|  |
| --- |
| FPT POLYTECHNIC |
| **DỰ ÁN 1: XÂY DỰNG PHẦN MỀM QUẢN LÍ NHÂN SỰ** |
| NGÀNH CÔNG NGHỆ THÔNG TIN (ỨNG DỤNG PHẦN MỀM) |



|  |
| --- |
| ĐÀ NẴNG 2021 |

|  |
| --- |
| GIẢNG VIÊN HƯỚNG DẪN: NGUYỄN VĂN ĐỊNH |
| SINH VIÊN THỰC HIỆN |
| 1. Đặng Quốc Chấn 1 (PD05208) |

MỤC LỤC

[1.3 Use case 2](#_Toc520132214)

[1 Phân tích 2](#_Toc520132211)

[1.1 Hiện trạng 2](#_Toc520132212)

[1.2 Yêu cầu hệ thống 2](#_Toc520132213)

[2 Thiết kế 2](#_Toc520132215)

[2.1 Mô hình triển khai 2](#_Toc520132216)

[2.2 Thiết kế CSDL 2](#_Toc520132217)

[2.2.1 Sơ đồ quan hệ thực thể 2](#_Toc520132218)

[2.2.2 Thiết kế chi tiết các thực thể 2](#_Toc520132219)

[2.3 Thiết kế giao diện 2](#_Toc520132220)

[2.3.1 Sơ đồ tổ chức giao diện 2](#_Toc520132221)

[2.3.2 Thiết kế giao diện cho các chức năng nghiệp vụ 2](#_Toc520132222)

[3 Thực hiện viết mã 2](#_Toc520132223)

[3.1 Viết mã tạo CSDL 2](#_Toc520132224)

[3.1.1 Tạo CSDL 2](#_Toc520132225)

[3.1.2 SQL truy vấn và thao tác 2](#_Toc520132226)

[3.1.3 Các thủ tục lưu tổng hợp thống kê 2](#_Toc520132227)

[3.2 Lập trình JDBC 3](#_Toc520132228)

[3.2.1 Lớp hỗ trợ 3](#_Toc520132229)

[3.2.2 Model class - Các lớp mô tả dữ liệu 3](#_Toc520132230)

[3.2.3 DAO Class - Các lớp truy xuất dữ liệu 3](#_Toc520132231)

[3.3 Viết mã cho ứng dụng 3](#_Toc520132232)

[3.3.1 Xử lý Form X 3](#_Toc520132233)

[3.3.2 Xử lý Form Y 3](#_Toc520132234)

[4 Kiểm thử 3](#_Toc520132235)

[4.1 Kiểm thử form X 3](#_Toc520132236)

[4.2 Kiểm thử form Y 3](#_Toc520132237)

[5 Đóng gói và triển khai 3](#_Toc520132238)

[5.1 Hướng dẫn chuyển đổi jar thành exe 3](#_Toc520132239)

[5.2 Hướng dẫn cài đặt triển khai 3](#_Toc520132240)

[5.3 Hướng dẫn sử dụng phần mềm 3](#_Toc520132241)

# Phân tích

## Hiện trạng:

1.1.1: Hệ thống:

Việc quản lí nhân sự của hệ thống công ty đang gặp khó khăn do số lượng nhân viên nhiều, dữ liệu công việc lớn khiến cho việc quản lí bằng cách thủ công gặp khó khăn, dễ sai sót. Để phục vụ cho việc lưu trữ số lượng lớn thì xây dựng dự án cung cấp ứng dụng Quản Lý Nhân Sự nhằm giải quyết những khó khăn về lưu trữ, sử lý thông tin và thống kê, báo cáo về tình trạng người dùng, công việc trong hệ thống sẽ đem lại sự tiện lợi cho người dùng.

1.1.1: Ưu điểm của ứng dụng :

**-** Giao diện thân thiện, dễ làm quen và sử dụng.

- Có khả năng lưu trữ cao, lưu trữ khối lượng thông tin lớn.

- Sử dụng ngôn ngữ Java nên có tính bảo mật cao.

- Thống kê, báo cáo chi tiết, rõ dàng.

- Quản lý nhân sự.

## Yêu cầu hệ thống :

### Yêu cầu lưu trữ

**-** Thông tin nhân viên.

- Thông tin trưởng phòng.

- Thông tin phòng ban.

- Thông tin công việc.

- Thông tin chấm công.

- Thông tin lương.

### Yêu cầu quản lý

**-** Quản lý thông tin nhân viên.

- Quản lý thông tin trưởng phòng.

- Quản lý công việc.

- Quản lý phòng ban.

- Quản lý chấm công.

- Quản lý lương.

### Yêu cầu thống kê

**-** Thống kê số lượng nhân viên theo năm.

**-** Thống kê công việc theo năm.

### Yêu cầu về bảo mật

**-**  Tất cả mọi thành viên phải đăng nhập mới sử dụng được phần mềm.

- Trưởng phòng được phép thực hiện tất cả các chức năng.

- Nhân viên không được phép xóa dữ liệu và cũng không được xem thông tin về doanh thu.

### Yêu cầu về môi trường công nghệ để phát triển ứng dụng

Phần mềm:

- Ứng dụng phải được thực với công nghệ Swing và JDBC chạy trên mọi hệ điều hành với môi trường JDK tối thiểu 1.8.

- Windown 10 trở lên

- Lập trình theo ngôn ngữ Java.

- Hệ quản trị CSDL SQL Server 2008 trở lên.

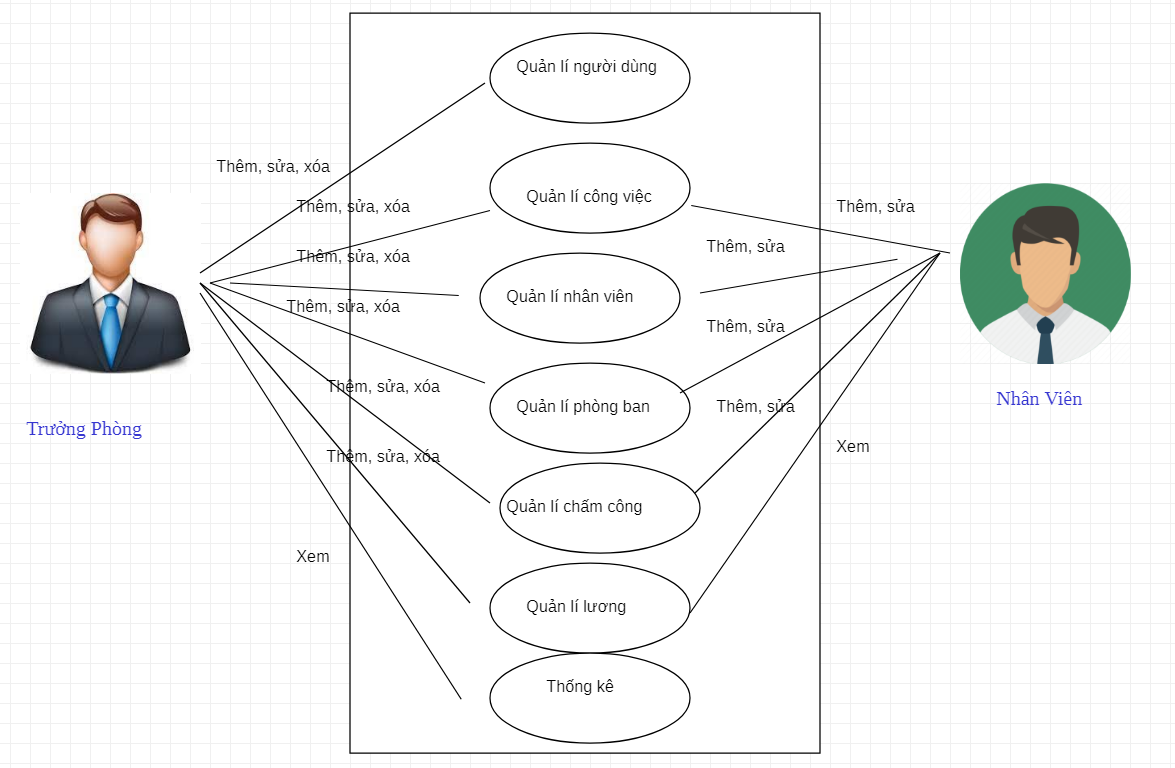
- Hệ điều hành windows 10 trở lên.

- Hệ quản trị CSDL SQL Server 2008 trở lên.

Phần cứng:

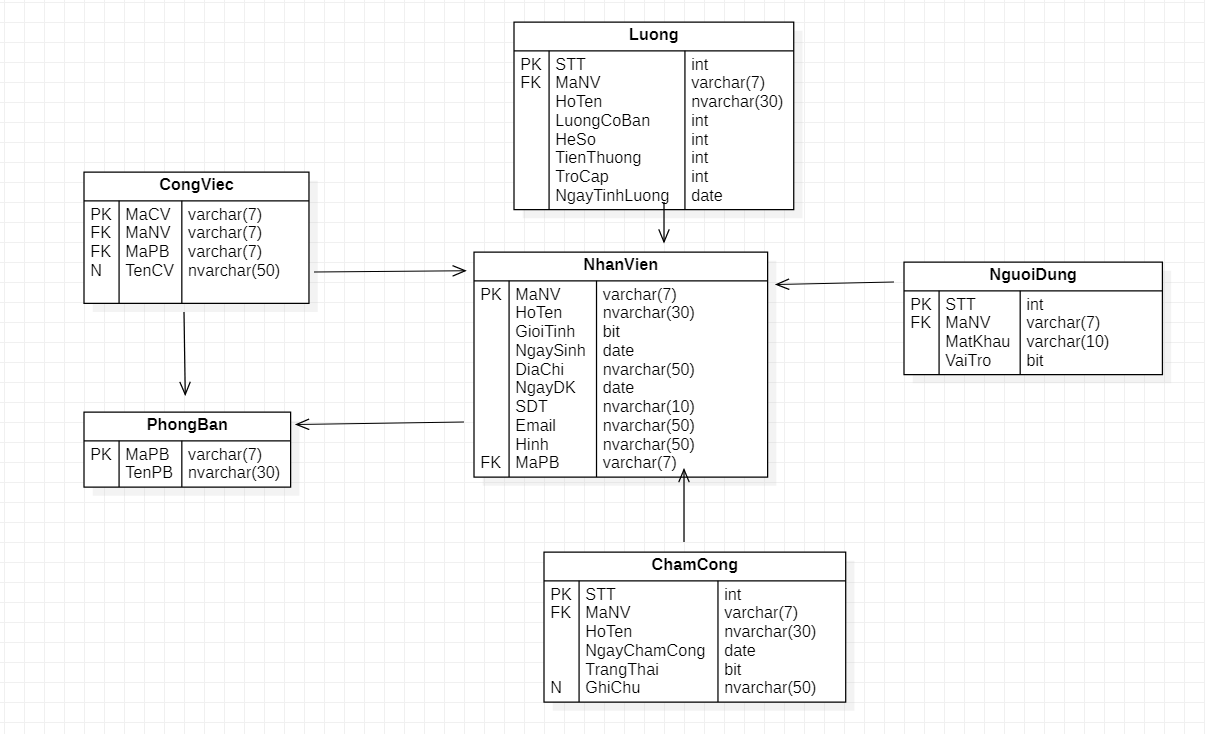
* Bộ vi xử lí I5 trở lên
* RAM 8GB
* Bộ nhớ tối thiểu 256GB

## Use case

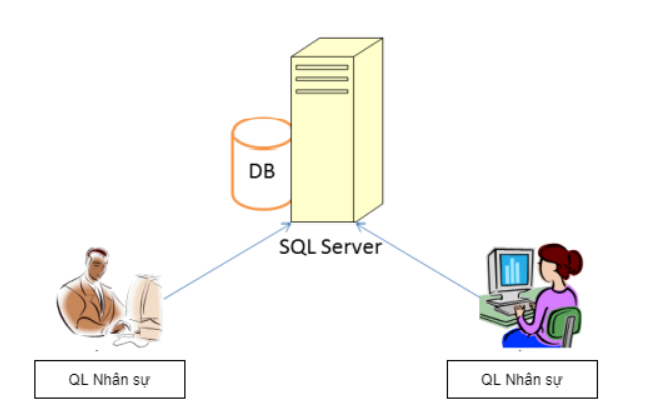


# Thiết kế

## Liệt kê các thực thể quan hệ



## Mô hình triển khai

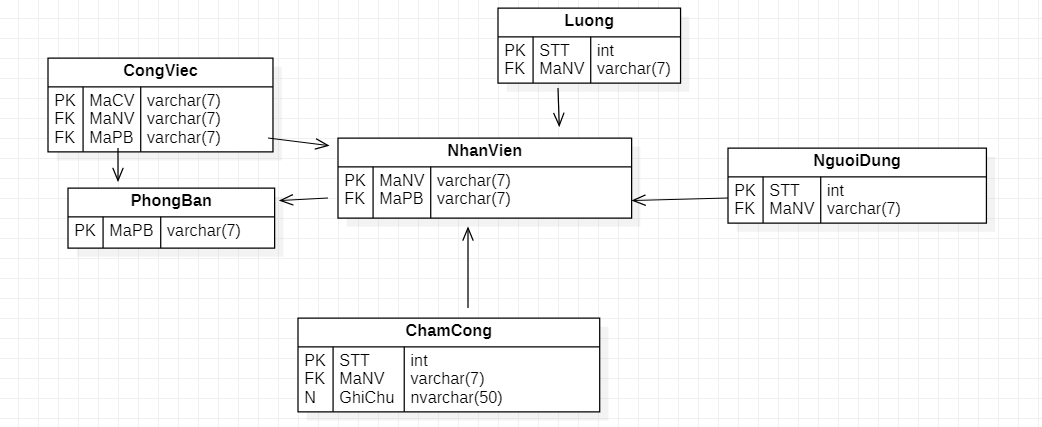


Ứng dụng được xây dựng để phục vụ cho nhiều chi nhánh nhưng cơ sở dữ liệu thì lưu trữ tập trung

* Cần một máy cài SQL Server 2008+. Máy này cần hệ điều hành window 10 trở lên
* Các máy nhân viên cài phần mềm Quản lí nhân sự. Các máy nhân viên cần JDK 1.8+ với hệ điều hành bất kì.

## Thiết kế CSDL

### Sơ đồ quan hệ thực thể



### Thiết kế chi tiết các thực thể

Công việc

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu** | **Ràng buộc** | **Ghi chú** |
| MaCV | Varchar(7) | PK, NOT NULL | Mã công việc |
| MaNV | Varchar(7) | FK, NOT NULL | Mã nhân viên |
| MaPB | Varchar(7) | FK, NOT NULL | Mã phòng ban |
| TenVC | Nvarchar(50) | NULL | Tên công việc |

Nhân viên

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu** | **Ràng buộc** | **Ghi chú** |
| MaNV | Varchar(7) | PK, NOT NULL | Mã nhân viên |
| HoTen | Nvarchar(30) | NOT NULL | Họ tên |
| GioiTinh | Bit | Default 0 | Giới tính |
| NgaySinh | Date | NOT NULL | Ngày sinh |
| DiaChi | Nvarchar(50) | NOT NULL | Địa chỉ |
| NgayDK | Date | NOT NULL | Ngày đăng kí |
| SDT | Nvarchar(10) | NOT NULL | Số điện thoại |
| Email | Nvarchar(50) | NOT NULL | Email |
| Hinh | Nvarchar(50) | NOT NULL | Hình |
| MaPB | Varchar(7) | FK, NOT NULL | Mã phòng ban |

Người dùng

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu** | **Ràng buộc** | **Ghi chú** |
| STT | int | PK, NOT NULL | Số thứ tự |
| MaNV | Varchar(7) | FK, NOT NULL | Mã nhân viên |
| MatKhau | Varchar(10) | NOT NULL | Mật khẩu |
| VaiTro | Bit | Default 0 | Vai trò |

Phòng ban

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu** | **Ràng buộc** | **Ghi chú** |
| MaPB | Varchar(7) | PK, NOT NULL | Mã phòng ban |
| TenPB | Nvarchar(30) | NOT NULL | Tên phòng ban |

Chấm công

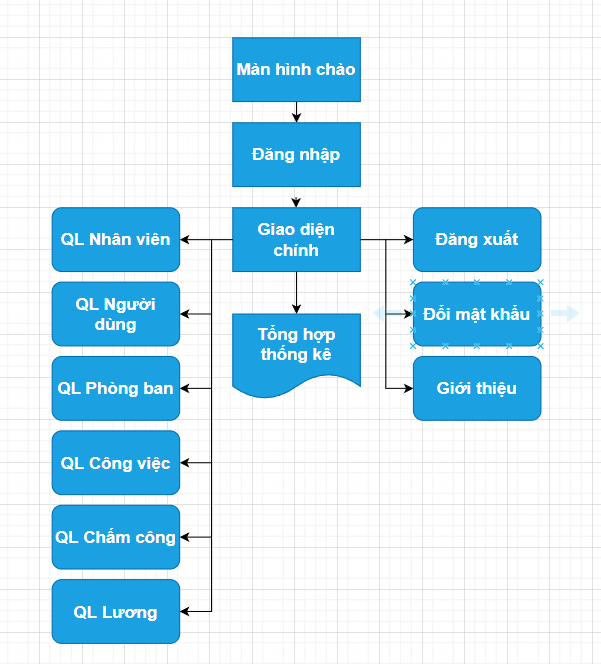
|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu** | **Ràng buộc** | **Ghi chú** |
| STT | int | PK, NOT NULL | Số thứ tự |
| MaNV | Varchar(7) | FK, NOT NULL | Mã nhân viên |
| HoTen | Nvarchar(30) | NOT NULL | Họ tên |
| NgayChamCong | date | NOT NULL | Ngày chấm công |
| TrangThai | bit | Default 0 | Trạng thái |
| GhiChu | Nvarchar(50) | NULL | Ghi chú |

Lương

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu** | **Ràng buộc** | **Ghi chú** |
| STT | int | PK, NOT NULL | Số thứ tự |
| MaNV | varchar(7) | FK, NOT NULL | Mã nhân viên |
| HoTen | Nvarchar(30) | NOT NULL | Họ tên |
| LuongCoBan | int | NOT NULL | Lương cơ bản |
| HeSo | int | NOT NULL | Hệ số |
| TienThuong | int | NULL | Tiền thưởng |
| TroCap | int | NULL | Trợ cấp |
| NgayTinhLuong | date | NOT NULL | Ngày tính lương |

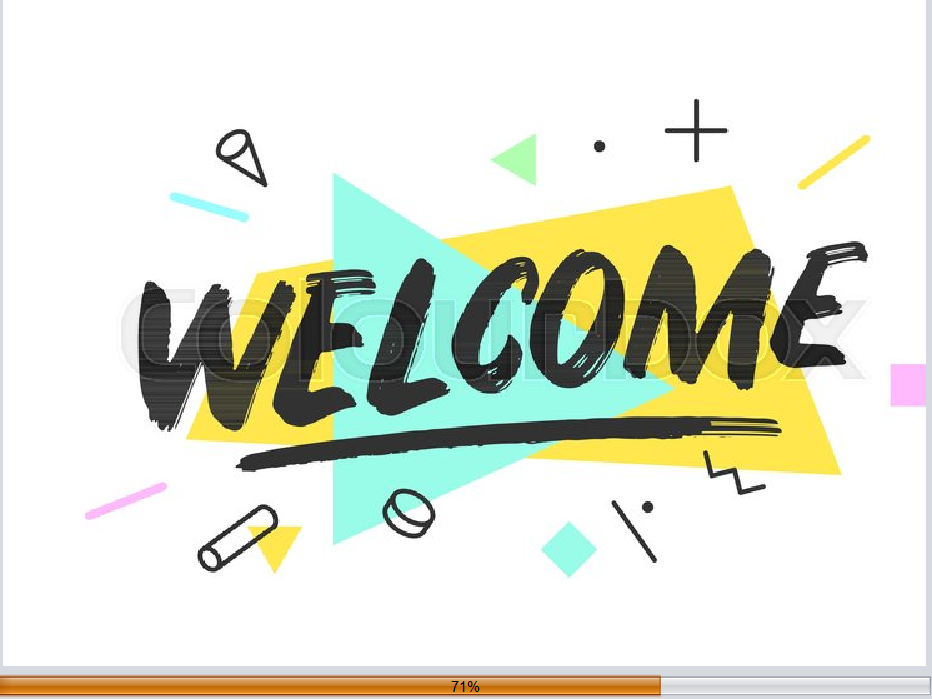
## Thiết kế giao diện

### Sơ đồ tổ chức giao diện



### Thiết kế giao diện cho các chức năng nghiệp vụ

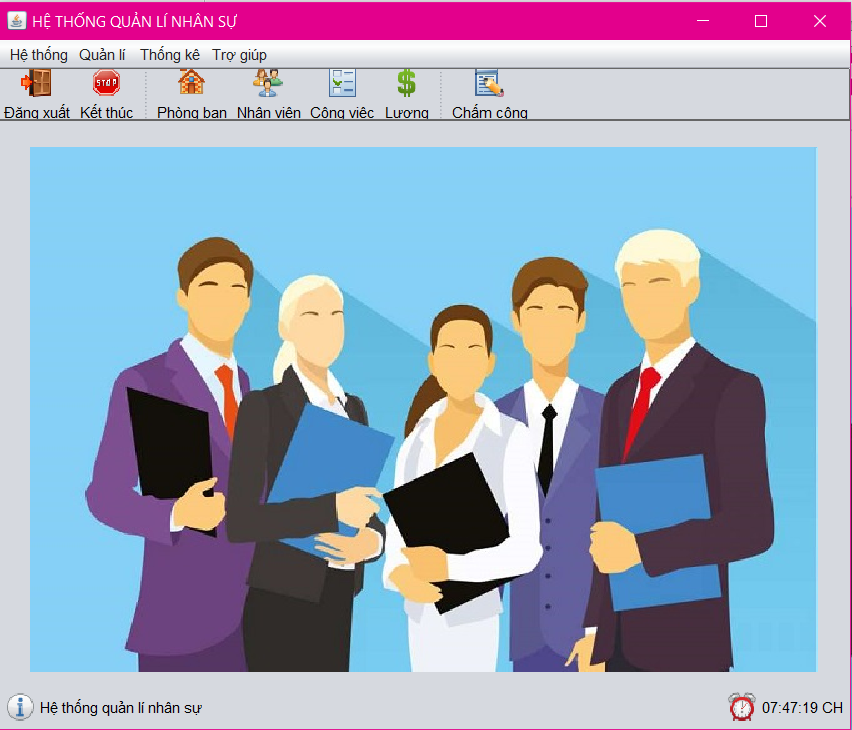
#### Thiết kế Form màn hình chào



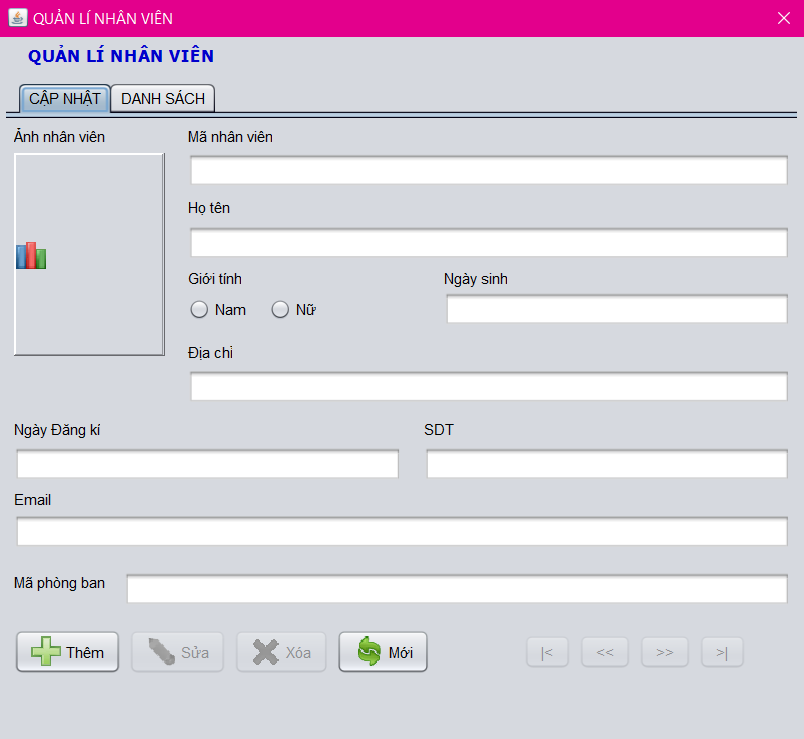
#### Thiết kế Form Đăng nhập

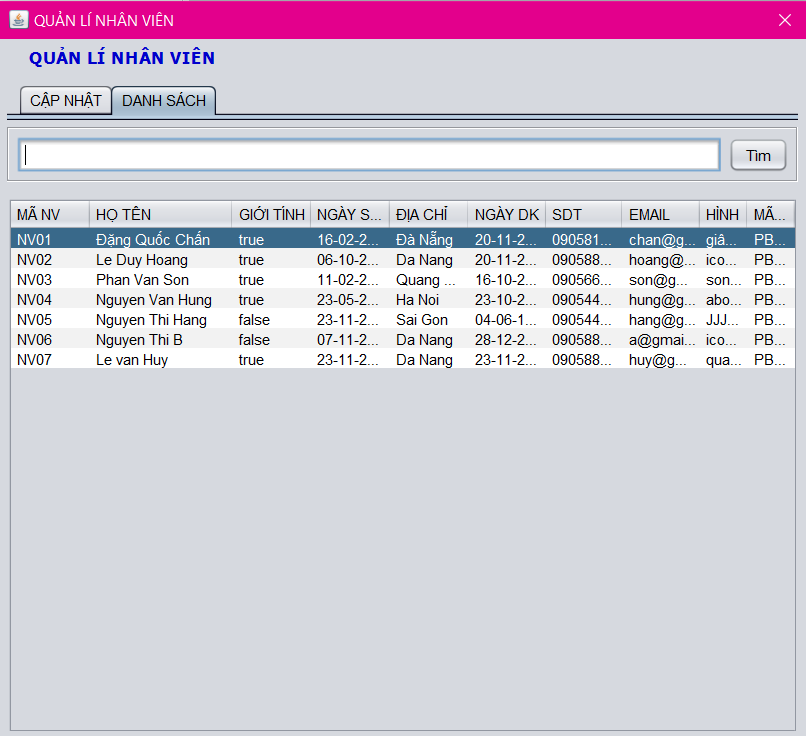


#### Thiết kế Form Giao diện chính

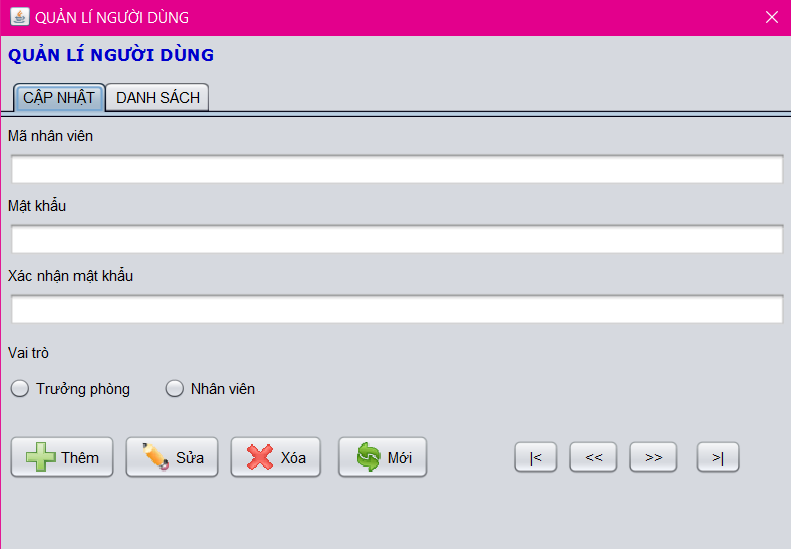


#### Thiết kế Form QL nhân viên

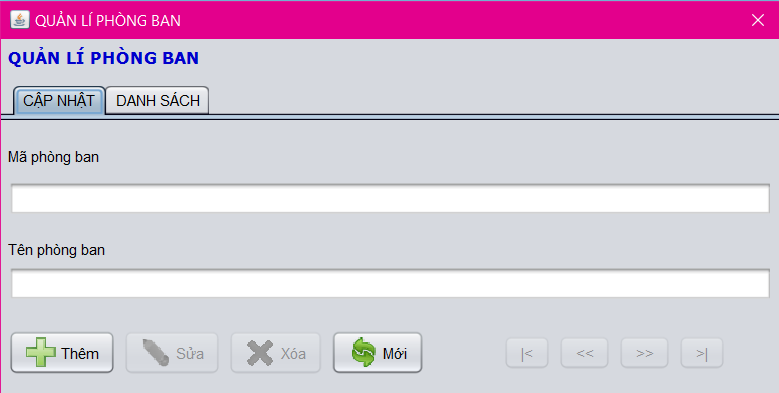




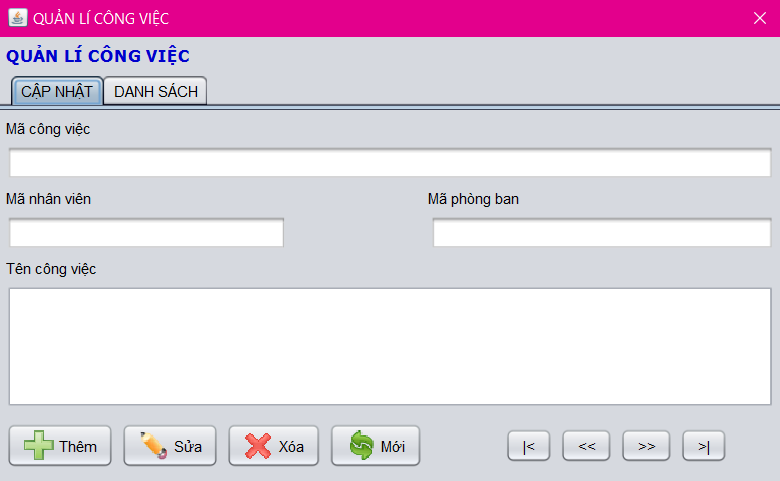
#### Thiết kế Form QL Người dùng



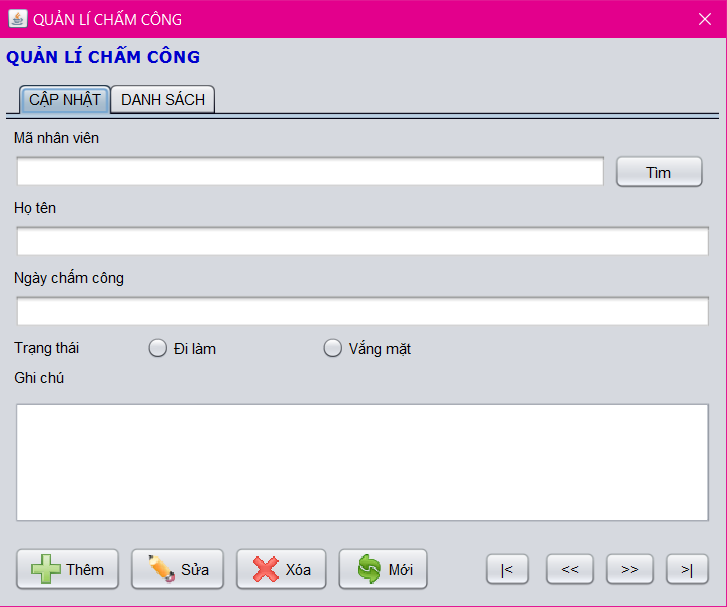
#### Thiết kế Form QL Phòng ban

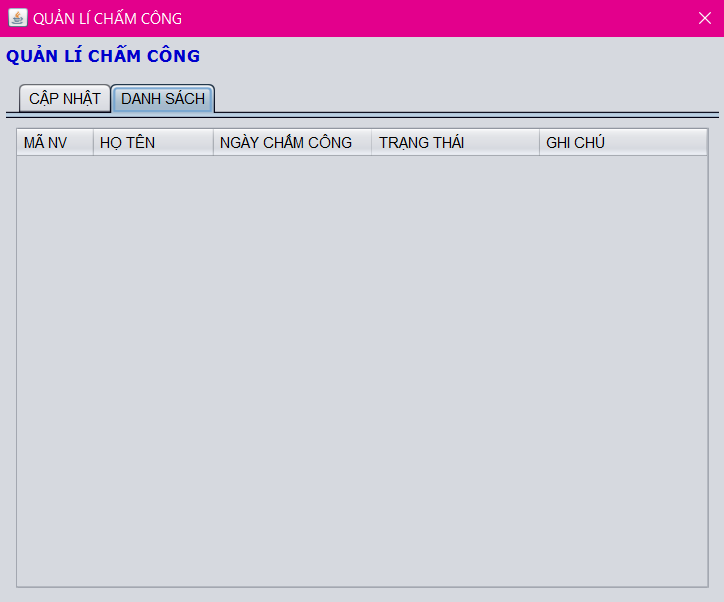


#### Thiết kế Form QL Công việc



#### Thiết kế Form QL Chấm công

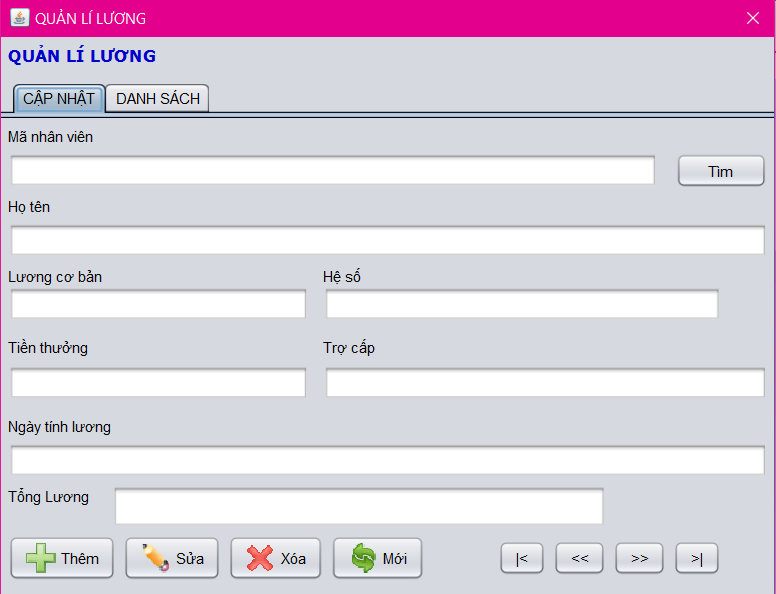


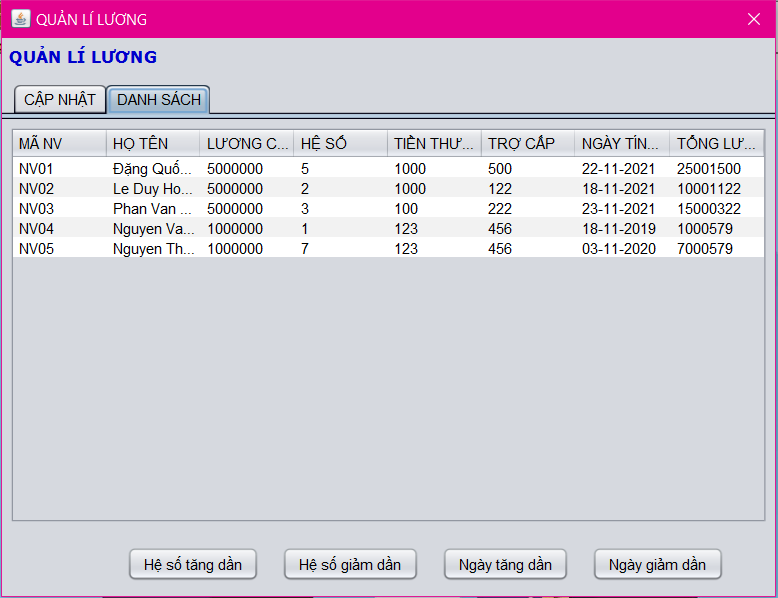


#### Thiết kế Form Đổi mật khẩu

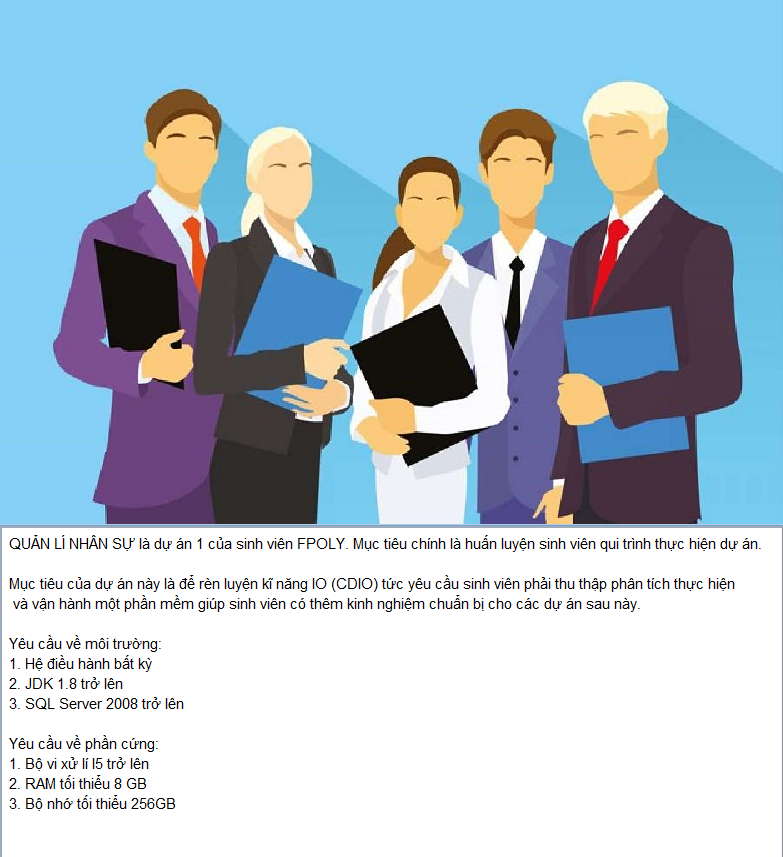


#### Thiết kế Form Quản Lí Lương

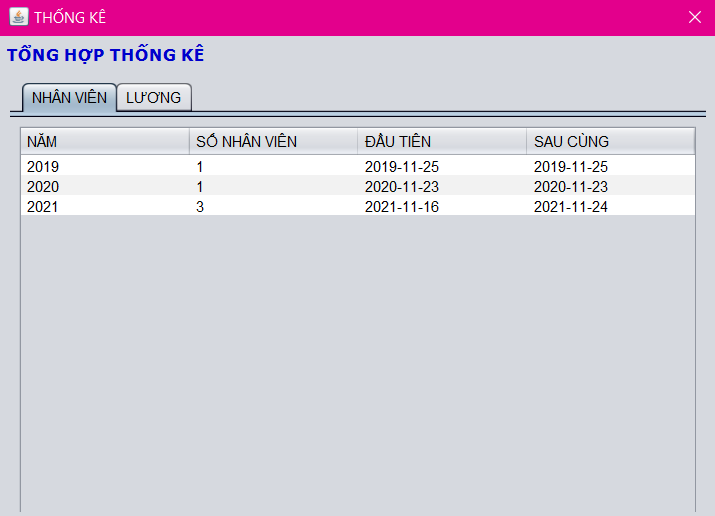


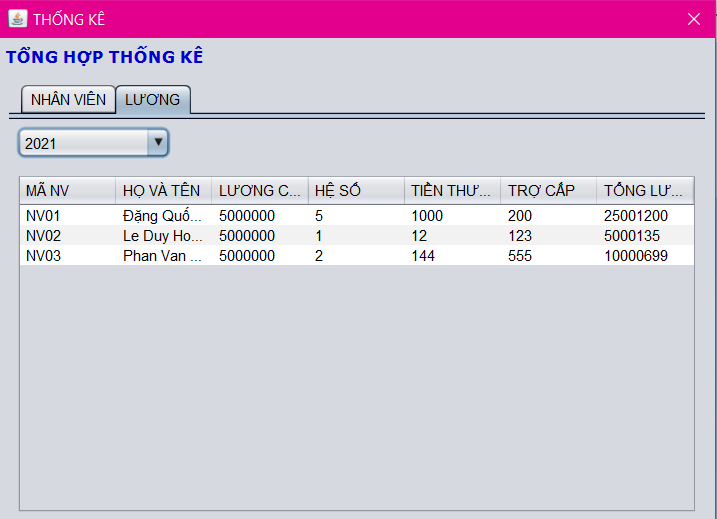


#### Thiết kế Form Giới thiệu



#### Thiết kế Form Chấm công





# Thực hiện viết mã

## Viết mã tạo CSDL

### Tạo CSDL

CREATE DATABASE QLNhanSu

GO

USE QLNhanSu

GO

CREATE TABLE PhongBan(

MaPB varchar(7) NOT NULL,

TenPB nvarchar(30) NOT NULL

PRIMARY KEY(MaPB)

)

GO

CREATE TABLE NhanVien(

MaNV varchar(7) NOT NULL,

HoTen nvarchar(30) NOT NULL,

GioiTinh bit NOT NULL Default 0,

NgaySinh date NOT NULL ,

DiaChi nvarchar(50) NOT NULL,

NgayDangKi date NOT NULL,

SDT nvarchar(10) NOT NULL,

Email nvarchar(50) NOT NULL,

Hinh nvarchar(50) NOT NULL,

MaPB varchar(7) NOT NULL

PRIMARY KEY(MaNV),

FOREIGN KEY (MaPB) REFERENCES PhongBan(MaPB)

)

GO

CREATE TABLE CongViec(

MaCV varchar(7) NOT NULL,

MaNV varchar(7) NOT NULL,

MaPB varchar(7) NOT NULL,

TenCV nvarchar(50) NULL

PRIMARY KEY(MaCV),

FOREIGN KEY (MaNV) REFERENCES NhanVien(MaNV) ON UPDATE CASCADE ,

FOREIGN KEY (MaPB) REFERENCES PhongBan(MaPB) ON UPDATE CASCADE

)

GO

CREATE TABLE ChamCong(

STT int identity(1,1) not null,

MaNV varchar(7) not null,

HoTen nvarchar(30) not null,

NgayChamCong date not null,

TrangThai bit default(0) not null,

GhiChu nvarchar(50) NULL

PRIMARY KEY (STT),

FOREIGN KEY (MaNV) REFERENCES NhanVien(MaNV) ON UPDATE CASCADE

);

GO

CREATE TABLE NguoiDung(

STT int identity (1,1),

MaNV varchar(7) NOT NULL,

MatKhau varchar(10) NOT NULL,

VaiTro bit NOT NULL DEFAULT 0,

PRIMARY KEY(STT),

FOREIGN KEY (MaNV) REFERENCES NhanVien(MaNV) ON UPDATE CASCADE,

)

GO

CREATE TABLE Luong(

STT int identity(1,1) not null,

MaNV varchar(7) not null,

HoTen nvarchar(30) NOT NULL,

LuongCoBan int not null,

HeSo int not null,

TienThuong int null,

TroCap int null,

NgayTinhLuong date not null,

PRIMARY KEY (STT),

FOREIGN KEY (MaNV) REFERENCES NhanVien(MaNV) ON UPDATE CASCADE

);

GO

### SQL truy vấn và thao tác

#### SQL đối với Bảng PhongBan

|  |  |
| --- | --- |
| Hành động | Câu lệnh SQL |
| Thêm mới | INSERT INTO PhongBan(MaPB,TenPB) VALUES (?,?) |
| Cập nhật theo mã | UPDATE PhongBan SET TenPB=? WHERE MaPB=? |
| Xóa theo mã | DELETE FROM PhongBan WHERE MaPB=? |
| Hiển thị tất cả | SELECT \* FROM PhongBan |
| Hiển thị theo mã | SELECT \* FROM PhongBan WHERE MaPB = ? |

#### SQL đối với Bảng NhanVien

|  |  |
| --- | --- |
| Hành động | Câu lệnh SQL |
| Thêm mới | INSERT INTO NhanVien(MaNV, HoTen, GioiTinh, NgaySinh, DiaChi,NgayDangKi, SDT, Email, Hinh, MaPB) VALUES (?,?,?,?,?,?,?,?,?,?) |
| Cập nhật theo mã | UPDATE NhanVien SET HoTen=?, GioiTinh=?, NgaySinh=?, DiaChi=?,NgayDangKi=?, SDT=?, Email=?, Hinh=?, MaPB=? WHERE MaNV=? |
| Xóa theo mã | DELETE FROM NhanVien WHERE MaNV=? |
| Hiển thị tất cả | SELECT \* FROM NhanVien |
| Hiển thị theo mã | SELECT \* FROM NhanVien WHERE MaNV = ? |

#### SQL đối với Bảng Luong

|  |  |
| --- | --- |
| Hành động | Câu lệnh SQL |
| Thêm mới | INSERT INTO Luong(MaNV,HoTen, LuongCoBan, HeSo, TienThuong, TroCap, NgayTinhLuong) VALUES (?,?,?,?,?,?,?) |
| Cập nhật theo mã | UPDATE Luong SET HoTen=?, LuongCoBan=?, HeSo=?, TienThuong=?, TroCap=?, NgayTinhLuong=? WHERE MaNV=? |
| Xóa theo mã | DELETE FROM Luong WHERE MaNV=? |
| Hiển thị tất cả | SELECT \* FROM Luong |
| Sắp xếp theo hệ số tăng dần | select \* from Luong order by HeSo asc |
| Sắp xếp theo hệ số giảm dần | select \* from Luong order by HeSo desc |
| Sắp xếp theo ngày giảm dần | select \* from Luong order by NgayTinhLuong desc |
| Sắp xếp theo ngày tăng dần | select \* from Luong order by NgayTinhLuong asc |
| Hiển thị theo mã | SELECT \* FROM Luong WHERE MaNV = ? |

#### SQL đối với Bảng ChamCong

|  |  |
| --- | --- |
| Hành động | Câu lệnh SQL |
| Thêm mới | INSERT INTO ChamCong(MaNV, HoTen, NgayChamCong, TrangThai, GhiChu) VALUES (?,?,?,?,?) |
| Cập nhật theo mã | UPDATE ChamCong SET HoTen=?, NgayChamCong=?, TrangThai=?, GhiChu=? WHERE MaNV=? |
| Xóa theo mã | DELETE FROM ChamCong WHERE MaNV=? |
| Hiển thị tất cả | SELECT \* FROM ChamCong |
| Sắp xếp theo ngày | select \* from ChamCong order by NgayChamCong ASC |
| Sắp xếp theo trạng thái Đi làm | select \* from ChamCong where TrangThai = 'true' |
| Sắp xếp theo trạng thái Vắng mặt | select \* from ChamCong where TrangThai = 'false' |
| Hiển thị theo mã | SELECT \* FROM ChamCong WHERE MaNV = ? |

#### SQL đối với Bảng NguoiDung

|  |  |
| --- | --- |
| Hành động | Câu lệnh SQL |
| Thêm mới | INSERT INTO NguoiDung(MaNV, MatKhau, VaiTro) VALUES (?,?,?) |
| Cập nhật theo mã | UPDATE NguoiDung SET MatKhau=?, VaiTro=? WHERE MaNV=? |
| Xóa theo mã | DELETE FROM NguoiDung WHERE MaNV=? |
| Hiển thị tất cả | SELECT \* FROM NguoiDung |
| Hiển thị theo mã | SELECT \* FROM NguoiDung WHERE MaNV = ? |

#### SQL đối với Bảng CongViec

|  |  |
| --- | --- |
| Hành động | Câu lệnh SQL |
| Thêm mới | INSERT INTO CongViec(MaCV, MaNV, MaPB, TenCV ) VALUES (?,?,?,?) |
| Cập nhật theo mã | UPDATE CongViec SET TenCV=? WHERE MaCV=? |
| Xóa theo mã | DELETE FROM CongViec WHERE MaCV=? |
| Hiển thị tất cả | SELECT \* FROM CongViec |
| Hiển thị theo mã | SELECT \* FROM CongViec WHERE MaCV = ? |

### Các thủ tục lưu tổng hợp thống kê

#### Procedure 1

create proc [dbo].[sp\_ThongKeLuong1]

(@year int)

as

begin

select

NhanVien.MaNV,

NhanVien.HoTen,

LuongCoBan,

HeSo,

TienThuong,

TroCap,

((LuongCoBan \* HeSo) + TienThuong + TroCap) TongTien

from Luong inner join NhanVien on Luong.MaNV = NhanVien.MaNV

where YEAR(NgayDangKi) = @year

group by NhanVien.MaNV, NhanVien.HoTen, LuongCoBan, HeSo, TienThuong, TroCap

end

#### Procedure 2

CREATE PROC [dbo].[sp\_ThongKeNhanVien]

AS BEGIN

SELECT

YEAR(NgayDangKi) Nam,

COUNT(\*) SoLuong,

MIN(NgayDangKi) DauTien,

MAX(NgayDangKi) CuoiCung

FROM NhanVien

GROUP BY YEAR(NgayDangKi)

END

## Lập trình JDBC

### Lớp hỗ trợ

#### Lớp tiện ích Auth

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package com.chandq.pd05208\_project1.utils;

import com.chandq.pd05208\_project1.entity.NguoiDung;

/\*\*

\*

\* @author ASUS

\*/

public class Auth {

//đối tượng này chứa thông tin người dùng sau khi đăng nhập

public static NguoiDung user = null;

//xóa thong tiin ngươi dùng khi có yeu cau đăng xuất

public static void clear(){

Auth.user = null;

}

//kiem tra dăng nhập hay chua

public static boolean isLogin(){

return Auth.user != null;

}

//kiem tra xem co phai la truong phong hay khong

public static boolean isManager(){

return Auth.isLogin() && user.isVaiTro();

}

}

#### Lớp tiện ích MsgBox

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package com.chandq.pd05208\_project1.utils;

import java.awt.Component;

import javax.swing.JOptionPane;

/\*\*

\*

\* @author ASUS

\*/

public class MsgBox {

//hiển thị thông báo cho người dùng

//parent là cửa sổ chứa thông báo

//mesage là thông báo

public static void alert(Component parent, String message){

JOptionPane.showMessageDialog(parent, message,

"Hệ thống quán lí nhân sự", JOptionPane.INFORMATION\_MESSAGE);

}

public static boolean confirm(Component parent, String message){

int result = JOptionPane.showConfirmDialog(parent, message,

"Hệ thống quán lí nhân sự", JOptionPane.YES\_NO\_OPTION, JOptionPane.QUESTION\_MESSAGE);

return result == JOptionPane.YES\_OPTION;

}

//hiển thị thông báo yêu cầu nhập dữ liệu

//parent là cửa sổ chứa thông báo

//mesage là thông báo nhắc nhở nhập

//return là kết quả nhận được

public static String promt(Component parent, String message){

return JOptionPane.showInputDialog(parent, message,

"Hệ thống quản lí nhân sự", JOptionPane.INFORMATION\_MESSAGE);

}

}

#### Lớp tiện ích Xdate

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package com.chandq.pd05208\_project1.utils;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.Date;

/\*\*

\*

\* @author ASUS

\*/

public class XDate {

static SimpleDateFormat formater = new SimpleDateFormat();

public static Date toDate(String date, String pattern){

try {

formater.applyPattern(pattern);

return formater.parse(date);

} catch (ParseException ex) {

throw new RuntimeException(ex);

}

}

public static String toString(Date date, String pattern){

formater.applyPattern(pattern);

return formater.format(date);

}

public static Date addDays(Date date, long days){

date.setTime(date.getTime()+ days\*24\*60\*60\*1000);

return date;

}

}

#### Lớp tiện ích Ximage

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package com.chandq.pd05208\_project1.utils;

import java.awt.Image;

import java.io.File;

import java.nio.file.Files;

import java.nio.file.Path;

import java.nio.file.Paths;

import java.nio.file.StandardCopyOption;

import javax.swing.ImageIcon;

/\*\*

\*

\* @author ASUS

\*/

public class XImage {

//ảnh của biểu tượng của ứng dụng xuất hiện trên mọi cửa sổ

public static Image getAppIcon(){

//String file = "/images/fpt.png/ ";

String file = "D:\\icon\\lee.pnj\\ ";

return new ImageIcon(XImage.class.getResource(file)).getImage();

}

public static void save(File src){

File dst = new File("D:\\icon\\", src.getName());

if(!dst.getParentFile().exists()){

dst.getParentFile().mkdirs();

}

try {

Path from = Paths.get(src.getAbsolutePath());

Path to = Paths.get(src.getAbsolutePath());

Files.copy(from,to, StandardCopyOption.REPLACE\_EXISTING);

} catch (Exception ex) {

throw new RuntimeException(ex);

}

}

public static ImageIcon read(String fileName){

File path = new File("D:\\icon\\"+ fileName);

return new ImageIcon(path.getAbsolutePath());

}

}

#### Lớp tiện ích XJdbc

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package com.chandq.pd05208\_project1.utils;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

/\*\*

\*

\* @author ASUS

\*/

public class XJdbc {

private static String driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";

private static String dburl = "jdbc:sqlserver://localhost:1433;database=QLNhanSu";

private static String username = "sa";

private static String password = "12345";

static {

try {

Class.forName(driver);

} catch (ClassNotFoundException ex) {

throw new RuntimeException(ex);

}

}

public static PreparedStatement getStmt(String sql, Object...args) throws SQLException{

Connection connection = DriverManager.getConnection(dburl, username, password);

PreparedStatement pstmt = null;

if(sql.trim().startsWith("{")){

pstmt = connection.prepareCall(sql);

}

else{

pstmt = connection.prepareStatement(sql);

}

for(int i = 0; i<args.length; i++){

pstmt.setObject(i+1, args[i]);

}

return pstmt;

}

public static void update(String sql, Object...args){

try {

PreparedStatement stmt = XJdbc.getStmt(sql, args);

try {

stmt.executeUpdate();

}finally{

stmt.getConnection().close();

}

} catch (SQLException e) {

throw new RuntimeException(e);

}

}

public static ResultSet query(String sql, Object...args){

try {

PreparedStatement stmt = XJdbc.getStmt(sql, args);

return stmt.executeQuery();

} catch (SQLException e) {

e.printStackTrace();

throw new RuntimeException(e);

}

}

public static Object value(String sql, Object...args){

try {

ResultSet rs = XJdbc.query(sql, args);

if(rs.next()){

return rs.getObject(0);

}

rs.getStatement().getConnection().close();

return null;

} catch (Exception e) {

throw new RuntimeException(e);

}

}

}

### Model class - Các lớp mô tả dữ liệu

#### Model ChamCong

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package com.chandq.pd05208\_project1.entity;

import java.util.Date;

/\*\*

\*

\* @author ASUS

\*/

public class ChamCong {

private int STT;

private String maNV;

private String hoTen;

private Date ngaychamCong = new Date();

private boolean trangThai = false;

private String ghiChu;

public int getSTT() {

return STT;

}

public void setSTT(int STT) {

this.STT = STT;

}

public String getMaNV() {

return maNV;

}

public void setMaNV(String maNV) {

this.maNV = maNV;

}

public String getHoTen() {

return hoTen;

}

public void setHoTen(String hoTen) {

this.hoTen = hoTen;

}

public Date getNgaychamCong() {

return ngaychamCong;

}

public void setNgaychamCong(Date ngaychamCong) {

this.ngaychamCong = ngaychamCong;

}

public boolean isTrangThai() {

return trangThai;

}

public void setTrangThai(boolean trangThai) {

this.trangThai = trangThai;

}

public String getGhiChu() {

return ghiChu;

}

public void setGhiChu(String ghiChu) {

this.ghiChu = ghiChu;

}

}

#### Model CongViec

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package com.chandq.pd05208\_project1.entity;

/\*\*

\*

\* @author ASUS

\*/

public class CongViec {

private String maCV;

private String maNV;

private String maPB;

private String tenCV;

public String getMaCV() {

return maCV;

}

public void setMaCV(String maCV) {

this.maCV = maCV;

}

public String getMaNV() {

return maNV;

}

public void setMaNV(String maNV) {

this.maNV = maNV;

}

public String getMaPB() {

return maPB;

}

public void setMaPB(String maPB) {

this.maPB = maPB;

}

public String getTenCV() {

return tenCV;

}

public void setTenCV(String tenCV) {

this.tenCV = tenCV;

}

}

#### Model Luong

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package com.chandq.pd05208\_project1.entity;

import java.util.Date;

/\*\*

\*

\* @author ASUS

\*/

public class Luong {

private int STT;

private String maNV;

private String hoTen;

private int luongcoBan;

private int heSo;

private int tienThuong;

private int troCap;

private Date ngaytinhLuong = new Date();

public int getSTT() {

return STT;

}

public void setSTT(int STT) {

this.STT = STT;

}

public String getMaNV() {

return maNV;

}

public void setMaNV(String maNV) {

this.maNV = maNV;

}

public String getHoTen() {

return hoTen;

}

public void setHoTen(String hoTen) {

this.hoTen = hoTen;

}

public int getLuongcoBan() {

return luongcoBan;

}

public void setLuongcoBan(int luongcoBan) {

this.luongcoBan = luongcoBan;

}

public int getHeSo() {

return heSo;

}

public void setHeSo(int heSo) {

this.heSo = heSo;

}

public int getTienThuong() {

return tienThuong;

}

public void setTienThuong(int tienThuong) {

this.tienThuong = tienThuong;

}

public int getTroCap() {

return troCap;

}

public void setTroCap(int troCap) {

this.troCap = troCap;

}

public Date getNgaytinhLuong() {

return ngaytinhLuong;

}

public void setNgaytinhLuong(Date ngaytinhLuong) {

this.ngaytinhLuong = ngaytinhLuong;

}

public int getTongLuong(){

return luongcoBan\* heSo + tienThuong + troCap ;

}

}

### DAO Class - Các lớp truy xuất dữ liệu

#### Lớp DAO làm việc với Bảng NhanVien

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package com.chandq.pd05208\_project1.dao;

import com.chandq.pd05208\_project1.entity.NhanVien;

import com.chandq.pd05208\_project1.utils.XJdbc;

import java.util.List;

import java.sql.\*;

import java.util.\*;

/\*\*

\*

\* @author ASUS

\*/

public class NhanVienDAO extends NhanSuDAO<NhanVien, String>{

public void insert(NhanVien model){

String sql = "INSERT INTO NhanVien(MaNV, HoTen, GioiTinh, NgaySinh, DiaChi,NgayDangKi, SDT, Email, Hinh, MaPB) VALUES (?,?,?,?,?,?,?,?,?,?)";

XJdbc.update(sql,

model.getMaNV(),

model.getHoTen(),

model.isGioiTinh(),

model.getNgaySinh(),

model.getDiaChi(),

model.getNgayDK(),

model.getSdt(),

model.getEmail(),

model.getHinh(),

model.getMaPB());

}

public void update(NhanVien model){

String sql= "UPDATE NhanVien SET HoTen=?, GioiTinh=?, NgaySinh=?, DiaChi=?,NgayDangKi=?, SDT=?, Email=?, Hinh=?, MaPB=? WHERE MaNV=? ";

XJdbc.update(sql,

model.getHoTen(),

model.isGioiTinh(),

model.getNgaySinh(),

model.getDiaChi(),

model.getNgayDK(),

model.getSdt(),

model.getEmail(),

model.getHinh(),

model.getMaPB(),

model.getMaNV());

}

public void delete(String MaNV){

String sql= "DELETE FROM NhanVien WHERE MaNV=?";

XJdbc.update(sql, MaNV);

}

public List<NhanVien> selectAll(){

String sql = "SELECT \* FROM NhanVien";

return selectBySql(sql);

}

public NhanVien selectById(String manv){

String sql= "SELECT \* FROM NhanVien WHERE MaNV = ? ";

List<NhanVien> list = selectBySql(sql, manv);

return list.size() > 0 ? list.get(0) : null;

}

protected List<NhanVien> selectBySql(String sql, Object...args){

List<NhanVien> list = new ArrayList<>();

try {

ResultSet rs = null;

try {

rs = XJdbc.query(sql, args);

while (rs.next()) {

NhanVien entity = new NhanVien();

entity.setMaNV(rs.getString("MaNV"));

entity.setHoTen(rs.getString("HoTen"));

entity.setGioiTinh(rs.getBoolean("GioiTinh"));

entity.setNgaySinh(rs.getDate("NgaySinh"));

entity.setDiaChi(rs.getString("DiaChi"));

entity.setNgayDK(rs.getDate("NgayDangKi"));

entity.setSdt(rs.getString("SDT"));

entity.setEmail(rs.getString("Email"));

entity.setHinh(rs.getString("Hinh"));

entity.setMaPB(rs.getString("MaPB"));

list.add(entity);

}

}

finally{

rs.getStatement().getConnection().close();

}

} catch (SQLException ex) {

throw new RuntimeException(ex);

}

return list;

}

public List<NhanVien> selectByKeyword(String keyword){ //tim kiem

String sql = "SELECT \* FROM NhanVien WHERE HoTen LIKE ?";

return this.selectBySql(sql, "%"+keyword+"%");

}

public List<NhanVien> selectByTimMa(String ma){ //tim kiem

String sql = "SELECT \* FROM NhanVien WHERE MaNV LIKE ?";

return this.selectBySql(sql, "%"+ma+"%");

}

}

#### Lớp DAO làm việc với Bảng ThongKe

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package com.chandq.pd05208\_project1.dao;

import com.chandq.pd05208\_project1.utils.XJdbc;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.List;

/\*\*

\*

\* @author ASUS

\*/

public class ThongKeDAO {

private List<Object[]> getListOfArray(String sql, String[] cols, Object...args) {

try {

List<Object[]> list = new ArrayList<>();

ResultSet rs = XJdbc.query(sql, args);

while (rs.next()){

Object[] vals = new Object[cols.length];

for(int i = 0; i < cols.length; i++){

vals[i] = rs.getObject(cols[i]);

}

list.add(vals);

}

rs.getStatement().getConnection().close();

return list;

} catch (SQLException ex) {

throw new RuntimeException(ex);

}

}

public List<Object[]> getNhanVien() {

String sql = "{CALL sp\_ThongKeNhanVien }";

String[] cols = {"Nam", "SoLuong", "DauTien", "CuoiCung"};

return this.getListOfArray(sql, cols);

}

public List<Object[]> getLuong(int nam) {

String sql = "{CALL sp\_ThongKeLuong1 (?)}";

String[] cols = {"MaNV", "HoTen", "LuongCoBan","HeSo", "TienThuong", "TroCap", "TongTien"};

return this.getListOfArray(sql, cols, nam);

}

}

#### Lớp DAO làm việc với Bảng ChamCong

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package com.chandq.pd05208\_project1.dao;

import com.chandq.pd05208\_project1.entity.ChamCong;

import com.chandq.pd05208\_project1.utils.XJdbc;

import java.util.ArrayList;

import java.util.List;

import java.sql.\*;

import java.util.\*;

/\*\*

\*

\* @author ASUS

\*/

public class ChamCongDAO extends NhanSuDAO<ChamCong, String> {

public void insert(ChamCong model) {

String sql = "INSERT INTO ChamCong(MaNV, HoTen, NgayChamCong, TrangThai, GhiChu) VALUES (?,?,?,?,?)";

XJdbc.update(sql,

model.getMaNV(),

model.getHoTen(),

model.getNgaychamCong(),

model.isTrangThai(),

model.getGhiChu());

}

public void update(ChamCong model) {

String sql = "UPDATE ChamCong SET HoTen=?, NgayChamCong=?, TrangThai=?, GhiChu=? WHERE MaNV=? ";

XJdbc.update(sql,

model.getHoTen(),

model.getNgaychamCong(),

model.isTrangThai(),

model.getGhiChu(),

model.getMaNV());

}

public void delete(String MaNV) {

String sql = "DELETE FROM ChamCong WHERE MaNV=?";

XJdbc.update(sql, MaNV);

}

public List<ChamCong> selectAll() {

String sql = "SELECT \* FROM ChamCong";

return selectBySql(sql);

}

public List<ChamCong> TheoNgay() {

String sql = "select \* from ChamCong order by NgayChamCong ASC";

return selectBySql(sql);

}

public List<ChamCong> DiLam() {

String sql = "select \* from ChamCong where TrangThai = 'true'";

return selectBySql(sql);

}

public List<ChamCong> VangMat() {

String sql = "select \* from ChamCong where TrangThai = 'false'";

return selectBySql(sql);

}

public ChamCong selectById(String manv) {

String sql = "SELECT \* FROM ChamCong WHERE MaNV = ? ";

List<ChamCong> list = selectBySql(sql, manv);

return list.size() > 0 ? list.get(0) : null;

}

protected List<ChamCong> selectBySql(String sql, Object... args) {

List<ChamCong> list = new ArrayList<>();

try {

ResultSet rs = null;

try {

rs = XJdbc.query(sql, args);

while (rs.next()) {

ChamCong entity = new ChamCong();

entity.setMaNV(rs.getString("MaNV"));

entity.setHoTen(rs.getString("HoTen"));

entity.setNgaychamCong(rs.getDate("NgayChamCong"));

entity.setTrangThai(rs.getBoolean("TrangThai"));

entity.setGhiChu(rs.getString("GhiChu"));

list.add(entity);

}

} finally {

rs.getStatement().getConnection().close();

}

} catch (SQLException ex) {

throw new RuntimeException(ex);

}

return list;

}

}

## Viết mã cho ứng dụng

### Xử lý Form DangNhap

public DangNhap(java.awt.Frame parent, boolean modal) {

super(parent, modal);

setTitle("HỆ THỐNG QUẢN LÍ NHÂN SỰ");

initComponents();

setLocationRelativeTo(null);

}

NguoiDungDAO dao = new NguoiDungDAO();

void DangNhap(){

String manv = txtMaNV.getText();

String matkhau = new String (txtMatKhau.getPassword());

NguoiDung nguoiDung = dao.selectById(manv);

if (nguoiDung == null) {

MsgBox.alert(this, "Sai tên đăng nhập!");

} else if (!matkhau.equals(nguoiDung.getMatKhau())) {

MsgBox.alert(this, "Sai mật khẩu!");

} else {

//MsgBox.alert(this, "Thành công!");

Auth.user = nguoiDung;

this.dispose();

//new GiaoDienChinh().setVisible(true);

}

}

void ketThuc() {

if (MsgBox.confirm(this, "Bạn muốn đóng ứng dụng?")) {

System.exit(0);

}

}

### Xử lý Form GiaoDienChinh

public GiaoDienChinh() {

new chao(this, true).setVisible(true);

new DangNhap(this, true).setVisible(true);

initComponents();

setLocationRelativeTo(null);

setTitle("HỆ THỐNG QUẢN LÍ NHÂN SỰ");

this.startDongHo();

}

void startDongHo(){

SimpleDateFormat formater = new SimpleDateFormat("hh:mm:ss a");

new Timer(1000, (ActionEvent e) ->{

lblDongHo.setText(formater.format(new Date()));

}).start();

}

void openDoiMatKhau(){

if(Auth.isLogin()){

new DoiMatKhau(this, true).setVisible(true);

}

else{

MsgBox.alert(this, "Vui lòng đăng nhập!");

}

}

void openGioiThieu(){

if(Auth.isLogin()){

new GioiThieu1(this,true).setVisible(true);

}

else{

MsgBox.alert(this, "Vui lòng đăng nhập!");

}

}

void dangXuat(){

Auth.clear();

new DangNhap(this, true).setVisible(true);

}

void ketThuc(){

if(MsgBox.confirm(this, "Bạn muốn kết thúc?")){

System.exit(0);

}

}

void openNhanVien(){

if(Auth.isLogin()){

new QLNhanVien(this, true).setVisible(true);

}

else{

MsgBox.alert(this, "Vui lòng đăng nhập!");

}

}

void openPhongBan(){

if(Auth.isLogin()){

new QLPhongBan(this, true).setVisible(true);

}

else{

MsgBox.alert(this, "Vui lòng đăng nhập!");

}

}

void openCongViec(){

if(Auth.isLogin()){

new QLCongViec(this, true).setVisible(true);

}

else{

MsgBox.alert(this, "Vui lòng đăng nhập!");

}

}

void openChamCong(){

if(Auth.isLogin()){

new QLChamCong(this, true).setVisible(true);

}

else{

MsgBox.alert(this, "Vui lòng đăng nhập!");

}

}

void openLuong(){

if(Auth.isLogin()){

new QLLuong(this, true).setVisible(true);

}

else{

MsgBox.alert(this, "Vui lòng đăng nhập!");

}

}

void openNguoiDung(){

if(Auth.isLogin()){

new QLNguoiDung(this, true).setVisible(true);

}

else{

MsgBox.alert(this, "Vui lòng đăng nhập!");

}

}

void openThongKe(){

if(!Auth.isManager()){

MsgBox.alert(this, "Bạn không có quyền xem thống kê");

}else{

new ThongKe(this, true).setVisible(true);

}

// if(Auth.isLogin()){

// new ThongKe(this, true).setVisible(true);

// }

// else{

// MsgBox.alert(this, "Vui lòng đăng nhập!");

// }

}

### Xử lý Form Chao

public chao(java.awt.Frame parent, boolean modal) {

super(parent, modal);

initComponents();

setLocationRelativeTo(null);

init();

}

void init(){

new Timer (10, new ActionListener() {

@Override

public void actionPerformed(ActionEvent e){

int value = jProgressBar2.getValue();

if(value < jProgressBar2.getMaximum()){

jProgressBar2.setValue(value +1);

//jProgressBar2.setStringPainted(true);

}

else{

chao.this.dispose();

}

}

}).start();

}

# Kiểm thử

## Kiểm thử form NhanVien

|  |  |  |  |
| --- | --- | --- | --- |
| Thuộc tính | Hoạt động | Kết quả | Trạng thái |
|  |  |  |  |
| Mã nhân viên | Không nhập mã nhân viên bấm nút Thêm | Hiển thị thông báo chưa nhập mã | Hoàn thành |
| Họ tên | Không nhập họ tên bấm nút Thêm | Hiển thị thông báo họ tên sinh viên chưa nhập | Hoàn thành |
| Ngày sinh | Không nhập ngày sinh | Hiển thị thông báo ngày sinh chưa được nhập | Hoàn thành |
| Địa chỉ | Không nhập địa chỉ | Hiển thị thông báo địa chỉ chưa nhập | Hoàn thành |
| Ngày đăng kí | Không nhập nhày đăng kí | Hiển thị thông báo ngày đăng kí chưa được nhập | Hoàn thành |
| SDT | Không nhập SDT, nhập sai định dạng | Hiển thị thông báo SDT chưa nhập, SDT phải đúng định dạng | Hoàn thành |
| Email | Không nhập Email, nhập sai định dạng | Hiển thị thông báo Email chưa nhập, Email phải đúng định dạng | Hoàn thành |
| Mã phòng ban | Không nhập mã phòng ban | Hiển thị thông báo phòng ban không được để trống | Hoàn thành |
| Nút thêm | Nhập đầy đủ các trường bấm nút Thêm | Hiển thị thống báo thêm mới thành công | Hoàn thành |
| Nút sửa | Chọn 1 hàng trong bảng, nhập thông tin cần thay đổi, bấm nút sửa | Hiển thị thông báo cập nhật mới thành công | Hoàn thành |
| Nút xóa | Chọn 1 hàng trong bảng, bấm nút xóa | Hiển thị thông báo xóa thành công | Hoàn thành |

## Kiểm thử form Luong

|  |  |  |  |
| --- | --- | --- | --- |
| Thuộc tính | Hoạt động | Kết quả | Trạng thái |
|  |  |  |  |
| Mã nhân viên | Không nhập mã nhân viên bấm nút Tìm kiếm | Hiển thị thông báo không tìm thấy mã nhân viên | Hoàn thành |
| Nhập mã nhân viên có trong bảng nhân viên | Đổ dữ liệu lên các trường tương ứng trong frame | Hoàn thành |
| Lương | Để trống | Hiển thị thông báo chưa nhập lương | Hoàn thành |
| Nhập kí tự không phải số hoặc số nhỏ hơn 1 | Hiển thị thông báo lương phải là số, lương phải lớn hơn 0 | Hoàn thành |
| Hệ số | Để trống | Hiển thị thông báo chưa nhập hệ số | Hoàn thành |
| Nhập kí tự không phải số hoặc số nhỏ hơn 1 | Hiển thị thông báohệ số phải là số, hệ số phải lớn hơn 0 | Hoàn thành |
| Tiền thưởng | Để trống | Hiển thị thông báo chưa nhập tiền thưởng | Hoàn thành |
| Trợ cấp | Để trống | Hiển thị thông báo chưa nhập trợ cấp | Hoàn thành |
| Ngày tính lương | Để trống | Hiển thị thông báo chưa nhập ngày | Hoàn thành |
| Nút thêm | Nhập đầy đủ các trường nhấn nút Thêm | Hiển thị thông báo Thêm mới thành công | Hoàn thành |
| Nút sửa | Chọn 1 hàng trong bảng, nhập thông tin cần thay đổi, bấm nút sửa | Hiển thị thông báo cập nhật mới thành công | Hoàn thành |
| Nút xóa | Chọn 1 hàng trong bảng, bấm nút xóa | Hiển thị thông báo xóa thành công | Hoàn thành |

## Kiểm thử form ChamCong

|  |  |  |  |
| --- | --- | --- | --- |
| Thuộc tính | Hoạt động | Kết quả | Trạng thái |
|  |  |  |  |
| Mã nhân viên | Không nhập mã nhân viên bấm nút Tìm kiếm | Hiển thị thông báo không tìm thấy mã nhân viên | Hoàn thành |
| Nhập mã nhân viên có trong bảng nhân viên | Đổ dữ liệu lên các trường tương ứng trong frame | Hoàn thành |
| Ngày chấm công | Để trống | Hiển thị thông báo chưa nhập ngày | Hoàn thành |
| Nút thêm | Nhập đầy đủ các trường nhấn nút Thêm | Hiển thị thông báo Thêm mới thành công | Hoàn thành |
| Nút sửa | Chọn 1 hàng trong bảng, nhập thông tin cần thay đổi, bấm nút sửa | Hiển thị thông báo cập nhật mới thành công | Hoàn thành |
| Nút xóa | Chọn 1 hàng trong bảng, bấm nút xóa | Hiển thị thông báo xóa thành công | Hoàn thành |
| Nút sắp xếp theo năm | Nhấn nút sắp xếp theo năm trong bảng | Hiển thị thông tin theo năm tăng dần | Hoàn thành |
| Nút đi làm | Nhấn nút sắp xếp theo tình trạng đi làm | Hiển thị thông tin nhân viên có trạng thái đi làm | Hoàn thành |
| Nút vắng mặt | Nhấn nút sắp xếp theo tình trang vắng mặt | Hiển thị thông tin nhân viên có trạng thái vắng mặt | Hoàn thành |

# Đóng gói và triển khai

## Hướng dẫn chuyển đổi jar thành exe

* Sử dụng phần mềm EXE4J để chuyển đổi jar sang exe
* Sử dụng phần mềm InnoSetup để đóng gói

## Hướng dẫn cài đặt triển khai

* Cài SQL Server 2008 trở lên
* Tạo CSDL QLNhanSu bằng cách chạy file NhanSu.sql
* Cài đặt JDK 1.8 trở lên
* Chạy file setup

## Hướng dẫn sử dụng phần mềm

* Đăng nhập với tài khoản username: admin và password:admin
* Cấp các tài khoản có vai trò là trưởng phòng
* Đăng nhập tài khoản mới và xóa tài khoản admin
* Xem hướng dẫn sử dụng trong menu trợ giúp của phần mềm