### PCS AUTOMATION

Project By

Name - M. Mukund Murthi

Reg No - S211165200307

# NIIT

PCS Automation

Batch Code - B210153

Start Date - 28-05-2021

End Date - 13-06-2021

Name Of the Coordinator - Dinesh Kumar

Name Of the Developer - M. Mukund Murthi

Date of Submission - 13.06.2021

**NIIT**

###### CERTIFICATE

This is to certify that this report, titled PCS Automation embodies the original work done by M. Mukund Murthi, in partial fulfillment of his course requirement at NIIT.

Coordinator - Dinesh Kumar

**Acknowledgement**

I would like to thank my mentor Dinesh Kumar and also the institute NIIT who helped me achieve this project on Professionet Consultancy Services. It helped me to do a lot on the research on the project and the environment.

I am really thankful for the staffs for helping me around in learning a lot of information during my learning process.

**Abstract**

PCS Automation project helps to manage the employee portal and helps the company to register new employees to their company and would help their Employees to update their professional qualification on the company portal.

This PCS Automation is build with core java concept, here java 8 is used and for client side JFrames are used to display .

Project structure is based on MVC Controller along with Data Access Object to implement CRUD Operation in this application . Here data of employee are stored in H2 Database and to retrive data from database JDBC is used. In

this project skill mapping is achieved as per the requirement of PCS and make their work flow easier way.

**Configuration**

Hardware - Dell (i3 processor) 4Gb RAM,

1TB Hard Disk.

Operating System - Windows 10.

Software - Eclipse,H2.

**Table of Contents**

Chapter 1- Introduction

1. 1.0 Aim
2. 1.1 Objectives
3. 1.2 Case Study

Chapter 2- Project Requirement Specification

1. Literature Research
2. Statement of Requirements

Chapter 3- Project Analysis

1. Project Plan
2. System Architecture
3. Business Process Model
4. Software Requirement Specification
5. High Level Use Case Diagrams

Chapter 4- Project Design

1. Low Level Use Case Diagrams
2. User Interface Design
3. Systems Input and Output Design
4. Database Structure
5. Entity Relationship Diagram
6. Data Model
7. User-centered Interface Designs
8. Personas
9. Paper Prototypes

**Aim**

The Main aim of this Project is to create an application that can map all the employees of a company. Through this application an User can also register their application for a job inside the company.

Objectives:

The Objectives that are to be achieved through Skill Mapping Application are,

1. User Registration
2. Skill Map
3. Job Postings
4. Profile validation
5. Recruitment.

Before this application, these process were done manually.

Literature Research

Introduction:

In order to map the Job seekers who could have be seeking for a job with a lot of talent in them, this application is developed.

With this Application the HR consultants could find lot more talent during the recruitment process. HR consultants were doing this process manually. In that manual process HR would take a lot of time in order to sort the proper talent.

Statement of Requirements

**Title**  - PCS AUTOMATION

**Subtitle** - EM portal

**Author** - M. Mukund Murthi

**Author’s Email** - m.mukundmurthi123@gmail.com

**Description** - Automated Consultancy service

**About Your company**:

Professionet consultancy services is a company which provides a wide range of business services to clients.

**Need of Process Automation**:

1. Saves a lot of time by making your process easy.
2. Improves the quality of your company
3. Helps to find the right talent
4. Adds quality to recruitment.

**Software Requirements**:

* Eclipse IDE
* H2 Database

**Benefits**:

* Pre-screening of candidates.
* Saves a lot of time to find the correct candidates.
* Skill Mapping

**Programming Language used:**

* Java 8.

**Project Life Cycle**

**The Initiation Phase :**

The initiation phase aims to define and authorize the project.

**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.

**DATABASE STRUCTURE:**

create table Employee (

empId int IDENTITY(1,1),

FirstName varchar(30) not null,

LastName varchar(30) not null,

UserId varchar(30) not null,

Passwords varchar(20) not null,

Roles varchar(3) not null,

Gender varchar(10) not null,

Active varchar(10) not null,

primary key(empId)

);

create table Skill(

SkillId int IDENTITY(1,1),

SkillName varchar(30) not null,

SkillDescription varchar(30) not null,

Active varchar(10) not null,

primary key(SkillId)

);

create table Job(

JobId int IDENTITY(1,1) ,

JobTitle varchar(30) not null,

JobDescription varchar(30) not null,

CompanyName varchar(30) not null,

Location varchar(30) not null,

KeySkill varchar(30) not null,

Salary Varchar(7),

Active varchar(10) not null,

primary key(JobId)

);

create table EmpSkill(

EsId int IDENTITY(1,1) ,

EmployeeId int,

SkillId int,

ExpYear int not null,

foreign key (EmployeeId) references Employee (empId),

foreign key (SkillId) references Skill(SkillId),

primary key(EsId)

);

create table EmpJoB(

EJId int IDENTITY(1,1),

EmployeeId int,

JobId int,

Recruited varchar(20) not null,

foreign key (EmployeeId) references Employee (empId),

foreign key (JobId) references Job (JobId),

primary key(EJId)

);

**ENTITY RELATIONSHIP DIAGRAM**:



**PACKAGES USED:**

* Swing components are used ( javax.swing.\*).
  + - * JFrame
      * JPanel,
      * JButton,
      * JTextField,
      * JOptionpane,
      * JLabel are used in project.
* JDBC (Java.sql.\*).
* Java.util.\*(list).

**PATTERN USED IN PROJECT**:

***MVC PATTERN*:**

MVC Pattern stands for Model-View-Controller Pattern. This pattern is used to separate application's concerns.

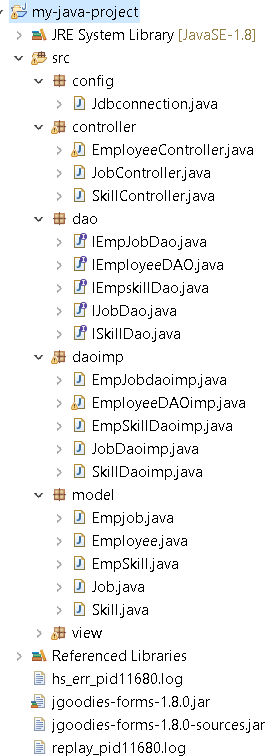
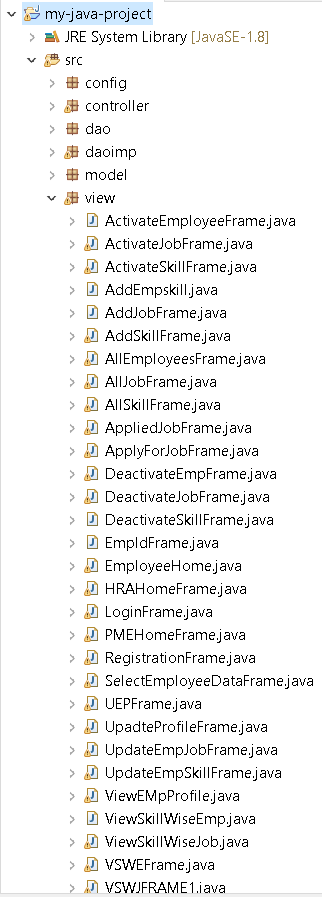
* **Model** - Model represents an object or JAVA POJO carrying data. It can also have logic to update controller if its data changes.
* **View** - View represents the visualization of the data that model contains.
* **Controller** - Controller acts on both model and view. It controls the data flow into model object and updates the view whenever data changes. It keeps view and model separate.

***DAO PATTERN***:

Data Access Object Pattern or DAO pattern is used to separate low level data accessing API or operations from high level business services. Following are the participants in Data Access Object Pattern.

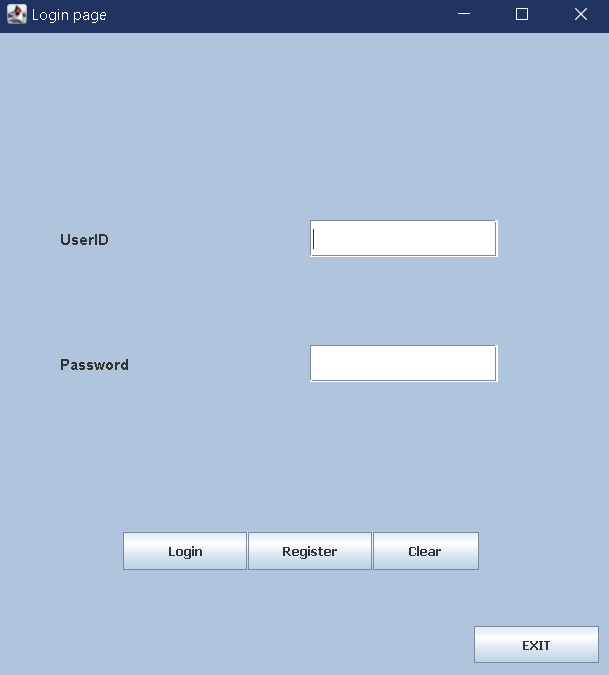
* **Data Access Object Interface** - This interface defines the standard operations to be performed on a model object(s).
* **Data Access Object concrete class** - This class implements above interface. This class is responsible to get data from a data source which can be database / xml or any other storage mechanism.
* **Model Object or Value Object** - This object is simple POJO containing get/set methods to store data retrieved using DAO class.

**PROJECT STRUCTURE**:

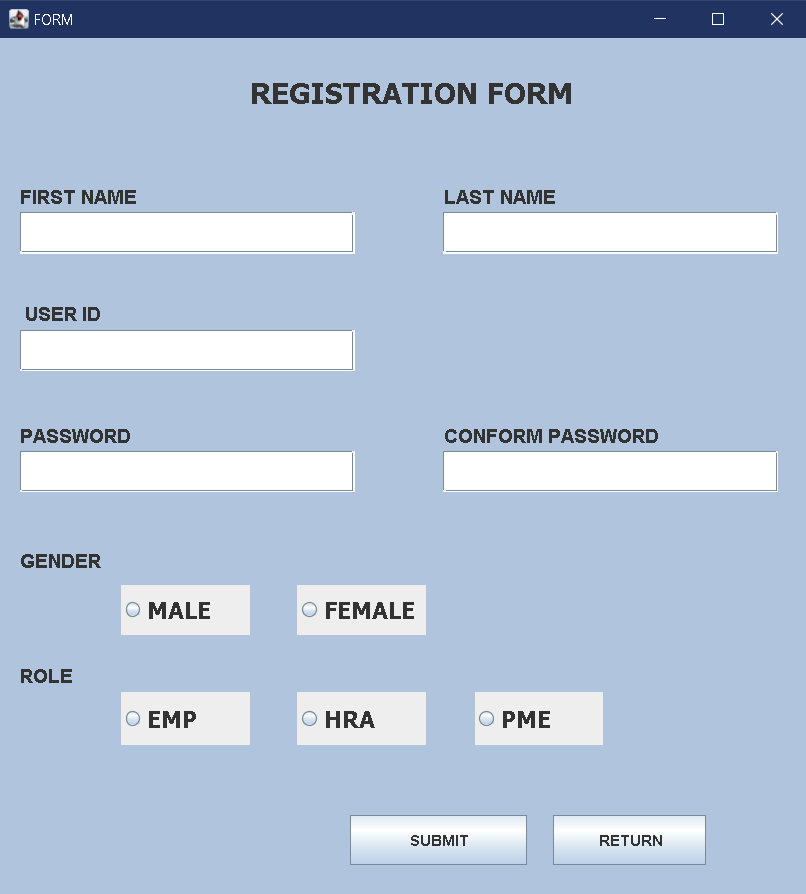
 

**User Interface Design**

Login Form:



Registration Form:



Employee Login:



PME Login:



HRA Home:

