

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC PHENIKAA**



QUẢN LÝ NHÂN SỰ CÔNG TY

Course/Học phần:	Lập trình hướng đối tượng
Code/Mã học phần:	CSE
Class/Lớp:	N02
Instructor/Giảng viên:	Nguyễn Thành Công

STT	Họ và tên	Mã sinh viên
1	Đàm Gia Phú	23010760
2	Lê Hồng Phúc	23010746

Hà Nội, tháng 3 năm 2025

MỤC LỤC

LỜI MỞ ĐẦU	3
1. Giới thiệu bài toán.....	4
2. Xây dựng sơ đồ lớp	6
a) Lớp NhanSu.....	6
b) Lớp CongViec.....	9
c) Lớp QuanLy.....	11
d) Lớp LoginFrame.....	12
e) Lớp MainFrame.....	13
3. Giao diện người dùng.....	17
4. Kết quả hoạt động	17
5. Tài liệu tham khảo.....	17

LỜI MỞ ĐẦU

Ngày nay, cùng với sự phát triển của khoa học công nghệ, các ứng dụng tin học đang là một phần không thể thiếu trong cuộc sống con người, nó góp phần đẩy mạnh công nghiệp hóa, hiện đại hóa. Máy tính cùng với những phần mềm là công cụ đắc lực giúp ta quản lý, tổ chức, sắp xếp và xử lý công việc một cách nhanh chóng và chính xác.

Ở Việt Nam hiện nay, máy tính điện tử, đặc biệt là máy vi tính trong nhiều năm qua đã được sử dụng rộng rãi. Sự phát triển của công nghệ phần mềm, phần cứng đã đưa chúng ta từng bước tiếp cận với công nghệ thông tin trong mọi lĩnh vực nhằm đáp ứng những nhu cầu con người. Một trong số đó là việc đưa tin học vào ứng dụng để quản lý rất hữu ích vì chúng ta bỏ ra ít thời gian mà lại thu được hiệu quả cao, chính xác và tiện lợi.

Quản lý công ty là một trong những công việc tương đối phức tạp, tốn nhiều thời gian và công sức. Chính vì vậy, tin học hóa trong lĩnh vực quản lý là một công việc yêu cầu tất yếu. Muốn quản lý tốt cần có được các phần mềm tốt, phần mềm phải đảm bảo được độ bảo mật cao, dễ sử dụng và nhiều tiện ích. Đề tài là một yêu cầu thiết thực trong quản lý của hầu hết các công ty đang hoạt động hiện nay

Từ lâu bài toán về quản lý đã được ứng dụng rộng rãi trong hầu hết các lĩnh vực. Hiểu được tầm quan trọng của môn học cùng với các kiến thức đã tích lũy trong suốt học kỳ, nhóm chúng em xin lựa chọn đề tài “Ứng dụng quản lý Công Ty”

1. Giới thiệu bài toán

Trong các công ty và xí nghiệp, việc quản lý nhân sự, công việc và lương là một yếu tố quan trọng nhằm đảm bảo các hoạt động được duy trì ổn định. Hệ thống này giúp giảm thiểu sai sót trong việc quản lý và tính toán lương, đồng thời tạo sự thuận tiện cho quản lý.

Báo cáo này trình bày cách xây dựng hệ thống quản lý công ty bằng ngôn ngữ nhập trình Java, áp dụng các nguyên tắc lập trình hướng đối tượng để tối ưu hóa quy trình quản lý và dễ dàng mở rộng hệ thống trong tương lai.

a. Tính năng chính

Ứng dụng cung cấp các tính năng sau:

- **Đăng nhập:** Người dùng cần đăng nhập với tài khoản hợp lệ để truy cập vào các chức năng của ứng dụng.
- **Quản lý nhân sự:** Nhập, sửa, xóa và lưu trữ thông tin nhân viên bao gồm:
 - Mã nhân viên
 - Tên nhân viên
 - Năm sinh
 - Giới tính
 - Phòng ban
 - Lương
 - Danh sách các công việc ứng với mã Nhân viên
- **Quản lý công việc:** Thêm, sửa xóa và lưu trữ thông tin công việc bao gồm:
 - Mã công việc
 - Tên công việc
 - Số giờ làm công việc
 - Mã nhân viên phụ trách
- **Quản lý công ty:** Tự động thêm, sửa, xóa và lưu trữ danh sách nhân sự, danh sách công việc
- **Tìm kiếm:** Tìm kiếm thông tin nhân viên, công việc dựa trên mã nhân viên, công việc tương ứng, hiển thị thông tin cần tìm kiếm một cách rõ ràng, chính xác

b. Thiết kế giao diện

CHƯƠNG TRÌNH QUẢN LÝ CÔNG TY

ĐĂNG NHẬP

Tên đăng nhập:

Mật khẩu:

ĐĂNG NHẬP

Hình 1: Giao diện đăng nhập vào ứng dụng

Nhân Viên Công Việc Tìm Kiếm

Mã Nhân Viên:

Tên Nhân Viên:

Năm Sinh:

Giới Tính: ☐ Nam ☐ Nữ

Phòng Ban:

Thêm Sửa Xóa

Mã NV	Tên NV	Năm Sinh	Giới Tính	Phòng Ban

Hình 2: Giao diện quản lý Nhân viên

Nhân Viên Công Việc Tìm Kiếm

Mã Công Việc:

Tên Công Việc:

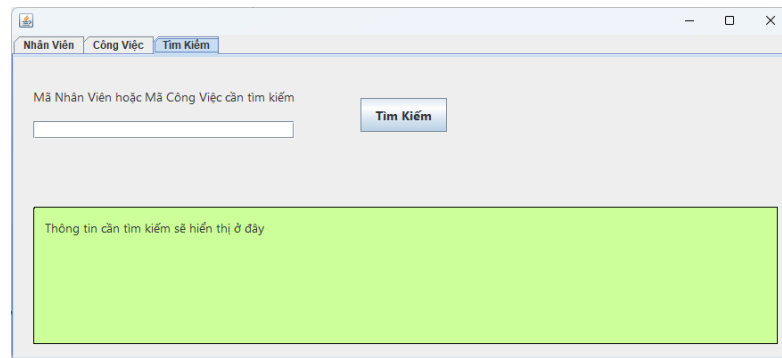
Số Giờ Làm Việc:

Mã NV Phụ Trách:

Thêm Sửa Xóa

Mã CV	Tên CV	Số Giờ Làm	Mã NV Phụ Trách

Hình 3: Giao diện quản lý Công việc



Hình 4: Giao diện tìm kiếm Nhân viên/ Công việc

Phần giao diện của ứng dụng được thiết kế với nguyên tắc đơn giản và dễ sử dụng, đảm bảo người dùng có trải nghiệm mượt mà và thuận tiện nhất có thể. Mọi thành phần, từ các nút chức năng đến các hộp thoại và menu đều được sắp xếp một cách logic và trực quan. Thông tin được hiển thị một cách rõ ràng và dễ hiểu, giúp người dùng nhanh chóng nắm bắt mọi thông tin họ cần.

2. Xây dựng sơ đồ lớp

a) Lớp NhanSu

- **Thuộc tính:**

- String idNhanVien: Mã nhân viên, kiểu dữ liệu String. Thuộc tính này lưu lại mã định danh của nhân viên.
- String tenNhanVien: Tên nhân viên, kiểu dữ liệu String. Thuộc tính này lưu lại tên của nhân viên
- int namSinh: Năm sinh của nhân viên, kiểu dữ liệu int. Thuộc tính này lưu trữ lại năm sinh của nhân viên.
- String gioiTinh: Giới tính của nhân viên, kiểu dữ liệu String. Thuộc tính lưu lại giới tính của nhân viên.
- String phongBan: Phòng mà nhân viên làm việc, kiểu dữ liệu String. Thuộc tính lưu trữ phòng làm việc của nhân viên.
- ArrayList <CongViec> dsCongViec: Danh sách công việc mà nhân viên đảm nhiệm. Lưu trữ lại những công việc mà nhân viên đã làm.

- **Constructor:**

Hàm tạo (Constructor) này khởi tạo đối tượng NhanVien với các giá trị được truyền vào:

- Constructor 1 gồm các thuộc tính:
 - idNhanVien: Mã nhân viên
 - tenNhanVien: Tên nhân viên
 - namSinh: Năm sinh
 - gioiTinh: Giới tính

- phongBan: Phòng ban
- dsCongViec: Danh sách công việc
- Constructor 2 gồm các thuộc tính:
 - idNhanVien: Mã nhân viên
 - tenNhanVien: Tên nhân viên
 - namSinh: Năm sinh
 - gioiTinh: Giới tính
 - phongBan: Phòng ban
 - luong: Lương nhân viên
 - dsCongViec: Danh sách công việc
- **Phương thức (Methods):**
 - **Getter Methods:** Các phương thức này trả về giá trị của các thuộc tính của đối tượng CanHo.
 - getIdNhanVien(): Trả về mã nhân viên
 - getTenNhanVien(): Trả về tên nhân viên
 - getNamSinh(): Trả về năm sinh
 - getGioiTinh(): Trả về giới tính của nhân viên
 - getPhongBan(): Trả về tên phòng ban của nhân viên.
 - getDsCongViec(): Trả về danh sách công việc mà nhân viên thực hiện
 - **Setter Methods:** Các phương thức này cho phép thay đổi giá trị thuộc tính của đối tượng:
 - setIdNhanVien (String idNhanVien): Cập nhật mã nhân viên
 - setTenNhanVien (String tenNhanVien): Cập nhật tên nhân viên
 - setNamSinh (int namSinh): Cập nhật năm sinh
 - setGioiTinh(String gioiTinh): Cập nhật giới tính của nhân viên
 - setPhongBan (String phongBan): Cập nhật phòng làm việc của nhân viên
 - setDsCongViec (ArrayList <CongViec> dsCongViec: Cập nhật danh sách công việc mà nhân viên đảm nhiệm
 - **Methods tínhLuong:** Tính lương của nhân viên. Lương của nhân viên được tính theo công thức:

$$Lương = \text{Lương cơ bản} + (\text{số giờ làm} * \text{số công việc đảm nhiệm})$$
 Sau khi làm tínhLuong() chạy thành công sẽ trả về lương của nhân viên.
 - **Mã code chương trình:**

```

1  public class NhanSu {
2      //1. Attributes
3      private String idNhanVien;
4      private String tenNhanVien;
5      private int NamSinh;
6      private String gioiTinh;
7      private String phongBan;
8      private float luong;
9      private ArrayList <CongViec> dsCongViec;
10
11     //2. Getters/Setters
12     public String getIdNhanVien() {
13         return idNhanVien;
14     }
15
16     public void setIdNhanVien(String idNhanVien) {
17         this.idNhanVien = idNhanVien;
18     }
19
20     public String getTenNhanVien() {
21         return tenNhanVien;
22     }
23
24     public void setTenNhanVien(String tenNhanVien) {
25         this.tenNhanVien = tenNhanVien;
26     }
27
28     public int getNamSinh() {
29         return NamSinh;
30     }
31
32     public void setNamSinh(int NamSinh) {
33         this.NamSinh = NamSinh;
34     }
35
36     public String getGioiTinh() {
37         return gioiTinh;
38     }
39
40     public void setGioiTinh(String gioiTinh) {
41         this.gioiTinh = gioiTinh;
42     }
43
44     public String getPhongBan() {
45         return phongBan;
46     }
47
48     public void setPhongBan(String phongBan) {
49         this.phongBan = phongB

```

Hình 5: Mô tả code lớp NhanSu

- **Giao diện quản lý nhân viên:**

Mã NV	Tên NV	Năm Sinh	Giới Tính	Phòng Ban
-------	--------	----------	-----------	-----------

Hình 6: Giao diện quản lý nhân viên

b) Lớp CongViec

• **Thuộc tính (Attribute):**

- maCV: mã công việc, kiểu dữ liệu String. Thuộc tính này dùng để lưu trữ lại mã định danh của công việc.
- tenCV: tên công việc, kiểu dữ liệu String. Thuộc tính này lưu trữ lại tên của công việc.
- soGioLam: số giờ làm việc, kiểu dữ liệu float. Thuộc tính này lưu trữ lại số giờ để nhân viên hoàn thành công việc.
- maNV: mã nhân viên, kiểu dữ liệu String. Thuộc tính này lưu trữ lại mã nhân viên phụ trách công việc.

• **Constructor:**

- Hàm tạo của class CongViec. Trong hàm này, tất cả các thuộc tính đều được khởi tạo, sẵn sàng để lưu trữ các thuộc tính của đối tượng.

• **Phương thức (Methods):**

- Các phương set, get toàn bộ các thuộc tính của lớp cho phép khởi tạo, trả về các thuộc tính của lớp bao gồm: maCV, tenCV, soGioLam và maNV

- Mã code chương trình

```
1
2 package company;
3
4 public class CongViec {
5     //1. Attributes
6     private String maCV;
7     private String tenCV;
8     private float soGioLam;
9     String maNV;
10
11     //2. Getters/Setters
12     public String getMaCV() {
13         return maCV;
14     }
15
16     public void setMaCV(String maCV) {
17         this.maCV = maCV;
18     }
19
20     public String getTenCV() {
21         return tenCV;
22     }
23
24     public void setTenCV(String tenCV) {
25         this.tenCV = tenCV;
26     }
27
28     public float getSoGioLam() {
29         return soGioLam;
30     }
31
32     public void setSoGioLam(float soGioLam) {
33         this.soGioLam = soGioLam;
34     }
35
36     public String getMaNV() {
37         return maNV;
38     }
39
40     public void setMaNV(String maNV) {
41         this.maNV = maNV;
42     }
43
44     //3. Constructors
45     public CongViec(String maCV, String tenCV, float soGioLam, String maNV) {
46         this.maCV = maCV;
47         this.tenCV = tenCV;
48         this.soGioLam = soGioLam;
49         this.maNV = maNV;
50     }
51 }
```

Hình 7 Mô tả code lớp CongViec

- **Giao diện quản lý công việc:**

Hình 8: Giao diện quản lý công việc

c) Lớp QuanLy

- **Thuộc tính (Attribute):**

- List<CongViec> dsCongViec: Danh sách công việc. Lưu trữ, quản lý toàn bộ danh sách công việc của công ty.
- List<NhanSu> dsNhanVien: Danh sách nhân viên. Lưu trữ, quản lý toàn bộ danh sách nhân viên của công ty.

- **Constructor:**

- Toàn bộ cả hai danh sách dsCongViec và dsCongTy đều được khởi tạo là các List trống, sẵn sàng lưu trữ các đối tượng NhanVien và CongViec

- **Phương thức (Methods):**

- Các phương thức set, get khởi tạo, lưu trữ và quản lý danh sách nhân viên và công việc của công ty.
- themNhanSu (): Giúp thêm nhân viên vào danh sách nhân viên
- themCongViec (): Giúp thêm công việc vào danh sách công việc
- timNhanSu (): Từ mã nhân viên người dung nhập vào, hàm giúp tìm kiếm và trả về nhân viên trong danh sách nhân viên có mã nhân viên trùng khớp.
- timCongViec(): Từ mã công việc người dung nhập vào, hàm giúp tìm kiếm và trả về công việc trong danh sách công việc có mã công việc trùng khớp.

- Mã code chương trình

```
1 package company;
2
3 import java.util.ArrayList;
4 import java.util.List;
5
6 public class QuanLy {
7     //1. Attribute
8
9     private List<CongViec> dsCongViec;
10    private List<NhanSu> dsNhanVien;
11
12    //2. Set, get
13    public List<NhanSu> getDsNhanSu(){
14        return dsNhanVien;
15    }
16
17    public List<CongViec> getDsCongViec() {
18        return dsCongViec;
19    }
20
21    //3. Constructor
22
23    public QuanLy() {
24        dsNhanVien = new ArrayList<>();
25        dsCongViec = new ArrayList<>();
26    }
27    public QuanLy(List<NhanSu> dsNhanSu, List<CongViec> dsCongViec) {
28        this.dsNhanVien = dsNhanSu;
29        this.dsCongViec = dsCongViec;
30    }
31
32    //4. Input, Output
33    public void themNhanSu(NhanSu nhanSu) {
34        dsNhanVien.add(nhanSu);
35    }
36
37    public void themCongViec(CongViec congViec){
38        dsCongViec.add(congViec);
39    }
40
41    public NhanSu timNhanSu(String maNV){
42        for(NhanSu nv : dsNhanVien) {
43            if(nv.getIdNhanVien().equals(maNV)) {
44                return nv;
45            }
46        }
47        return null;
48    }
49
50    public CongViec timCongViec(String maCV) {
51        for(CongViec cv : dsCongViec) {
52            if(cv.getMaCV().equals(maCV)) {
53                return cv;
54            }
55        }
56        return null;
57    }
58 }
59
```

Hình 9: Mô tả code lớp QuanLy

d) Lớp LoginFrame

- Thuộc tính (Attributes):

- JTextField usernameField: Một trường văn bản để người dùng nhập vào tên đăng nhập.
- JPasswordField passwordField: Một trường mật khẩu giúp người dùng nhập vào mật khẩu đăng nhập vào hệ thống.

- JButton loginButton: Nút giúp người dùng thực hiện xác nhận đăng nhập vào hệ thống.
- **Phương thức (Methods):**
 - Hàm tạo nhận vào giá trị username và password tương ứng với hai trường nhập vào dữ liệu là usernameField và passwordField. Người dùng xác nhận bằng cách bấm vào nút “Đăng nhập”
 - Hàm kiểm tra tên đăng nhập, mật khẩu:
 - Mật khẩu và tên đăng nhập sau khi nhập vào sẽ được lưu trữ vào hai biến username và password.
 - Hàm loginButtonActionPerformed có nhiệm vụ lấy thông tin nhập vào, lưu trữ và kiểm tra tên đăng nhập, mật khẩu có khớp với hệ thống hay không. Nếu khớp, hệ thống sẽ gửi thông báo “Đăng nhập thành công” và thực hiện mở cửa sổ MainFrame, ngược lại thông báo “Kiểm tra lại tên đăng nhập, mật khẩu” và yêu cầu người dùng nhập lại tới khi nào đúng thì thôi.
- **Giao diện:**

Hình 10: Giao diện đăng nhập

e) Lớp MainFrame

- **Thuộc tính (Attributes):**
 - dtmNV, dtmCV: để quản lý dữ liệu trong bảng NhanVien và bảng CongViec.
 - dsNhanVien, dsCongViec: để quản lý, lưu trữ danh sách nhân viên và danh sách công việc.

- quanLy: để quản lý các nhân viên, công việc có trong công ty.
- Ngoài ra còn nhiều phương thức khác giúp lấy, lưu trữ và đưa ra màn hình thông tin cần theo dõi.

- **Constructors:**

- Hàm tạo này khởi tạo đối tượng MainFrame với các thuộc tính:
 - quanLy
 - dtmNV, dtmCV
 - dsNhanVien
 - dsCongViec

- **Phương thức:**

- **themNVButtonActionPerformed:** Phương thức giúp thêm nhân viên vào danh sách nhân viên để quản lý:
 - Các dữ liệu được lấy vào:
 - maNV từ maNVField
 - tenNV từ tenNVField
 - namSinh từ JButtonGroup1 gồm namRadioButton và nuRadioButton: Người dùng chỉ được chọn 1 trong 2 giới tính là Nam hoặc nữ
 - phongBan từ phongComboBox: Người dùng chọn 1 trong 4 phòng ban có sẵn bao gồm : Giám Đốc, Nhân Sự, Kế Toán và Kế Hoạch
 - Khi người dùng bỏ trống 1 trong số các trường thông tin, hệ thống ngay lập tức gửi thông báo đến người dùng. Đặc biệt, thuộc tính năm sinh mà người dùng nhập vào bắt buộc phải ở dạng số nguyên (int)
 - Người dùng xác nhận thêm thông tin nhân viên bằng cách bấm vào nút “Thêm” trên màn hình.
 - Thuộc tính dtmNV lấy ra mô hình dữ liệu của bảng NVTable sau đó thêm các trường thông tin theo thứ tự có sẵn.
 - Đối tượng nhanSu mới được thiết lập và thêm vào dsNhanSu cũng như quanLy.
 - Thông báo “Thêm nhân viên thành công” hiện ra và toàn bộ nội dung trên các trường văn bản bị xoá bỏ.
- **NVTableMouseClicked:** Phương thức giúp lấy ra dữ liệu trong bảng khi nhấp chuột vào. Lúc này, nút “Thêm” sẽ bị vô hiệu hoá để tránh trường hợp một nhân viên xuất hiện hai lần.
- **xoaNVButtonActionPerformed:** Phương thức giúp xoá bỏ nhân viên đã nhấp chuột
 - Xoá dữ liệu trong bảng: Ta chỉ cần câu lệnh dtmNV.removeRow(row) để xoá đi hàng dữ liệu đó.

- Xóa dữ liệu trong list dsNhanVien: Ta cho mã nhân viên cần xóa chạy trong dsNhanVien, đến mã nhân viên trùng với mã nhân viên cần xóa, ta sử dụng hàm remove(nv).
 - Sau khi xóa thông báo “Xóa nhân viên thành công” sẽ hiển thị trên màn hình, đồng thời các trường dữ liệu nhập vào cũng sẽ bị xóa bỏ và nút “Thêm” trở lại trạng thái bình thường.
- **suaNVButtonActionPerformed:** Phương thức giúp sửa thông tin nhân viên
- Cập nhật thông tin trong bảng: Hàm setValueAt giúp thay đổi thông tin ở dòng được nhấp chuột lần lượt ở các cột 0, 1, 2, 3, 4.
 - Cập nhật thông tin trong dsNhanVien: cho i chạy trong dsNhanVien tới khi gặp được mã nhân viên trùng khớp với mã nhân viên cần sửa, nó sẽ thay đổi các thuộc tính của đối tượng này.
 - Sau khi cập nhật cũng sẽ có thông báo và trạng thái nút “Thêm” trở lại bình thường.

Mã NV	Tên NV	Năm Sinh	Giới Tính	Phòng Ban
23010746	Lê Hồng Phúc	2005	Nam	Giám Đốc
23010762	Đàm Gia Phú	2004	Nam	Kế Hoạch

Hình 11: Giao diện thêm, sửa, xóa nhân viên

- **themCVButtonActionPerformed, CVTableMouseClicked, xoaCVButtonActionPerformed, suaCVButtonActionPerformed:** cũng giống như thêm, sửa, xóa và lấy dữ liệu khi nhấp chuột của Nhân viên. Chỉ khác ở chỗ:
- Mã nhân viên phụ trách phải trùng với mã Nhân viên có trong danh sách nhân viên.

The screenshot shows a window with three tabs: 'Nhân Viên', 'Công Việc', and 'Tìm Kiếm'. The 'Công Việc' tab is active. It contains four input fields: 'Mã Công Việc:' with value '23010746KHD', 'Tên Công Việc:' with value 'Ký hợp đồng', 'Số Giờ Làm Việc:' with value '14', and 'Mã NV Phụ Trách:' with value '23010746'. Below these are three buttons: 'Thêm', 'Sửa', and 'Xóa'. To the right is a table with the following data:

Mã CV	Tên CV	Số Giờ Làm	Mã NV Phụ Trách
23010746KHD	Ký hợp đồng	14	23010746

Hình 12: Giao diện thêm, sửa, xóa công việc

- **clearDataNV, clearDataCV:** Giúp xóa thông tin trên các trường dữ liệu nhập vào.
- **searchButtonActionPerformed:** Từ mã nhân viên/ mã công việc nhập vào, phương thức thực hiện tìm kiếm thông tin có trong dsNhanVien/ dsCongViec để đưa ra kết quả. <html>,
 để hiển thị văn bản dưới dạng HTML
 - Với công việc, các thông tin đưa ra bao gồm: Tên công việc, mã công việc, số giờ làm, tên nhân viên phụ trách, mã nhân viên phụ trách.
 - Với nhân viên, thông tin đưa ra bao gồm: tên nhân viên, mã nhân viên, năm sinh, giới tính, phòng ban, số công việc phụ trách và lương
 - Nếu mã nhập vào không trùng khớp với mã nhân viên, mã công việc nào thì thông báo “Không tìm thấy thông tin hiện ra”, dữ liệu trong ô tìm kiếm sẽ bị xóa bỏ.

The screenshot shows the 'Tìm Kiếm' tab active. It has a label 'Mã Nhân Viên hoặc Mã Công Việc cần tìm kiếm' above an input field containing '23010746' and a 'Tìm Kiếm' button. Below the input field is a green box containing the following text:

Tên nhân viên: Lê Hồng Phúc
 Mã nhân viên: 23010746
 Năm sinh: 2005
 Giới tính: Nam
 Phòng ban: Giám Đốc
 Số công việc phụ trách: 1
 Lương: 6400000

Hình 13: Giao diện hiển thị tìm kiếm nhân viên



Hình 14: Giao diện hiển thị tìm kiếm công việc

3. Giao diện người dùng

- Giao diện hệ thống được thiết kế đơn giản, dễ sử dụng với các chức năng:
 - Thêm, sửa, xoá danh sách nhân viên.
 - Thêm, sửa, xoá danh sách công việc
 - Tìm kiếm nhân viên, công việc theo mã định danh.
 - Tính lương nhân viên dựa trên số công việc và số giờ làm.

4. Mã nguồn dự án

- Dự án đã được nhóm commit lên trên Github để thầy cô và các bạn tiện theo dõi: https://github.com/honpphuc/OOP_N02_CompanyManagement.git

5. Kết quả hoạt động

- Ứng dụng hiện tại vẫn còn một số hạn chế và thiếu sót do bản thân nhóm chưa có nhiều kinh nghiệm trong việc phát triển phần mềm. Tuy nhiên, hệ thống đáp ứng các nhu cầu cơ bản và có thể mở rộng trong tương lai.

6. Tài liệu tham khảo

- Github Copilot, Chat GPT
- Khoá học lập trình hướng đối tượng của Trung Tâm Java.
- Khoá học JavaSwing của Trung tâm Java.
- [Bài tập lớp OOP N02 Nhóm 21](#)