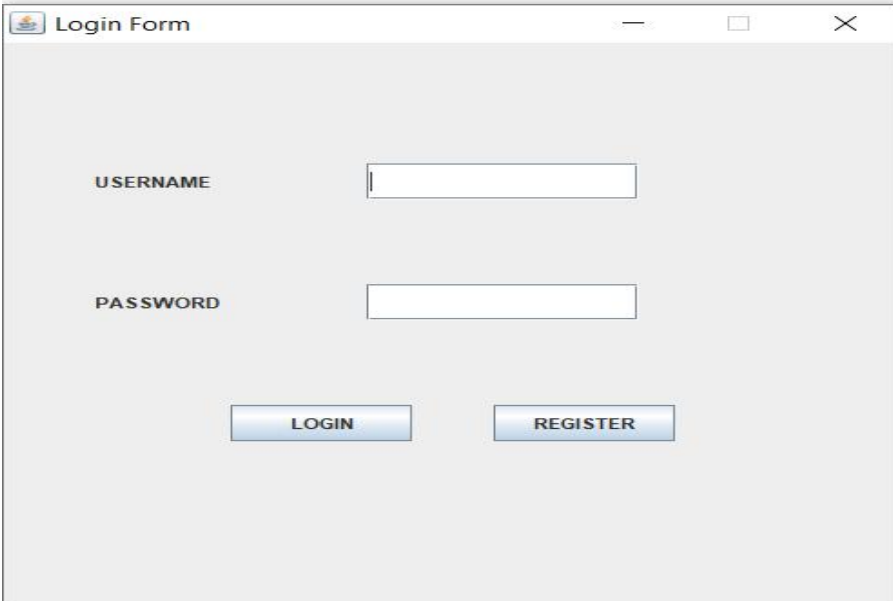
PME

HRA





USER INTERFACE DESIGN

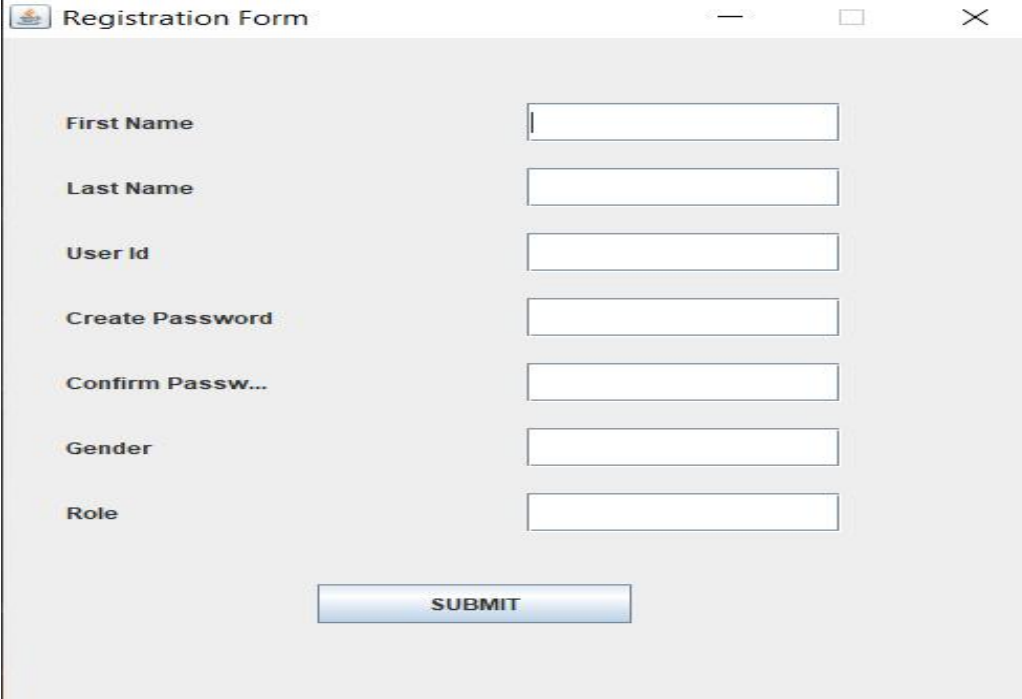


A screenshot of a web browser window titled "Login Form". The window has a light gray background and a standard title bar with a minimize button, a maximize button, and a close button. The form contains two input fields: "USERNAME" and "PASSWORD". Below the input fields are two buttons: "LOGIN" and "REGISTER".

Username:

Password:

LOGIN **REGISTER**



A screenshot of a web browser window titled "Registration Form". The window has a light gray background and a standard title bar with a minimize button, a maximize button, and a close button. The form contains seven input fields: "First Name", "Last Name", "User Id", "Create Password", "Confirm Passw...", "Gender", and "Role". Below the input fields is a "SUBMIT" button.

First Name:

Last Name:

User Id:

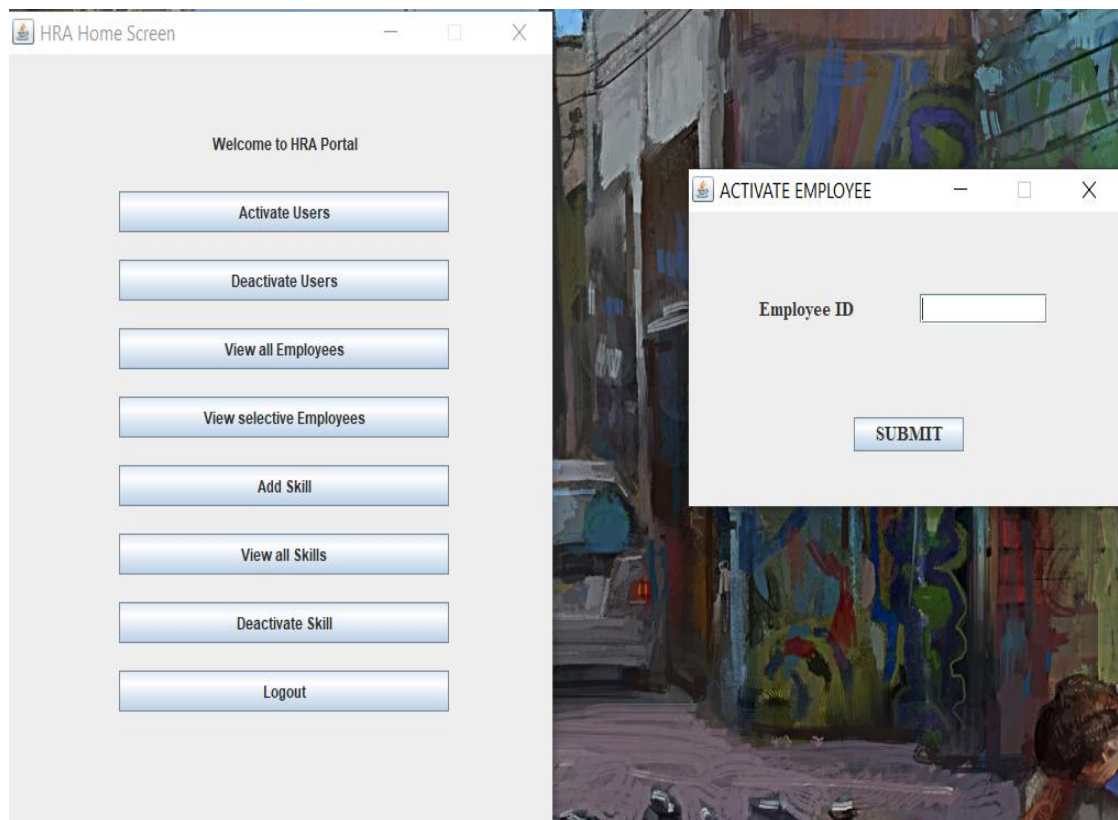
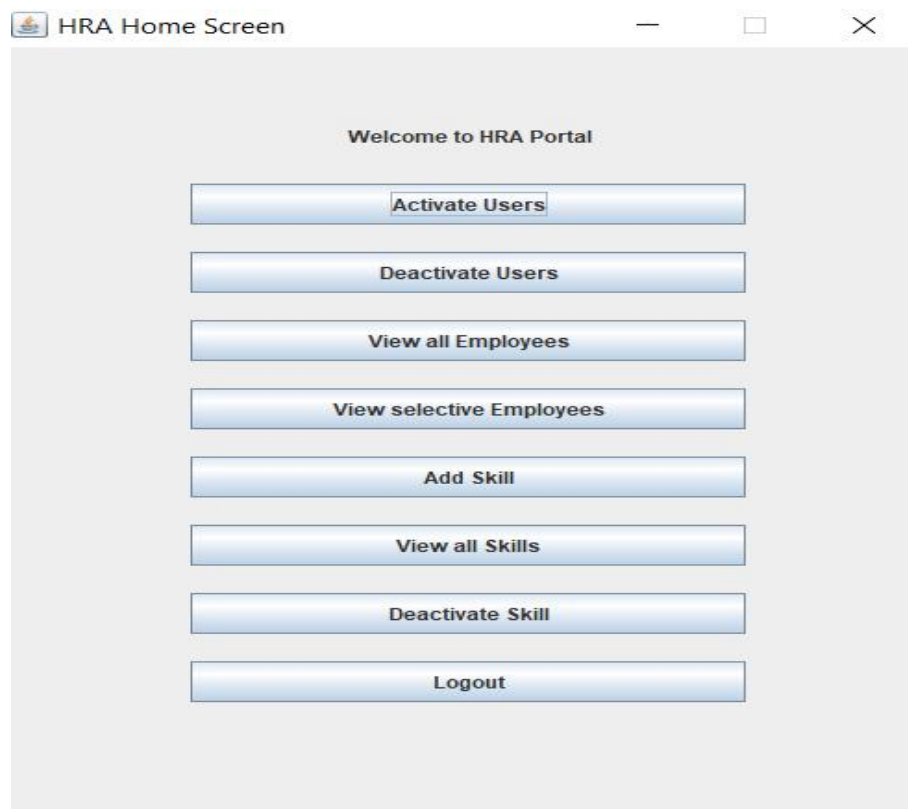
Create Password:

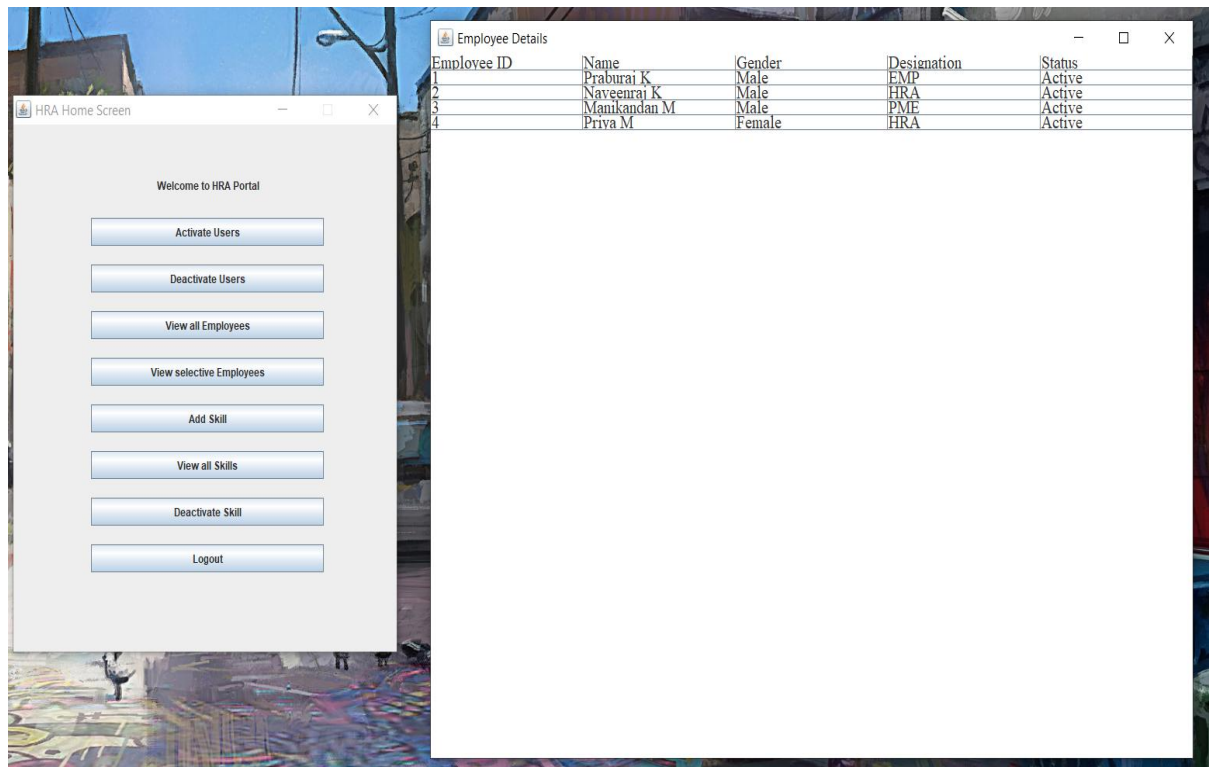
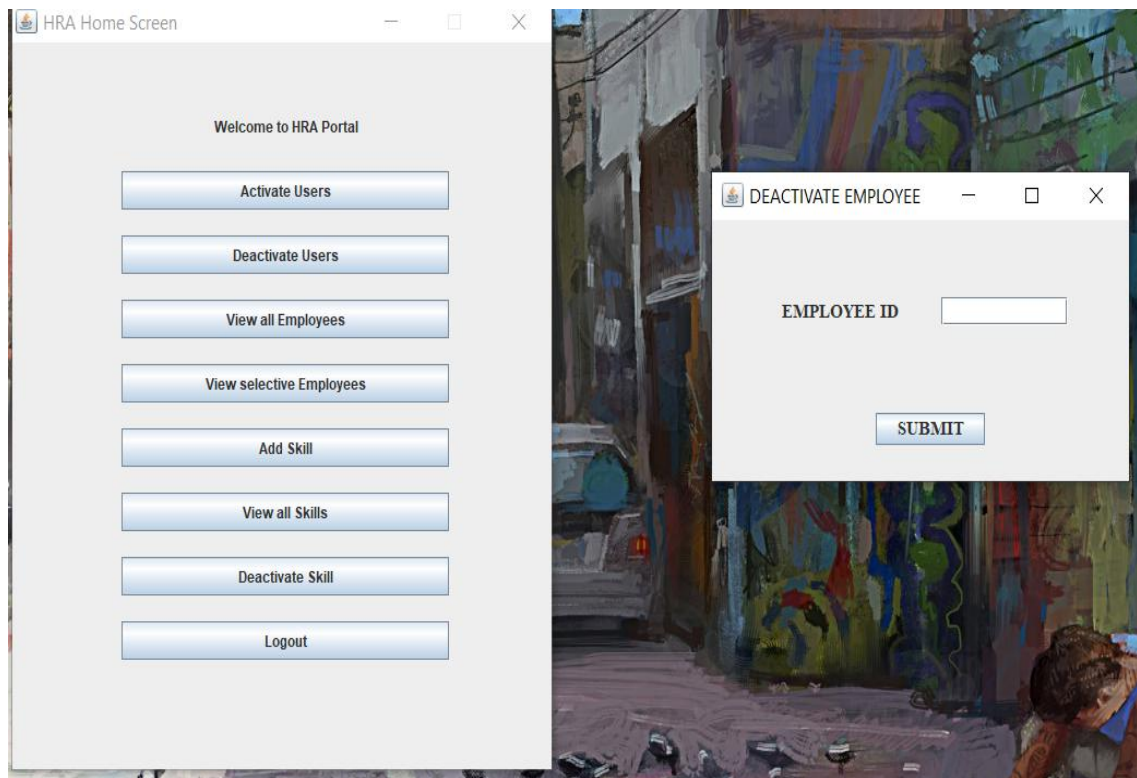
Confirm Passw...:


Gender:

Role:

SUBMIT







—


□

×

Employee Id

Submit

Employee Id	Name	Gender	Designation	Status
1	Praburaj K	Male	EMP	Active



Add Skill

—

□

×

Skill Name

Skill Description

SUBMIT

Employee Details			
SKILL ID	SKILL NAME	SKILL DESCRIPTION	STATUS
1	Management	Planning, organizing, staffin...	Active
2	Presentation	Delivering effective and eng...	Deactive

DEACTIVATE SKILL

SKILL ID

SUBMIT

EMP Home Screen

Welcome to Employee Portal

View Profile

Update profile

Update Skill

Apply Job

Logout

EMP Home Screen

Welcome to Employee Portal

View Profile

Update profile

Update Skill

Apply Job

Logout

PROFILE SELECTION

USER ID prabu14

Submit

PROFILE

EMPLOYEE ID	1
NAME	Praburaj
USER ID	prabu14
GENDER	Male
ROLE	EMP
STATUS	Active



The screenshot shows a window titled "Skill Details" with a standard Windows-style title bar. The main content area displays a table with four columns: "SKILL ID", "SKILL NAME", "SKILL DESCRIPTION", and "STATUS". There are two data rows. The first row has values: 1, Management, Planning, organizing, staffin..., and Deactive. The second row has values: 2, Presentation, Delivering effective and eng..., and Deactive.

SKILL ID	SKILL NAME	SKILL DESCRIPTION	STATUS
1	Management	Planning, organizing, staffin...	Deactive
2	Presentation	Delivering effective and eng...	Deactive

PME Home Screen

Welcome to PME Portal

View Skill wise Employee

View All Skills

Add an Job

View All Jobs

View Skill wise Jobs

View Employee who applied for Job

Deactivate Job

Logout

Add Job

Job Title

Job Description

Company Name

Location

Key Skill

Salary

SUBMIT

Job Details							
JOB ID	JOB TITLE	JOB DESCR...	COMPANY ...	LOCATION	KEY SKILL	SALARY	STATUS
1	Human resou...	Recurting	TCS	Chennai	Management	50000	Deactive
2	Manager	Managing E...	Amazon	Banaglore	Decision-ma...	100000	Deactive

PME Home Screen

Welcome to PME Portal

View Skill wise Employee

View All Skills

Add an Job

View All Jobs

View Skill wise Jobs

View Employee who applied for Job

Deactivate Job

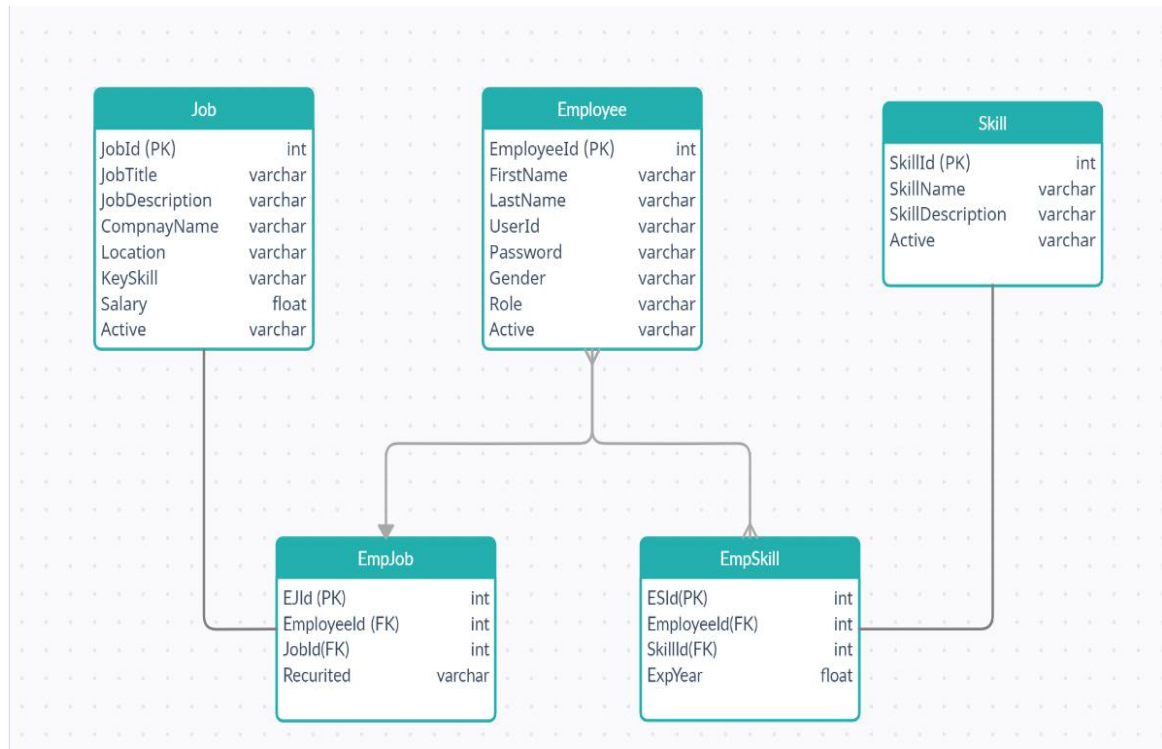
Logout

DEACTIVATE JOB

JOB ID

SUBMIT

DATABASE STRUCTURE



```
create database PCSDB;
```

```
use PCSDB;
```

```
create table Employee(  
  EmployeeId int auto_increment ,  
  FirstName varchar(30) not null,  
  LastName varchar(30)not null,  
  UserId varchar(30)not null,  
  Password varchar(20)not null,  
  Gender varchar(10)not null,  
  Role varchar(3)not null,  
  Active varchar(10)not null,  
  primary key(EmployeeId)  
);
```

```
create table Skill(  
SkillId int auto_increment ,  
SkillName varchar(20)not null,  
SkillDescription varchar(100)not null,  
Active varchar(10)not null,  
primary key(SkillId)  
);
```

```
create table Job(  
JobId int auto_increment ,  
JobTitle varchar(20)not null,  
JobDescription varchar(100)not null,  
CompanyName varchar(50)not null,  
Location varchar(20)not null,  
KeySkill varchar(20)not null,  
Salary float,  
Active varchar(10)not null,  
primary key (JobId)  
);
```

```
create table EmpSkill(  
ESId int auto_increment ,  
EmployeeId int ,  
SkillId int ,  
ExpYear float,  
primary key(ESID),  
foreign key(EmployeeId) references Employee(EmployeeId),  
foreign key(SkillId) references Skill(SkillId)  
);
```

```
create table EmpJob(  
  EJId int auto_increment ,  
  EmployeeId int ,  
  JobId int ,  
  Recruited varchar(10)not null,  
  primary key(EJId),  
  foreign key(EmployeeId) references Employee(EmployeeId),  
  foreign key(JobId) references Job(JobId)  
);
```

```
Select * from Employee;
```

```
Select * from Skill;
```

```
Select * from Job;
```

```
Select * from EmpSkill;
```

```
Select * from EmpJob;
```

SAMPLE CODE

HRA

```
package view;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.SQLException;
import javax.swing.*;
import controller.EmployeeController;
import controller.JobController;
import controller.SkillController;
import model.Employee;
import java.awt.Dialog;
public class HRAHome extends JFrame {

    Container container;
    JLabel lTitle;
    JButton
bSetActive,bSetDeactive,bViewAllEmp,bViewSele
ctiveEmp,bAddSkill,bViewAllSkill,bSetDeactiveS
kill,bLogout;

    JFrame f;
```

```
EmployeeController empController=null;
JobController jobController=null;
SkillController skillController=null;
```

```
public HRAHome() throws
ClassNotFoundException, SQLException {
    Employee emp=new Employee();
    container=getContentPane();
    empController=new EmployeeController();
    jobController=new JobController();
    skillController=new SkillController();
    lTitle=new JLabel("Welcome to HRA
Portal");
```

```
    bSetActive=new JButton("Activate Users");
    bSetActive.addActionListener(new
ActionListener(){
```

```
        @Override
        public void actionPerformed(ActionEvent e)
        {
            try {
                new ActivateEmployee();
            } catch (ClassNotFoundException e1) {

                e1.printStackTrace();
            } catch (SQLException e1) {

                e1.printStackTrace();
            }
        }
    }
}
```

```

        }

    }

});

    bSetDeactive=new JButton("Deactivate
Users");
    bSetDeactive.addActionListener(new
ActionListener(){

        @Override
        public void actionPerformed(ActionEvent e)
        {

            try {
                new DeactivateEmployee();
            } catch (ClassNotFoundException e1) {

                e1.printStackTrace();
            } catch (SQLException e1) {

                e1.printStackTrace();
            }

        }

    });

```



```

        bViewAllEmp=new JButton("View all
Employees");
        bViewAllEmp.addActionListener(new
        ActionListener(){

            private ViewAllEmployees x;

            @Override
            public void actionPerformed(ActionEvent e)
            {

                try {

                    //x = new ViewAllEmployees();
                    //x.showEmployeeData();
                    new EmployeeDetails();
                } catch (ClassNotFoundException e1) {

                    e1.printStackTrace();
                } catch (SQLException e1) {

                    e1.printStackTrace();
                }
            }
        });

```

```

        bViewSelectiveEmp=new JButton("View
selective Employees");
        bViewSelectiveEmp.addActionListener(new
        ActionListener){

```

```

@Override
public void actionPerformed(ActionEvent e)
{
    try {
        new ViewEmployeeById();
    } catch (ClassNotFoundException e1) {
        // TODO Auto-generated catch block
        e1.printStackTrace();
    } catch (SQLException e1) {
        // TODO Auto-generated catch block
        e1.printStackTrace();
    }
}
});

```

```

bAddSkill=new JButton("Add Skill");
bAddSkill.addActionListener(new
ActionListener(){

```

```

@Override
public void actionPerformed(ActionEvent e)
{

    try {
        new AddSkillFrame();
    } catch (ClassNotFoundException e1)
{
        // TODO Auto-generated catch
block

```

```

        e1.printStackTrace();
    } catch (SQLException e1) {
        // TODO Auto-generated catch
block
        e1.printStackTrace();
    }
});

bViewAllSkill=new JButton("View all
Skills");
bViewAllSkill.addActionListener(new
ActionListener(){

    @Override
    public void actionPerformed(ActionEvent e)
    {

        try {
            new ViewAllSkills();
        } catch (ClassNotFoundException e1) {
            // TODO Auto-generated catch block
            e1.printStackTrace();
        } catch (SQLException e1) {
            // TODO Auto-generated catch block
            e1.printStackTrace();
        }
    }
});

```

```

        bSetDeactiveSkill=new JButton("Deactivate
Skill");
        bSetDeactiveSkill.addActionListener(new
ActionListener(){

            @Override
            public void actionPerformed(ActionEvent e)
            {

                try {
                    new DeactivateSkill();
                } catch (ClassNotFoundException e1) {
                    // TODO Auto-generated catch block
                    e1.printStackTrace();
                } catch (SQLException e1) {
                    // TODO Auto-generated catch block
                    e1.printStackTrace();
                }
            }
        });

        bLogout=new JButton("Logout");

        bLogout.addActionListener(new
java.awt.event.ActionListener() {
            public void
actionPerformed(java.awt.event.ActionEvent evt) {
                bLogoutActionPerformed(evt);
            }
        });

```

```

        setLayoutManager();
        setLocationAndSize();
        addComponentsToContainer();
        this.setTitle("HRA Home Screen");
        this.setVisible(true);
        this.setBounds(10,10,500,600);

        //this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setResizable(false);
        this.setLocationRelativeTo(null);
    }

    public void addComponentsToContainer() {
        container.setLayout(null);

    }

    public void setLocationAndSize() {
        lTitle.setBounds(185, 50, 300, 30);
        bSetActive.setBounds(100,100,300,30);
        bSetDeactive.setBounds(100,150,300,30);
        bViewAllEmp.setBounds(100,200,300,30);

        bViewSelectiveEmp.setBounds(100,250,300,30)
;
        bAddSkill.setBounds(100,300,300,30);
        bViewAllSkill.setBounds(100,350,300,30);
    }

```

```
bSetDeactiveSkill.setBounds(100,400,300,30);
    bLogout.setBounds(100,450,300,30);

}

public void setLayoutManager() {
    container.add(lTitle);
    container.add(bSetActive);
    container.add(bSetDeactive);
    container.add(bViewAllEmp);
    container.add(bViewSelectiveEmp);
    container.add(bAddSkill);
    container.add(bViewAllSkill);
    container.add(bSetDeactiveSkill);
    container.add(bLogout);
}

private void
bLogoutActionPerformed(ActionEvent evt) {
    System.exit(0);
}
}
```

EMP

```
package view;
```

```
import java.awt.*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import javax.swing.*;  
import controller.EmployeeController;  
import model.Employee;  
import java.sql.*;
```

```
public class EMPHome extends JFrame {  
    Container container=null;  
    JLabel lTitle;  
    JButton  
bViewprofile,bUpdateProfile,bUpdateSkill,bApply  
Job,bLogout;  
    private EmployeeController empController;
```

```
    public EMPHome(Employee emp) {  
  
        container=getContentPane();  
        try {  
            empController = new  
EmployeeController();  
        } catch (ClassNotFoundException e1) {  
            // TODO Auto-generated catch block  
            e1.printStackTrace();  
        } catch (SQLException e1) {
```

```

        // TODO Auto-generated catch block
        e1.printStackTrace();
    }
    lTitle=new JLabel("Welcome to Employee
Portal");
    bViewprofile=new JButton("View Profile");
    bViewprofile.addActionListener(new
ActionListener(){

        @Override
        public void actionPerformed(ActionEvent e)
        {

            try {
                new ViewProfile();
            } catch (ClassNotFoundException e1) {
                // TODO Auto-generated catch block
                e1.printStackTrace();
            } catch (SQLException e1) {
                // TODO Auto-generated catch block
                e1.printStackTrace();
            }
        }
    });
    bUpdateProfile=new JButton("Update
profile");
    bUpdateProfile.addActionListener(new
ActionListener(){

        @Override

```



```
        public void actionPerformed(ActionEvent e)
    {

        }

    });
    bUpdateSkill=new JButton("Update Skill");
    bUpdateSkill.addActionListener(new
ActionListener(){
```

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

        }

    });
    bApplyJob=new JButton("Apply Job");
    bApplyJob.addActionListener(new
ActionListener(){
```

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

        }

    });
    bLogout=new JButton("Logout");
```

```
        bLogout.addActionListener(new  
java.awt.event.ActionListener() {  
            public void  
actionPerformed(java.awt.event.ActionEvent evt) {  
                bLogoutActionPerformed(evt);  
            }  
        });
```

```
        setLayoutManager();  
        setLocationAndSize();  
        addComponentsToContainer();  
        this.setTitle("EMP Home Screen");  
        this.setVisible(true);  
        this.setBounds(10,10,500,450);
```

```
        //this.setDefaultCloseOperation(JFrame.EXIT_O  
N_CLOSE);  
        this.setResizable(false);  
        this.setLocationRelativeTo(null);  
    }  
    public void setLayoutManager() {  
        container.setLayout(null);  
    }  
    public void setLocationAndSize() {  
        lTitle.setBounds(180,50,300,30);  
        bViewprofile.setBounds(100, 100, 300, 30);  
        bUpdateProfile.setBounds(100, 150, 300, 30);  
        bUpdateSkill.setBounds(100, 200, 300, 30);  
        bApplyJob.setBounds(100, 250, 300, 30);  
        bLogout.setBounds(100, 300, 300, 30);
```

```
}  
public void addComponentsToContainer() {  
    container.add(lTitle);  
    container.add(bViewprofile);  
    container.add(bUpdateProfile);  
    container.add(bUpdateSkill);  
    container.add(bApplyJob);  
    container.add(bLogout);  
  
}  
    private void  
bLogoutActionPerformed(ActionEvent evt) {  
        System.exit(0);  
    }  
}
```

PME

```
package view;
```

```
import java.awt.*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.sql.SQLException;
```

```
import javax.swing.*;
```

```
public class PMEHome extends JFrame{
```

```
    Container container=null;
```

```
    JLabel lTitle;
```

```
    JButton
```

```
bViewSkEmp,bViewAllSkill,bAddJob,bViewAllJo  
bs,bViewSkJobs,bViewEmpApJob,bSetDeactiveJo  
b,bLogout;
```

```
    public PMEHome() {
```

```
        container=getContentPane();
```

```
        lTitle=new JLabel("Welcome to PME  
Portal");
```

```
        bViewSkEmp=new JButton("View Skill wise  
Employee");
```

```
        bViewSkEmp.addActionListener(new  
        ActionListener(){
```

```

        @Override
        public void actionPerformed(ActionEvent e)
        {

            }

        });

```

```

        bViewAllSkill=new JButton("View All
Skills");
        bViewAllSkill.addActionListener(new
        ActionListener(){

```

```

        @Override
        public void actionPerformed(ActionEvent e)
        {

            try {
                new ViewAllSkills();
            } catch (ClassNotFoundException e1) {
                // TODO Auto-generated catch block
                e1.printStackTrace();
            } catch (SQLException e1) {
                // TODO Auto-generated catch block
                e1.printStackTrace();
            }

        }

    });

```

```
        bAddJob=new JButton("Add an Job");
        bAddJob.addActionListener(new
ActionListener(){

            @Override
            public void actionPerformed(ActionEvent e)
            {

                try {
                    new AddJobFrame();
                } catch (ClassNotFoundException e1) {
                    // TODO Auto-generated catch block
                    e1.printStackTrace();
                } catch (SQLException e1) {
                    // TODO Auto-generated catch block
                    e1.printStackTrace();
                }
            }
        });
```

```
        bViewAllJobs=new JButton("View All
Jobs");
        bViewAllJobs.addActionListener(new
ActionListener(){
```

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

                try {
                    new ViewAllJobs();
```

```
        } catch (ClassNotFoundException e1) {  
            // TODO Auto-generated catch block  
            e1.printStackTrace();  
        } catch (SQLException e1) {  
            // TODO Auto-generated catch block  
            e1.printStackTrace();  
        }  
    }  
});
```

```
    bViewSkJobs=new JButton("View Skill wise  
Jobs");
```

```
    bViewSkJobs.addActionListener(new  
ActionListener(){
```

```
        @Override  
        public void actionPerformed(ActionEvent e)  
        {  
  
        }  
    });
```

```
    bViewEmpApJob=new JButton("View  
Employee who applied for Job");
```

```
    bViewEmpApJob.addActionListener(new  
ActionListener(){
```

```
        @Override
```

```
        public void actionPerformed(ActionEvent e)
    {

        }

    });
```

```
        bSetDeactiveJob=new JButton("Deactivate
Job");
        bSetDeactiveJob.addActionListener(new
ActionListener(){
```

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

            try {
                new DeactivateJob();
            } catch (ClassNotFoundException e1) {

                e1.printStackTrace();
            } catch (SQLException e1) {

                e1.printStackTrace();
            }

        }

    });
```



```
        bLogout=new JButton("Logout");
        bLogout.addActionListener(new
java.awt.event.ActionListener() {
            public void
actionPerformed(java.awt.event.ActionEvent evt) {

bLogoutActionPerformed(evt);
            }
        });
```

```
        setLayoutManager();
        setLocationAndSize();
        addComponentsToContainer();
        this.setTitle("PME Home Screen");
        this.setVisible(true);
        this.setBounds(10,10,500,600);
```

```
        //this.setDefaultCloseOperation(JFrame.EXIT_O
N_CLOSE);
        this.setResizable(false);
    }
```

```
    private void addComponentsToContainer() {
        container.setLayout(null);

    }
```

```

private void setLocationAndSize() {
    lTitle.setBounds(185, 50, 300, 30);
    bViewSkEmp.setBounds(100,100,300,30);
    bViewAllSkill.setBounds(100,150,300,30);
    bAddJob.setBounds(100,200,300,30);
    bViewAllJobs.setBounds(100,250,300,30);
    bViewSkJobs.setBounds(100,300,300,30);

    bViewEmpApJob.setBounds(100,350,300,30);
    bSetDeactiveJob.setBounds(100,400,300,30);
    bLogout.setBounds(100,450,300,30);

}

private void setLayoutManager() {
    container.add(lTitle);
    container.add(bViewSkEmp);
    container.add(bViewAllSkill);
    container.add(bAddJob);
    container.add(bViewAllJobs);
    container.add(bViewSkJobs);
    container.add(bViewEmpApJob);
    container.add(bSetDeactiveJob);
    container.add(bLogout);
}

private void
bLogoutActionPerformed(ActionEvent evt) {
    System.exit(0);
}
}

```

