**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

---------------------------------------

BÁO CÁO BÀI TẬP LỚN HỌC PHẦN

LẬP TRÌNH JAVA

**ĐỀ TÀI**

**PHẦN MỀM QUẢN LÝ NHÂN SỰ TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

|  |  |  |
| --- | --- | --- |
| **GVHD:** | ***ThS. Vũ Thị Dương*** |  |
| **Sinh viên:** | **Trần Hà Bắc**  **Hoàng Trọng Nghĩa**  **Hoàng Hữu Kim**  **Nguyễn Văn Huy**  **Trần Hữu Vinh** |  |
| **Nhóm:**  **Lớp:** | **03**  **20212IT6019007** | **Khóa: 14** |

**Hà Nội - Năm 2021**

MỤC LỤC

[PHẦN 1. MỞ ĐẦU 4](#_Toc105370248)

[PHẦN 2. KẾT QUẢ NGHIÊN CỨU 5](#_Toc105370249)

[2.1 Giới thiệu: 5](#_Toc105370250)

[2.2 Định nghĩa yêu cầu – Bước 1: 6](#_Toc105370251)

[2.2.1 Khảo sát sơ bộ: 6](#_Toc105370252)

[2.2.2 Tài liệu đặc tả yêu cầu người dùng: 8](#_Toc105370253)

[2.3 Thiết kế hệ thống – Bước 2: 9](#_Toc105370254)

[2.3.1 Mô hình hóa chức năng hệ thống: 9](#_Toc105370255)

[2.3.1.1 Phân tích use case Đăng nhập (Hoàng Hữu Kim): 10](#_Toc105370256)

[2.3.1.2 Phân tích use case Quản lý thông tin nhân sự(Nguyễn Văn Huy):4 12](#_Toc105370257)

[2.3.1.3 Phân tích use case Phân chia công việc (Hoàng Trọng Nghĩa): 15](#_Toc105370258)

[2.3.1.4 Phân tích use case Tuyển dụng(Trần Hữu Vinh): 20](#_Toc105370259)

[2.3.1.5 Phân tích use case Quản lý lương (Trần Hà Bắc): 23](#_Toc105370260)

[2.3.1.6 Phân tích use case Quản lý tài khoản (Hoàng Hữu Kim): 27](#_Toc105370261)

[2.3.2 Mô hình hóa dữ liệu của hệ thống: 29](#_Toc105370262)

[2.3.3 Thiết kế giao diện: 31](#_Toc105370263)

[2.3.4 Thiết kế dữ liệu – Ánh xạ lớp sáng bảng: 52](#_Toc105370264)

[2.5 Cài đặt – Bước 3: 60](#_Toc105370265)

[2.4.1 Hoàng Hữu Kim – Form đăng nhập: 60](#_Toc105370266)

[2.4.2 Trần Hữu Vinh – Form Tuyển Dụng 62](#_Toc105370267)

[2.4.3 Hoàng Trọng Nghĩa – Form Phân Chia Công Việc 69](#_Toc105370268)

[2.4.4 Trần Hà Bắc – Form quản lý lương: 72](#_Toc105370269)

[2.4.5 Nguyễn Văn Huy – Form quản thông tin nhân sự: 75](#_Toc105370270)

[2.4.6 Hoàng Hữu Kim – Form quản lý tài khoản: 79](#_Toc105370271)

[PHẦN 3. KẾT LUẬN VÀ BÀI HỌC KINH NGHIỆM 84](#_Toc105370272)

[3.1 Nội dung đã thực hiện: 84](#_Toc105370273)

[3.2 Hướng phát triển: 84](#_Toc105370274)

PHẦN 1. MỞ ĐẦU

Đi cùng cuộc cách mạng công nghệ 4.0, việc số hóa các dữ liệu đang dần trở nên phố biến hơn không chỉ ở trong các cơ quan mà ngay cả trong các hệ thống của các trường đại học cũng rất cần thiết. Khác với việc quản lý hồ sơ, sổ sách, lương bậc theo phương pháp thủ công công truyền thống, quản lý bởi máy tính hạn chế được những rủi ro hơn việc quản lý theo phương pháp truyền thống, đó là giảm được số lượng người tham gia quản lý, tốc độ nhanh hơn, thông tin được tập trung gọn nhẹ và không cồng kềnh. Việc tính toán bằng máy tính cũng giảm tối thiểu những sai sót.

Vì vậy ứng dụng tin học trong công tác quản lý trường học, cụ thể là đại học Công nghiệp Hà Nội, và đem lại những tiện lợi mới trong công tác quản lý nhân sự tại trường như: việc quản lí hồ sơ, lương bậc các giáo viên, phòng ban, viên chức…. đối với những nhân sự trong trường là rất tốt.

Để hoàn thành được phần mềm này chúng em đã áp dụng mô hình thác nước để xây dựng phần mềm. Bên cạnh đó chúng em áp dụng các kiến thức đã học từ môn lập trình Java để viết chương trình.

Trong thời gian có hạn với sự hướng dẫn tận tình của cô Vũ Thị Dương chúng em đã hoàn thành được phần mềm quản lý nhân sự trường Đại học Công nghiệp Hà Nội. Chúng em rất mong nhận được những lời góp ý, nhận xét của thầy cô để hoàn thiện kiến thức, kỹ năng của bản thân hơn.

# PHẦN 2. KẾT QUẢ NGHIÊN CỨU

## 2.1 Giới thiệu:

- Tên dự án cần xây dựng: Xây dựng phần mềm quản lý nhân sự trường Đại học Công nghiệp Hà Nội.

- Mô hình phát triển phần mềm: mô hình thác nước để xây dựng phần mềm.

|  |  |  |
| --- | --- | --- |
|  | **Mô hình thác nước** | **Mô tả chi tiết** |
| Bước 1 | Định nghĩa yêu cầu: Phân tích và xác định yêu cầu | Thu thập, phân tích và xác định yêu cầu phần mềm:  - Yêu cầu chức năng:  + Yêu cầu nghiệp vụ  + Yêu cầu hệ thống - Yêu cầu phi chức năng  - Miền ứng dụng |
| Bước 2 | Thiết kế hệ thống | - Thiết kế kiến trúc – Architectural design  - Đặc tả trừu tượng – Abstract specification  - Thiết kế giao diện – Interface design  - Thiết kế thành phần – Component design  - Thiết kế cấu trúc dữ liệu – Data structure design  - Thiết kế thuật toán – Algorithm design |
| Bước 3 | Cài đặt và kiểm thử đơn vị | - Cài đặt  - Kiểm thử đơn vị |
| Bước 4 | Tích hợp và kiểm thử hệ thống | - Kiểm thử tích hợp  - Kiểm thử hệ thống |
| Bước 5 | Vận hành và bảo trì | Bàn giao cho khách hàng (Bàn gian, vận hành và bảo trì) |

- Sản phẩm sau khi hoàn thành không phải chỉ là một sản phẩm lý thuyết chứng minh vấn đề hay sản phẩm giả lập mà là một sản phẩm ứng dụng chạy được, hoàn toàn có thể đưa vào thực tế để sử dụng.

- Phần mềm được viết bằng công cụ NetBeans. Bởi NetBeans là một IDE nguồn mở, có thể chạy trên hệ điều hành khác nhau, dễ dàng cài đặt và sử dụng. Đặc biệt NetBeans rất phù hợp để viết Java Swing.

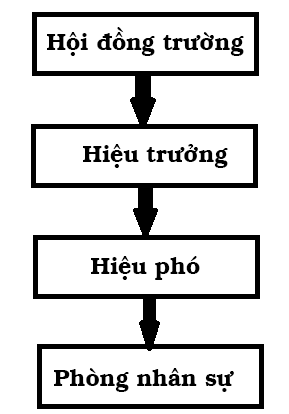
## 2.2 Định nghĩa yêu cầu – Bước 1:

### Khảo sát sơ bộ:

- Giới thiệu bài toán

* Việc quản lý nhân sự trường đại học Công nghiệp Hà Nội là công việc của bộ phận nhân sự. Trong quá trình vận hành nhà trường, hồ sơ, dữ liệu của nhân viên sẽ được ghi lại, lưu trữ. Ngoài ra việc tính lương cũng phải phụ thuộc vào nhiều yêu tố như bậc lương, phụ cấp, ... Qua thời gian, hồ sơ, dữ liệu có thể được nhập thêm, sửa, xoá. Tuy nhiên khi dữ liệu, hồ sơ qua năm tháng ngày càng nhiều hơn, cũng như việc tính lương trở nên phức tập thì làm thủ công có thể xảy ra những sai xót, cũng như việc lưu trữ trở nên khó khăn.
* Từ những yêu cầu thực tế đã cho ta thấy rằng ứng dụng tin học vào việc quản lí nhân sự là rất cần thiết. Phần mềm sẽ được ứng dụng nội bộ trong trường đại học Công nghiệp Hà Nội.

- Sơ đồ tổ chức của đơn vị đặt hàng

****

- Nội dung phỏng vấn:

|  |  |
| --- | --- |
| Câu hỏi | Trả lời |
| Người hỏi: Thanh viên nhóm 3 | Người trả lời |
| Câu 1: Mục đích chính của việc quản lý nhân sự là gì? | Mục đích: Xác định được số lượng nhân sự, hồ sơ nhân sự, tổ chức nhân sự, phòng ban, trả lương. |
| Câu 2: Cô hãy cho biết, hiện nay việc quản lý nhân sự ở trường Đại học Công nghiệp Hà Nội được thực hiện như thế nào? | Hiện nay, ở trường Đại học Công nghiệp Hà Nội thực hiện việc quản lý nhân sự thông qua việc quản lý:   * Quản lý hồ sơ nhân sự. * Quản lý tổ chức nhân sự. * Quản lý trả lương cho nhân sự. |
| Câu 3: Phòng nhân sự thực hiện việc tiếp nhận và quản lý nhân sự như thế nào? | Phòng nhân sự thực hiện việc tiếp nhận và quản lý nhân khẩu thường trú thông qua việc quản lý hồ sơ nhân sự và việc chuyển đi, chuyển đến của nhân sự: lập mới hồ sơ nhân sự, lưu trữ, trả lương. |
| Câu 4: Việc lưu trữ nhân sự hiện nay thực hiện qua hình thức nào, có khó khăn và thuận lợi ra sao? | Hiện nay, việc lưu trữ thông tin vẫn thông qua lưu trữ sổ sách, dẫn tới việc tra cứu thông tin tốn nhiều thời gian, rườm ra. |
| Câu 5: Nếu việc quản lý nhân sự được tin học hóa thì cô có yêu cầu gì đối với hệ thống? | Hệ thống phải đảm bảo đầy đủ các chức năng nêu trên, dễ dàng sử dụng, đảm bảo tính chính xác và không tốn thời gian. |

- Các vấn đề của bài toán cần giải quyết :

* Quản lý các thông tin về nhân sự như : hồ sơ, bậc lương, phòng ban, thông tin tuyển dụng. Cho phép xem, thêm, sửa, xoá.
* Có chức năng phân chia người về các phòng ban.
* Quản lý tài khoản của người quản trị.
* Thực hiện tính lương theo yêu cầu đã đề ra.
* In ra thông tin, bảng lương.

### 2.2.2 Tài liệu đặc tả yêu cầu người dùng:

* Yêu cầu chung

- Tổ chức : Phần mềm được sử dụng bởi phòng nhân sự của trường.

- Dữ liệu : Phần mềm có mọi dữ liệu liên quan đến nhân sự như : hồ sơ, lý lịch, thời gian công tác, trình độ học vấn, các phòng ban, hệ số lương, ….

- Thiết bị : Phần mềm có thể cài đặt trên máy tính để bàn, máy tính xách tay, máy tính bảng sử dụng hệ điều hành window.

* Yêu cầu chức năng

- Đối với bộ phận nhân sự

* Có thể đăng nhập vào hệ thống với vai trò quản trị.
* Có thể Xem, thêm, sửa, xóa các thông tin về nhân sự(giáo viên và cán bộ công nhân viên), phòng ban, chức vụ, tài khoản đăng nhập….
* Có thể thực hiện tính lương cho từng cá nhân, chỉnh sửa công thức tính lương.
* Yêu cầu phi chức năng

- Hệ thống xử lý nhanh gọn, chính xác và thuận tiện.

- Giao diện dễ nhìn, dễ sử dụng.

- Quy trình phát triển phần mềm phù hợp để dễ dàng bảo trì, nâng cấp.

- Độ bảo mật, chính xác cao.

- Cho phép cập nhật xử lí các cơ sở dữ liệu.

- Thời gian xử lý các thao tác truy vấn dữ liệu trễ nhất 5s.

## 2.3 Thiết kế hệ thống – Bước 2:

### 2.3.1 Mô hình hóa chức năng hệ thống:

- Các tác nhân của hệ thống:

* Nhân viên phòng nhân sự - người quản lý(người dùng): Đăng nhập vào hệ thống và có các chức năng: quản lý hồ sơ, quản lý phòng ban, quản lý chức vụ, quản lý lương, quản lý tài khoản đăng nhập.
* Đăng nhập: Use case cho phép người quản lý đăng nhập vào hệ thống, ngoài ra còn có thể xem thông tin chi tiết tài khoản, sửa thông tin cá nhân và đổi mật khẩu…
* Tuyển Dụng: Use case cho phép người dùng quản lý các hồ sợ tuyển dụng cũng như duyệt những hồ sơ tuyển dụng.
* Phân chia công việc: Use case cho phép người dùng quản lý các phòng ban, chia người về các phòng ban,…
* Quản lý thông tin nhân sự: Use case cho phép người dùng quản lý thông tin xem, thêm, sửa, xóa nhân sự.
* Quản lý lương: Use case cho phép người dùng quản lý lương.

- Biểu đồ use case:



Hình 2.1:Biểu đồ quan hệ giữa các use case

#### 2.3.1.1 Phân tích use case Đăng nhập (Hoàng Hữu Kim):

* **Mô tả vắn tắt:** Use case này cho phép người quản lý(người dùng) truy cập vào hệ thống để thực hiện các chức năng quản lý.
* **Luồng sự kiện:**
* **Luồng cơ bản:**

1. Use case bắt đầu khi người quản lý bắt đầu vào phần mềm. Hệ thống sẽ hiển thị màn hình đăng nhập để yêu cầu người quản trị nhập tên tài khoản và mật khẩu.

2. Khi người quản lý kích nút "Đăng nhập" hệ thống sẽ lấy thông tin nhập vào so sánh với dữ liệu từ tệp account.txt, nếu hợp lệ sẽ hiển thị sang màn hình các chức năng quản lý và tên người quản lý.

3. Sau khi đăng nhập nếu ấn vào nút log out, sẽ đăng xuất tài khoản trở lại màn hình đăng nhập.

Use case kết thúc.

* **Luồng rẽ nhánh:**

1. Tại bước 2 người quản lý nhập thiếu tài khoản. Hệ thống thông báo nhập tên tài khoản.

2. Tại bước 2 người quản lý nhập thiếu mật khẩu. Hệ thống thông báo nhập mật khẩu.

3. Tại bước 2 người quản lý nhập sai tài khoản hoặc mật khẩu. Hệ thống thông báo sai thông tin đăng nhập.

4. Tại bước 2 người quản lý nhập sai mật khẩu. Hệ thống thông báo sai mật khẩu.

* **Các yêu cầu đặc biệt:** Không có.
* **Tiền điều kiện:** Không có.
* **Hậu điều kiện:** Không có.
* **Điểm mở rộng:** Không có.

#### 2.3.1.2 Phân tích use case Quản lý thông tin nhân sự(Nguyễn Văn Huy):4

* **Mô tả vắn tắt:** Use case này cho phép người quản lý(người dùng) truy cập vào hệ thống để thực hiện các chức năng xem, thêm, sửa, xóa các hồ sơ nhân sự.
* **Luồng sự kiện:**
* **Luồng cơ bản:**

1. Use case bắt đầu khi người quản lý kích vào nút “Quản lý thông tin nhân sự” trên menu quản trị. Hệ thống sẽ lấy thông tin chi tiết của các hồ sơ nhân sự (mã nhân sự, họ tên nhân sự, ngày sinh, giới tính, số điện thoại, quê quán, dân tộc, trình độ, chuyên ngành, chức vụ, Số CCCD, Số điện thoại, loại nhân sự, ngày tham gia, hạn hợp đồng) từ file NhanSu.txt và hiển thị lên màn hình danh sách thông tin các nhân sự.
2. Thêm hồ sơ nhân sự:
3. Người quản lý kích nút “Thêm”: Hệ thống hiển thị màn hình yêu cầu nhập thông tin chi tiết cho hồ sơ nhân sự(mã nhân sự, họ tên nhân sự, số CCND, ngày sinh,quê quán, giới tính, số điện thoại, dân tộc, trình độ, chuyên ngành, ngày tham gia, loại nhân sự, hạn hợp đồng).
4. Người quản lý nhập thông tin cho một hồ sơ mới. Nếu người quản lý kích nút “Thêm” thì hệ thống sẽ tạo một bản ghi mới vào tệp NhanSu.txtvà hiển thị danh sách các hồ sơ được cập nhật.
5. Nếu người quản lý kích nút “Hủy” thì hệ thống sẽ không có bản ghi mới nào được thêm vào file NhanSu.txt và hiển thị lại danh sách ban đầu.
6. Xóa hồ sơ nhân sự:
7. Người quản lý có thể chọn một hoặc nhiều hồ sơ trong danh sách rồi kích nút “Xóa”. Màn hình sẽ hiển thị một thông báo xác nhận xóa
8. Nếu người quản lý kích nút “Yes” thì hệ thống sẽ tiếp hành xóa các bản ghi đó trong file NhanSu.txt, đưa ra thông báo xác nhận đã xóa thành công và hiển thị lại danh sách sau khi xóa.
9. Nếu người quản lý kích nút “No” thì hệ thống sẽ không xóa bản ghi nào đã chọn.
10. Sửa hồ sơ nhân sự:
11. Người quản lý có thể chọn một hoặc nhiều hồ sơ trong danh sách rồi kích nút “Sửa”, hệ thống sẽ hiển thị màn hình chi tiết về hồ sơ gồm: ảnh, mã nhân sự, họ tên nhân sự, số CCND, ngày sinh, giới tính, số điện thoại, quê quán, dân tộc, trình độ, chuyên ngành, phòng ban, chức vụ, chính trị, chuyên ngành, thời gian công tác, loại nhân sự, hạn hợp đồng, luong. Tuy nhiên người quản lý không thể sửa được mã nhân sự.
12. Người quản lý sửa các thông tin của hồ sơ. Nếu kích nút “Lưu” thì hệ thống sẽ cập nhật thông tin hồ sơ vào file NhanSu.txt, ghi lại tên người quản lý vừa sửa hồ sơ và hiển thị lại danh sách hồ sơ đã cập nhật.
13. Nếu người quản lý kích nút “Hủy” thì hệ thống sẽ hiển thị lại danh sách như cũ.
14. Xem hồ sơ nhân sự:
    1. Người quản lý có thể chọn một hoặc nhiều hồ sơ trong danh sách rồi kích nút “Xem”, hệ thống sẽ hiển thị màn hình chi tiết về hồ sơ gồm: ảnh, mã nhân sự, họ tên nhân sự, số CCND, ngày sinh, giới tính, số điện thoại, quê quán, dân tộc, trình độ, chuyên ngành, phòng ban, chức vụ, chính trị, chuyên ngành, thời gian công tác, loại nhân sự, hạn hợp đồng,lương.

Use case kết thúc.

* **Luồng rẽ nhánh:**

1.Tại bước 3 trong luồng cơ bản khi người quản lý nhập sai kiểu dữ liệu hoặc để trống không nhập (ngoại trừ đoàn thể và chính trị được để trống), sau đó kích nút “Thêm” thì hệ thống sẽ hiển thị thông báo lỗi miền đó chưa nhập hoặc nhập kiểu dữ liệu sai và đặt con trỏ vào trường đang nhập lỗi.

2.. Tại bước 3 trong luồng cơ bản khi người quản lý nhập trùng mã nhân sự đã có trước đó thì hệ thống sẽ hiển thị thông báo “Mã nhân sự này đã tồn tại”.

3. Tại bước 4 trong luồng cơ bản khi người quản lý nhập sai kiểu dữ liệu hoặc để trống không nhập, sau đó kích nút “Lưu” thì hệ thống sẽ hiển thị thông báo lỗi miền đó chưa nhập hoặc nhập kiểu dữ liệu sai và đặt con trỏ vào trường đang nhập lỗi.

4. Tại bước 3, 4, 5 nếu chưa chọn nhân sự nào thì hệ thống sẽ báo lỗi chưa chọn nhân sự nào.

* **Các yêu cầu đặc biệt:** Cần kiểm soát quyền thực hiện use case này để đảm bảo tính an toàn và bảo mật.
* **Tiền điều kiện:** Người quản lý cần phải đăng nhập vào tài khoản quản trị.
* **Hậu điều kiện:** Sau khi use case kết thúc thành công thì các thông tin về nhân sự cần được cập nhật vào file NhanSu.txt.
* **Điểm mở rộng:** Không có.

#### 2.3.1.3 Phân tích use case Phân chia công việc (Hoàng Trọng Nghĩa):

* **Mô tả vắn tắt:** Use case này cho phép người quản lý (người dùng) truy cập vào hệ thống để thực hiện các chức năng xem, thêm, sửa, xóa các phòng ban và thêm nhân sự vào các phòng ban
* **Luồng sự kiện:**
* **Luồng cơ bản:**

1. Use case này bắt đầu khi người quản lý kích vào nút “Phân chia công việc” trên menu quản trị, hệ thống sẽ lấy thông tin bao gồm: STT, Mã phòng, Tên phòng và số ban từ file phong.txt và ban.txt để hiển thị lên màn hình danh sách các phòng hiện có.
2. Xem tất cả:
3. Người quản lý chọn vào nút “ Xem tất cả” hiển thị trên màn hình, hệ thống sẽ lấy dữ liệu của tất cả các phòng ban bao gồm các thuộc tính: Phòng, Ban, Mã nhân sự, Tên nhân sự, Chức vụ hiện có từ file phong.txt và ban.txt để hiển thị lên màn hình.
4. Thêm phòng:
5. Người quản lý kích vào nút “Thêm” hệ thống sẽ cho phép người quản trị nhập mã phòng ban mới, tên phòng ban mới.
6. Nếu người quản lý kích nút “Lưu” thì hệ thống sẽ hiển thị thông báo “Thêm thành công”, thêm 1 bản ghi mới vào file phong.txt.
7. Nếu người quản lý kích nút “Hủy” thì hệ thống sẽ không có thay đổi nào xảy ra
8. Sửa phòng:
9. Người quản lý chọn tên phòng ban muốn sửa ở combo box hệ thống sẽ hiển thị thông tin của phòng ban đó và cho phép sửa các thông tin: mã phòng ban, tên phòng ban.
10. Người quản lý kích nút “Sửa” hệ thống sẽ cập nhật thông tin phòng ban trong bảng phong.txt, hiện thông báo “Sửa thành công”.
11. Nếu người quản lý kích nút “Hủy” thì hệ thống sẽ không có thay đổi nào xảy ra.
12. Xóa phòng :
13. Người quản lý chọn tên phòng ban muốn xóa ở combo box và kích nút “Xóa” hệ thống sẽ hiện thông báo xác nhận muốn xóa hay không.
14. Nếu người quản lý kích nút “Yes”. Hệ thống sẽ xóa phòng khỏi file phong.txt và cập nhật lại danh sách phòng ban trong combo box.
15. Nếu người quản lý kích nút “No” thì hệ thống không có thay đổi gì.
16. Xem phòng:
    1. Người quản lý chọn tên phòng muốn xem ban ở combo box và kích nút “Xem phòng” hệ thống sẽ hiển thị danh sách ban ở trong phòng đó bao gồm: Mã ban, Tên ban, Số người trong ban, Thuộc phòng. Dữ liệu được lấy từ file ban.txt.
    2. Thêm phòng:
    3. Người quản lý kích vào nút “Thêm” hệ thống sẽ cho phép người quản trị nhập mã ban mới, tên ban mới.
    4. Nếu người quản lý kích nút “Lưu” thì hệ thống sẽ hiển thị thông báo “Thêm thành công”, thêm 1 bản ghi mới vào file ban.txt.
    5. Nếu người quản lý kích nút “Hủy” thì hệ thống sẽ không có thay đổi nào xảy ra
    6. . Sửa phòng:
    7. Người quản lý chọn tên phòng ban muốn sửa ở combo box hệ thống sẽ hiển thị thông tin của phòng ban đó và cho phép sửa các thông tin: mã ban, tên ban.
    8. Người quản lý kích nút “Sửa” hệ thống sẽ cập nhật thông tin phòng ban trong bảng ban.txt, hiện thông báo “Sửa thành công”.
    9. Nếu người quản lý kích nút “Hủy” thì hệ thống sẽ không có thay đổi nào xảy ra.
    10. . Xóa phòng :
    11. Người quản lý chọn tên phòng ban muốn xóa ở combo box và kích nút “Xóa” hệ thống sẽ hiện thông báo xác nhận muốn xóa hay không.
    12. Nếu người quản lý kích nút “Yes”. Hệ thống sẽ xóa phòng khỏi file ban.txt và cập nhật lại danh sách phòng ban trong combo box.
    13. Nếu người quản lý kích nút “No” thì hệ thống không có thay đổi gì.

6.4. Xem chi tiết ban:

a. Người quản lý chọn tên phòng muốn xem ban ở combo box và kích nút “Xem ban” hệ thống sẽ hiển thị danh sách ban ở trong phòng đó bao gồm: Mã ban, Tên ban, Số người trong ban, Thuộc phòng. Dữ liệu được lấy từ file ban.txt.

6.4.1. Thêm người vào ban:

a. Người quản lý chọn nút “Thêm nhân sự vào ban” hệ thống sẽ hiển thị bảng thêm người vào ban, người dùng chọn nhân sự và chọn các phòng ban đã có để thêm người vào trong phòng ban và chỉ định các chức vụ bao gồm: Trưởng phòng, phó phòng, trưởng ban, phó ban, Giảng Viên trong phòng ban đó cho nhân sự cần sắp xếp.

b. Nếu người dùng ấn “Thêm nhân sự vào ban” hệ thống sẽ thêm dữ liệu vào file ban\_nhansu.txt

c. Nếu người dùng ấn “Thoát” sẽ không có nhân sự nào được thêm và trở về màn hình trước đó.

6.4.2. Xóa người khỏi ban:

a. Người quản lý chọn nhân sự đang có tên trong danh sách hiển thị sau đó ấn “Xóa”. Hệ thống sẽ hỏi có xác nhận xóa không.

b. Nếu ấn “Yes”, dữ liệu sẽ bị xóa khỏi dữ liệu trong file ban\_nhansu.txt và không hiển thị lại lên màn hình.

c. Nếu ấn “No”, sẽ không có gì xảy ra và trở về màn hình trước đó.

6.4.3. Sửa thông tin người trong ban:

a. Người quản lý chọn nhân sự đang có tên trong danh sách hiển thị sau đó ấn “Sửa”. Hệ thống sẽ hiển thị bảng sửa cho người quản lý sửa.

b. Người quản lý được phép sửa vị trí phòng ban và chức vụ của nhân sự hiện có. Nếu ấn “Sửa” Thì dữ liệu sẽ được cập nhật lại vào file ban\_nhansu.txt.

c. Nếu ấn “Thoát”, sẽ không có gì xảy ra và trở về màn hình trước đó.

Use case kết thúc.

* **Luồng rẽ nhánh:**

1.Tại bước 3, 6.1 và 6.4.1 trong luồng cơ bản khi người quản lý nhập sai kiểu dữ liệu hoặc để trống không nhập, sau đó kích nút “Lưu” thì hệ thống sẽ hiển thị thông báo lỗi và đặt con trỏ chuột vào trường đang nhập lỗi.

2. Tại bước 3, 6.1 trong luồng cơ bản khi người quản lý nhập trùng với mã phòng ban đã có, sau đó kích nút “Lưu” thì hệ thống sẽ hiển thị thông báo “Mã phòng ban này đã tồn tại”.

3. Tại bước 4, 6.2 trong luồng cơ bản khi người quản lý nhập sai kiểu dữ liệu hoặc để trống không nhập, sau đó kích nút “Lưu” thì hệ thống sẽ hiển thị thông báo “Thông tin sửa chưa hợp lệ”.

4. Tại bước 4, 6.2, 6.4.2 trong luồng cơ bản khi người quản lý chưa chọn thông tin nào trong bảng dữ liệu thì hẹ thống sẽ hiện “Chưa chọn đối tượng nào”.

5. Tại bước 6, 6.4 nếu chưa chọn phòng hoặc ban nào để xem thì hệ thống sẽ hiện “Chưa chọn phòng/ban”.

* **Các yêu cầu đặc biệt:** Cần kiểm soát quyền thực hiện use case này để đảm bảo tính an toàn và bảo mật.
* **Tiền điều kiện:** Người quản lý cần phải đăng nhập vào tài khoản quản trị.
* **Hậu điều kiện:** Sau khi use case kết thúc thành công thì các thông tin về phòng ban và nhân sự cần được cập nhật vào file phong.txt, ban.txt và ban\_nhansu.txt.
* **Điểm mở rộng:** Không có.

#### 2.3.1.4 Phân tích use case Tuyển dụng(Trần Hữu Vinh):

* **Mô tả vắn tắt:** Use case này cho phép người quản lý (người dùng) truy cập vào hệ thống để thực hiện các chức năng xem, thêm, sửa, xóa các đơn tuyển dụng.
* **Luồng sự kiện:**
* **Luồng cơ bản:**

1. Use case này bắt đầu khi người quản lý kích vào nút “Tuyển dụng” trên menu quản trị, hệ thống sẽ lấy thông tin chi tiết từ file TuyenDung.txt bao gồm: STT, Mã TD, Họ tên, Vị trí và Trạng Thái để hiển thị lên màn hình.
2. Xem đơn tuyển dụng:
3. Người quản lý chọn đơn muốn xem ở combo box. Hệ thống sẽ lấy thông tin của đơn đó từ file TuyenDung.txt và hiển thị lên màn hình.
4. Thêm đơn tuyển dụng:
5. Người quản lý kích vào nút “Thêm” hệ thống sẽ cho phép người quản trị nhập mã tuyển dụng, họ tên người tuyển dụng, số CCCD, Ngày Sinh, Quê quán, Giưới tính, Dân tộc, Số điện thoại, Chức vụ, Trình Độ, Chuyên ngành, Ngày Nộp Đơn, Tình Trạng. Chọn chức vụ bằng cách ấn vào mũi tên cạnh ô chức vụ.
6. Nếu người quản lý kích nút “Thêm” thì hệ thống sẽ hiển thị thông báo “Thêm thành công”, thêm 1 bản ghi mới trong file TuyenDung.txt và hiển thị ra màn hình.
7. Nếu người quản lý kích nút “Hủy” thì hệ thống sẽ không có thay đổi nào xảy ra.
8. Sửa đơn tuyển dụng:
   1. Người quản lý kích vào nút “Thêm” hệ thống sẽ cho phép người quản trị nhập mã tuyển dụng, họ tên người tuyển dụng, số CCCD, Ngày Sinh, Quê quán, Giưới tính, Dân tộc, Số điện thoại, Chức vụ, Trình Độ, Chuyên ngành, Ngày Nộp Đơn, Tình Trạng. Chọn chức vụ bằng cách ấn vào mũi tên cạnh ô chức vụ.
   2. Người quản lý kích nút “Sửa” hệ thống sẽ cập nhật thông tin phòng ban trong file TuyenDung.txt, hiện thông báo “Sửa thành công”, ghi lại thông tin tuyển dụng so với ban đầu và hiển thị thông tin phòng ban sau khi sửa lên màn hình.
9. Xóa đơn tuyển dụng
10. Người quản lý chọn tên chức vụ muốn xóa ở combo box và kích nút “Xóa” hệ thống sẽ hiện thông báo xác nhận muốn xóa hay không.
11. Nếu người quản lý kích nút “Yes”. Hệ thống sẽ đơn đó khỏi file TuyenDung.txt và cập nhật lại danh sách chức vụ trong combo box.
12. Nếu người quản lý kích nút “No” thì hệ thống không có thay đổi gì.

Use case kết thúc.

* **Luồng rẽ nhánh:**

1.Tại bước 3 trong luồng cơ bản khi người quản lý nhập sai kiểu dữ liệu hoặc để trống không nhập, sau đó kích nút “Lưu” thì hệ thống sẽ hiển thị thông báo lỗi và đặt con trỏ chuột vào trường đang nhập lỗi.

2. Tại bước 3 trong luồng cơ bản khi người quản lý nhập trùng với mã chức vụ đã có, sau đó kích nút “Lưu” thì hệ thống sẽ hiển thị thông báo “Mã tuyển dụng này đã tồn tại”.

3. Tại bước 4 trong luồng cơ bản khi người quản lý nhập sai kiểu dữ liệu hoặc để trống không nhập, sau đó kích nút “Lưu” thì hệ thống sẽ hiển thị thông báo “Thông tin sửa chưa hợp lệ”.

4. Ở bước 2, 4, 5 nếu chưa chọn đơn nào hệ thống sẽ báo chưa chọn đơn tuyển dụng.

* **Các yêu cầu đặc biệt:** Cần kiểm soát quyền thực hiện use case này để đảm bảo tính an toàn và bảo mật.
* **Tiền điều kiện:** Người quản lý cần phải đăng nhập vào tài khoản quản trị.
* **Hậu điều kiện:** Sau khi use case kết thúc thành công thì các thông tin về chức vụ và nhân sự cần được cập nhật vào file TuyenDung.txt.
* **Điểm mở rộng:** Không có.

#### 2.3.1.5 Phân tích use case Quản lý lương (Trần Hà Bắc)

* **Mô tả vắn tắt:** Use case này cho phép người quản lý(người dùng) truy cập vào hệ thống để thực hiện các chức năng chỉnh sửa thông số tính lương, thêm loại viên chức cho nhân sự, sửa loại viên chức, xem loại viên chức của nhân sự, xóa lương cho nhân sự
* **Luồng sự kiện:**
* **Luồng cơ bản:**

1. Use case này bắt đầu khi người quản lý kích vào nút “Quản Lý Lương” trên menu quản trị, hệ thống sẽ hiển thị lên màn hình một danh sách có các thông tin: STT, Mã NS, Họ Tên, Loại Viên Chức, Lương(Nghìn VND).
2. Sửa mức lương cho nhân sự:
3. Người quản lý chọn vào một nhân sự sau đó ấn chọn nút “Sửa”. Màn hình hiện ra sẽ cho người dùng chọn lại loại nhân viên.
4. Nếu ấn “Sửa” thì hệ thóng sẽ cập nhật lại vào file luong\_nhansu.txt.
5. Nếu người quản lý kích nút “Hủy” thì hệ thống sẽ không có thay đổi gì xảy ra.
6. Thêm nhân sự vào bảng tính lương:
7. Người quản lý kích vào nút “Thêm” hệ thống sẽ hiển thị màn hình các thành phần để thêm bậc lương cho nhân sự gồm 1 dòng chọn nhân sự và 1 dòng chọn loại viên chức để t lương.
8. Nếu ấn “Xác nhận” thì hệ thống sẽ cập nhật thêm thông tin vào file luong\_nhansu.txt.
9. Nếu người quản lý kích nút “Thoát” thì hệ thống sẽ không có thay đổi gì xảy ra và quay lại màn hình trước đó.
10. Xóa lương nhân sự
    1. Người quản lý chọn 1 nhân sự sau đó ấn vào nút "Xóa” hệ thống sẽ hỏi có chắc chắn xóa hay không.
    2. Nếu ấn “Yes” thì hệ thống sẽ xóa nhân sự đó và cập nhật lại thông tin vào file luong\_nhansu.txt.
    3. Nếu ấn “No” thì không có gì xảy ra và quay về màn hình trước đó.
11. Xem loại nhân sự:
12. Người quản lý kích vào nút “Xem loại nhân sự” hệ thống sẽ hiển thị màn hình các thành phần để tính lương cho từng loại NS gồm: Hệ Số Lương, Lương Cơ Sở, Phụ Cấp và Bảo Hiểm.

5.1. Thêm loại nhân sự

a. Người quản lý ấn vào nút “Thêm” xuất hiện bảng thêm để thêm loại nhân sự mới gồm: Loại Nhân Sự, Mức Lương Cơ Sở, Hệ Số Lương, Phụ Cấp, Bảo Hiểm.

b. Nếu ấn “Lưu” hệ thống sẽ cập nhật lại dữ liệu và lưu vào file luong.txt

c. Nếu ấn “Trở về” sẽ không có gì xảy ra và trở về màn hình trước đó.

5.2. Sửa loại nhân sự

a. Người quản lý chọn vào loại nhân sự cần sửa và ấn vào nút “Sửa” xuất hiện bảng thêm để thêm loại nhân sự mới gồm: Mức Lương Cơ Sở, Hệ Số Lương, Phụ Cấp, Bảo Hiểm.

b. Nếu ấn “Lưu” hệ thống sẽ cập nhật lại dữ liệu và lưu vào file luong.txt

c. Nếu ấn “Trở về” sẽ không có gì xảy ra và trở về màn hình trước đó.

5.3. Xóa loại nhân sự

a. Người quản lý chọn vào loại nhân sự cần sửa và ấn vào nút “Xóa” hệ thống sẽ hỏi có chắc chắn muốn xóa không.

b. Nếu ấn “Yes” hệ thống sẽ xóa loại nhân sự ,cập nhật lại dữ liệu và lưu vào file luong.txt

c. Nếu ấn “Trở về” sẽ không có gì xảy ra và trở về màn hình trước đó.

Use case kết thúc.

* **Luồng rẽ nhánh:**

1. Tại bước 2 trong luồng cơ bản khi người quản lý nhập sai kiểu dữ liệu hoặc để trống không nhập, sau đó kích nút “Lưu” thì hệ thống sẽ hiển thị thông báo lỗi và đặt con trỏ chuột vào trường đang nhập lỗi.

3. Tại bước 3 trong luồng cơ bản khi người quản lý nhập sai kiểu dữ liệu hoặc để trống không nhập, sau đó kích nút “Xác nhận” thì hệ thống sẽ hiển thị thông báo lỗi và đặt con trỏ chuột vào trường đang nhập lỗi.

4. Tại bước 2, 4, 5.2, 5.3 nếu chưa chọn nhân sự nào sẽ hiển thị là bạn chưa chọn nhân sự nào.

* **Các yêu cầu đặc biệt:** Cần kiểm soát quyền thực hiện use case này để đảm bảo tính an toàn và bảo mật.
* **Tiền điều kiện:** Người quản lý cần phải đăng nhập vào tài khoản quản trị.
* **Hậu điều kiện:** Sau khi use case kết thúc thành công thì các thông tin về thông số tính lương cần được cập nhật vào file lương.txt và luong\_nhansu.txt
* **Điểm mở rộng:** Không có.

#### 2.3.1.6 Phân tích use case Quản lý tài khoản (Hoàng Hữu Kim):

* **Mô tả vắn tắt:**
* Use case này cho phép người quản lý(người dùng) với tư cách là người quản lý chính truy cập vào hệ thống để thực hiện các chức năng đổi mật khẩu, đăng xuất và quản lý tất cả các tài khoản có trong hệ thống(thêm tài khoản, sửa tài khoản, xóa tài khoản).
* **Luồng sự kiện:**
* **Luồng cơ bản:**

1. Use case này bắt đầu khi người quản lý kích vào icon “Tài khoản” trên menu quản trị, hệ thống sẽ hiển thị lên màn hình tài khoản đang đăng nhập(tên tài khoản, chức vụ và họ tên) từ file account.txt.
2. Đổi mật khẩu(dùng bởi tất cả người quản lý):
3. Người quản lý nhập mật khẩu cũ và mật khẩu mới rồi ấn nút “Đổi mật khẩu”, hệ thống sẽ cập nhật thông tin tài khoản trong file account.txt, hiển thị thông báo “Đổi mật khẩu thành công” và xóa mật khẩu cũ và mật khẩu mới trên màn hình.
4. Đăng xuất(dùng bởi tất cả người quản lý):
5. Người quản lý kích nút “Đăng xuất” hệ thống sẽ thoát khỏi phần mềm và hiển thị màn hình đăng nhập.
6. Xem danh sách tài khoản
   1. Người dùng ấn vào “Xem Danh Sách Tài Khoản Nhân Viên”, hệ thống sẽ lấy dữ liệu từ file account.txt sau đó hiển thị danh sách tài khoản gồm “Tên tài khoản”, “Chức vụ” và “Họ Tên”.
   2. Thêm tài khoản(dùng bởi người quản lý chính):
7. Người quản lý chính kích nút “Thêm”, hệ thống hiển thị màn hình thêm tài khoản gồm “Tên tài khoản”, “Chức vụ”, “Họ Tên” và “Mật khẩu”
8. Người quản lý kích nút “Lưu”, hệ thống lưu lại thông tin tài khoản vào file account.txt.
9. Nếu người dùng ấn nút “Trở về” hệ thống sẽ tắt màn hình hiện tại và chuyển về màn hình trước đó.
   1. Sửa tài khoản(dùng bởi người quản lý chính):
10. Người quản lý chính chọn tài khoản muốn sửa trong danh sách tài khoản, hệ thống sẽ cho phép sửa họ và tên, tên tài khoản và mật khẩu của người quản lý được chọn.
11. Người quản lý chính kích nút “Lưu”, hệ thống cập nhật thông tin tài khoản trong file account.txt, hiện thông báo “Sửa thông tin thành công” và hiển thị lại danh sách tài khoản cập nhật.
    1. Xóa tài khoản(dùng bởi người quản lý chính):
12. Người quản lý chính chọn tài khoản muốn xóa trong danh sách tài khoản rồi kích nút “Xóa tài khoản”, hệ thống sẽ xóa tài khoản trong file account.txt, hiện thông báo “Xóa thành công” và hiển thị lại danh sách tài khoản cập nhật.

Use case kết thúc.

* **Luồng rẽ nhánh:**

1.Tại bước 2 trong luồng cơ bản khi người quản lý để trống không nhập mật khẩu cũ hoặc mật khẩu mới, sau đó kích nút “Đổi mật khẩu” thì hệ thống sẽ hiển thị thông báo các trường đó không được để trống.

2. Tại bước 4 trong luồng cơ bản khi người quản lý chính để trống không nhập, sau đó kích nút “Lưu” thì hệ thống sẽ thông báo không được để trống và đưa con trỏ chuột vào trường đang lỗi nhập.

3. Tại bước 4 trong luồng cơ bản nếu đã tồn tại tên tài khoản định thêm thì khi người quản lý chính kích nút “Lưu” hệ thống sẽ đưa ra thông báo “Tài khoản này đã tồn tại!”.

4. Tại bước 4.2 trong luồng cơ bản khi người quản lý để trống không nhập họ tên và ảnh, sau đó kích nút “Sửa tài khoản” thì hệ thống sẽ hiển thị thông báo các trường đó không được để trống.

* **Các yêu cầu đặc biệt:** Cần kiểm soát quyền thực hiện use case này để đảm bảo tính an toàn và bảo mật.
* **Tiền điều kiện:** Người quản lý cần phải đăng nhập vào tài khoản quản trị.
* **Hậu điều kiện:** Sau khi use case kết thúc thành công thì các thông tin về tài khoản đăng nhập cần được cập nhật vào file account.txt.
* **Điểm mở rộng:** Không có.

### 2.3.2 Mô hình hóa dữ liệu của hệ thống:

* Dựa vào use case quản lý tài khoản ta xác định được:
* Lớp ACCOUNT: Lưu trữ thông tin của tài khoản đăng nhập vào hệ thống: username, password, ten, chucVu
* Dựa vào use case quản lý thông tin nhân sự ta xác định được:
* Lớp nhân sự(NhanSu): Lưu trữ thông tin của nhân sự: mã nhân sự, họ tên, ngày sinh, giới tính, quê quán, căn cước, số điện thoại, dân tộc, chức vụ, trình độ, chuyên ngành, loại NS, ngày tham gia, hạn hợp đồng.
* Dựa vào use case phân chia công việc ta xác định được:
* Lớp phòng: Lưu trữ thông tin của phòng : mã phòng , tên phòng.
* Lớp ban: Lưu trữ thông tin của ban: mã ban , tên ban.
* Dựa vào use case tuyển dụng ta xác định được:
* Lớp tuyển dụng: Mã tuyển dụng, họ tên, ngày sinh, giới tính, quê quán, căn cước, số điện thoại, dân tộc, vị trí tuyển dụng, trình độ, chuyên ngành, ngày nộp đơn, trạng thái.
* Dựa vào use case quản lý lương ta xác định được:
* Lớp bậc lương(BacLuong): Lưu trữ thông tin của bậc lương: loại nhân sự, mã lương cơ sở, hệ số lương, phụ cấp, bảo hiểm.
* Dựa vào các use case quản lý thông tin nhân sự, quản lý phòng ban, quản lý lương ta xác định được:
* Lớp BAN\_NHANSU: Lưu trữ thông tin: mã ban, mã nhân sự, chức vụ
* Lớp LUONG\_NHANSU: Lưu trữ thông tin: mã nhân sự, loại nhân sự



Hinh 2.2: Các lớp dữ liệu

### 2.3.3 Thiết kế giao diện:

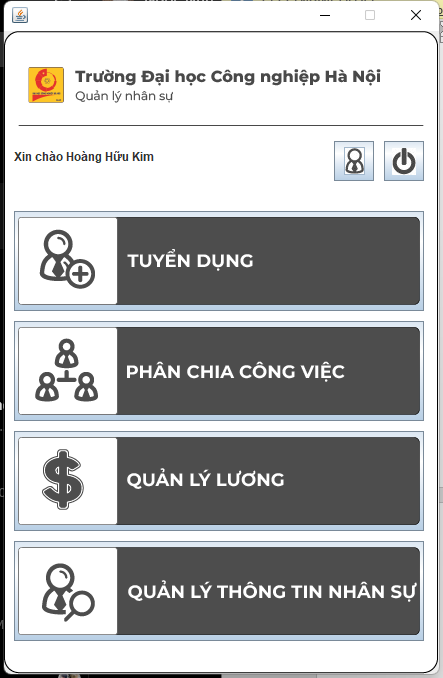
- Dựa vào phần phân tích use case đăng nhập xây dựng nên:

* “Giao diện đăng nhập” gồm các trường thông tin: tên tài khoản, mật khẩu và nút “Đăng nhập”.



Hình 2.3: Giao diện đăng nhập

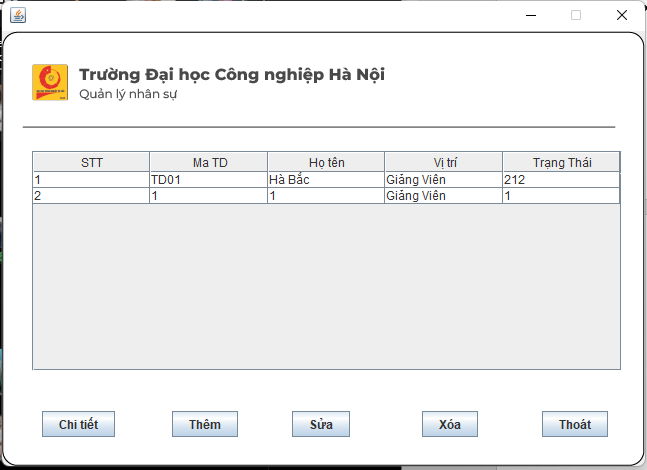
* “Giao diện các chức năng quản lý” gồm các nút bấm đi đến các chức năng quản lý trong chương trình.



Hình 2.4: Giao diện các chức năng quản lý

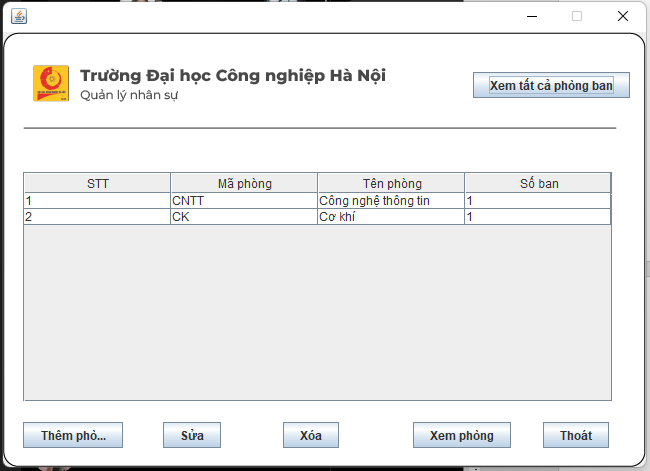
- Dựa vào phần phân tích use case quản lý hồ sơ xây dựng nên:

* “Giao diện tuyển dụng” :



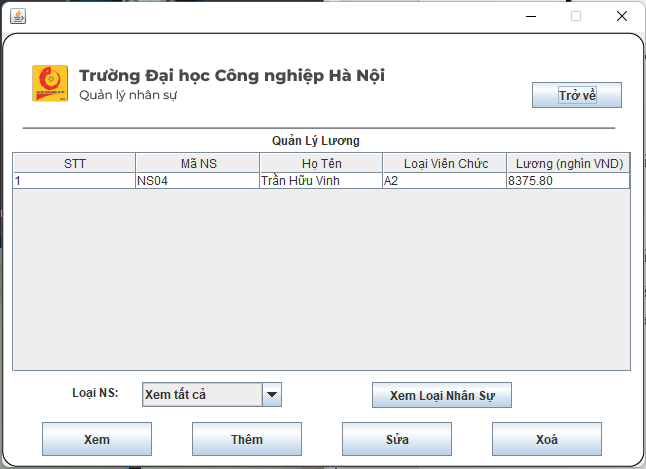
Hình 2.5: Giao diện tuyển dụng

* Giao diện “Phân chia công việc”:



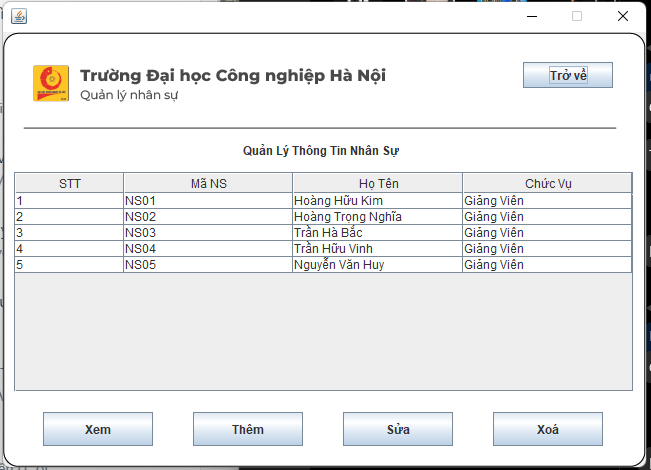
Hình 2.6: Giao diện phân chia công việc

* “Giao diện Quản lý lương”



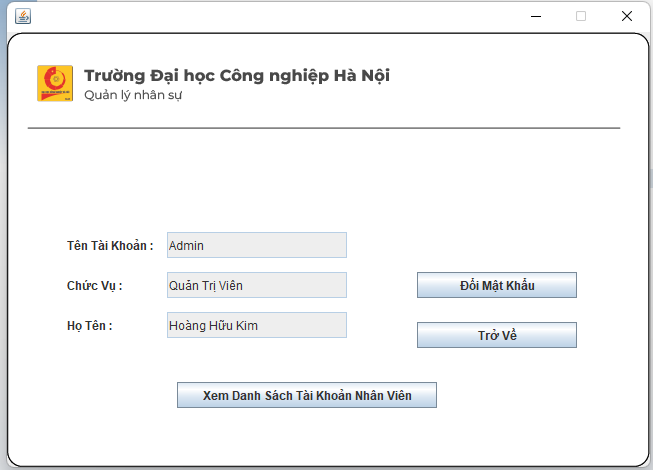
Hình 2.7: Giao diện quản lý lương

* “Giao diện quản lý thông tin nhân sự”



Hình 2.8: Giao diện quản lý thông tin nhân sự

* “Giao diện Xem thông tin chi tiết tài khoản”:



Hình 2.9: Giao diện thông tin tài khoản

### 2.3.4 Thiết kế dữ liệu – Ánh xạ lớp sáng bảng:

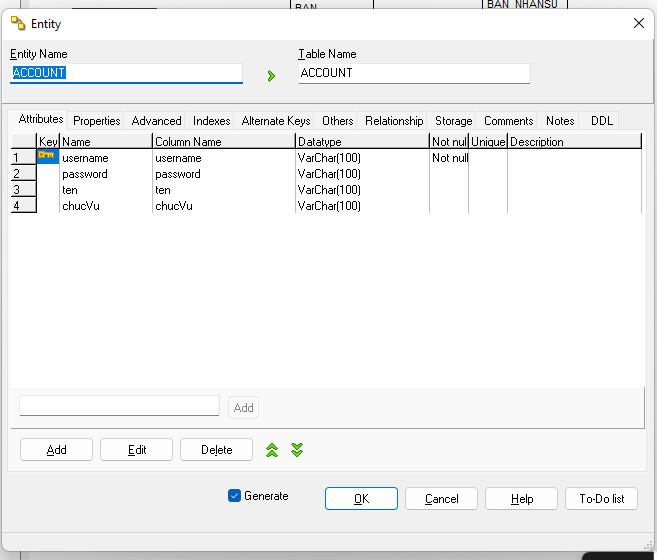
Ánh xạ các lớp trong biểu đồ thực thể theo nguyên tắc:

- Mỗi lớp thực thể sẽ được ánh xạ thành một bảng cơ sở dữ liệu.

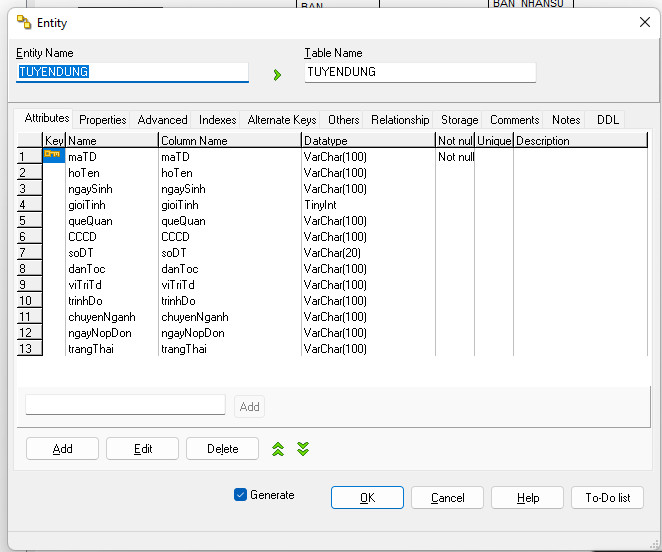
- Xử lý kiểu liên kết **:**

* Quy tắc 1 : với kiểu liên kết 1:1
* Cách 1: Chuyển khóa chính của Bảng này sang làm khóa ngoài của Bảng kia hoặc ngược lại.
* Cách 2: Nhập 2 kiểu thực thể và mối liên kết thành 1 Bảng, chọn khóa chính cho phù hợp.
* Quy tắc 2 : với kiểu liên kết 1:n Chuyển khóa chính của Bảng bên 1 (cha) sang làm khóa ngoại của BẢNG bên nhiều (con)
* Quy tắc 3: Với kiểu liên kết n:n Chuyển mối liên kết thành một BẢNG có thuộc tính là thuộc tính của mối liên kết, thêm các thuộc tính khóa chính của các BẢNG có liên quan, khóa chính của BẢNG mới này là các thuộc tính mới thêm vào.
* Quy tắc 4: Với thuộc tính đa trị Tách thuộc tính đa trị ra khỏi kiểu thực thể ban đầu, thêm một kiểu thực thể mới chứa thuộc tính đa trị này, xác định lại mối liên kết của kiểu thực thể này với kiểu thực thể ban đầu. Áp dụng các quy tắc từ 1 đến quy tắc 3.

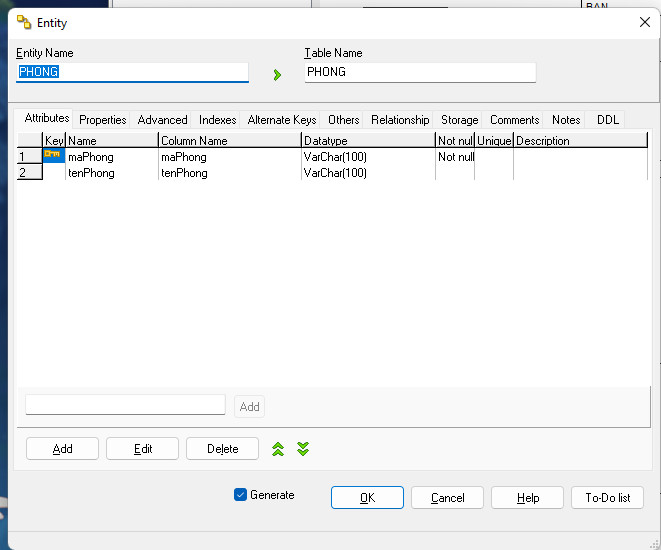
- Lớp tài khoản 🡪 bảng TAIKHOAN:



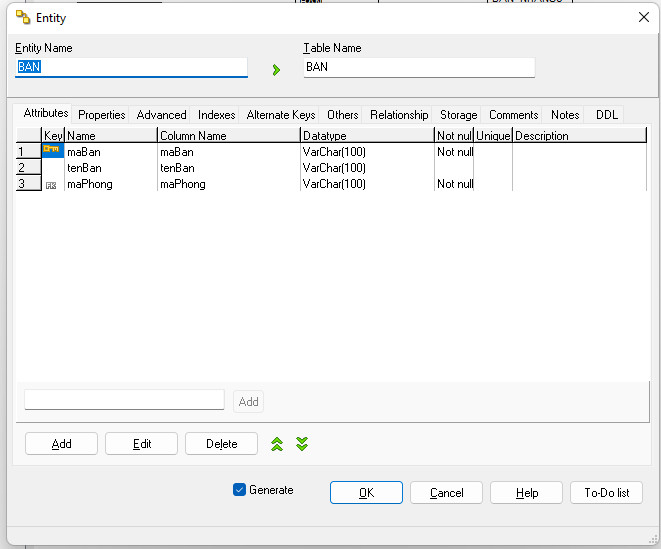
- Lớp tuyển dụng 🡪 bảng TUYỂN DỤNG:



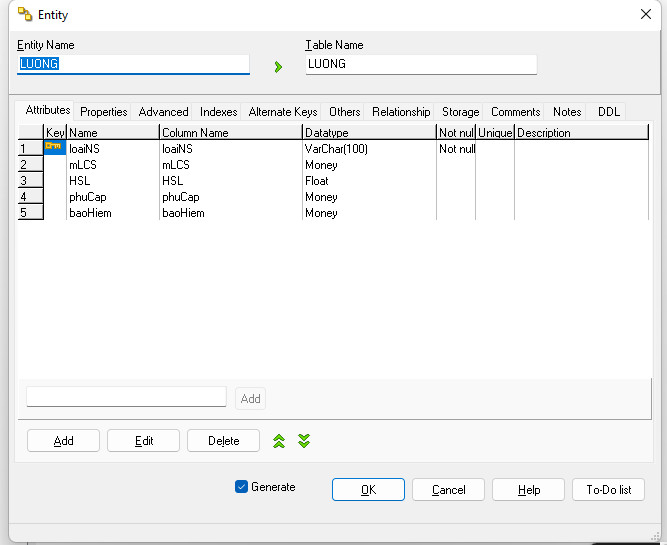
- Lớp phòng 🡪 bảng Phòng



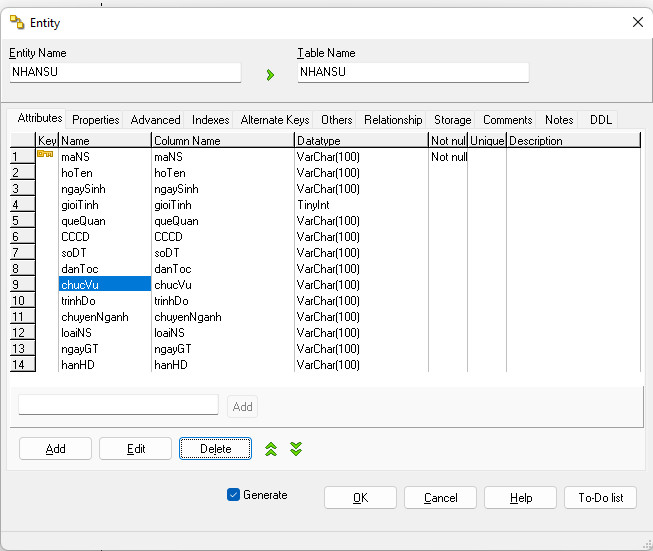
- Lớp ban 🡪 bảng Ban:



- Lớp lương -> bảng Lương:



- Lớp nhân sự -> bảng Nhân Sự:



- Thiết kế cơ sở dữ liệu:

* Lớp PHONG có quan hệ 1-n với lớp BAN nên ta lấy thuộc tính khóa của bảng PHONG ánh xạ thành khóa phụ của bảng BAN
* Lớp BAN có thể có quan hệ n-n với lớp NHANSU sinh ra một bảng con lấy thuộc tính khóa của hai bảng BAN và NHANSU thành khóa phụ đồng thời có thêm 1 thuộc tính là chucVu.
* Lớp LUONG có thể có quan hệ n-n với lớp NHANSU sinh ra một bảng con lấy thuộc tính khóa của hai bảng LUONG và NHANSU thành khóa phụ.
* - Các bảng tìm đươc:

PHONG(**maPhong,** tenPhong)

BAN(**maBan**, **maPhong**, tenBam)

BAN\_NHANSU(**maBan,maNS**, chucVu)

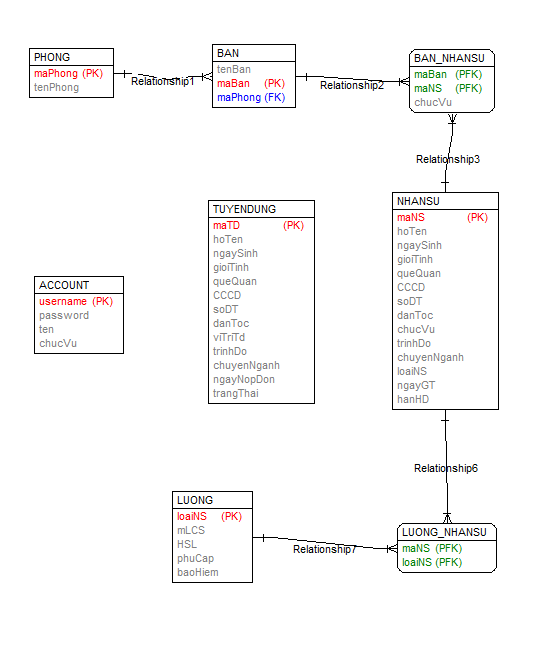
NHANSU(**maNS**, hoTen, ngaySinh, queQuan, CCCD, soDT, danToc, chucVu, trinhDo, chuyenNganh, loaiNS, ngayGT, hanHD)

LUONG(**loaiNS**, maLCS, HSL, phuCap, baoHiem)

LUONG\_NHANSU(**maNS, loaiNS**)

ACCOUNT(**username**, password, ten, chucVu)

TUYENDUNG(**maTD**, hoTen, ngaySinh, gioiTinh, queQuan, CCCD, soDT, danToc, viTriTd, trinhDo, chuyenNganh, ngayNopDon, trangThai)



Hinh 2.10: Các bảng dữ liệu

## 2.4 Cài đặt – Bước 3:

## 2.4.1 Hoàng Hữu Kim – Form đăng nhập:

* Thiết kế giao diện vào/ra:



Hinh 2.11: Đăng nhập

* Đối tượng có trên màn hình: tên đăng nhập và mật khẩu
* Thuật toán kiểm tra tính hợp lệ dữ liệu đầu vào:
* Điền đầy đủ thông tin của tên tài khoản và mật khẩu nếu không sẽ yêu cầu nhập lại
* Kiểm tra xem thông tin vừa nhập có tồn tại trong file hay không. Nếu có thì sẽ chuyển sang màn hình các chức năng quản lý. Nếu sai thì sẽ hiện thông báo cho người dùng biết.
* Ví dụ xử lí tính hợp lệ tại sự kiện bấm nút đăng nhập:

|  |  |
| --- | --- |
| Bắt lỗi nhập tài khoản mật khẩu | try {  danhsach = TaiKhoanController.danh\_sach\_TK();  } catch (IOException ex) {  Logger.getLogger(Login.class.getName()).log(Level.SEVERE, null, ex);  }  try {  if (txt\_taiKhoan.getText().equals("")) {  lb\_thongbao.setText("Vui lòng nhập tài khoản");  txt\_taiKhoan.requestFocus();  } else if (String.valueOf(txt\_matKhau.getPassword()).equals("")) {  lb\_thongbao.setText("Vui lòng nhập mật khẩu");  txt\_matKhau.requestFocus();  } else {  AccountModel A = new AccountModel(tk);  if(danhsach.contains(A)){  if(tk.equals(danhsach.get(danhsach.indexOf(A)).getUsername()) && mk.equals(danhsach.get(danhsach.indexOf(A)).getPassword())){  lb\_thongbao.setText("Thành công");  Main f = new Main(tk);  f.setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);  f.setLocationRelativeTo(null);  f.setResizable(false);  f.setVisible(true);  this.dispose();  }else {  lb\_thongbao.setText("Sai mật khẩu");  }    }else{  lb\_thongbao.setText("Sai thông tin đăng nhập");  }  }  } catch (IOException e) {  JOptionPane.showMessageDialog(this, e.getMessage(), "Lỗi Đăng Nhập", JOptionPane.ERROR\_MESSAGE);  } |
| **Thao tác với file** | |
| Đọc các dữ liệu từ file với scanner | ArrayList<AccountModel> list = new ArrayList<>();  String url = "account.txt";  // Đọc dữ liệu từ File với Scanner  FileInputStream fileInputStream = new FileInputStream(url);  Scanner scanner = new Scanner(fileInputStream);  try {  while (scanner.hasNextLine()) {  String s = scanner.nextLine();  String[] acc = s.split(";");  if (acc.length == 4) {  AccountModel a = new AccountModel();  a.setUsername(acc[0]);  a.setPassword(acc[1]);  a.setTen(acc[2]);  a.setChucvu(acc[3]);  list.add(a);  }  else {  System.out.println(url + " === Lỗi dữ liệu === " + s);  }      }  } finally {  try {  scanner.close();  fileInputStream.close();  } catch (IOException ex) {  java.util.logging.Logger.getLogger(Login.class.getName()).log(Level.SEVERE, null, ex);  }  }  return list;  }  public static void WriteToFile(ArrayList<AccountModel> list) throws FileNotFoundException, IOException{  String url = "Account.txt";  // Đọc dữ liệu từ File với Scanner  try {  FileWriter fw = new FileWriter(url);  BufferedWriter bw = new BufferedWriter(fw);  for(AccountModel o: list){  bw.write(o.getUsername() + ";" + o.getPassword() + ";" + o.getTen() + ";" + o.getChucvu());  bw.newLine();  }  bw.close();  fw.close();  } catch (IOException ex) {  } |
| Ghi các dữ liệu vào file | public static void WriteToFile(ArrayList<DoanhNghiepModel> list) throws FileNotFoundException, IOException{  String url = "doanhnghiep.txt";  // Đọc dữ liệu từ File với Scanner  try {  FileWriter fw = new FileWriter(url);  BufferedWriter bw = new BufferedWriter(fw);  for(DoanhNghiepModel o: list){  bw.write(o.getMaCT());  bw.newLine();  bw.write(o.getTenCT());  bw.newLine();  bw.write(o.getDiaChi());  bw.newLine();  }  bw.close();  fw.close();  } catch (IOException ex) {  }  String them = "congty\_nhom.txt";  // Đọc dữ liệu từ File với Scanner  try {  FileWriter fw = new FileWriter(them);  BufferedWriter bw = new BufferedWriter(fw);  for(DoanhNghiepModel o: list){    bw.write(o.getTenCT());  bw.newLine();  }  bw.close();  fw.close();  } catch (IOException ex) {  }  } |

* Thuật toán xử lý tương ứng với các tác động của người dùng:



Hinh 2.12: Sơ đồ đăng nhập

* Hướng dẫn sử dụng các thư viện chuẩn:
  + java.awt.Toolkit

Ví dụ:

Dimension dim = Toolkit.getDefaultToolkit().getScreenSize();

Lấy kích thước màn hình.

* + java.awt.Color

Ví dụ:

this.setBackground(new java.awt.Color(255, 255, 255, 0));

Tạo màu trong suốt.

* + java.awt.Color

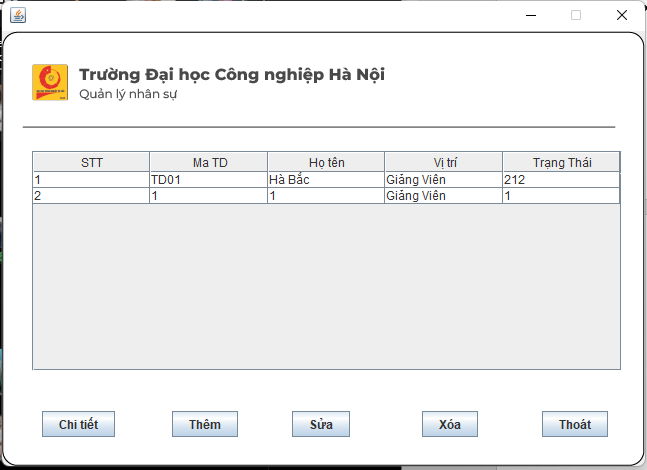
Ví dụ:

this.setBackground(new java.awt.Color(255, 255, 255, 0));

Tạo màu trong suốt.

### 2.4.2 Trần Hữu Vinh – Form Tuyển Dụng

* Thiết kế giao diện vào/ ra:
* “Giao diện tuyển dụng” :



Hình 2.13: Giao diện tuyển dụng

* Đối tượng có trên màn hình:
* Màn hình quản lý tuyển dụng: bảng danh sách các đơn tuyển dụng, các nút xem chi tiết, thêm, sửa, xóa.
* Màn hình chi tiết tuyển dụng: nhập mã tuyển dụng, họ tên người tuyển dụng, số CCCD, Ngày Sinh, Quê quán, Giưới tính, Dân tộc, Số điện thoại, Chức vụ, Trình Độ, Chuyên ngành, Ngày Nộp Đơn, Tình Trạng.
* Màn hinh thêm hồ sơ tuyển dụng: nhập mã tuyển dụng, họ tên người tuyển dụng, số CCCD, Ngày Sinh, Quê quán, Giưới tính, Dân tộc, Số điện thoại, Chức vụ, Trình Độ, Chuyên ngành, Ngày Nộp Đơn, Tình Trạng.
* Màn hình giao diện sửa đơn tuyển dụng: nhập mã tuyển dụng, họ tên người tuyển dụng, số CCCD, Ngày Sinh, Quê quán, Giưới tính, Dân tộc, Số điện thoại, Chức vụ, Trình Độ, Chuyên ngành, Ngày Nộp Đơn, Tình Trạng
* Thuật toán kiểm tra tính hợp lệ dữ liệu đầu vào:
* Khi người dùng muốn xem thông tin chi tiết của 1 hồ sơ cần kích vào hồ sơ cần xem trên danh sách và ấn vào nút xem ở màn hình quản lý hồ sơ
* Khi người dùng muốn thêm 1 hồ sơ thì cần kích vào “Thêm”, điền đủ thông tin(ngoại trừ chính trị và đoàn thể có thể bỏ), mã nhân sự không trùng trong file và kích nút “Lưu” để lưu thông tin vào file.
* Khi người dùng muốn sửa thông tin 1 hồ sơ cần chọn vào hồ sơ muốn sửa và ấn nút sửa, nhập lại các thông tin (ngoại trừ mã nhân sự và thời gian công tác). Sau đó kích nút “Lưu” để cập nhật các thay đổi trong CSDL.
* Khi người dùng muốn xóa hồ sơ cần kích chọn 1 hoặc nhiều hồ sơ trong danh sách rồi kích nút “Xóa”.
* Thuật toán xử lý tương ứng với các tác động của người dùng:

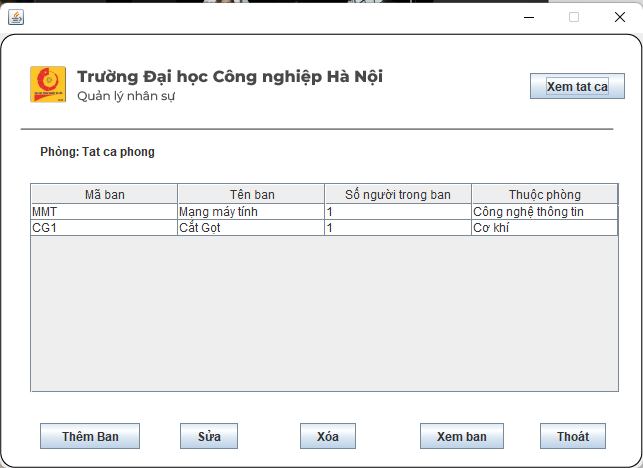
|  |  |
| --- | --- |
| Bắt lỗi thêm tuyển dụng | try{  if(txt\_mTD.getText().equals("")){  txt\_mTD.requestFocus();  throw new Exception("Mã nhân sự không được để trống!");  }  if(txt\_hoTen.getText().equals("")){  txt\_hoTen.requestFocus();  throw new Exception("Họ tên không được để trống!");  }  if(txt\_cCCD.getText().equals("")){  txt\_cCCD.requestFocus();  throw new Exception("Số căn cước không được để trống!");  }  if(!txt\_cCCD.getText().matches("[0-9]+")){  txt\_cCCD.requestFocus();  throw new Exception("Số căn cước chỉ bao gồm số!");  }  if(txt\_ngaySinh.getText()==null)  throw new Exception("Ngày sinh không được để trống!");  if(txt\_queQuan.getText().equals("")){  txt\_queQuan.requestFocus();  throw new Exception("Quê quán không được để trống!");  }  if(txt\_danToc.getText().equals("")){  txt\_danToc.requestFocus();  throw new Exception("Dân tộc không được để trống!");  }  if(txt\_soDT.getText().equals("")){  txt\_soDT.requestFocus();  throw new Exception("Số điện thoại không được để trống!");  }  if(!txt\_soDT.getText().matches("[0-9]+")){  txt\_soDT.requestFocus();  throw new Exception("Số điện thoại chỉ bao gồm số!");  }  if(txt\_trinhDo.getText().equals("")){  txt\_trinhDo.requestFocus();  throw new Exception("Trình độ không được để trống!");  }  if(txt\_chuyenNganh.getText().equals("")){  txt\_chuyenNganh.requestFocus();  throw new Exception("Chuyên ngành không được để trống!");  }  if(txt\_ngayNopDon.getText()==null)  throw new Exception("Ngày nop don không được để trống!");  if(txt\_tinhTrang.getText()==null)  throw new Exception("Tinh trang không được để trống!");  TuyenDungModel ns = new TuyenDungModel();  ns.setMaTD(txt\_mTD.getText());  if(danhsachtd.contains(ns)){  throw new Exception("Mã tuyen dung đã tồn tại!");  }  ns.setMaTD(txt\_mTD.getText());//1  ns.setHoTen(txt\_hoTen.getText());//2  ns.setNgaySinh(txt\_ngaySinh.getText());//3  ns.setGioiTinh(gioitinh\_nam.isSelected()?true:false);//4  ns.setQueQuan(txt\_queQuan.getText());//5  ns.setCCCD(txt\_cCCD.getText());//6  ns.setSoDT(txt\_soDT.getText());//7  ns.setDanToc(txt\_danToc.getText());//8  ns.setViTriTd((String)txt\_chucVu.getSelectedItem());//9  ns.setTrinhDo(txt\_trinhDo.getText());//10  ns.setChuyenNganh(txt\_chuyenNganh.getText());//11  ns.setNgayNopDon(txt\_ngayNopDon.getText());//12  ns.setTrangThai(txt\_tinhTrang.getText());//13  System.out.println(ns.getTrangThai());  danhsachtd.add(ns);  TuyenDungController.WriteToFile(danhsachtd);  JOptionPane.showMessageDialog(this,"Thêm thành công!");  TuyenDung f = new TuyenDung(id);  f.setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);  f.setLocationRelativeTo(null);  f.setResizable(false);  f.setVisible(true);  this.dispose();  this.dispose();  }  catch(Exception e){  JOptionPane.showMessageDialog(this, e.getMessage(), "Thêm thất bại!", JOptionPane.ERROR\_MESSAGE);  }  } |
| Bắt lỗi sửa trùng mã | TuyenDungModel ns = new TuyenDungModel();  ns.setMaTD(txt\_mTD.getText());  if(danhsachtd.contains(ns)){  throw new Exception("Mã nhân sự đã tồn tại!");  } if (txtDC.getText().equals("")) {  txtDC.requestFocus();  throw new Exception("Địa chỉ không được để trống!");  } |
| **Thao tác với file** | |
| Đọc các dữ liệu từ file với scanner | ArrayList<TuyenDungModel> listtd = new ArrayList<>();  String url = "TuyenDung.txt";  // Đọc dữ liệu từ File với Scanner  FileInputStream fileInputStream = new FileInputStream(url);  Scanner scanner = new Scanner(fileInputStream);  try {  String maTD, hoTen, ngaySinh, queQuan, CCCD, soDT, danToc,  viTriTd, trinhDo, chuyenNganh, ngayNopDon, TrangThai;  boolean gioiTinh;  while (scanner.hasNextLine()) {  String s = scanner.nextLine();  String[] ns = s.split(";");  if (ns.length == 13) {  maTD = ns[0];  hoTen = ns[1];  ngaySinh = ns[2];  gioiTinh = Boolean.parseBoolean(ns[3]);  queQuan = ns[4];  CCCD = ns[5];  soDT = ns[6];  danToc = ns[7];  viTriTd = ns[8];  trinhDo = ns[9];  chuyenNganh = ns[10];  ngayNopDon = ns[11];  TrangThai = ns[12];  TuyenDungModel a = new TuyenDungModel(maTD, hoTen, ngaySinh, gioiTinh, queQuan, CCCD, soDT, danToc,  viTriTd, trinhDo, chuyenNganh, ngayNopDon, TrangThai);  listtd.add(a);  } else {  System.out.println(url + " === L?i d? li?u === " + s);  }  }  } finally {  try {  scanner.close();  fileInputStream.close();  } catch (IOException ex) {  }  }  return listtd;  } |
| Ghi các dữ liệu vào file | public static void WriteToFile(ArrayList<TuyenDungModel> list) throws FileNotFoundException, IOException {  String url = "TuyenDung.txt";  // Đọc dữ liệu từ File với Scanner  try {  FileWriter fw = new FileWriter(url);  BufferedWriter bw = new BufferedWriter(fw);  for (TuyenDungModel o : list) {  bw.write(o.toString());  bw.newLine();  }  bw.close();  fw.close();  } catch (IOException ex) {  }  }  public static NhanSuModel TimNhanSuTheoMa(ArrayList<NhanSuModel> list, String id) throws IOException {  TuyenDungModel timNhanSu = new TuyenDungModel(id);  if (!list.contains(timNhanSu)) {  throw new IOException("Không tìm th?y phòng có mã: " + id);  }  return list.get(list.indexOf(timNhanSu));  }  } |
| **Mã nguồn chính các chức năng:** | |
| Truyền dữ liệu doanh nghiệp vào bảng hiển thị | public class TuyenDung extends javax.swing.JFrame {  static ArrayList<TuyenDungModel> ReadToFile() throws FileNotFoundException, IOException {  ArrayList<TuyenDungModel> list = new ArrayList<>();  String url = "TuyenDung.txt";  // Đọc dữ liệu từ File với Scanner  FileInputStream fileInputStream = new FileInputStream(url);  Scanner scanner = new Scanner(fileInputStream);  try {  String maTD, hoTen, ngaySinh, queQuan, CCCD, soDT, danToc,  viTriTd, trinhDo, chuyenNganh, ngayNopDon, TrangThai;  boolean gioiTinh;  while (scanner.hasNextLine()) {  String s = scanner.nextLine();  String[] ns = s.split(";");  maTD = ns[0];  hoTen = ns[1];  ngaySinh = ns[2];  gioiTinh = Boolean.parseBoolean(ns[3]);  queQuan = ns[4];  CCCD = ns[5];  soDT = ns[6];  danToc = ns[7];  viTriTd = ns[8];  trinhDo = ns[9];  chuyenNganh = ns[10];  ngayNopDon = ns[11];  TrangThai = ns[12];  TuyenDungModel a = new TuyenDungModel(maTD, hoTen, ngaySinh, gioiTinh, queQuan, CCCD, soDT, danToc,  viTriTd, trinhDo, chuyenNganh, ngayNopDon, TrangThai);  list.add(a);  }  } finally {  try {  scanner.close();  fileInputStream.close();  } catch (IOException ex) {  }  }  return list;  } |
| *Chức năng xóa* | try {  int index = tb\_TuyenDung.getSelectedRow();  if(index==-1){  throw new IOException("Bạn chưa chọn nhân sự muốn xoá!");  }  if(JOptionPane.showConfirmDialog (null, "Bạn có chắc muốn xoá người này?","Cảnh báo",JOptionPane.YES\_NO\_OPTION)==JOptionPane.YES\_OPTION){  danhsachtd.remove(index);  TuyenDungController.WriteToFile(danhsachtd);  TuyenDung f = new TuyenDung(id);  f.setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);  f.setLocationRelativeTo(null);  f.setResizable(false);  f.setVisible(true);  this.dispose();  }  } catch (IOException ex) {  JOptionPane.showMessageDialog(this, ex.getMessage(), "Thao tác thất bại!", JOptionPane.ERROR\_MESSAGE);  }  } |
| Chức năng thêm | TuyenDungModel ns = new TuyenDungModel();  ns.setMaTD(txt\_mTD.getText());  if(danhsachtd.contains(ns)){  throw new Exception("Mã tuyen dung đã tồn tại!");  }  ns.setMaTD(txt\_mTD.getText());//1  ns.setHoTen(txt\_hoTen.getText());//2  ns.setNgaySinh(txt\_ngaySinh.getText());//3  ns.setGioiTinh(gioitinh\_nam.isSelected()?true:false);//4  ns.setQueQuan(txt\_queQuan.getText());//5  ns.setCCCD(txt\_cCCD.getText());//6  ns.setSoDT(txt\_soDT.getText());//7  ns.setDanToc(txt\_danToc.getText());//8  ns.setViTriTd((String)txt\_chucVu.getSelectedItem());//9  ns.setTrinhDo(txt\_trinhDo.getText());//10  ns.setChuyenNganh(txt\_chuyenNganh.getText());//11  ns.setNgayNopDon(txt\_ngayNopDon.getText());//12  ns.setTrangThai(txt\_tinhTrang.getText());//13  System.out.println(ns.getTrangThai());  danhsachtd.add(ns);  TuyenDungController.WriteToFile(danhsachtd);  JOptionPane.showMessageDialog(this,"Thêm thành công!");  TuyenDung f = new TuyenDung(id);  f.setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);  f.setLocationRelativeTo(null);  f.setResizable(false);  f.setVisible(true);  this.dispose();  this.dispose(); |
| Chức năng sửa | TuyenDungModel ns = new TuyenDungModel();  ns.setMaTD(txt\_mTD.getText());  if(danhsachtd.contains(ns)){  throw new Exception("Mã nhân sự đã tồn tại!");  }  ns.setHoTen(txt\_hoTen.getText());  ns.setNgaySinh(txt\_ngaySinh.getText());  ns.setGioiTinh(gioitinh\_nam.isSelected()?true:false);  ns.setQueQuan(txt\_queQuan.getText());  ns.setCCCD(txt\_cCCD.getText());  ns.setSoDT(txt\_soDT.getText());  ns.setDanToc(txt\_danToc.getText());  ns.setViTriTd((String)txt\_chucVu.getSelectedItem());  ns.setTrinhDo(txt\_trinhDo.getText());  ns.setChuyenNganh(txt\_chuyenNganh.getText());  ns.setNgayNopDon(txt\_ngayNopDon.getText());  ns.setTrangThai(txt\_tinhTrang.getText());  danhsachtd.add(ns);  TuyenDungController.WriteToFile(danhsachtd);  JOptionPane.showMessageDialog(this,"Thêm thành công!");  TuyenDung f = new TuyenDung(id);  f.setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);  f.setLocationRelativeTo(null);  f.setResizable(false);  f.setVisible(true);  this.dispose();  this.dispose();  } |



Hình 2.14: Sơ đồ quản lý tuyển dụng

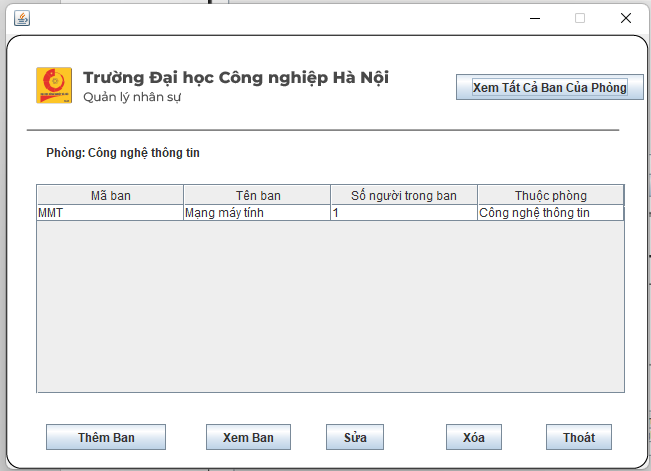
### 2.4.3 Hoàng Trọng Nghĩa – Form Phân Chia Công Việc

* Thiết kế giao diện vào/ra:

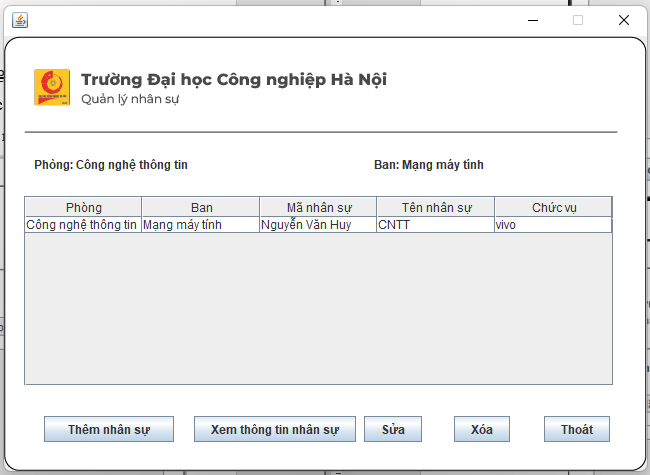


Hình 2.15: Màn hình phân chia công việc

* Đối tượng có trên màn hình: mã phòng ban, tên phòng ban, số người trong ban, thuộc phòng.
* Thuật toán kiểm tra tính hợp lệ dữ liệu đầu vào:
  + Người dùng có thể xem tất cả cá phòng ban bằng chức năng xem tất cả trên góc màn hình.
* Khi người dùng muốn xem 1 phòng ban thì cần chọn tên phòng ban muốn xem rồi bấm xem phòng
* Khi người dùng muốn thêm phòng ban thì kích vào nút “Thêm”, màn hình sẽ xuất hiện thêm nút “Lưu” và nút “Hủy”. Người dùng cần nhập đúng kiểu dữ liệu của các thông tin cần thiết như mã phòng ban(không trùng với mã đã có trong file) tên. Sau đó kích nút “Lưu” để hoàn tất.
* Khi người dùng muốn sửa phòng ban thì có thể sửa đúng kiểu dữ liệu của các thông tin như tên phòng ban Sau đó kích nút “Sửa” để hoàn tất.
* Khi người dùng muốn xóa một phòng ban thì chọn tên phòng ban cần xóa trong combo box rồi kích nút “Xóa”, sau đó kích “Yes” để xác nhận xóa.
* Khi người dùng muốn xem chi tiết các ban có trong phòng chọn vào phần xem ban. Trong phần xem ban cũng có các chức năng thêm sửa xóa nhưu theo tác với phòng. Chức năng có giao diện như sau:



* Tiếp tục muốn xem các nhân sự có trong ban đó ấn chọn “Xem ban” danh sách các nhân sự trong ban sẽ hiện ra, tại đây cũng có các thao tác thêm sửa, xóa và xem thông tin của nhân sự đó. Chức năng có giao diện:



* Ví dụ kiểm tra tính hợp lệ dữ liệu đầu vào khi thêm phòng ban:

|  |  |
| --- | --- |
| Bắt lỗi thêm | try {  if (txt\_TenBanMoi.getText().compareTo("") == 0 || txt\_MaBanMoi.getText().compareTo("") == 0) {  throw new IOException("Tên ban và mã ban không được để trống!");  } else {  BanModel newBan = new BanModel(txt\_MaBanMoi.getText(), txt\_TenBanMoi.getText(), dsPhong.get(cbbox\_dsPhong.getSelectedIndex()));  if (dsBan.contains(newBan) == true) {  throw new IOException("Tên ban hoặc mã ban đã tồn tại!"); |
| Bắt lỗi sửa | ArrayList<BanModel> dsBanCheck = new ArrayList<>();  dsBanCheck.addAll(dsBan);  try {  if (txt\_TenBanMoi.getText().compareTo("") == 0 || txt\_MaBanMoi.getText().compareTo("") == 0) {  throw new IOException("Tên ban và mã ban không được để trống!");  } else {  dsBanCheck.remove(banChinhSua);  BanModel newBan = new BanModel(txt\_MaBanMoi.getText(), txt\_TenBanMoi.getText(), dsPhong.get(cbbox\_dsPhong.getSelectedIndex()));  if (dsBanCheck.contains(newBan) == true) {  throw new IOException("Tên ban ho?c mã ban dã t?n t?i!");  } else {  int viTri = dsBan.indexOf(banChinhSua);  ArrayList<Ban\_NhanSuModel> list = Ban\_NhanSuController.ReadToFile();  for (Ban\_NhanSuModel bx : list) {  if (bx.getBan().equals(banChinhSua)) {  bx.setBan(newBan);  }  } Ban\_NhanSuController.WriteToFile(list);  dsBan.remove(banChinhSua);  dsBan.add(viTri, newBan);  BanController.WriteToFile(dsBan); Ban\_NhanSuController.SuaBan\_NhanSuTheoBan(banChinhSua, newBan);  try {  XemBan f = new XemBan(id, thuocPhong); f.setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);  f.setLocationRelativeTo(null);  f.setResizable(false);  f.setVisible(true);  this.dispose();  } catch (IOException ex) { Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null, ex);  }  }  }  } catch (IOException ex) {  JOptionPane.showMessageDialog(this, ex.getMessage(), "Thêm thất bại!", JOptionPane.ERROR\_MESSAGE);  } |
| **Thao tác với file** | |
| Đọc các dữ liệu từ file với scanner | public static ArrayList ReadToFile() throws FileNotFoundException, IOException {  ArrayList<BanModel> list = new ArrayList<>();  ArrayList<PhongModel> dsPhong = PhongController.ReadToFile();  String url = "ban.txt";  // Đọc dữ liệu từ File với Scanner  FileInputStream fileInputStream = new FileInputStream(url);  Scanner scanner = new Scanner(fileInputStream);  try {  String maBan;  String tenBan;  PhongModel thuocPhong;  while (scanner.hasNextLine()) {  String s = scanner.nextLine();  String[] phong = s.split(";");  if (phong.length == 3) {  maBan = phong[0];  tenBan = phong[1];  thuocPhong = PhongController.TimPhongTheoMa(dsPhong, phong[2]);  BanModel a = new BanModel(maBan, tenBan, thuocPhong);  BanModel.class.getClass().getModule();  list.add(a);  } else {  System.out.println(url + " === L?i d? li?u === " + s);  }  }  } finally {  try {  scanner.close();  fileInputStream.close();  } catch (IOException ex) {  }  }  return list;  } |
| Ghi các dữ liệu vào file | public static void WriteToFile(ArrayList<BanModel> list) throws FileNotFoundException, IOException {  String url = "ban.txt";  // Đọc dữ liệu từ File với Scanner  try {  FileWriter fw = new FileWriter(url);  BufferedWriter bw = new BufferedWriter(fw);  for (BanModel o : list) {  bw.write(o.getMaBan() + ";" + o.getTenBan() + ";" + o.getThuocPhong().getMaPhong());  bw.newLine();  }  bw.close();  fw.close();  } catch (IOException ex) {  }  } |
| **Mã nguồn chính các chức năng:** | |
| Xem Ban | public XemBan(String id, PhongModel thuocPhong) throws IOException {  initComponents();  this.id = id;  this.thuocPhong = thuocPhong;  try {  dsBan = BanController.ReadToFile();  dsBan\_NhanSu = Ban\_NhanSuController.ReadToFile();  if(thuocPhong != null){  dsBanThuocPhong = BanController.TimBanTheoPhong(dsBan, thuocPhong);  text\_Phong.setText(text\_Phong.getText() + thuocPhong.getTenPhong());  }  else {  dsBanThuocPhong.addAll(dsBan);  text\_Phong.setText(text\_Phong.getText() + "Tat ca phong");  }  } catch (IOException ex) {  }  model = (DefaultTableModel) tableDanhSachBan.getModel();  showresult();  }  public void showresult() {  for (BanModel s : dsBanThuocPhong) {  model.addRow(new Object[]{  s.getMaBan(), s.getTenBan(), Ban\_NhanSuController.TimNhanSuTheoBan(dsBan\_NhanSu, s).size(), s.getThuocPhong().getTenPhong()  });  }  } |
| *Chức năng xóa* | try {  int index = tableDanhSachBan.getSelectedRow();  if (index == -1) {  throw new IOException("Bạn chưa chọn ban muốn xoá!");  }  if (JOptionPane.showConfirmDialog(null, "Bạn có chắc muốn xoá ban " + dsBanThuocPhong.get(index).getTenBan() + " ?", "Cảnh báo", JOptionPane.YES\_NO\_OPTION) == JOptionPane.YES\_OPTION) {  BanModel banXoa = dsBanThuocPhong.get(index);  dsBan.remove(banXoa);  BanController.WriteToFile(dsBan);  XoaBan\_NhanSuTheoBan(banXoa);    }  } catch (IOException ex) {  JOptionPane.showMessageDialog(this, ex.getMessage(), "Xoa thất bại!", JOptionPane.ERROR\_MESSAGE);  }  try {  XemBan f = new XemBan(id, thuocPhong);  f.setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);  f.setLocationRelativeTo(null);  f.setResizable(false);  f.setVisible(true);  this.dispose();  } catch (IOException ex) {  Logger.getLogger(XemBan.class.getName()).log(Level.SEVERE, null, ex);  }  public void XoaBan\_NhanSuTheoBan(BanModel banXoa) {  try {  ArrayList<Ban\_NhanSuModel> list = new ArrayList<>();  list.addAll(dsBan\_NhanSu);  for (int i = 0; i < list.size(); i++) {  if (list.get(i).getBan().equals(banXoa)) {  list.remove(list.get(i));  i--;  }  }  Ban\_NhanSuController.WriteToFile(list);  } catch (IOException e) {  System.out.println("Loi b");  }  } |
| Thêm ban | public ThemBan(String id, PhongModel thuocPhong) throws IOException {  initComponents();  this.id = id;  this.thuocPhong = thuocPhong;  try {  dsPhong = PhongController.ReadToFile();  dsBan = BanController.ReadToFile();  } catch (IOException ex) {  }  for (PhongModel p : dsPhong) {  cbbox\_dsPhong.addItem(p.getMaPhong() + " - " + p.getTenPhong());  }  if (thuocPhong != null) {  cbbox\_dsPhong.setSelectedIndex(dsPhong.indexOf(thuocPhong));  }  } |
| Sửa ban | *public SuaBan(String id, PhongModel thuocPhong, BanModel banChinhSua) throws IOException {*  *initComponents();*  *this.id = id;*  *this.thuocPhong = thuocPhong;*  *this.banChinhSua = banChinhSua;*  *try {*  *dsBan = BanController.ReadToFile();*  *dsPhong = PhongController.ReadToFile();*  *} catch (IOException ex) {*  *}*  *for (PhongModel p : dsPhong) {*  *cbbox\_dsPhong.addItem(p.getMaPhong() + " - " + p.getTenPhong());*  *}*  *if (thuocPhong != null) {*  *cbbox\_dsPhong.setSelectedIndex(dsPhong.indexOf(thuocPhong));*  *}*  *showResult();*  *}*  *public void showResult() {*  *txt\_MaBanMoi.setText(banChinhSua.getMaBan());*  *txt\_TenBanMoi.setText(banChinhSua.getTenBan());*  *cbbox\_dsPhong.setSelectedIndex(dsPhong.indexOf(banChinhSua.getThuocPhong()));*  *}* |

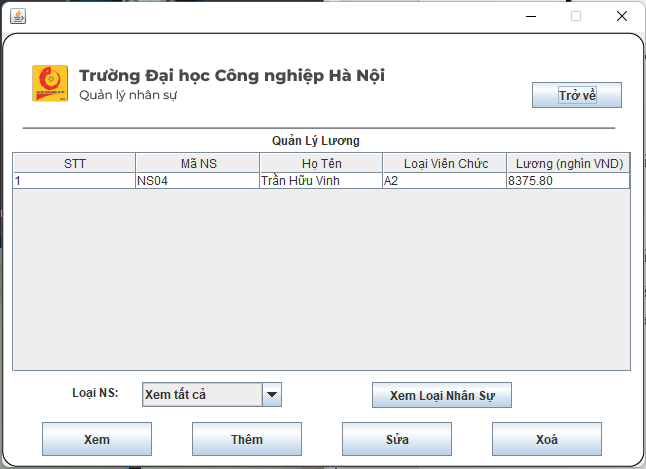
* Thuật toán xử lý tương ứng với các tác động của người dùng:



Hình 2.16: Sơ đồ quản lý phòng ban

### 2.4.4 Trần Hà Bắc – Form quản lý lương:

* Thiết kế giao diện vào/ra:



Hinh 2.17: Màn hình quản lý lương

* Đối tượng có trên màn hình: STT, Mã NS, Họ Tên, Loại Viên Chức, Lương
* Thuật toán kiểm tra tính hợp lệ dữ liệu đầu vào:
* Khi người dùng muốn xem lương của nhân sự thì chonj nhân sự và ấn xem
* Khi người dùng muốn thêm lương cho nhân sự thì ấn lương và hoàn thành các thao tác chọn nhập để hệ thống sẽ tự tính và in ra lương
* Khi người dùng muốn sửa lương cho nhân sự thì chọn nhân sự và ấn vào sửa chọn lại loại nhân sự để tính lương
* Khi người dùng muốn xóa lương thì chọn tên nhân sự cần xóa trong combo box rồi kích nút “Xóa”, sau đó kích “Yes” để xác nhận xóa.
* Khi muốn thêm sửa xóa loại nhân sự thì ấn vào Xem loại nhân sự để thực hiện thêm sửa xóa loại nhân sự.
* Ví dụ kiểm tra tính hợp lệ của dữ liệu đầu vào khi thêm chức vụ:

|  |  |
| --- | --- |
| Bắt lỗi thêm loại nhân sự | private void btn\_suaMKActionPerformed(java.awt.event.ActionEvent evt) {  ArrayList<LuongModel> listL = new ArrayList<>();  try {  listL = LuongController.danh\_sach\_L();  } catch (IOException ex) {  Logger.getLogger(ThemLoaiNS.class.getName()).log(Level.SEVERE, null, ex);  }  try {  if (txt\_loaiNS.getText().equals("")) {  txt\_loaiNS.requestFocus();  throw new Exception("Loại nhân sự không được để trống!");  }  if (txt\_HSL.getText().equals("")) {  txt\_HSL.requestFocus();  throw new Exception("Hệ số lương không được để trống!");  }  if (txt\_MLCS.getText().equals("")) {  txt\_MLCS.requestFocus();  throw new Exception("Mức lương cơ sở không được để trống!");  }  if (txt\_PC.getText().equals("")) {  txt\_PC.requestFocus();  throw new Exception("Mật khẩu không được để trống!");  }  if (txt\_BH.getText().equals("")) {  txt\_BH.requestFocus();  throw new Exception("Xác nhận mật không được để trống!");  }  LuongModel a = new LuongModel(txt\_loaiNS.getText());  if (listL.contains(a)) {  txt\_loaiNS.requestFocus();  throw new Exception("Loại nhân sự đã tồn tại!");  }  a.setmLCS(Double.parseDouble(txt\_MLCS.getText()));  a.setHSL(Double.parseDouble(txt\_HSL.getText()));  a.setPhuCap(Double.parseDouble(txt\_PC.getText()));  a.setBaohiem(Double.parseDouble(txt\_BH.getText()));  listL.add(a);  try {  LuongController.WriteToFile(listL);  throw new Exception("Thêm thành công");  } catch (Exception eX) {  JOptionPane.showMessageDialog(this, eX.getMessage(), "Thông báo!", JOptionPane.NO\_OPTION);  }  XemLoaiNS f;  try {  f = new XemLoaiNS(id);  f.setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);  f.setLocationRelativeTo(null);  f.setResizable(false);  f.setVisible(true);  this.dispose();  } catch (IOException ex) {  Logger.getLogger(QuanLyLuong.class.getName()).log(Level.SEVERE, null, ex);  }  } catch (Exception e) {  JOptionPane.showMessageDialog(this, e.getMessage(), "Thêm thất bại!", JOptionPane.ERROR\_MESSAGE);  }  } |
| Bắt lỗi sửa loại nhân sự | private void btn\_luuActionPerformed(java.awt.event.ActionEvent evt) {  ArrayList<LuongModel> listL = new ArrayList<>();  try {  listL = LuongController.danh\_sach\_L();  } catch (IOException ex) {  java.util.logging.Logger.getLogger(SuaLoaiNS.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);  }  try {  if (txt\_loaiNS.getText().equals("")) {  txt\_loaiNS.requestFocus();  throw new Exception("Tên tài khoản không được để trống!");  }  if (txt\_HSL.getText().equals("")) {  txt\_HSL.requestFocus();  throw new Exception("Họ tên không được để trống!");  }  if (txt\_MLCS.getText().equals("")) {  txt\_MLCS.requestFocus();  throw new Exception("Chức vụ không được để trống!");  }  if (txt\_PC.getText().equals("")) {  txt\_PC.requestFocus();  throw new Exception("Mật khẩu không được để trống!");  }  if (txt\_BH.getText().equals("")) {  txt\_BH.requestFocus();  throw new Exception("Xác nhận mật không được để trống!");  }  int index = listL.indexOf(luong);  LuongModel a = new LuongModel(txt\_loaiNS.getText());  a.setmLCS(Double.parseDouble(txt\_MLCS.getText()));  a.setHSL(Double.parseDouble(txt\_HSL.getText()));  a.setPhuCap(Double.parseDouble(txt\_PC.getText()));  a.setBaohiem(Double.parseDouble(txt\_BH.getText()));  listL.remove(luong);  listL.add(index, a);  LuongController.WriteToFile(listL);  Luong\_NhanSuController.SuaLuong\_NhanSuTheoLuong(luong, a);  XemLoaiNS f;  try {  f = new XemLoaiNS(id);  f.setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);  f.setLocationRelativeTo(null);  f.setResizable(false);  f.setVisible(true);  this.dispose();  } catch (IOException ex) {  java.util.logging.Logger.getLogger(SuaLoaiNS.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);  }  } catch (Exception e) {  JOptionPane.showMessageDialog(this, e.getMessage(), "Thêm thất bại!", JOptionPane.ERROR\_MESSAGE);  }  } |
| *Bắt lỗi xóa loại nhân sự* | try {  int index = DSNS.getSelectedRow();  if(index==-1){  throw new IOException("Bạn chưa chọn Loại NS muốn xoá!");  }  ArrayList<Luong\_NhanSuModel> dsNhanSu = Luong\_NhanSuController.ReadToFile();  ArrayList<Luong\_NhanSuModel> luong\_NhanSuXoa = Luong\_NhanSuController.TimNhanSuTheoLuong(dsNhanSu, danhsach.get(index));  if(JOptionPane.showConfirmDialog (null, "Bạn có chắc muốn xoá "+ danhsach.get(index).getLoaiNS() + " \n"  + "Ðông thoi xóa " + luong\_NhanSuXoa.size() + " du lieu luong nhan su","Cảnh báo",JOptionPane.YES\_NO\_OPTION)==JOptionPane.YES\_OPTION){  dsNhanSu.removeAll(luong\_NhanSuXoa);  Luong\_NhanSuController.WriteToFile(dsNhanSu);    danhsach.remove(index);  LuongController.WriteToFile(danhsach);  XemLoaiNS f = new XemLoaiNS(id);  f.setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);  f.setLocationRelativeTo(null);  f.setResizable(false);  f.setVisible(true);  this.dispose();  }  } catch (IOException ex) {  JOptionPane.showMessageDialog(this, ex.getMessage(), "Thao tác thất bại!", JOptionPane.ERROR\_MESSAGE);  }  } |
| **Thao tác với file** | |
| Đọc các dữ liệu từ file với scanner | public static ArrayList danh\_sach\_L() throws FileNotFoundException, IOException {  ArrayList<LuongModel> list = new ArrayList<>();  String url = "luong.txt";  // Đọc dữ liệu từ File với Scanner  FileInputStream fileInputStream = new FileInputStream(url);  Scanner scanner = new Scanner(fileInputStream);  try {  while (scanner.hasNextLine()) {  LuongModel a = new LuongModel();  String s = scanner.nextLine();  String[] luong = s.split(";");  if (luong.length == 5) {  a.setLoaiNS(luong[0]);  a.setmLCS(Double.parseDouble(luong[1]));  a.setHSL(Double.parseDouble(luong[2]));  a.setPhuCap(Double.parseDouble(luong[3]));  a.setBaohiem(Double.parseDouble(luong[4]));  list.add(a);  }  else {  System.out.println(url + " === L?i d? li?u === " + s);  }  }  } finally {  try {  scanner.close();  fileInputStream.close();  } catch (IOException ex) {  }  }  return list;  } |
| Ghi các dữ liệu vào file | public static void WriteToFile(ArrayList<LuongModel> list) throws FileNotFoundException, IOException {  String url = "luong.txt";  // Đọc dữ liệu từ File với Scanner  try {  FileWriter fw = new FileWriter(url);  BufferedWriter bw = new BufferedWriter(fw);  for (LuongModel o : list) {  bw.write(o.getLoaiNS() + ";" + o.getmLCS().toString() + ";" + o.getHSL().toString() + ";" + o.getPhuCap().toString() + ";" + o.getBaohiem().toString());  bw.newLine();  }  bw.close();  fw.close();  } catch (IOException ex) {  }  } |
| **Mã nguồn chính các chức năng:** | |
| Truyền dữ liệu lương vào bảng hiển thị | public final class QuanLyLuong extends javax.swing.JFrame {  /\*\*  \* Creates new form QuanLyLuong  \*/  ArrayList<NhanSuModel> listNS = new ArrayList<>();  ArrayList<Luong\_NhanSuModel> listLNS = new ArrayList<>();  ArrayList<Luong\_NhanSuModel> listLNSHienThi = new ArrayList<>();  ArrayList<LuongModel> listL = new ArrayList<>();  DefaultTableModel model;  String id;  public QuanLyLuong(String id) throws IOException {  this.id = id;  initComponents();  try {  listNS = NhanSuController.ReadToFile();  listL = LuongController.danh\_sach\_L();  listLNS = Luong\_NhanSuController.ReadToFile();  } catch (IOException ex) {  }  for (LuongModel p : listL) {  jcb\_LoaiNhanSu.addItem(p.getLoaiNS());  }  jcb\_LoaiNhanSu.addItem("Xem tất cả");  jcb\_LoaiNhanSu.setSelectedIndex(listL.size());  listLNSHienThi.clear();  listLNSHienThi.addAll(listLNS);  showresult();  }  public void showresult() {  model = (DefaultTableModel) DSNS.getModel();  model.setRowCount(0);  int dem = 1;  for (Luong\_NhanSuModel s : listLNSHienThi) {  model.addRow(new Object[]{  dem++, s.getNhanSu().getMaNS(), s.getNhanSu().getHoTen(), s.getLuong().getLoaiNS(), String.format("%.2f", s.getLuong().TinhLuong())  });  }  } |
| *Chức năng xóa* | try {  int index = DSNS.getSelectedRow();  if(index==-1){  throw new IOException("Bạn chưa chọn Loại NS muốn xoá!");  }  ArrayList<Luong\_NhanSuModel> dsNhanSu = Luong\_NhanSuController.ReadToFile();  ArrayList<Luong\_NhanSuModel> luong\_NhanSuXoa = Luong\_NhanSuController.TimNhanSuTheoLuong(dsNhanSu, danhsach.get(index));  if(JOptionPane.showConfirmDialog (null, "Bạn có chắc muốn xoá "+ danhsach.get(index).getLoaiNS() + " \n"  + "Ðông thoi xóa " + luong\_NhanSuXoa.size() + " du lieu luong nhan su","Cảnh báo",JOptionPane.YES\_NO\_OPTION)==JOptionPane.YES\_OPTION){  dsNhanSu.removeAll(luong\_NhanSuXoa);  Luong\_NhanSuController.WriteToFile(dsNhanSu);    danhsach.remove(index);  LuongController.WriteToFile(danhsach);  XemLoaiNS f = new XemLoaiNS(id);  f.setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);  f.setLocationRelativeTo(null);  f.setResizable(false);  f.setVisible(true);  this.dispose();  }  } catch (IOException ex) {  JOptionPane.showMessageDialog(this, ex.getMessage(), "Thao tác thất bại!", JOptionPane.ERROR\_MESSAGE);  }  } |
| Chức năng thêm lương nhân viên | public class ThemLuongNhanVien extends javax.swing.JFrame {  ArrayList<LuongModel> dsLuong = new ArrayList<>();  ArrayList<Luong\_NhanSuModel> dsLuong\_NhanSu = new ArrayList<>();  ArrayList<NhanSuModel> dsNhanSu = new ArrayList<>();  ArrayList<NhanSuModel> dsNhanSu\_Loc = new ArrayList<>();  NhanSuModel nhanSuDuocChon = null;  DefaultTableModel model;  java.awt.Frame bb;  String id;  public ThemLuongNhanVien(String id) throws IOException {  initComponents();  this.id = id;  try {  dsLuong = LuongController.danh\_sach\_L();  dsLuong\_NhanSu = Luong\_NhanSuController.ReadToFile();  dsNhanSu = NhanSuController.ReadToFile();  dsNhanSu\_Loc.addAll(dsNhanSu);  } catch (IOException ex) {  System.out.println("Loi");  }  model = (DefaultTableModel) table\_NhanSu.getModel();  hienThiDsNhanSu();  for (LuongModel b : dsLuong) {  cbbox\_dsLuong.addItem(b.getLoaiNS());  }  }  public void hienThiDsNhanSu() {  model.setRowCount(0);  for (NhanSuModel n : dsNhanSu\_Loc) {  model.addRow(new Object[]{  n.getMaNS(), n.getHoTen(), n.getChuyenNganh()  });  }  }  public void timKiemDsNhanSu() {  dsNhanSu\_Loc.clear();  String keywordChuyennganh = text\_KeywordChuyenNganh.getText();  String keywordTen = text\_KeywordTen.getText();  String keywordMa = text\_KeywordMa.getText();  for (NhanSuModel n : dsNhanSu) {  if (n.getMaNS().toLowerCase().contains(keywordMa.toLowerCase())  && n.getHoTen().toLowerCase().contains(keywordTen.toLowerCase())  && n.getChuyenNganh().toLowerCase().contains(keywordChuyennganh.toLowerCase())) {  dsNhanSu\_Loc.add(n);  }  }  hienThiDsNhanSu();  } |
| Chức năng thêm loại nhân sự | private void btn\_suaMKActionPerformed(java.awt.event.ActionEvent evt) {  ArrayList<LuongModel> listL = new ArrayList<>();  try {  listL = LuongController.danh\_sach\_L();  } catch (IOException ex) {  Logger.getLogger(ThemLoaiNS.class.getName()).log(Level.SEVERE, null, ex);  }  try {  if (txt\_loaiNS.getText().equals("")) {  txt\_loaiNS.requestFocus();  throw new Exception("Loại nhân sự không được để trống!");  }  if (txt\_HSL.getText().equals("")) {  txt\_HSL.requestFocus();  throw new Exception("Hệ số lương không được để trống!");  }  if (txt\_MLCS.getText().equals("")) {  txt\_MLCS.requestFocus();  throw new Exception("Mức lương cơ sở không được để trống!");  }  if (txt\_PC.getText().equals("")) {  txt\_PC.requestFocus();  throw new Exception("Mật khẩu không được để trống!");  }  if (txt\_BH.getText().equals("")) {  txt\_BH.requestFocus();  throw new Exception("Xác nhận mật không được để trống!");  }  LuongModel a = new LuongModel(txt\_loaiNS.getText());  if (listL.contains(a)) {  txt\_loaiNS.requestFocus();  throw new Exception("Loại nhân sự đã tồn tại!");  }  a.setmLCS(Double.parseDouble(txt\_MLCS.getText()));  a.setHSL(Double.parseDouble(txt\_HSL.getText()));  a.setPhuCap(Double.parseDouble(txt\_PC.getText()));  a.setBaohiem(Double.parseDouble(txt\_BH.getText()));  listL.add(a);  try {  LuongController.WriteToFile(listL);  throw new Exception("Thêm thành công");  } catch (Exception eX) {  JOptionPane.showMessageDialog(this, eX.getMessage(), "Thông báo!", JOptionPane.NO\_OPTION);  }  XemLoaiNS f;  try {  f = new XemLoaiNS(id);  f.setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);  f.setLocationRelativeTo(null);  f.setResizable(false);  f.setVisible(true);  this.dispose();  } catch (IOException ex) {  Logger.getLogger(QuanLyLuong.class.getName()).log(Level.SEVERE, null, ex);  }  } catch (Exception e) {  JOptionPane.showMessageDialog(this, e.getMessage(), "Thêm thất bại!", JOptionPane.ERROR\_MESSAGE);  }  } |
| Chức năng sửa loại nhân sự | public SuaLoaiNS(String id, LuongModel luong) throws IOException {  this.id = id;  this.luong = luong;  initComponents();  khoitao();  }  public void khoitao() throws IOException {  ArrayList<LuongModel> listL = new ArrayList<>();  listL = LuongController.danh\_sach\_L();  luong = listL.get(listL.indexOf(luong));  txt\_loaiNS.setText(luong.getLoaiNS());  txt\_MLCS.setText(luong.getmLCS().toString());  txt\_HSL.setText(luong.getHSL().toString());  txt\_PC.setText(luong.getPhuCap().toString());  txt\_BH.setText(luong.getBaohiem().toString());  } |

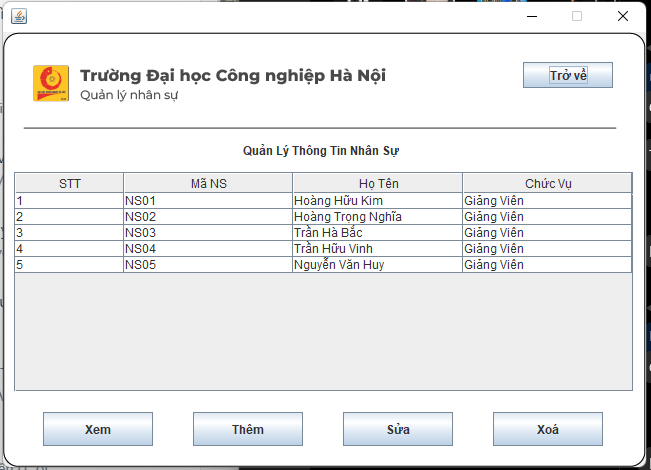
* Thuật toán xử lý tương ứng với các tác động của người dùng:



Hình 2.18: Sơ đồ quản lý lương

### 2.4.5 Nguyễn Văn Huy – Form quản thông tin nhân sự:

* Thiết kế giao diện vào/ra:



Hinh 2.19 Màn hình quản lý thông tin nhân sự

* Đối tượng có trên màn hình:
* Màn hình quản lý thông tin nhân sự: STT, Mã NS, Họ Tên, Chức vụ
* Thuật toán kiểm tra tính hợp lệ dữ liệu đầu vào:
* Khi người dùng muốn xem thông tin cụ thể của một nhân sự thì kích vào dòng chứa nhân sự đó và ấn nút xem.
* Khi người dùng muốn thêm một nhân sự mới thì ấn nút thêm và đảm bảo mã nhân sự không trùng với dữ liệu đã có trong file. Ấn xác nhận để thêm
* Khi người dùng muốn xóa nhân sự thì chọn nhân sự cần xóa và ấn nút xóa nhân sự
* Khi người dùng cần sửa nhân sự thì ấn nút sửa và nhập cho các dữ liệu hợp lệ và ấn xác nhận
* Ví dụ kiểm tra tính hợp lệ của dữ liệu đầu vào:

|  |  |
| --- | --- |
| Bắt lỗi khi thêm nhân sự mới | try{  if(txt\_mNS.getText().equals("")){  txt\_mNS.requestFocus();  throw new Exception("Mã nhân sự không được để trống!");  }  if(txt\_hoTen.getText().equals("")){  txt\_hoTen.requestFocus();  throw new Exception("Họ tên không được để trống!");  }  if(txt\_cCCD.getText().equals("")){  txt\_cCCD.requestFocus();  throw new Exception("Số căn cước không được để trống!");  }  if(!txt\_cCCD.getText().matches("[0-9]+")){  txt\_cCCD.requestFocus();  throw new Exception("Số căn cước chỉ bao gồm số!");  }  if(txt\_ngaySinh.getText()==null)  throw new Exception("Ngày sinh không được để trống!");  if(txt\_queQuan.getText().equals("")){  txt\_queQuan.requestFocus();  throw new Exception("Quê quán không được để trống!");  }  if(txt\_danToc.getText().equals("")){  txt\_danToc.requestFocus();  throw new Exception("Dân tộc không được để trống!");  }  if(txt\_soDT.getText().equals("")){  txt\_soDT.requestFocus();  throw new Exception("Số điện thoại không được để trống!");  }  if(!txt\_soDT.getText().matches("[0-9]+")){  txt\_soDT.requestFocus();  throw new Exception("Số điện thoại chỉ bao gồm số!");  }  if(txt\_trinhDo.getText().equals("")){  txt\_trinhDo.requestFocus();  throw new Exception("Trình độ không được để trống!");  }  if(txt\_chuyenNganh.getText().equals("")){  txt\_chuyenNganh.requestFocus();  throw new Exception("Chuyên ngành không được để trống!");  }  if(txt\_ngayTG.getText()==null)  throw new Exception("Ngày tham gia không được để trống!");  if(txt\_hanHD.getText()==null)  throw new Exception("Hạn hợp đồng không được để trống!");  NhanSuModel ns = new NhanSuModel();  ns.setMaNS(txt\_mNS.getText());  if(danhsach.contains(ns)){  throw new Exception("Mã nhân sự đã tồn tại!");  }  ns.setHoTen(txt\_hoTen.getText());  ns.setNgaySinh(txt\_ngaySinh.getText());  ns.setGioiTinh(gioitinh\_nam.isSelected()?true:false);  ns.setQueQuan(txt\_queQuan.getText());  ns.setCCCD(txt\_cCCD.getText());  ns.setSoDT(txt\_soDT.getText());  ns.setDanToc(txt\_danToc.getText());  ns.setMaPB("Chưa có");  ns.setChucVu((String)txt\_chucVu.getSelectedItem());  ns.setTrinhDo(txt\_trinhDo.getText());  ns.setChuyenNganh(txt\_chuyenNganh.getText());  ns.setLoaiNS(hopDong\_yes.isSelected()?true:false);  ns.setNgayTG(txt\_ngayTG.getText());  ns.setHanHD(txt\_hanHD.getText());  danhsach.add(ns);  NhanSuController.WriteToFile(danhsach);  JOptionPane.showMessageDialog(this,"Thêm thành công!");  QuanLyThongTinNhanSu f = new QuanLyThongTinNhanSu(id);  f.setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);  f.setLocationRelativeTo(null);  f.setResizable(false);  f.setVisible(true);  this.dispose();  this.dispose();  }  catch(Exception e){  JOptionPane.showMessageDialog(this, e.getMessage(), "Thêm thất bại!", JOptionPane.ERROR\_MESSAGE);  } |
| Bắt lỗi khi sửa thông tin nhân sự | try {  if (txt\_hoTen.getText().equals("")) {  txt\_hoTen.requestFocus();  throw new Exception("Họ tên không được để trống!");  }  if (txt\_cCCD.getText().equals("")) {  txt\_cCCD.requestFocus();  throw new Exception("Số căn cước không được để trống!");  }  if (!txt\_cCCD.getText().matches("[0-9]+")) {  txt\_cCCD.requestFocus();  throw new Exception("Số căn cước chỉ bao gồm số!");  }  if (txt\_ngaySinh.getText() == null) {  throw new Exception("Ngày sinh không được để trống!");  }  if (txt\_queQuan.getText().equals("")) {  txt\_queQuan.requestFocus();  throw new Exception("Quê quán không được để trống!");  }  if (txt\_danToc.getText().equals("")) {  txt\_danToc.requestFocus();  throw new Exception("Dân tộc không được để trống!");  }  if (txt\_soDT.getText().equals("")) {  txt\_soDT.requestFocus();  throw new Exception("Số điện thoại không được để trống!");  }  if (!txt\_soDT.getText().matches("[0-9]+")) {  txt\_soDT.requestFocus();  throw new Exception("Số điện thoại chỉ bao gồm số!");  }  if (txt\_trinhDo.getText().equals("")) {  txt\_trinhDo.requestFocus();  throw new Exception("Trình độ không được để trống!");  }  if (txt\_chuyenNganh.getText().equals("")) {  txt\_chuyenNganh.requestFocus();  throw new Exception("Chuyên ngành không được để trống!");  }  if (txt\_ngayTG.getText() == null) {  throw new Exception("Ngày tham gia không được để trống!");  }  if (txt\_Luong.getText() == null) {  throw new Exception("Hạn hợp đồng không được để trống!");  }  NhanSuModel ns = new NhanSuModel();  ns.setMaNS(txt\_mNS.getText());  ns.setHoTen(txt\_hoTen.getText());  ns.setNgaySinh(txt\_ngaySinh.getText());  ns.setGioiTinh(gioitinh\_nam.isSelected() ? true : false);  ns.setQueQuan(txt\_queQuan.getText());  ns.setCCCD(txt\_cCCD.getText());  ns.setSoDT(txt\_soDT.getText());  ns.setDanToc(txt\_danToc.getText());  ns.setMaPB("Chưa có");  ns.setChucVu((String) txt\_chucVu.getSelectedItem());  ns.setTrinhDo(txt\_trinhDo.getText());  ns.setChuyenNganh(txt\_chuyenNganh.getText());  ns.setLoaiNS(hopDong\_yes.isSelected() ? true : false);  ns.setNgayTG(txt\_ngayTG.getText());  ns.setHanHD(txt\_Luong.getText());  danhsach.add(ns);  NhanSuController.WriteToFile(danhsach);  JOptionPane.showMessageDialog(this, "Sửa thành công!");  QuanLyThongTinNhanSu f = new QuanLyThongTinNhanSu(id);  f.setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);  f.setLocationRelativeTo(null);  f.setResizable(false);  f.setVisible(true);  this.dispose();  this.dispose();  } catch (Exception e) {  JOptionPane.showMessageDialog(this, e.getMessage(), "Sửa thất bại!", JOptionPane.ERROR\_MESSAGE);  } |
| Bắt lỗi khi xoá nhân sự | try {  int index = DSNS.getSelectedRow();  if (index == -1) {  throw new IOException("Bạn chưa chọn nhân sự muốn xoá!");  }  if (JOptionPane.showConfirmDialog(null, "Bạn có chắc muốn xoá người này?", "Cảnh báo", JOptionPane.YES\_NO\_OPTION) == JOptionPane.YES\_OPTION) {  ArrayList<Luong\_NhanSuModel> dsLuong = Luong\_NhanSuController.ReadToFile();  dsLuong.remove(dsLuong.indexOf(new Luong\_NhanSuModel(danhsach.get(index))));  danhsach.remove(index);  Luong\_NhanSuController.WriteToFile(dsLuong);  NhanSuController.WriteToFile(danhsach);  QuanLyThongTinNhanSu f = new QuanLyThongTinNhanSu(id);  f.setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);  f.setLocationRelativeTo(null);  f.setResizable(false);  f.setVisible(true);  this.dispose();  }  } catch (IOException ex) {  JOptionPane.showMessageDialog(this, ex.getMessage(), "Thao tác thất bại!", JOptionPane.ERROR\_MESSAGE);  } |
| Thao tác với file | |
| Đọc file dữ liệu thông tin nhân sự | public static ArrayList ReadToFile() throws FileNotFoundException, IOException {  ArrayList<NhanSuModel> list = new ArrayList<>();  String url = "NhanSu.txt";  // Đọc dữ liệu từ File với Scanner  FileInputStream fileInputStream = new FileInputStream(url);  Scanner scanner = new Scanner(fileInputStream);  try {  String maNS, hoTen, ngaySinh, queQuan, CCCD, soDT, danToc,  maPB, chucVu, trinhDo, chuyenNganh, ngayTG, hanHD;  boolean gioiTinh, loaiNS;  while (scanner.hasNextLine()) {  String s = scanner.nextLine();  String[] ns = s.split(";");  if (ns.length == 15) {  maNS = ns[0];  hoTen = ns[1];  ngaySinh = ns[2];  gioiTinh = Boolean.parseBoolean(ns[3]);  queQuan = ns[4];  CCCD = ns[5];  soDT = ns[6];  danToc = ns[7];  maPB = ns[8];  chucVu = ns[9];  trinhDo = ns[10];  chuyenNganh = ns[11];  loaiNS = Boolean.parseBoolean(ns[12]);  ngayTG = ns[13];  hanHD = ns[14];  NhanSuModel a = new NhanSuModel(maNS, hoTen, ngaySinh, gioiTinh, queQuan, CCCD, soDT, danToc,  maPB, chucVu, trinhDo, chuyenNganh, loaiNS, ngayTG, hanHD);  list.add(a);  } else {  System.out.println(url + " === L?i d? li?u === " + s);  }  }  } finally {  try {  scanner.close();  fileInputStream.close();  } catch (IOException ex) {  }  }  return list;  } |
| Ghi dữ liệu mới vào file nhân sự | public static void WriteToFile(ArrayList<NhanSuModel> list) throws FileNotFoundException, IOException {  String url = "NhanSu.txt";  // Đọc dữ liệu từ File với Scanner  try {  FileWriter fw = new FileWriter(url);  BufferedWriter bw = new BufferedWriter(fw);  for (NhanSuModel o : list) {  bw.write(o.toString());  bw.newLine();  }  bw.close();  fw.close();  } catch (IOException ex) {  }  } |
| Mã nguồn chính các chức năng | |
| Hiển thị giao diện quản lý thông tin nhân sự | public class QuanLyThongTinNhanSu extends javax.swing.JFrame {  ArrayList<NhanSuModel> danhsach = new ArrayList<>();  DefaultTableModel model;  java.awt.Frame bb;  String id;  public QuanLyThongTinNhanSu(String id) throws IOException {  this.id = id;  initComponents();  try {  danhsach = NhanSuController.ReadToFile();  } catch (IOException ex) {  }  model = (DefaultTableModel) DSNS.getModel();  showresult();  }  int t = 1;  public void showresult() {  for (NhanSuModel s : danhsach) {  model.addRow(new Object[]{  t++, s.getMaNS(), s.getHoTen(), s.getChucVu()  });  }  }  private void btn\_themActionPerformed(java.awt.event.ActionEvent evt) {  ThemNhanSu f = new ThemNhanSu(id);  f.setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);  f.setLocationRelativeTo(null);  f.setResizable(false);  f.setVisible(true);  this.dispose();  }  private void btn\_suaActionPerformed(java.awt.event.ActionEvent evt) {  try {  int index = DSNS.getSelectedRow();  if (index == -1) {  throw new IOException("Bạn chưa chọn nhân sự muốn sửa!");  }  String t = (String) model.getValueAt(index, 1);  SuaThongTinNhanSu f = new SuaThongTinNhanSu(id, t);  f.setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);  f.setLocationRelativeTo(null);  f.setResizable(false);  f.setVisible(true);  this.dispose();  } catch (IOException ex) {  JOptionPane.showMessageDialog(this, ex.getMessage(), "Thao tác thất bại!", JOptionPane.ERROR\_MESSAGE);  }  }  private void btn\_xemActionPerformed(java.awt.event.ActionEvent evt) {  try {  int index = DSNS.getSelectedRow();  if (index == -1) {  throw new IOException("Bạn chưa chọn nhân sự muốn xem!");  }  String t = (String) model.getValueAt(index, 1);  ChiTietNhanSu f = new ChiTietNhanSu(id, t,"QLTTNS");  f.setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);  f.setLocationRelativeTo(null);  f.setResizable(false);  f.setVisible(true);  this.dispose();  } catch (IOException ex) {  JOptionPane.showMessageDialog(this, ex.getMessage(), "Thao tác thất bại!", JOptionPane.ERROR\_MESSAGE);  }  }  private void btn\_xoaActionPerformed(java.awt.event.ActionEvent evt) {  try {  int index = DSNS.getSelectedRow();  if (index == -1) {  throw new IOException("Bạn chưa chọn nhân sự muốn xoá!");  }  if (JOptionPane.showConfirmDialog(null, "Bạn có chắc muốn xoá người này?", "Cảnh báo", JOptionPane.YES\_NO\_OPTION) == JOptionPane.YES\_OPTION) {  ArrayList<Luong\_NhanSuModel> dsLuong = Luong\_NhanSuController.ReadToFile();  dsLuong.remove(dsLuong.indexOf(new Luong\_NhanSuModel(danhsach.get(index))));  danhsach.remove(index);  Luong\_NhanSuController.WriteToFile(dsLuong);  NhanSuController.WriteToFile(danhsach);  QuanLyThongTinNhanSu f = new QuanLyThongTinNhanSu(id);  f.setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);  f.setLocationRelativeTo(null);  f.setResizable(false);  f.setVisible(true);  this.dispose();  }  } catch (IOException ex) {  JOptionPane.showMessageDialog(this, ex.getMessage(), "Thao tác thất bại!", JOptionPane.ERROR\_MESSAGE);  }  }  private void btn\_troVeActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  Main f;  try {  f = new Main(id);  f.setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);  f.setLocationRelativeTo(null);  f.setResizable(false);  f.setVisible(true);  this.dispose();  } catch (IOException ex) {  Logger.getLogger(ChiTietTaiKhoan.class.getName()).log(Level.SEVERE, null, ex);  }  } |
| Chức  Năng thêm nhân sự | public class ThemNhanSu extends javax.swing.JFrame {  ArrayList<NhanSuModel> danhsach = new ArrayList<>(); String id;  public ThemNhanSu(String id) {  this.id = id;  initComponents();  try {  danhsach = NhanSuController.ReadToFile();  } catch (IOException ex) {    }  }  private void btn\_themActionPerformed(java.awt.event.ActionEvent evt) {  try{  if(txt\_mNS.getText().equals("")){  txt\_mNS.requestFocus();  throw new Exception("Mã nhân sự không được để trống!");  }  if(txt\_hoTen.getText().equals("")){  txt\_hoTen.requestFocus();  throw new Exception("Họ tên không được để trống!");  }  if(txt\_cCCD.getText().equals("")){  txt\_cCCD.requestFocus();  throw new Exception("Số căn cước không được để trống!");  }  if(!txt\_cCCD.getText().matches("[0-9]+")){  txt\_cCCD.requestFocus();  throw new Exception("Số căn cước chỉ bao gồm số!");  }  if(txt\_ngaySinh.getText()==null)  throw new Exception("Ngày sinh không được để trống!");  if(txt\_queQuan.getText().equals("")){  txt\_queQuan.requestFocus();  throw new Exception("Quê quán không được để trống!");  }  if(txt\_danToc.getText().equals("")){  txt\_danToc.requestFocus();  throw new Exception("Dân tộc không được để trống!");  }  if(txt\_soDT.getText().equals("")){  txt\_soDT.requestFocus();  throw new Exception("Số điện thoại không được để trống!");  }  if(!txt\_soDT.getText().matches("[0-9]+")){  txt\_soDT.requestFocus();  throw new Exception("Số điện thoại chỉ bao gồm số!");  }  if(txt\_trinhDo.getText().equals("")){  txt\_trinhDo.requestFocus();  throw new Exception("Trình độ không được để trống!");  }  if(txt\_chuyenNganh.getText().equals("")){  txt\_chuyenNganh.requestFocus();  throw new Exception("Chuyên ngành không được để trống!");  }  if(txt\_ngayTG.getText()==null)  throw new Exception("Ngày tham gia không được để trống!");  if(txt\_hanHD.getText()==null)  throw new Exception("Hạn hợp đồng không được để trống!");  NhanSuModel ns = new NhanSuModel();  ns.setMaNS(txt\_mNS.getText());  if(danhsach.contains(ns)){  throw new Exception("Mã nhân sự đã tồn tại!");  }  ns.setHoTen(txt\_hoTen.getText());  ns.setNgaySinh(txt\_ngaySinh.getText());  ns.setGioiTinh(gioitinh\_nam.isSelected()?true:false);  ns.setQueQuan(txt\_queQuan.getText());  ns.setCCCD(txt\_cCCD.getText());  ns.setSoDT(txt\_soDT.getText());  ns.setDanToc(txt\_danToc.getText());  ns.setMaPB("Chưa có");  ns.setChucVu((String)txt\_chucVu.getSelectedItem());  ns.setTrinhDo(txt\_trinhDo.getText());  ns.setChuyenNganh(txt\_chuyenNganh.getText());  ns.setLoaiNS(hopDong\_yes.isSelected()?true:false);  ns.setNgayTG(txt\_ngayTG.getText());  ns.setHanHD(txt\_hanHD.getText());  danhsach.add(ns);  NhanSuController.WriteToFile(danhsach);  JOptionPane.showMessageDialog(this,"Thêm thành công!");  QuanLyThongTinNhanSu f = new QuanLyThongTinNhanSu(id);  f.setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);  f.setLocationRelativeTo(null);  f.setResizable(false);  f.setVisible(true);  this.dispose();  this.dispose();  }  catch(Exception e){  JOptionPane.showMessageDialog(this, e.getMessage(), "Thêm thất bại!", JOptionPane.ERROR\_MESSAGE);  }  }  private void btn\_huyActionPerformed(java.awt.event.ActionEvent evt) {  QuanLyThongTinNhanSu f;  try {  f = new QuanLyThongTinNhanSu(id);  f.setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);  f.setLocationRelativeTo(null);  f.setResizable(false);  f.setVisible(true);  this.dispose();  } catch (IOException ex) {  Logger.getLogger(ThemNhanSu.class.getName()).log(Level.SEVERE, null, ex);  }  } |
| Chức năng sửa thông tin nhân sự | public class SuaThongTinNhanSu extends javax.swing.JFrame {  ArrayList<NhanSuModel> danhsach = new ArrayList<>();  ArrayList<Luong\_NhanSuModel> dsLNS = new ArrayList<>();  String id;  NhanSuModel nhanSuSua;  Luong\_NhanSuModel luong\_NhanSuSua;  public SuaThongTinNhanSu(String id, String maNS) {  this.id = id;  initComponents();  try {  danhsach = NhanSuController.ReadToFile();  dsLNS = Luong\_NhanSuController.ReadToFile();  } catch (IOException ex) {  }  showresult(maNS);  }  public void showresult(String maNS) {  NhanSuModel s = new NhanSuModel(maNS);  if (danhsach.contains(s)) {  NhanSuModel ns = danhsach.get(danhsach.indexOf(s));  nhanSuSua = ns;  txt\_mNS.setText(ns.getMaNS());  txt\_hoTen.setText(ns.getHoTen());  txt\_ngaySinh.setText(ns.getNgaySinh());  if (ns.isGioiTinh()) {  gioitinh\_nam.isSelected();  } else {  gioitinh\_nu.isSelected();  }  txt\_chucVu.setSelectedItem(ns.getChucVu());  txt\_queQuan.setText(ns.getQueQuan());  txt\_cCCD.setText(ns.getCCCD());  txt\_soDT.setText(ns.getSoDT());  txt\_danToc.setText(ns.getDanToc());  txt\_trinhDo.setText(ns.getTrinhDo());  txt\_chuyenNganh.setText(ns.getChuyenNganh());  if (ns.isLoaiNS()) {  hopDong\_yes.isSelected();  } else {  hopDong\_no.isSelected();  }  txt\_ngayTG.setText(ns.getNgayTG());  txt\_Luong.setText(ns.getHanHD());  danhsach.remove(ns);  ArrayList<Luong\_NhanSuModel> dsLuongCuaNs = new ArrayList<>();  dsLuongCuaNs.addAll(Luong\_NhanSuController.TimLuongTheoNhanSu(dsLNS, ns));  if (dsLuongCuaNs.size() <= 0) {  txt\_Luong.setText("Chua co");  luong\_NhanSuSua = null;  } else {  luong\_NhanSuSua = dsLuongCuaNs.get(0);  txt\_Luong.setText(dsLuongCuaNs.get(0).getLuong().getLoaiNS());  }  }  }  private void btn\_suaActionPerformed(java.awt.event.ActionEvent evt) {  try {  if (txt\_hoTen.getText().equals("")) {  txt\_hoTen.requestFocus();  throw new Exception("Họ tên không được để trống!");  }  if (txt\_cCCD.getText().equals("")) {  txt\_cCCD.requestFocus();  throw new Exception("Số căn cước không được để trống!");  }  if (!txt\_cCCD.getText().matches("[0-9]+")) {  txt\_cCCD.requestFocus();  throw new Exception("Số căn cước chỉ bao gồm số!");  }  if (txt\_ngaySinh.getText() == null) {  throw new Exception("Ngày sinh không được để trống!");  }  if (txt\_queQuan.getText().equals("")) {  txt\_queQuan.requestFocus();  throw new Exception("Quê quán không được để trống!");  }  if (txt\_danToc.getText().equals("")) {  txt\_danToc.requestFocus();  throw new Exception("Dân tộc không được để trống!");  }  if (txt\_soDT.getText().equals("")) {  txt\_soDT.requestFocus();  throw new Exception("Số điện thoại không được để trống!");  }  if (!txt\_soDT.getText().matches("[0-9]+")) {  txt\_soDT.requestFocus();  throw new Exception("Số điện thoại chỉ bao gồm số!");  }  if (txt\_trinhDo.getText().equals("")) {  txt\_trinhDo.requestFocus();  throw new Exception("Trình độ không được để trống!");  }  if (txt\_chuyenNganh.getText().equals("")) {  txt\_chuyenNganh.requestFocus();  throw new Exception("Chuyên ngành không được để trống!");  }  if (txt\_ngayTG.getText() == null) {  throw new Exception("Ngày tham gia không được để trống!");  }  if (txt\_Luong.getText() == null) {  throw new Exception("Hạn hợp đồng không được để trống!");  }  NhanSuModel ns = new NhanSuModel();  ns.setMaNS(txt\_mNS.getText());  ns.setHoTen(txt\_hoTen.getText());  ns.setNgaySinh(txt\_ngaySinh.getText());  ns.setGioiTinh(gioitinh\_nam.isSelected() ? true : false);  ns.setQueQuan(txt\_queQuan.getText());  ns.setCCCD(txt\_cCCD.getText());  ns.setSoDT(txt\_soDT.getText());  ns.setDanToc(txt\_danToc.getText());  ns.setMaPB("Chưa có");  ns.setChucVu((String) txt\_chucVu.getSelectedItem());  ns.setTrinhDo(txt\_trinhDo.getText());  ns.setChuyenNganh(txt\_chuyenNganh.getText());  ns.setLoaiNS(hopDong\_yes.isSelected() ? true : false);  ns.setNgayTG(txt\_ngayTG.getText());  ns.setHanHD(txt\_Luong.getText());  danhsach.add(ns);  NhanSuController.WriteToFile(danhsach);  JOptionPane.showMessageDialog(this, "Sửa thành công!");  QuanLyThongTinNhanSu f = new QuanLyThongTinNhanSu(id);  f.setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);  f.setLocationRelativeTo(null);  f.setResizable(false);  f.setVisible(true);  this.dispose();  this.dispose();  } catch (Exception e) {  JOptionPane.showMessageDialog(this, e.getMessage(), "Sửa thất bại!", JOptionPane.ERROR\_MESSAGE);  }  }  private void btn\_huyActionPerformed(java.awt.event.ActionEvent evt) {  QuanLyThongTinNhanSu f;  try {  f = new QuanLyThongTinNhanSu(id);  f.setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);  f.setLocationRelativeTo(null);  f.setResizable(false);  f.setVisible(true);  this.dispose();  } catch (IOException ex) {  Logger.getLogger(SuaThongTinNhanSu.class.getName()).log(Level.SEVERE, null, ex);  }  } |
| Chức năng xoá nhân sự | private void btn\_xoaActionPerformed(java.awt.event.ActionEvent evt) {  try {  int index = DSNS.getSelectedRow();  if (index == -1) {  throw new IOException("Bạn chưa chọn nhân sự muốn xoá!");  }  if (JOptionPane.showConfirmDialog(null, "Bạn có chắc muốn xoá người này?", "Cảnh báo", JOptionPane.YES\_NO\_OPTION) == JOptionPane.YES\_OPTION) {  ArrayList<Luong\_NhanSuModel> dsLuong = Luong\_NhanSuController.ReadToFile();  dsLuong.remove(dsLuong.indexOf(new Luong\_NhanSuModel(danhsach.get(index))));  danhsach.remove(index);  Luong\_NhanSuController.WriteToFile(dsLuong);  NhanSuController.WriteToFile(danhsach);  QuanLyThongTinNhanSu f = new QuanLyThongTinNhanSu(id);  f.setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);  f.setLocationRelativeTo(null);  f.setResizable(false);  f.setVisible(true);  this.dispose();  }  } catch (IOException ex) {  JOptionPane.showMessageDialog(this, ex.getMessage(), "Thao tác thất bại!", JOptionPane.ERROR\_MESSAGE);  }  } |
| Chức năng xem thông tin chi tiết nhân sự | public class ChiTietNhanSu extends javax.swing.JFrame {  ArrayList<NhanSuModel> danhsach = new ArrayList<>();  ArrayList<Luong\_NhanSuModel> dsLNS = new ArrayList<>();  String id;  String MaTV;  public ChiTietNhanSu(String id, String MaNS,String MaTV) throws IOException {  this.id = id;  this.MaTV = MaTV;  initComponents();  danhsach = NhanSuController.ReadToFile();  dsLNS = Luong\_NhanSuController.ReadToFile();  showresult(MaNS);  }  public void showresult(String maNS) {  NhanSuModel s = new NhanSuModel(maNS);  if (danhsach.contains(s)) {  NhanSuModel ns = danhsach.get(danhsach.indexOf(s));  txt\_mNS.setText(ns.getMaNS());  txt\_hoTen.setText(ns.getHoTen());  txt\_ngaySinh.setText(ns.getNgaySinh());  if (ns.isGioiTinh()) {  txt\_gioiTinh.setText("Nam");  } else {  txt\_gioiTinh.setText("Nữ");  }  txt\_queQuan.setText(ns.getQueQuan());  txt\_cCCD.setText(ns.getCCCD());  txt\_soDT.setText(ns.getSoDT());  txt\_danToc.setText(ns.getDanToc());  txt\_phongBan.setText("Chưa có");  txt\_chucVu.setText(ns.getChucVu());  txt\_trinhDo.setText(ns.getTrinhDo());  txt\_chuyenNganh.setText(ns.getChuyenNganh());  if (ns.isLoaiNS()) {  txt\_loaiNS.setText("Hợp đồng");  } else {  txt\_loaiNS.setText("Không hợp đồng");  }  txt\_ngayTG.setText(ns.getNgayTG());  txt\_hanHD.setText(ns.getHanHD());  ArrayList<Luong\_NhanSuModel> dsLuongCuaNs = new ArrayList<>();  dsLuongCuaNs.addAll(Luong\_NhanSuController.TimLuongTheoNhanSu(dsLNS, ns));  if (dsLuongCuaNs.size() <= 0) {  txt\_Luong.setText("Chua co");  } else {  txt\_Luong.setText(dsLuongCuaNs.get(0).getLuong().getLoaiNS());  }  }  private void btn\_thoatActionPerformed(java.awt.event.ActionEvent evt) {  switch (MaTV) {  case "QLL" -> {  QuanLyLuong f;  try {  f = new QuanLyLuong(id);  f.setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);  f.setLocationRelativeTo(null);  f.setResizable(false);  f.setVisible(true);  this.dispose();  } catch (IOException ex) {  Logger.getLogger(ChiTietNhanSu.class.getName()).log(Level.SEVERE, null, ex);  }  }  case "XNSTB" -> {  XemNhanSuTrongBan f;  try {  f = new XemNhanSuTrongBan(id,new BanModel("","",null),null);  f.setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);  f.setLocationRelativeTo(null);  f.setResizable(false);  f.setVisible(true);  this.dispose();  } catch (IOException ex) {  Logger.getLogger(ChiTietNhanSu.class.getName()).log(Level.SEVERE, null, ex);  }  }  case "QLTTNS" -> {  QuanLyThongTinNhanSu f;  try {  f = new QuanLyThongTinNhanSu(id);  f.setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);  f.setLocationRelativeTo(null);  f.setResizable(false);  f.setVisible(true);  this.dispose();  } catch (IOException ex) {  Logger.getLogger(ChiTietNhanSu.class.getName()).log(Level.SEVERE, null, ex);  }  }  default -> {  Main f;  try {  f = new Main(id);  f.setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);  f.setLocationRelativeTo(null);  f.setResizable(false);  f.setVisible(true);  this.dispose();  } catch (IOException ex) {  Logger.getLogger(ChiTietNhanSu.class.getName()).log(Level.SEVERE, null, ex);  }  }  }  } |

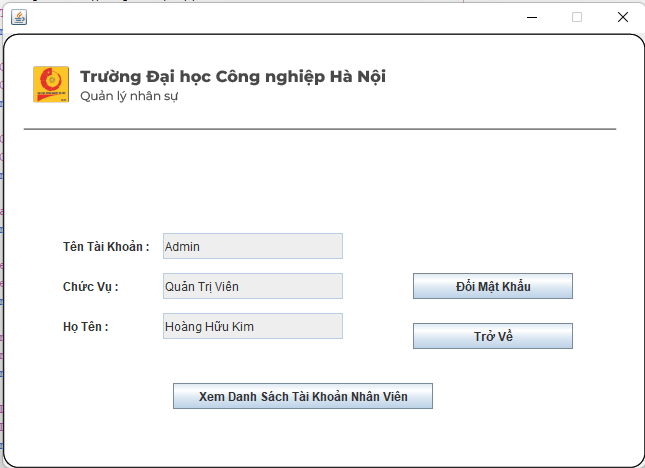
* Thuật toán xử lý tương ứng với các tác động của người dùng:



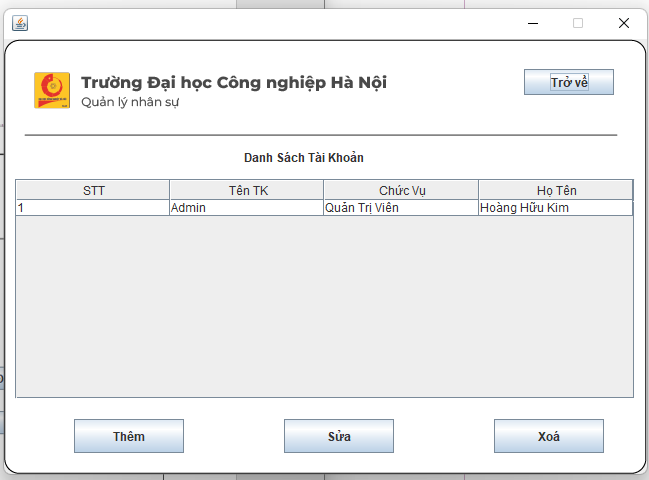
Hình 2.20: Sơ đồ quản lý thông tin nhân sự

### 2.4.6 Hoàng Hữu Kim – Form quản lý tài khoản:

* Thiết kế giao diện vào/ra:



Hình 2.21: Màn hình quản lý tài khoản



Hình 2.22: Màn hình quản lý danh sách tài khoản

* Đối tượng có trên màn hình:
* Màn hình quản lý tài khoản: Tên tài khoản, chức vụ, họ tên.
* Màn hình quản lý danh sách tài khoản: STT, Tên TK, Chức Vụ, Họ Tên
* Thuật toán kiểm tra tính hợp lệ dữ liệu đầu vào:
* Khi người dùng muốn đổi mật khẩu thì cần nhập mật khẩu cũ và mật khẩu mới rồi kích nút “Đổi mật khẩu”.
* Khi người dùng(với quyền người quản lý chính) muốn thêm tài khoản thì kích vào nút “Thêm” tại màn hình quản lý danh sách tài khoản. Lúc này màn hình sẽ hiện thêm nút “Lưu” và “Hủy”. Người dùng cần nhập tên tài khoản mới, họ tên người quản lý, mật khẩu mới và ảnh rồi kích nút “Lưu” để hoàn tất.
* Khi người dùng(với quyền người quản lý chính) muốn sửa tài khoản thì kích vào nút “Sửa” ở màn hình Quản lý danh sách tài khoản. Người dùng có thể thay đổi tên tài khoản mới, họ tên người quản lý và ảnh rồi kích nút “Hoàn tất” để hoàn tất.
* Khi người dùng(với quyền người quản lý chính) muốn xóa tài khoản thì kích vào nút “Xóa” tại màn hình quản lý danh sách tài khoản.
* Ví dụ kiểm tra tính hợp lệ của dữ liệu đầu vào khi thêm tài khoản:

|  |  |
| --- | --- |
| *Bắt lỗi sửa mật khẩu* | String mkc = txt\_matKhauCu.getText();  String mkm = txt\_matKhauMoi.getText();  String mkxc = txt\_xacNhan.getText();  try {  listAc = TaiKhoanController.danh\_sach\_TK();  } catch (IOException ex) {  java.util.logging.Logger.getLogger(SuaMatKhau.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);  }  AccountModel accountModel = new AccountModel(id);  accountModel = listAc.get(listAc.indexOf(accountModel));  try {  if (mkc.equals("")) {  txt\_matKhauCu.requestFocus();  throw new Exception("Mật khẩu không được để trống!");  }  if (mkm.equals("")) {  txt\_matKhauMoi.requestFocus();  throw new Exception("Mật khẩu mới không được để trống!");  }  if (mkxc.equals("")) {  txt\_xacNhan.requestFocus();  throw new Exception("Mật khẩu xác nhận không được để trống!");  }  if (mkc.equals(accountModel.getPassword())) {  String pattern = "(?=.\*[0-9])(?=.\*[a-z])(?=.\*[A-Z])(?=\\S+$).{8,}";  if (mkm.equals(mkxc)) {  if (mkm.matches(pattern)) {  accountModel.setPassword(mkm);  try {  TaiKhoanController.WriteToFile(listAc);  throw new Exception("Đổi mật khẩu thành công");  } catch (Exception e) {  JOptionPane.showMessageDialog(this, e.getMessage(), "Thông báo", JOptionPane.NO\_OPTION);  }  try {  ChiTietTaiKhoan f;  f = new ChiTietTaiKhoan(id);  f.setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);  f.setLocationRelativeTo(null);  f.setResizable(false);  f.setVisible(true);  this.dispose();  } catch (IOException ex) {  java.util.logging.Logger.getLogger(Main.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);  }  } else {  txt\_matKhauMoi.requestFocus();  throw new Exception("Mật khẩu phải bao gồm cả chữ hoa, chữ thường, số và ít nhất 8 ký tự");  }  } else {  txt\_xacNhan.requestFocus();  throw new Exception("Mật khẩu không trùng nhau");  }  } else {  txt\_matKhauCu.requestFocus();  throw new Exception("Sai mật khẩu");  }  } catch (Exception ex) {  JOptionPane.showMessageDialog(this, ex.getMessage(), "Mật khẩu không đúng định dạng", JOptionPane.ERROR\_MESSAGE);  } |
| Bắt lỗi thêm tài khoản | try {  if (txt\_tenTK.getText().equals("")) {  txt\_tenTK.requestFocus();  throw new Exception("Tên tài khoản không được để trống!");  }  if (txt\_ten.getText().equals("")) {  txt\_ten.requestFocus();  throw new Exception("Họ tên không được để trống!");  }  if (txt\_chucVu.getText().equals("")) {  txt\_chucVu.requestFocus();  throw new Exception("Chức vụ không được để trống!");  }  if (txt\_matKhau.getText().equals("")) {  txt\_chucVu.requestFocus();  throw new Exception("Mật khẩu không được để trống!");  }  if (txt\_xacNhan.getText().equals("")) {  txt\_chucVu.requestFocus();  throw new Exception("Xác nhận mật không được để trống!");  }  AccountModel a = new AccountModel(txt\_tenTK.getText());  if (listAc.contains(a)) {  txt\_tenTK.requestFocus();  throw new Exception("Tên tài khoản đã tồn tại!");  }  String mkm = txt\_matKhau.getText();  String mkxc = txt\_xacNhan.getText();  String pattern = "(?=.\*[0-9])(?=.\*[a-z])(?=.\*[A-Z])(?=\\S+$).{8,}";  if (mkm.equals(mkxc)) {  if (mkm.matches(pattern)) {  a.setTen(txt\_ten.getText());  a.setPassword(mkm);  a.setChucvu(txt\_chucVu.getText());  listAc.add(a);  try {  TaiKhoanController.WriteToFile(listAc);  throw new Exception("Thêm tài khoản thành công");  } catch (Exception ex) {  JOptionPane.showMessageDialog(this, ex.getMessage(), "Thông báo", JOptionPane.NO\_OPTION);  }  try {  DanhSachTK f;  f = new DanhSachTK(id);  f.setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);  f.setLocationRelativeTo(null);  f.setResizable(false);  f.setVisible(true);  this.dispose();  } catch (IOException ex) {  java.util.logging.Logger.getLogger(Main.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);  }  } else {  txt\_matKhau.requestFocus();  throw new Exception("Mật khẩu không đúng định dạng: bao gồm số, chữ hoa, chữ thường, tối thiểu 8 ký tự");  }  } else {  txt\_xacNhan.requestFocus();  throw new Exception("Mat khau khong trung nhau");  }  } catch (Exception e) {  JOptionPane.showMessageDialog(this, e.getMessage(), "Thêm thất bại!", JOptionPane.ERROR\_MESSAGE);  } |
| Bắt lỗi sửa thông tin tài khoản | try {  if (txt\_tenTK.getText().equals("")) {  txt\_tenTK.requestFocus();  throw new Exception("Tên tài khoản không được để trống!");  }  if (txt\_ten.getText().equals("")) {  txt\_ten.requestFocus();  throw new Exception("Họ tên không được để trống!");  }  if (txt\_chucVu.getText().equals("")) {  txt\_chucVu.requestFocus();  throw new Exception("Chức vụ không được để trống!");  }  if (txt\_matKhau.getText().equals("")) {  txt\_chucVu.requestFocus();  throw new Exception("Mật khẩu không được để trống!");  }  if (txt\_xacNhan.getText().equals("")) {  txt\_chucVu.requestFocus();  throw new Exception("Xác nhận mật không được để trống!");  }  AccountModel a = new AccountModel(txt\_tenTK.getText());  a = listAc.get(listAc.indexOf(a));  if (a.getChucvu().equals("Quản Trị Viên")) {  throw new IOException("Bạn không thể sửa người này!");  }  String mkm = txt\_matKhau.getText();  String mkxc = txt\_xacNhan.getText();  String pattern = "(?=.\*[0-9])(?=.\*[a-z])(?=.\*[A-Z])(?=\\S+$).{8,}";  if (mkm.equals(mkxc)) {  if (mkm.matches(pattern)) {  a.setTen(txt\_ten.getText());  a.setPassword(mkm);  a.setChucvu(txt\_chucVu.getText());  try {  TaiKhoanController.WriteToFile(listAc);  throw new Exception("Sửa thông tin tài khoản thành công");  } catch (Exception ex) {  JOptionPane.showMessageDialog(this, ex.getMessage(), "Thông báo", JOptionPane.NO\_OPTION);  }  try {  DanhSachTK f;  f = new DanhSachTK(id);  f.setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);  f.setLocationRelativeTo(null);  f.setResizable(false);  f.setVisible(true);  this.dispose();  } catch (IOException ex) {  java.util.logging.Logger.getLogger(Main.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);  }  } else {  txt\_matKhau.requestFocus();  throw new Exception("Mật khẩu không đúng định dạng: bao gồm số, chữ hoa, chữ thường, tối thiểu 8 ký tự");  }  } else {  txt\_xacNhan.requestFocus();  throw new Exception("Mật khẩu không trùng nhau");  }  } catch (Exception e) {  JOptionPane.showMessageDialog(this, e.getMessage(), "Thêm thất bại!", JOptionPane.ERROR\_MESSAGE);  } |
| Bắt lỗi xoá tài khoản | try {  int index = DSTK.getSelectedRow();  if (index == -1) {  throw new IOException("Bạn chưa chọn tài khoản muốn xoá!");  }  String t = (String) model.getValueAt(index, 1);  AccountModel a = new AccountModel(t);  a = danhsach.get(danhsach.indexOf(a));  if (a.getChucvu().equals("Quản Trị Viên")) {  throw new IOException("Bạn không thể xoá người này!");  }  if (JOptionPane.showConfirmDialog(null, "Bạn có chắc muốn xoá người này?", "Cảnh báo", JOptionPane.YES\_NO\_OPTION) == JOptionPane.YES\_OPTION) {  danhsach.remove(index);  try {  TaiKhoanController.WriteToFile(danhsach);  throw new Exception("Xoá thành công!");  } catch (Exception e) {  JOptionPane.showMessageDialog(this, e.getMessage(), "Thông báo", JOptionPane.NO\_OPTION);  }  try {  DanhSachTK f;  f = new DanhSachTK(id);  f.setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);  f.setLocationRelativeTo(null);  f.setResizable(false);  f.setVisible(true);  this.dispose();  } catch (IOException ex) {  java.util.logging.Logger.getLogger(Main.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);  }  }  } catch (IOException ex) {  JOptionPane.showMessageDialog(this, ex.getMessage(), "Thao tác thất bại!", JOptionPane.ERROR\_MESSAGE);  } |
| **Thao tác với file** | |
| Đọc các dữ liệu từ file với scanner | ArrayList<AccountModel> list = new ArrayList<>();  String url = "account.txt";  // Đọc dữ liệu từ File với Scanner  FileInputStream fileInputStream = new FileInputStream(url);  Scanner scanner = new Scanner(fileInputStream);  try {  while (scanner.hasNextLine()) {  String s = scanner.nextLine();  String[] acc = s.split(";");  if (acc.length == 4) {  AccountModel a = new AccountModel();  a.setUsername(acc[0]);  a.setPassword(acc[1]);  a.setTen(acc[2]);  a.setChucvu(acc[3]);  list.add(a);  }  else {  System.out.println(url + " === Lỗi dữ liệu === " + s);  }      }  } finally {  try {  scanner.close();  fileInputStream.close();  } catch (IOException ex) {  java.util.logging.Logger.getLogger(Login.class.getName()).log(Level.SEVERE, null, ex);  }  }  return list;  }  public static void WriteToFile(ArrayList<AccountModel> list) throws FileNotFoundException, IOException{  String url = "Account.txt";  // Đọc dữ liệu từ File với Scanner  try {  FileWriter fw = new FileWriter(url);  BufferedWriter bw = new BufferedWriter(fw);  for(AccountModel o: list){  bw.write(o.getUsername() + ";" + o.getPassword() + ";" + o.getTen() + ";" + o.getChucvu());  bw.newLine();  }  bw.close();  fw.close();  } catch (IOException ex) {  } |
| Ghi các dữ liệu vào file | public static void WriteToFile(ArrayList<DoanhNghiepModel> list) throws FileNotFoundException, IOException{  String url = "doanhnghiep.txt";  // Đọc dữ liệu từ File với Scanner  try {  FileWriter fw = new FileWriter(url);  BufferedWriter bw = new BufferedWriter(fw);  for(DoanhNghiepModel o: list){  bw.write(o.getMaCT());  bw.newLine();  bw.write(o.getTenCT());  bw.newLine();  bw.write(o.getDiaChi());  bw.newLine();  }  bw.close();  fw.close();  } catch (IOException ex) {  }  String them = "congty\_nhom.txt";  // Đọc dữ liệu từ File với Scanner  try {  FileWriter fw = new FileWriter(them);  BufferedWriter bw = new BufferedWriter(fw);  for(DoanhNghiepModel o: list){    bw.write(o.getTenCT());  bw.newLine();  }  bw.close();  fw.close();  } catch (IOException ex) {  }  } |
|  |  |

* Thuật toán xử lý tương ứng với các tác động của người dùng:



Hình 2.33: Sơ đồ quản lý tài khoản

# PHẦN 3. KẾT LUẬN VÀ BÀI HỌC KINH NGHIỆM

## 3.1 Nội dung đã thực hiện:

- Sau khi tìm hiểu và thực hiện bài tập lớn của mình em đã có cái nhìn tổng quan về Java, đã hiểu được các tác vụ trong một dự án thực tế như thêm, sửa, xóa, tìm kiếm, … nhằm quản lý dễ dàng hơn, các kỹ năng về tạo dựng cơ sở dữ liệu từ nghiệp vụ thực tế nhờ đi khảo sát tại các đơn vị, kỹ năng quản lý dự án…

- Các chuẩn đầu ra:

* Đã có thể tự xây dựng được Cơ sở dữ liệu dựa trên thực tế.
* Đã có thể tự cài đặt được một chương trình đơn giản.
* Báo cáo, hướng dẫn sử dụng rõ ràng, rành mạch.

- Sau khi hoàn thành bài tập lớn chúng em thấy rằng muốn làm được sản phẩm tốt, phù hợp với khách hàng cũng như xã hội thì cần bám sát thực tế do đó phải khảo sát một cách kỹ lưỡng để đưa ra các tác vụ cần làm giúp ích cho cộng đồng: Giao diện dễ nhìn, dễ sử dụng, nội dung phù hợp với thực tế …

## 3.2 Hướng phát triển:

Qua quá trình học tập tại nhà trường và tìm hiểu kiến thức trên mạng chúng em đã có thể làm ra sản phẩm thực tế với các chức năng thiết thực dễ sử dụng.

Qua đề tài lần này em đã có thêm kinh nghiệm khảo sát thực tế, các nghiệp vụ được rõ ràng phù hợp với công việc được giao. Được học hỏi thêm về cách quản lý nhân sự quy củ, chuyên nghiệp ở trường đại học Công nghiệp Hà Nội từ đó mở mang kiến thức về xã hội hơn.

Chúng em xin chân thành cảm ơn tới cô **Vũ Thị Dương** thạc sĩ trường đại học Công Nghiệp Hà Nội đã nhiệt tình giúp đỡ em trong thời gian thực hiện đề tài để có thể tạo ra được sản phẩm tốt. Chúng em xin chân thành cảm ơn!