**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

**---------------------------------------**

BÁO CÁO THÍ NGHIỆM/THỰC NGHIỆM

LẬP TRÌNH JAVA

**XÂY DỰNG PHẦN MỀM ĐĂNG KÝ MÔN HỌC**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**GVHD: *ThS. Vũ Thị Dương***

**Sinh viên:**

1. Nguyễn Thành Đạt
2. Nguyễn Văn Hậu (NT)
3. Vy Văn Hùng
4. Lê Văn Tuyến

**Nhóm: 12**

**Lớp:** 20212IT6019001

**Khóa:** 14

**Hà Nội – Năm 2022**

MỤC LỤC

[PHẦN 1. MỞ ĐẦU 4](#_Toc105366968)

[1.1 Mục đích 4](#_Toc105366969)

[1.2. Lý do chọn đề tài 4](#_Toc105366970)

[1.3. Các kiến thức cơ bản 5](#_Toc105366971)

[1.4. Các kỹ năng đã có để thực hiện chủ đề nghiên cứu 5](#_Toc105366972)

[PHẦN 2. KẾT QUẢ NGHIÊN CỨU 7](#_Toc105366973)

[2.1. Giới thiệu 7](#_Toc105366974)

[2.2. Khảo sát hệ thống. 7](#_Toc105366975)

[2.2.1. Khảo sát sơ bộ. 7](#_Toc105366976)

[2.2.2. Tài liệu đặc tả yêu cầu. 9](#_Toc105366977)

[2.3. Phân tích hệ thống. 11](#_Toc105366978)

[2.3.1. Mô hình hóa chức năng hệ thống. 11](#_Toc105366979)

[2.3.2. Mô hình hóa dữ liệu hệ thống. 32](#_Toc105366980)

[2.4. Thực hiện bài toán. 35](#_Toc105366981)

[2.4.1. Nguyễn Văn Hậu – Chức năng Quản lý Khoa, Quản lý Lớp Học Phần và Giao diện trang chủ 35](#_Toc105366982)

[2.4.2. Nguyễn Thành Đạt, quản lý giảng viên, quản lý chuyên ngành, quản lý học phần 45](#_Toc105366983)

[2.4.3. Vy Văn Hùng – Quản lí đăng kí học phần – Xem lịch học – Thay đổi mật khẩu 64](#_Toc105366984)

[2.4.4. Lê Văn Tuyến – Chức năng quản lý sinh viên, quản lý lớp, quản lý đăng nhập 80](#_Toc105366985)

[PHẦN 3. KẾT LUẬN VÀ BÀI HỌC KINH NGHIỆM 93](#_Toc105366986)

[3.1. Nội dung đã thực hiện. 93](#_Toc105366987)

[3.2. Hướng phát triển. 93](#_Toc105366988)

# PHẦN 1. MỞ ĐẦU

## Mục đích

Trong thời đại công nghệ thông tin phát triển mạnh mẽ như hiện nay, việc tin học hóa công việc giảng dạy trong các trường đại học là rất cần thiết, từ thiết bị đến cách thức giảng dạy và học tập phải không ngừng được đầu tư, hiện đại. Việc ứng dụng công nghệ thông tin sẽ tiết kiệm thời gian, chi phí cho nhà trường cũng như sinh viên mà kết quả đạt được lại rất cao. Với mục đích tạo ra một hệ thống đăng ký học phần, hỗ trợ tốt công việc quản lý đăng ký học phần của phòng đào tạo cũng như sinh viên trong mỗi học kỳ nên nhóm chúng em chọn đề tài xây dựng phần mềm đăng ký học phần của trường Đại học Công nghiệp Hà Nội.

## Lý do chọn đề tài

Một trong những công đoạn quan trọng là công tác đăng ký học phần cho sinh viên vào mỗi học kỳ. Tất cả các trường học đã áp dụng phương pháp đăng ký học phần trực tuyến để giảm thiểu công việc cho phòng đào tạo, cho sinh viên, giảm chi phí, thời gian cũng như công sức. Chính vì vậy, xây dựng một hệ thống đăng ký học phần là vô cùng cần thiết.

Xây dựng hệ thống đăng ký học phần này giúp chúng em tổng kết lại những kiến thức mình có được trong quá trình học tập và nghiên cứu ở trường, ứng dụng vào xây dựng một hệ thống thực tế.

Hệ thống đăng ký học phần này có thể là một sản phẩm ứng dụng vào thực tế cho bất cứ trường đại học nào có nhu cầu. Từ việc nghiên cứu hệ thống đăng ký học phần, nghiên cứu quy trình, quy chế đăng ký học phần của trường Đại học Công nghiệp Hà Nội cùng với kiến thức chuyên môn mà chúng em có được trong quá trình học tập, chúng em hi vọng hệ thống đăng ký học phần sẽ là một sản phẩm chất lượng và phục vụ tốt chức năng của nó khi đưa vào áp dụng thực tế.

## Các kiến thức cơ bản

Tổng hợp các kiến thức cơ bản cần có để thực hiện xây dựng hệ thống gồm:

* Kiến thức về lập trình hướng đối tượng để thiết kế các đối tượng cần thiết và sử dụng ngôn ngữ lập trình Java để trình bày.
* Kiến thức cơ bản về Java bao gồm: lớp, đối tượng, gom rác, bẫy lỗi, xử lý giao diện, kết nối với cơ sở dữ liệu, …

## Các kỹ năng đã có để thực hiện chủ đề nghiên cứu

Nhóm thực hiện mô tả và xây dựng hệ thống đăng ký học phần cùng với những kiến thức và kỹ năng dưới đây:

Về kiến thức:

* Kiến thức cơ bản lập trình Java: lớp, đối tượng, gom rác, bẫy lỗi, xử lý giao diện, kết nối với cơ sở dữ liệu.
* Thành thạo ngôn ngữ Java và một số công cụ viết mã nguồn Eclipse, Netbeans…, công cụ Microsoft Word làm báo cáo công việc, công cụ IBM Rational Rose để thiết kế phần mềm.
* Ngoài việc sử dụng môn “Lập trình Java” còn sử dụng các kiến thức ở môn “Phân tích thiết kế hệ thống”, Thiết kế cơ sở dữ liệu” để khảo sát, mô tả, phân tích, thiết kế cơ sở dữ liệu và giao diện.
* Áp dụng môn học “Quản lý dự án” lên kế hoạch chi tiết và triển khai từng công việc để thực hiện đề tài.

Về kỹ năng:

* Các kỹ năng làm việc nhóm, thuyết trình, tổng hợp kiến thức, lập bảng biểu thực hiện công việc.

Bài tập lớn được hoàn thành từ các kiến thức, kỹ năng có được ở trên và là triển khai của mô hình xây dựng bản mẫu. Xây dựng phần mềm theo mô hình thu thập bản mẫu là cách tiếp cận thực tế nhất, phù hợp với hệ thống vừa và nhỏ, đặc biệt khi chưa có yêu cầu rõ ràng như hệ thống đăng ký học phần mà nhóm thực hiện. Đồng thời, từ những mẫu thử ban đầu và đưa cho khách hàng, qua quá trình tinh chỉnh mẫu thử qua nhiều phiên bản nhóm chúng em tin rằng sẽ làm thỏa mãn yêu cầu của người sử dụng và tạo ra được kết quả tốt nhất.

# PHẦN 2. KẾT QUẢ NGHIÊN CỨU

## Giới thiệu

Tên bài toán sẽ xây dựng: Xây dựng phần mềm đăng ký môn học Trường đại học Công nghiệp Hà Nội.

Quy trình áp dụng triển khai bài tập lớn: Xây dựng phần mềm theo mô hình thu thập bản mẫu.

Thực hiện quy trình gồm 6 bước:

+ Bước 1: Thu thập và phân tích yêu cầu phần mềm thông qua nhiểu hình thức (Khảo sát, phỏng vấn).

+ Bước 2: Thiết kế nhanh: phân tích các chức năng thành các use case.

+ Bước 3: Xây dựng thực hiện bài toán.

+ Bước 4: Chuyển giao kết quả cho khách hàng và nhận ý kiến đánh giá của khách hàng.

+ Bước 5: Nâng cấp, chỉnh sửa, bổ sung các chức năng.

+ Bước 6: Bàn giao cho khách hàng sản phẩm cuối cùng.

Hình thức sản phẩm: Sản phẩm ứng dụng.

Kết quả đạt được: Cài đặt và triển khai thành công phần mềm.

## Khảo sát hệ thống.

### 2.2.1. Khảo sát sơ bộ.

Đối tượng: Trường đại học Công nghiệp Hà Nội.

Khi một sinh viên đăng ký, thay đổi hoặc hủy môn học thì thông tin sẽ được lưu tại phòng đào tạo theo khuôn mẫu. Vì số lượng sinh viên rất lớn nên dẫn đến việc sai sót thông tin có thể xảy ra rất cao, việc làm khá thủ công cần một lượng cán bộ nhân sự tương đối lớn của trường, đồng thời cũng tốn rất nhiều thời gian và công sức của sinh viên trong mỗi lần đăng ký học phần.

Trước khi tiến hành khảo sát sơ bộ, chúng tôi xác định mục tiêu đạt được là hiểu được cách đăng ký môn học, cách quản lý thông tin môn học, thông tin sinh viên.

Để kết quả của quá trình khảo sát được chính xác và khách quan, chúng tôi quyết định khảo sát bằng phương pháp phỏng vấn trực tiếp bộ phận quản lý thông qua phiếu phỏng vấn dưới đây:

|  |  |
| --- | --- |
| Kế Hoạch Phỏng Vấn | |
| Người được phỏng vấn:  Đại diện trường Đại học Công Nghiệp Hà Nội. | Người phỏng vấn: |
| Địa chỉ: Cơ sở 1 trường Đại học Công Nghiệp Hà Nội. | Thời gian hẹn: 24/04/2022  Thời gian bắt đầu: 8h  Thời gian kết thúc: 8h50p |
| Cần thu thập thông tin về:  - Thông tin sinh viên  - Thông tin môn học | Trình độ: Hiểu rõ về quản lý sinh viên và môn học |
| Nội dung:  Giới thiệu tổng quan về dự án  Câu hỏi 1: Nhà trường có yêu cầu gì về hệ thống đăng kí học phần?  Câu hỏi 2: Hệ thống cần làm có những tính năng gì đặc biệt ngoài những tính năng cơ bản như: Đăng nhập, đăng kí môn học hay không?  Tổng hợp các nội dung chính ý kiến của người được hỏi  Kết thúc (thỏa thuận) | Dự kiến thời gian:  10 phút  10 phút  10 phút  20 phút   * Dự kiến tổng cộng 50 phút |

### 2.2.2. Tài liệu đặc tả yêu cầu.

**a. Mô tả hoạt động của hệ thống**

Quy trình hoạt động của việc đăng ký học phần bắt đầu bằng việc phòng đào tạo sẽ lấy thông tin về các môn học (thường là mã môn học, tên môn học, số tín chỉ, học phí) và sinh viên (mã sinh viên, tên sinh viên, lớp, khoa). Sau khi nhập hết thông tin môn học và sinh viên vào sổ sách thì việc nhập thông tin kết thúc. Thường sẽ có rất nhiều môn học, sinh viên và được chia ra làm nhiều khoa nên sẽ có thể có sai sót.

Khi sinh viên đến đăng ký học phần sẽ phải đến phòng đào tạo xác minh mình là sinh viên trường và thuộc lớp nào, khoa nào. Sau khi xác minh thì thực hiện đăng ký môn học, phòng đào tạo sẽ lưu lại thông tin sinh viên vào lớp dạy môn học đó.

Phòng đào tạo sẽ phải kiểm soát số lượng đăng ký vào lớp (dễ sai sót).

Sau khi hết thời gian đăng ký môn học, Phòng Đào Tạo sẽ thống kê lại số lượng lớp đủ điều kiện để mở, số lượng sinh viên trong lớp và tính ra học phí của sinh viên (không được nợ quá 10 triệu).

Công việc quản lý đăng ký môn học có rất nhiều vấn đề vì vậy việc quản lý bằng sổ sách có rất nhiều nhược điểm:

- Tốn thời gian và công sức khi thống kê.

- Quản lý rất khó khăn.

- Cần số lượng lớn nhân sự để có thể hoàn thành được sổ sách.

**b. Yêu cầu chức năng**

Phần mềm đòi hỏi phải khắc phục được các hạn chế trên và đảm bảo công việc quản lý quá trình đăng ký học phần của sinh viên với những chức năng sau đây. Sinh viên sau khi đăng nhập vào hệ thống phần mềm bằng tài khoản sinh viên sẽ thực hiện được những chức năng:

- Đăng ký học phần.

- Hủy học phần đã đăng ký.

- Xem thông tin đăng ký học phần.

Phòng đào tạo sau khi đăng nhập vào hệ thống phần mềm sẽ quản lý việc đăng ký học phần của sinh viên bằng các chức năng sau đây:

- Quản lý khoa, chuyên ngành, lớp của sinh viên.

- Quản lý sinh viên.

- Quản lý giảng viên.

- Quản lý học phần.

- Quản lý danh sách các lớp học phần để sinh viên chuẩn bị đăng ký.

**c. Yêu cầu phi chức năng**

- Về yêu cầu xử lý:

+ Phần mềm dễ sử dụng, truy cập nhanh, chính xác, khả năng hoạt động ổn định.

+ Đáp ứng đầy đủ nhu cầu của người sử dụng.

- Về giao diện người sử dụng:

+ Các chức năng phải được sắp xếp hợp lý, logic.

+ Tương thích với cấu hình điện thoại và máy tính.

- Về khả năng thích nghi: Chạy được trên các phiên bản windows thông dụng.

- Về tính tái sử dụng: Phần mềm có thể được nâng cấp dễ dàng.

## Phân tích hệ thống.

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

|  |  |
| --- | --- |
| Actor | Mô tả |
| Sinh viên | Là người đăng ký, hủy và kiểm tra học phần đã đăng ký của mình. |
| Người quản trị | Là người quản lý khoa, chuyên ngành, lớp, giảng viên, sinh viên, học phần và danh sách lớp học phần được mở cho sinh viên đăng ký. |

Bảng 1. Mô tả tổng quát chức năng người dùng

Diagram

Description automatically generated

*Hình 2.3.1.1. Biểu đồ use case*

**Mô tả vắn tắt các use case**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Use Case** | **Mô tả Use Case** | **Ghi chú** |
| 1 | Đăng nhập | Cho phép sinh viên và quản trị viên đăng nhập vào hệ thống. | Sinh viên, người quản trị |
| 2 | Đăng ký học phần | Cho phép sinh viên đăng ký, hủy đăng ký và xem lịch học các lớp học phần đã đăng ký. | Sinh viên |
| 3 | Xem lịch học | Cho phép sinh viên xem lịch học | Sinh viên |
| 4 | Quản lý sinh viên | Cho phép người quản trị quản lý thông tin sinh viên bằng cách thêm, sửa, xóa sinh viên. | Người quản trị |
| 5 | Quản lý giảng viên | Cho phép người quản trị xem, thêm, sửa, xóa danh sách giảng viên trong danh sách giảng viên. | Người quản trị |
| 6 | Quản lý học phần | Cho phép quản trị viên xem, thêm, sửa xóa, tìm kiếm danh sách học phần. | Người quản trị |
| 7 | Quản lý lớp học phần | Cho phép quản trị viên xem, thêm, sửa xóa, tìm kiếm, làm mới danh sách các lớp học phần mở để sinh viên đăng ký. | Người quản trị |
| 8 | Quản lý lớp | Cho phép người quản trị xem, thêm, sửa, xóa danh sách lớp học trong danh sách lớp. | Người quản trị |
| 9 | Quản lý khoa | Cho phép người quản trị xem, thêm, sửa, xóa danh sách khoa trong danh sách khoa. | Người quản trị |
| 10 | Quản lý chuyên ngành | Cho phép người quản trị xem, thêm, sửa, xóa danh sách chuyên ngành trong danh sách chuyên ngành. | Người quản tri |

Bảng 2. Bảng mô tả chi tiết use case

**Mô tả chi tiết các use case**

1. **Use case Quản lý lớp học phần**

* Tên use case

Quản lý lớp học phần

* Mô tả vắn tắt

Use case này cho phép quản trị viên xem, thêm, sửa xóa, tìm kiếm danh sách các lớp học phần mở để sinh viên đăng ký.

* Luồng sự kiện
* Luồng cơ bản

1. Use case này bắt đầu khi quản trị viên nhấn nút “Quản lý lớp học phần” trên menu chính. Hệ thống lấy thông tin chi tiết từ danh sách lớp học phần trong CSDL và hiển thị lên màn hình cùng với các lựa chọn thêm, sửa, xóa, tìm kiếm lớp học phần.
2. Thêm lớp học phần: Người quản trị nhập thông tin của lớp học phần muốn thêm và nhấn nút “Thêm”, hệ thống kiểm tra tính hợp lệ của thông tin người quản trị cung cấp và tạo ra một bản ghi mới trong danh sách lớp học phần trong CSDL. Hệ thống hiển thị lại danh sách lớp học phần sau khi cập nhật.
3. Sửa lớp học phần: Người quản trị chọn dòng thông tin trong bảng dữ liệu muốn sửa, nhập lại thông tin mới. Người quản trị nhấn nút “Sửa”, hệ thống kiểm tra tính hợp lệ của thông tin người quản trị cung cấp và sửa bản ghi trong danh sách lớp học phần trong CSDL. Hệ thống hiển thị lại danh sách lớp học phần sau khi cập nhật.
4. Xóa lớp học phần: Người quản trị điền thông tin mã lớp học phần cần xóa hoặc chọn dòng thông tin trong bảng dữ liệu muốn xóa, hệ thống sẽ hiển thị lên một thông báo để người quản trị xác nhận lại. Người quản trị nhấn vào nút “Đồng ý” , hệ thống sẽ xóa lớp học phần trong danh sách lớp học phần trong CSDL và hiển thị lại danh sách lớp học phần sau khi cập nhật.
5. Tìm kiếm học phần: Người quản trị điền mã học phần vào ô tìm kiếm và bấm nút “Tìm kiếm”. Hệ thống kiểm tra danh sách lớp học phần trong CSDL và hiển thị lên màn hình thông tin lớp học phần đó.Làm mới: Người quản trị nhấn vào nút “làm mới” thì hệ thống sẽ xóa trắng các ô nhập dữ liệu.

* Luồng rẽ nhánh

1. Tại bước 2, 3 trong luồng cơ bản, nếu trong quá trình thêm hoặc sửa thông tin lớp học phần mà người quản trị nhập thông tin lớp học phần không hợp lệ thì hệ thống sẽ hiển thị thông báo lỗi yêu cầu người quản trị nhập lại.
2. Tại bước 4 trong luồng cơ bản, người quản trị nhấn vào nút “Hủy” thì hệ thống sẽ bỏ thao tác xóa và hiển thị lại danh sách các lớp học phần có trong danh sách lớp học phần.
3. Tại bất kỳ thời điểm nào trong quá trình use case, nếu không kết nối được với CSDL thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

* Các yêu cầu đặc biệt

Use case chỉ cho phép người quản trị hệ thống thực hiện.

* Tiền điều kiện

Người quản trị phải đăng nhập vào hệ thống.

* Hậu điều kiện

Thông tin của tài khoản được lưu vào danh sách lớp học phần.

* Điểm mở rộng

Không.

1. **Use case Đăng ký học phần**

* Tên use case

Đăng ký học phần.

* Mô tả vắn tắt

Use case này cho phép sinh viên đăng ký, hủy đăng ký và xem lịch học các lớp học phần đã đăng ký.

* Luồng sự kiện
  + Luồng cơ bản

Use case này bắt đầu khi sinh viên nhấn nút “Đăng ký học phần” trên menu chính. Hệ thống lấy thông tin chi tiết của từng lớp học phần từ danh sách lớp học phần trong CSDL và hiển thị lên màn hình cùng với các lựa chọn đăng ký học phần, hủy đăng ký học phần và xem lịch học.

Đăng ký học phần: Sinh viên chọn học phần muốn đăng ký trong danh sách lớp học phần hiển thị trên màn hình và nhấn nút “Đăng ký”. Hệ thống kiểm tra tính hợp lệ của lớp học phần này và thông báo “Đăng ký thành công”, thông tin đăng ký sẽ được lưu vào danh sách đăng ký.

Hủy học phần: Sinh viên chọn học phần muốn hủy đăng ký trong danh sách lớp học phần hiển thị trên màn hình và nhấn nút “Hủy đăng ký”, hệ thống kiểm tra sinh viên đăng ký lớp học phần này trong danh sách đăng ký. Nếu có, hệ thống thông báo “Hủy đăng ký thành công”, thông tin đăng ký sẽ được xóa trong danh sách đăng ký.

Xem lịch học: Sinh viên nhấn nút “Xem lịch học”, hệ thống sẽ lấy toàn bộ thông tin các lớp học phần mà sinh viên đã đăng ký từ danh sách đăng ký và hiển thị lên màn hình.

* + Luồng rẽ nhánh

1. Tại bước 2 trong luồng cơ bản, nếu lớp học phần muốn đăng ký trùng lịch (ngày học, tiết học) với lớp học phần đã đăng ký trước đó, lớp học phần có mã học phần trùng với mã học phần đã đăng ký, lớp học phần muốn đăng ký đã đủ 60 người, tổng số tiền nợ quá 10 triệu, vi phạm 1 trong 4 điều trên thì hệ thống sẽ thông báo không đăng ký thành công.
2. Tại bước 3 trong luồng cơ bản, nếu lớp học phần chưa được sinh viên đăng ký mà sinh viên nhấn vào nút “Hủy đăng ký” thì hệ thống sẽ thông báo sinh viên chưa đăng ký học phần này.
3. Tại bất kỳ thời điểm nào trong quá trình use case nếu không kết nối được với CSDL thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

* Các yêu cầu đặc biệt

Use case chỉ cho phép sinh viên thực hiện.

* Tiền điều kiện

Sinh viên phải đăng nhập vào hệ thống.

* Hậu điều kiện

Thông tin đăng ký được lưu vào danh sách đăng ký.

* Điểm mở rộng

Không.

1. **Use case Quản lý sinh viên**

* Tên use case

Quản lý sinh viên.

* Mô tả vắn tắt

Use case này cho phép người quản trị quản lý thông tin sinh viên bằng cách thêm, sửa, xóa sinh viên.

* Luồng sự kiện
  + Luồng cơ bản

Use case này bắt đầu khi người quản trị nhấn nút “Quản lý sinh viên” trên menu chính. Hệ thống lấy thông tin chi tiết của sinh viên từ danh sách sinh viên trong CSDL và hiển thị lên màn hình cùng với các lựa chọn thêm sinh viên, sửa sinh viên, xóa sinh viên.

Thêm sinh viên: Người quản trị nhập đầy đủ thông tin sinh viên cần thêm và nhấn nút “Thêm”. Hệ thống kiểm tra tính hợp lệ của thông tin người quản trị cung cấp và tạo ra một bản ghi mới trong danh sách sinh viên trong CSDL và hiển thị lại danh sách sinh viên sau khi thêm.

Sửa sinh viên: Người quản trị chọn vào dòng thông tin sinh viên muốn sửa trong bảng sinh viên hiển thị trên màn hình, nhập lại thông tin của sinh viên và nhấn nút “Sửa”. Hệ thống cập nhật lại thông tin của sinh viên vừa sửa vào danh sách sinh viên trong CSDL và hiển thị lại danh sách sinh viên sau khi sửa.

Xóa sinh viên: Người quản trị chọn vào dòng thông tin sinh viên muốn xóa trong bảng sinh viên hiển thị trên màn hình và nhấn nút “Xóa”, hệ thống hiển thị thông báo xác nhận xóa sinh viên. Người quản trị nhấn nút “Đồng ý”, hệ thống xóa thông tin của sinh viên trong danh sách sinh viên trong CSDL và hiển thị lại danh sách sinh viên sau khi xóa.

* + Luồng rẽ nhánh

1. Tại bước 2 trong luồng cơ bản, nếu người quản trị chưa nhập đầy đủ thông tin sinh viên hoặc thông tin nhập không hợp lệ mà nhấn nút “Thêm” thì hệ thống sẽ gửi một thông báo yêu cầu nhập lại thông tin của sinh viên trước khi thêm.
2. Tại bước 3 trong luồng cơ bản, nếu người quản trị chưa nhập đầy đủ thông tin sinh viên hoặc thông tin nhập không hợp lệ mà nhấn nút “Sửa” thì hệ thống sẽ gửi một thông báo yêu cầu nhập lại thông tin của sinh viên trước khi sửa.
3. Tại bước 4 trong luồng cơ bản, hệ thống hiển thị thông báo xác nhận xóa sinh viên, người quản trị nhấn nút “Hủy bỏ”, thao tác xóa sinh viên không được thực hiện và hệ thống hiển thị lại danh sách sinh viên có trong danh sách sinh viên trong CSDL.
4. Tại bất kỳ thời điểm nào trong quá trình use case nếu không kết nối được với CSDL thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

* Các yêu cầu đặc biệt

Use case chỉ cho phép người quản trị hệ thống thực hiện.

* Tiền điều kiện

Người quản trị phải đăng nhập vào hệ thống.

* Hậu điều kiện

Thông tin sinh viên được lưu vào danh sách sinh viên.

* Điểm mở rộng

Không.

1. **Use case Xem lịch học**

* Tên use case

Xem lịch học

* Mô tả vắn tắt

Use case này cho phép sinh viên xem lịch học.

* Luồng sự kiện
* Luồng cơ bản

1. Use case này bắt đầu khi sinh viên nhấn nút “Xem lịch học” trên menu chính. Hệ thống sẽ lấy thông tin của lớp học phần sinh viên đã đăng ký từ danh sách đăng ký và danh sách lớp học phần trong CSDL rồi hiển thị lên màn hình.

* Luồng rẽ nhánh

1. Tại bất kỳ thời điểm nào trong quá trình use case, nếu không kết nối được với CSDL thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

* Các yêu cầu đặc biệt

Không.

* Tiền điều kiện

Sinh viên phải đăng nhập vào hệ thống bằng tài khoản của mình.

* Hậu điều kiện

Không.

* Điểm mở rộng

Không.

1. **Use case Quản lý học phần**

* Tên use case

Quản lý học phần

* Mô tả vắn tắt

Use case này cho phép quản trị viên xem, thêm, sửa xóa, tìm kiếm danh sách các học phần.

* Luồng sự kiện
* Luồng cơ bản

1. Use case này bắt đầu khi quản trị viên nhấn nút “Quản lý học phần” trên menu chính. Hệ thống lấy thông tin chi tiết từ danh sách học phần trong CSDL và hiển thị lên màn hình cùng với các lựa chọn thêm, sửa, xóa học phần.
2. Thêm học phần: Người quản trị nhập thông tin của học phần muốn thêm và nhấn nút “Thêm”, hệ thống kiểm tra tính hợp lệ của thông tin người quản trị cung cấp và tạo ra một bản ghi mới trong danh sách học phần trong CSDL. Hệ thống hiển thị lại danh sách học phần sau khi cập nhật.
3. Sửa học phần: Người quản trị chọn dòng thông tin trong bảng dữ liệu muốn sửa, nhập lại thông tin mới. Nếu người quản trị không chọn dòng trong bảng dữ liệu thì chỉ có thể sửa thông tin cho những mã học phần đã có. Người quản trị nhấn nút “Sửa”, hệ thống kiểm tra tính hợp lệ của thông tin người quản trị cung cấp và sửa bản ghi trong danh sách học phần trong CSDL. Hệ thống hiển thị lại danh sách học phần sau khi cập nhật.
4. Xóa học phần: Người quản trị điền thông tin mã học phần cần xóa hoặc chọn dòng thông tin trong bảng dữ liệu muốn xóa, hệ thống sẽ hiển thị lên một thông báo để người quản trị xác nhận lại. Người quản trị nhấn vào nút “Đồng ý” , hệ thống sẽ xóa học phần trong danh sách học phần trong CSDL và hiển thị lại danh sách học phần sau khi cập nhật.

* Luồng rẽ nhánh

1. Tại bước 2 trong luồng cơ bản, nếu trong quá trình thêm thông tin học phần mà người quản trị nhập thông tin học phần không hợp lệ hoặc nhập thiếu thông tin của học phần thì hệ thống sẽ hiển thị thông báo lỗi yêu cầu người quản trị nhập lại.
2. Tại bước 3 trong luồng cơ bản, nếu người quản trị nhập mã học phần không tồn tại trong danh sách học phần thì hệ thống sẽ hiển thị thông báo lỗi và yêu cầu nhập lại.
3. Tại bước 4 trong luồng cơ bản, người quản trị nhấn vào nút “Hủy” thì hệ thống sẽ bỏ thao tác xóa và hiển thị lại danh sách các học phần có trong danh sách học phần.
4. Tại bất kỳ thời điểm nào trong quá trình use case, nếu không kết nối được với CSDL thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

* Các yêu cầu đặc biệt

Use case chỉ cho phép người quản trị hệ thống thực hiện.

* Tiền điều kiện

Người quản trị phải đăng nhập vào hệ thống.

* Hậu điều kiện

Thông tin của học phần được lưu vào danh sách học phần.

* Điểm mở rộng

Không.

1. **Use case Đăng nhập**

* Tên use case

Đăng nhập

* Mô tả vắn tắt

Use case này cho phép sinh viên và quản trị viên đăng nhập vào hệ thống.

* Luồng sự kiện:
* Luồng cơ bản

1. Use case này bắt đầu khi người dùng điền thông tin đăng nhập và nhấn vào nút “đăng nhập”. Hệ thống xác thực thông tin đăng nhập từ danh sách tài khoản trong CSDL và cho phép truy cập vào ứng dụng.

* Luồng rẽ nhánh

1. Tại bước 1 trong luồng cơ bản, nếu người dùng nhập thông tin sai hoặc không hợp lệ thì hệ thống sẽ hiển thị thông báo lỗi yêu cầu nhập lại
2. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với CSDL thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc

Use case kết thúc.

* Các yêu cầu đặc biệt

Không có.

* Tiền điều kiện

Tài khoản phải được cung cấp

Thiết bị của người dùng phải kết nối với cơ sở dữ liệu

* Hậu điều kiện

Không có.

* Điểm mở rộng

Không có.

1. **Use case Quản lý giảng viên**

* Tên use case

Quản lý giảng viên

* Mô tả vắn tắt

Use case này cho phép người quản trị xem, thêm, sửa, xóa danh sách giảng viên trong danh sách giảng viên.

* Luồng sự kiện:
  + Luồng cơ bản:

1. Use case này bắt đầu khi người quản trị nhấn vào nút “Quản Lý Giáo Viên” trên menu chính. Hệ thống lấy chi tiết các thông tin chi tiết các giảng viên từ danh sách giảng viên trong CSDL và hiển thị lên cùng với các lựa chọn thêm giảng viên, sửa giảng viên, xóa giảng viên, làm mới.
2. Thêm giảng viên:

Người quản trị nhập thông tin giảng viên mới và nhấn vào nút “thêm giảng viên”. Hệ thống sẽ cập nhập danh sách giảng viên mới, hiển thị và lưu vào CSDL.

1. Sửa giảng viên:

Người quản trị nhấn chọn dòng thông tin trong danh sách giảng viên muốn sửa và nhập lại thông tin giảng viên. Người quản trị nhấn vào nút “sửa”, hệ thống cập nhập lại danh sách giảng viên và hiển thị.

1. Xóa giảng viên:

Người quản trị nhấn chọn trong danh sách giảng viên và nhấn vào nút “xóa”. Hệ thống sẽ hiển thị thông báo xác nhận xóa. Nhấn nút “đồng ý” thì hệ thống sẽ xóa giảng viên, cập nhập lại danh sách hiển thị.

1. Làm mới

Người quản trị nhấn vào nút “làm mới” thì hệ thống sẽ xóa trắng các ô dữ liệu đang nhập.

* + Luồng rẽ nhánh:

1. Tại bước 2 và 3 trong luồng cơ bản nếu người quản trị nhập thông tin giảng viên không hợp lệ thì hệ thống sẽ hiển thị thông báo lỗi yêu cầu nhập lại. Người quản trị có thể nhập lại để tiếp tục.
2. Tại bước 4 trong luồng cơ bản người quản trị kích vào nút “không đồng ý”, hệ thống sẽ bỏ qua thao tác xóa và hiển thị danh sách giảng viên ban đầu
3. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case này nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi.

Use case kết thúc.

* Các yêu cầu đặc biệt:

Use case này chỉ cho phép người quản trị hệ thống thực hiện.

* Tiền điều kiện:

Người quản trị cần đăng nhập với vai trò quản trị hệ thống trước khi có thể thực hiện Use case.

* Hậu điều kiện:

Nếu Use case kết thúc thành công thì thông tin giảng viên sẽ cập nhật trong cơ sở dữ liệu.

* Điểm mở rộng:

Không có.

1. **Mô tả use case Quản Lý Khoa**

* Tên use case

Quản lý khoa.

* Mô tả vắn tắt

Use case này cho phép người quản trị xem, thêm, sửa, xóa danh sách khoa trong danh sách khoa.

* Luồng sự kiện:
  + Luồng cơ bản:

1. Use case này bắt đầu khi người quản trị nhấn vào nút “Quản Lý Khoa” trên menu chính. Hệ thống lấy chi tiết các thông tin chi tiết của khoa từ danh sách khoa trong CSDL để hiển thị cùng với các lựa chọn thêm, sửa, xóa khoa và làm mới.
2. Thêm Khoa:

Người quản trị nhập thông tin khoa mới và nhấn vào nút “thêm”. Hệ thống sẽ cập nhập danh sách khoa mới, hiển thị và lưu vào cơ sở dữ liệu.

1. Sửa Khoa:

Người quản trị nhấn chọn dòng thông tin khoa trong danh sách khoa và nhập lại thông tin khoa. Người quản trị nhấn vào nút “sửa”, hệ thống cập nhập lại danh sách khoa và hiển thị thông tin đã sửa lên danh sách.

1. Xóa Khoa:

Người quản trị nhấn chọn dòng thông tin khoa muốn xóa trong danh sách khoa và nhấn vào nút “xóa”. Hệ thống sẽ hiển thị thông báo xác nhận xóa, người quản trị nhấn nút “đồng ý” thì hệ thống sẽ xóa khoa, cập nhập lại danh sách và hiển thị lên danh sách.

1. Làm mới Khoa:

Người quản trị nhấn vào nút “làm mới” thì hệ thống sẽ xóa trắng các ô dữ liệu đang nhập.

Use case kết thúc.

* + Luồng rẽ nhánh:

1. Tại bước 2 và 3 trong luồng cơ bản nếu người quản trị nhập thông tin khoa không hợp lệ thì hệ thống sẽ hiển thị thông báo lỗi yêu cầu nhập lại. Người quản trị có thể nhập lại để tiếp tục
2. Tại bước 4 trong luồng cơ bản người quản trị nhấn vào nút “không đồng ý” hệ thống sẽ bỏ qua thao tác xóa và hiển thị danh sách khoa ban đầu.
3. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case này nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi.

Use case kết thúc.

* Các yêu cầu đặc biệt

Use case này chỉ cho phép người quản trị hệ thống thực hiện.

* Tiền điều kiện

Người quản trị cần đăng nhập với vai trò quản trị hệ thống trước khi có thể thực hiện Use case.

* Hậu điều kiện

Nếu Use case kết thúc thành công thì thông tin khoa sẽ cập nhật trong CSDL.

* Điểm mở rộng

Không có.

1. **Mô tả use case Quản Lý Chuyên Ngành**

* Tên use case

Quản lý chuyên ngành.

* Mô tả vắn tắt

Use case này cho phép người quản trị xem, thêm, sửa, xóa danh sách chuyên ngành trong danh sách chuyên ngành.

* Luồng sự kiện
  + Luồng cơ bản:

1. Use case này bắt đầu khi người quản trị nhấn vào nút “Quản Lý Chuyên Ngành” trên menu chính. Hệ thống lấy chi tiết các thông tin chi tiết các chuyên ngành từ danh sách chuyên ngành trong CSDL để hiển thị cùng với các lựa chọn thêm, sửa, xóa chuyên ngành và làm mới.
2. Thêm Chuyên Ngành:

Người quản trị nhập thông tin chuyên ngành mới và nhấn vào nút “thêm”. Hệ thống sẽ cập nhập danh sách chuyên ngành mới, hiển thị và lưu vào cơ sở dữ liệu.

1. Sửa Chuyên Ngành:

Người quản trị nhấn chọn dòng chuyên ngành muốn sửa trong danh sách chuyên ngành, thông tin sẽ được hiển thị vào ô nhập dữ liệu. Người quản trị thay đổi thông tin đang hiển thị và nhấn vào nút “sửa”. Hệ thống cập nhập lại danh sách chuyên ngành và hiển thị thông tin đã sửa lên danh sách.

1. Xóa Chuyên Ngành:

Người quản trị nhấn chọn dòng thông tin chuyên ngành trong danh sách chuyên ngành muốn xóa và nhấn vào nút “xóa”. Hệ thống sẽ hiển thị thông báo xác nhận xóa. Nhấn nút “đồng ý” thì hệ thống sẽ xóa chuyên ngành, cập nhập lại danh sách và hiển thị lên danh sách.

1. Làm mới:

Người quản trị nhấn vào nút “làm mới” thì hệ thống sẽ xóa trắng các ô dữ liệu đang nhập.

Use case kết thúc.

* + Luồng rẽ nhánh:

1. Tại bước 2 và 3 trong luồng cơ bản nếu người quản trị nhập thông tin chuyên ngành không hợp lệ thì hệ thống sẽ hiển thị thông báo lỗi yêu cầu nhập lại. Người quản trị có thể nhập lại để tiếp tục
2. Tại bước 4 trong luồng cơ bản người quản trị nhấn vào nút “không đồng ý” hệ thống sẽ bỏ qua thao tác xóa và hiển thị danh sách chuyên ngành ban đầu.
3. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case này nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi.

Use case kết thúc.

* Các yêu cầu đặc biệt

Use case này chỉ cho phép người quản trị hệ thống thực hiện.

* Tiền điều kiện

Người quản trị cần đăng nhập với vai trò quản trị hệ thống trước khi có thể thực hiện Use case.

* Hậu điều kiện

Nếu Use case kết thúc thành công thì thông tin chuyên ngành sẽ cập nhật trong CSDL.

* Điểm mở rộng

Không có.

1. **Mô tả use case Quản Lý Lớp**

* Tên use case

Quản lý lớp

* Mô tả vắn tắt

Use case này cho phép người quản trị xem, thêm, sửa, xóa danh sách lớp học trong danh sách lớp.

* Luồng sự kiện
  + Luồng cơ bản:

1. Use case này bắt đầu khi người quản trị nhấn vào nút “Quản Lý Lớp” trên menu chính. Hệ thống lấy chi tiết các thông tin chi tiết các lớp từ danh sách lớp trong CSDL hiển thị cùng các lựa chọn thêm lớp, sửa lớp, xóa lớp, làm mới.
2. Thêm lớp:

Người quản trị nhập thông tin lớp mới và nhấn vào nút “thêm”. Hệ thống sẽ cập nhập danh sách lớp mới, hiển thị và lưu vào cơ sở dữ liệu.

1. Sửa lớp:

Người quản trị nhấn chọn dòng thông tin lớp muốn sửa trong danh sách lớp, thông tin sẽ được hiển thị vào ô nhập dữ liệu. Người quản trị thay đổi thông tin đang hiển thị và nhấn vào nút “sửa”. Hệ thống cập nhập lại danh sách lớp và hiển thị thông tin đã sửa lên danh sách.

1. Xóa lớp:

Người quản trị nhấn chọn trong danh sách lớp và nhấn vào nút “xóa”. Hệ thống sẽ hiển thị thông báo xác nhận xóa. Nhấn nút “đồng ý” thì hệ thống sẽ xóa lớp, cập nhập lại danh sách và hiển thị lên danh sách.

1. Làm mới:

Người quản trị nhấn vào nút “làm mới” thì hệ thống sẽ xóa trắng các ô dữ liệu đang nhập.

Use case kết thúc.

* + Luồng rẽ nhánh:

1. Tại bước 2 và 3 trong luồng cơ bản nếu người quản trị nhập thông tin lớp không hợp lệ thì hệ thống sẽ hiển thị thông báo lỗi yêu cầu nhập lại. Người quản trị có thể nhập lại để tiếp tục
2. Tại bước 4 trong luồng cơ bản người quản trị nhấn vào nút “không đồng ý” hệ thống sẽ bỏ qua thao tác xóa và hiển thị danh sách lớp ban đầu.
3. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case này nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi.

Use case kết thúc.

* Các yêu cầu đặc biệt

Use case này chỉ cho phép người quản trị hệ thống thực hiện.

* Tiền điều kiện

Người quản trị cần đăng nhập với vai trò quản trị hệ thống trước khi có thể thực hiện Use case.

* Hậu điều kiện

Nếu Use case kết thúc thành công thì thông tin lớp sẽ cập nhật trong cơ sở dữ liệu.

* Điểm mở rộng

Không có.

### 2.3.2. Mô hình hóa dữ liệu hệ thống.

**a.** **Thông tin cần lưu trữ**

Từ mô tả hệ thống và các Use Case bên trên, ta có các thông tin cần phải lưu lại trong hệ thống là:

* Thông tin về khoa (use case Quản lý khoa)
* Thông tin về chuyên ngành (use case Quản lý chuyên ngành)
* Thông tin về lớp (use case Quản lý lớp)
* Thông tin về học phần (use case Quản lý học phần)
* Thông tin về giảng viên (use case Quản lý giáo viên)
* Thông tin về sinh viên (use case Quản lý sinh viên)
* Thông tin về lớp học phần (use case Quản lý lớp học phần)
* Thông tin đăng ký học phần của sinh viên (use case Đăng ký học phần)
* Thông tin về tài khoản (use case Quản lý Đổi mật khẩu)

Từ đó, ta có các lớp tương ứng là:

* TAIKHOAN (thể hiện thông tin tài khoản người dùng)
* KHOA (thể hiện thông tin về khoa)
* CHUYENNGANH (thể hiện thông tin về chuyên ngành)
* LOP (thể hiện thông tin về lớp học)
* SINHVIEN (thể hiện thông tin về sinh viên)
* GIANGVIEN (thể hiện thông tin về giảng viên)
* HOCPHAN (thể hiện thông tin về học phần)
* LOPHOCPHAN (thể hiện thông tin về lớp học phần)
* DANGKY (thể hiện thông tin về sự đăng ký học phần của sinh viên)

**b. Thuộc tính của lớp:**

* Lớp TAIKHOAN:

MaTaiKhoan: là thuộc tính khóa mô tả tên đăng nhập của người dùng

MatKhau: là thuộc tính để xác thực người dùng

* Lớp KHOA:

MaKhoa: là thuộc tính khóa, xác định mã khoa

TenKhoa: tên khoa

* Lớp CHUYENNGANH:

MaCN: là thuộc tính khóa của chuyên ngành

TenCN: tên của chuyên ngành

* Lớp LOP:

MaLop: là thuộc tính khóa, xác định mã lớp

TenLop: tên lớp

SiSo: sĩ số lớp

KhoaHoc: khóa học lớp

* Lớp SINHVIEN:

MaSinhVien: là thuộc tính khóa, xác định sinh viên

HoTen: tên sinhvien

NgaySinh: ngày sinh của sinh viên

GioiTinh: giới tính của sinh viên

DiaChi: địa chỉ của sinh viên

HeDaoTao: Hệ đào tạo của sinh viên

* Lớp GIANGVIEN:

MaGiangVien: là thuộc tính khóa, xác định giảng viên

TenGiangVien: tên giảng viên

* Lớp HOCPHAN:

MaHocPhan: là thuộc tính khóa, xác định học phần

TenHocPhan: tên học phần

SoTCLT: số tín chỉ lý thuyết

SoTCTH: số tín chỉ thực hành

DonGiaTinChi: đơn giá một tín chỉ

* Lớp LOPHOCPHAN

MaLopHocPhan: là thuộc tính khóa, xác định lớp học phần

TietDay: tiết học

NgayHoc: ngày học trong tuần

NgayBatDau: ngày bắt đầu học

DiaDiem: địa điểm học

## Thực hiện bài toán.

### Nguyễn Văn Hậu – Chức năng Quản lý Khoa, Quản lý Lớp Học Phần và Giao diện trang chủ admin

* **Cài đặt và thiết kế giao diện trang chủ admin**

Màn hình giao diện



*Hình ảnh 2.4.1.1 Màn hình trang chủ admin*

* Mô tả các đối tượng trên màn hình

+Hiển thị dữ liệu chung về sinh viên, lớp học và lớp học phần về tổng số lượng

**Bắt lỗi:**

+ Bắt lỗi khi đọc ghi file

* **Giao diện và file đã dùng :**

Giao diện Trang chủ dùng dùng các chữ jlabel, các button ,… và sử dụng file

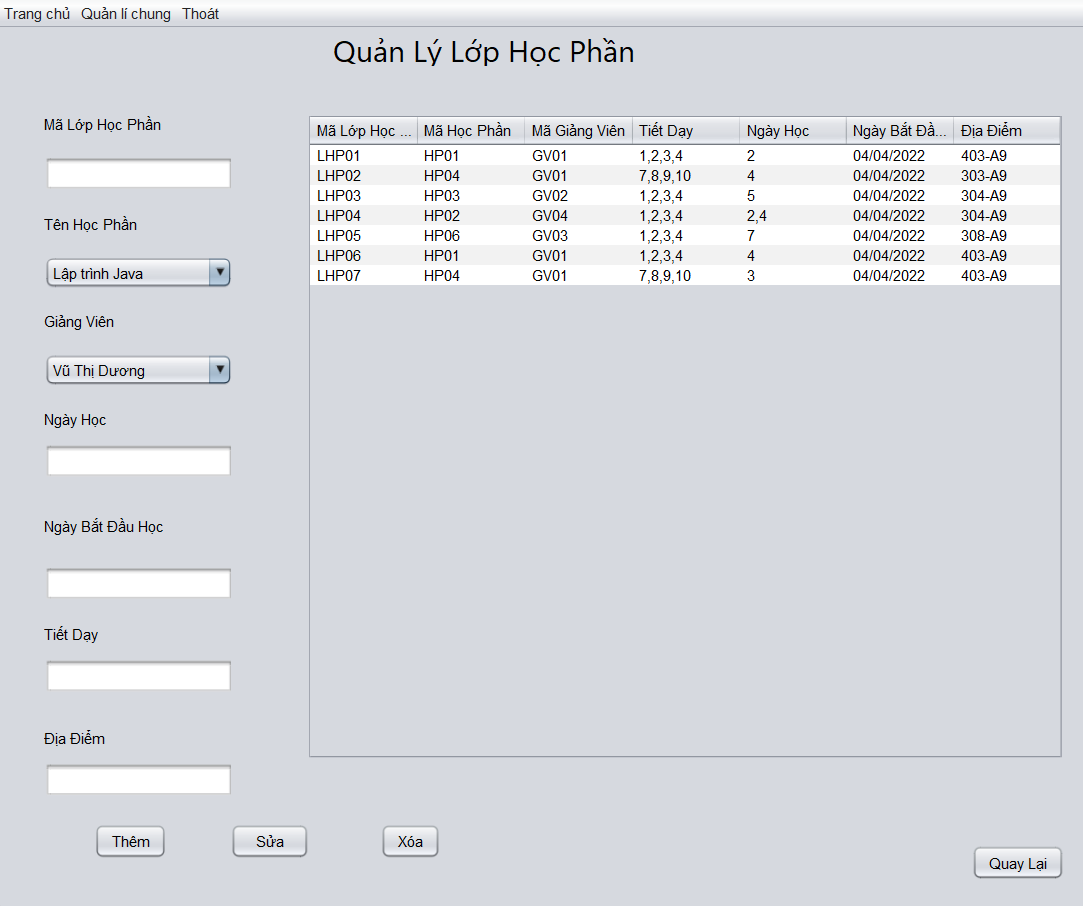
+file Lop.txt

+file SinhVien.txt

+file LopHocPhan.txt

* **Cài đặt và thiết kế giao diện Quản lý Lớp học phần**

Màn hình giao diện



*Hình ảnh 2.4.1.2 Màn hình quản lí lớp học phần*

Mô tả các đối tượng trên màn hình

* Các ô nhập thông tin lớp học phần: mã lớp học phần, tên học phần, giảng viên, tiết dạy, ngày học, ngày bắt đầu học, địa điểm
* Bảng hiển thị danh sách các lớp học phần

Danh sách các Lớp được load từ file LopHocPhan.txt

* Nút Thêm: Người quản trị nhập thông tin lớp học phần muốn thêm vào. Hệ thống sẽ kiểm tra thông tin nhập nếu hợp lệ thì thêm thành công và lưu vào file. Nếu không hợp lệ hoặc trùng mã lớp học phần thì hệ thống sẽ báo lỗi

Code xử lý nút Thêm:

1. private void jButtonThemActionPerformed(java.awt.event.ActionEvent evt) {
2. // TODO add your handling code here:
3. if(jTextMaLHP.getText().trim().equals("") ||jTextTietDay.getText().trim().equals("") ||jTextNgayHoc.getText().trim().equals("") ||jTextNgayBD.getText().trim().equals("") || jTextDD.getText().trim().equals("")){
4. JOptionPane.showMessageDialog(this,"Chua du thong tin de them");
5. }else{
6. if(kiemtra(jTextMaLHP.getText())){
7. lhp = new LopHocPhan(jTextMaLHP.getText(),mahp,magv, jTextTietDay.getText(),jTextNgayHoc.getText(),jTextNgayBD.getText(),jTextDD.getText());
8. dslhp.add(lhp);
9. nhapfileLHP();
10. loadtb();
11. }
12. else{
13. JOptionPane.showMessageDialog(this,"Đã Tồn Tại Mã :"+jTextMaLHP.getText());
14. }
15. }
16. }

* Nút Sửa: Người quản trị chọn thông tin của lớp học phần cần sửa, nhập lại đầy đủ thông tin lớp học phần. Nếu vi phạm những điều kiện về dữ liệu trên thì hệ thống hiển thị thông báo lỗi. Nếu không vi phạm điều kiện dữ liệu nào thì hệ thống thông báo sửa thông tin lớp học phần thành công.

Code xử lý nút Sửa:

1. private void jButtonSuaActionPerformed(java.awt.event.ActionEvent evt) {
2. // TODO add your handling code here:
3. dong = jTableLHP.getSelectedRow();
4. if(dong !=-1){
5. if(jTextMaLHP.getText().trim().equals("") ||jTextTietDay.getText().trim().equals("") ||jTextNgayHoc.getText().trim().equals("") ||jTextNgayBD.getText().trim().equals("") || jTextDD.getText().trim().equals("")){
6. JOptionPane.showMessageDialog(this,"Chua du thong tin de them");
7. }else{
8. if(kiemtra(jTextMaLHP.getText())==false){
9. if(jTextMaLHP.getText().trim().equals(dslhp.get(dong).getMaLopHocPhan().trim())){
10. lhp = new LopHocPhan(jTextMaLHP.getText(),mahp,magv, jTextTietDay.getText(),jTextNgayHoc.getText(),jTextNgayBD.getText(),jTextDD.getText());
11. dslhp.set(dong,lhp);
12. nhapfileLHP();
13. loadtb();
14. }
15. else{
16. JOptionPane.showMessageDialog(this,"Đã có mã: "+jTextMaLHP.getText()+" Ở một dòng khác");
17. }
18. }
19. else{
20. lhp = new LopHocPhan(jTextMaLHP.getText(),mahp,magv, jTextTietDay.getText(),jTextNgayHoc.getText(),jTextNgayBD.getText(),jTextDD.getText());
21. dslhp.set(dong,lhp);
22. nhapfileLHP();
23. loadtb();
24. }
25. }
26. }
27. }

* Nút Xoá: Người quản trị chọn thông tin của lớp học phần cần xóa, hệ thống sẽ thực hiện xoá lớp lớp học phần.

Code xử lý nút Xoá

1. private void jButtonXoaActionPerformed(java.awt.event.ActionEvent evt) {
2. // TODO add your handling code here:
3. dong= jTableLHP.getSelectedRow();
4. if(dong != -1){
5. dslhp.remove(dong);
6. nhapfileLHP();
7. loadtb();
9. }
10. }

* Nút Quay lại: Trở lại menu Quản lý của admin

Code xử lý nút Quay lại:

1. private void jButtonqlActionPerformed(java.awt.event.ActionEvent evt) {
2. // TODO add your handling code here:
3. MenuQuanLyAdmin MN= new MenuQuanLyAdmin();
4. MN.setVisible(true);
5. dispose();
6. }

**Hướng đối tượng :**

+ Chia bài toán thành các đối tượng mỗi đối tượng sẽ có thuộc tính và phương thức, áp dụng các tính các tính chất hướng đối tượng vào bài toán như sau:

+ Tính đóng gói : chia các class có tính chất giống nhau vào các package , kết hợp với các access modifier(public,protected,private) để bảo vệ dữ liệu

+ tính kế thừa : tạo ra class cha có thuộc tính và phương thức chung để các class con kế thừa và sử dụng chung

+ Tính trừu tượng: sử dụng các abstract class(AbstractTableModel) để áp dụng tính trừu tượng, các class extend class abstract sẽ override để cụ thể hóa các logic khác nhau tại mỗi class.

**Bắt lỗi :**

+ bắt lỗi khi đọc ghi file.

**Collection sử dụng trong code:**

Sử dụng các Arraylist để lưu trữ danh sách dữ liệu và sử dụng một số hành động sau

+ lấy dữ liệu đối tượng trong danh sách(get)

+ thêm dữ liệu đối tượng vào danh sách(add)

+ xóa dữ liệu đôi tượng khỏi danh sách(remove)

+ trả về int vị trí của đối tượng trong danh sách(indexof())

+ Kiểm tra đối tượng tồn tại trong array list không(contain).

**Giao diện và file đã dùng:**

Giao diện Lớp học phần dùng dùng các chữ jlabel, các button ,… và sử dụng file

+file LopHocPhan.txt

+file HocPhan.txt

+file GiangVien.txt

* **Cài đặt và thiết kế giao diện Quản lý khoa**

Màn hình giao diện



*Hình ảnh 2.4.1.3 Màn hình quản lí khoa*

* Mô tả các đối tượng trên màn hình
* Các ô nhập dữ liệu của giảng viên: mã khoa, tên khoa.
* Bảng hiển thị danh sách khoa.

Danh sách các Lớp được load từ file Khoa.txt

* Nút thêm: Người quản trị nhập thông tin của khoa cần thêm, thông tin khoa phải nhập đầy đủ, mã khoa không được trùng. Nếu vi phạm những điều kiện về dữ liệu trên thì hệ thống hiển thị thông báo lỗi. Nếu không vi phạm điều kiện dữ liệu nào thì hệ thống sẽ thêm khoa vào và hiển thị thêm thành công.

Code xử lý nút Thêm:

1. private void jButtonThemActionPerformed(java.awt.event.ActionEvent evt) {
2. // TODO add your handling code here:
3. if(jTextMaK.getText().trim().equals("") ||jTextTenK.getText().trim().equals("")){
4. JOptionPane.showMessageDialog(this,"Chua du thong tin de them");
5. }else{
6. if(kiemtra(jTextMaK.getText())){
7. kh = new Khoa(jTextMaK.getText(), jTextTenK.getText());
8. ds.add(kh);
9. nhapfile();
10. loadtb();
11. }
12. else{
13. JOptionPane.showMessageDialog(this,"Đã Tồn Tại Mã :"+jTextMaK.getText());
14. }
15. }
16. }

* Nút xóa: Người quản trị chọn khoa muốn xóa trên bảng rồi ấn xóa hệ thống sẽ xóa khoa được chọn.

Code xử lý nút Xóa :

1. private void jButtonXoaActionPerformed(java.awt.event.ActionEvent evt) {
2. // TODO add your handling code here:
3. dong= jTableKhoa.getSelectedRow();
4. if(dong != -1){
5. ds.remove(dong);
6. nhapfile();
7. loadtb();
8. }
9. }

* Nút sửa: Người quản trị chọn thông tin của khoa cần sửa, nhập lại đầy đủ thông tin khoa. Nếu vi phạm những điều kiện về dữ liệu thì hệ thống hiển thị thông báo. Nếu không vi phạm điều kiện dữ liệu nào thì hệ thống sẽ sửa lại thông tin khoa.

Code xử lý nút Sửa :

1. private void jButtonSuaActionPerformed(java.awt.event.ActionEvent evt) {
2. // TODO add your handling code here:
3. dong = jTableKhoa.getSelectedRow();
4. if(dong !=-1){
5. if(jTextMaK.getText().trim().equals("") ||jTextTenK.getText().trim().equals("")){
6. JOptionPane.showMessageDialog(this,"Chua du thong tin de them");
7. }else{
8. if(kiemtra(jTextMaK.getText())==false){
9. if(jTextMaK.getText().trim().equals(ds.get(dong).getMaKhoa().trim())){
10. kh = new Khoa(jTextMaK.getText(), jTextTenK.getText());
11. ds.set(dong,kh);
12. nhapfile();
13. loadtb();
14. }
15. else{
16. JOptionPane.showMessageDialog(this,"Đã có mã: "+jTextMaK.getText()+" Ở một dòng khác");
17. }
18. }
19. else{
20. kh = new Khoa(jTextMaK.getText(), jTextTenK.getText());
21. ds.set(dong,kh);
22. nhapfile();
23. loadtb();
24. }
25. }
26. }
27. }

* Nút quay lại: cho phép admin quay trở lại menu quản lý.

1. private void jButtonqlActionPerformed(java.awt.event.ActionEvent evt) {
2. // TODO add your handling code here:
3. MenuQuanLyAdmin MN= new MenuQuanLyAdmin();
4. MN.setVisible(true);
5. dispose();
6. }

**Hướng đối tượng :**

+ Chia bài toán thành các đối tượng mỗi đối tượng sẽ có thuộc tính và phương thức, áp dụng các tính các tính chất hướng đối tượng vào bài toán như sau:

+ Tính đóng gói : chia các class có tính chất giống nhau vào các package , kết hợp với các access modifier(public,protected,private) để bảo vệ dữ liệu

+ Tính kế thừa : tạo ra class cha có thuộc tính và phương thức chung để các class con kế thừa và sử dụng chung

+ Tính trừu tượng: sử dụng các abstract class(AbstractTableModel) để áp dụng tính trừu tượng, các class extend class abstract sẽ override để cụ thể hóa các logic khác nhau tại mỗi class.

**Bắt lỗi :**

+ bắt lỗi khi đọc ghi file.

* Collection sử dụng trong code :

Sử dụng các Arraylist để lưu trữ danh sách dữ liệu và sử dụng một số hành động sau

+ lấy dữ liệu đối tượng trong danh sách(get)

+ thêm dữ liệu đối tượng vào danh sách(add)

+ xóa dữ liệu đôi tượng khỏi danh sách(remove)

+ trả về int vị trí của đối tượng trong danh sách(indexof())

+ Kiểm tra đối tượng tồn tại trong array list không(contain).

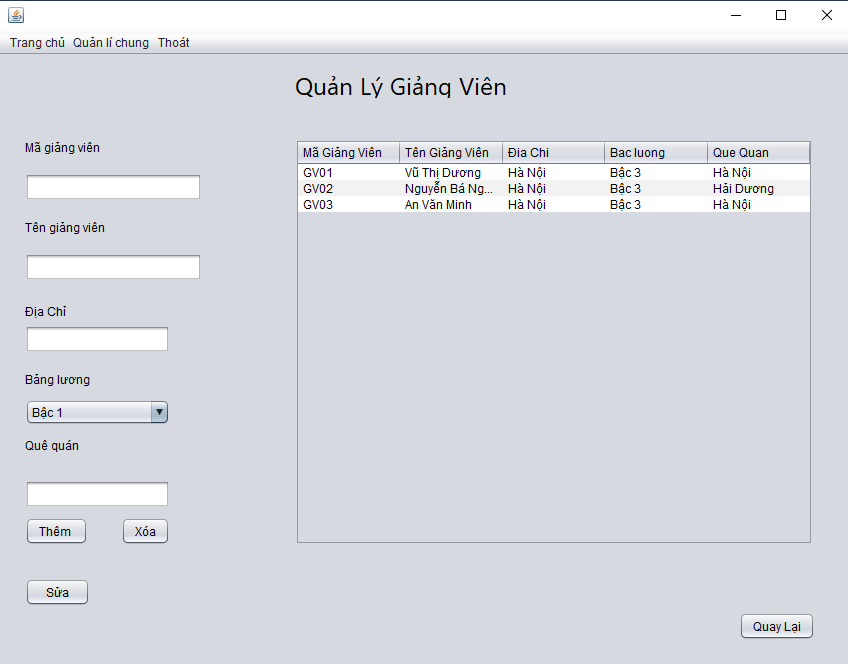
* **Giao diện và file đã dùng :**

Giao diện Khoa dùng dùng các chữ jlabel, các button ,… và sử dụng file

+file Khoa.txt

### **Nguyễn Thành Đạt\_ Quản lý giảng viên, quản lý chuyên ngành, quản lý học phần**

* Quản lý Giảng Viên
* Giao diện quản lý giảng viên



*Hình ảnh 2.4.2.1 Màn hình quản lí giảng viên*

* Mô tả các đối tượng trên màn hình
  + Các ô nhập dữ liệu của giảng viên: mã giảng viên, tên giảng viên, địa chỉ, quê quán.
  + Bảng hiển thị danh sách giảng viên.
  + Nút thêm: Người quản trị nhập thông tin của giảng viên cần thêm thông tin giảng viên phải nhập đầy đủ, mã giảng viên không được trùng. Nếu vi phạm những điều kiện về dữ liệu trên thì hệ thống hiển thị thông báo lỗi. Nếu không vi phạm điều kiện dữ liệu nào thì hệ thống sẽ thêm giảng viên vào và hiển thị thêm thành công.
  + Nút xóa: Người quản trị chọn giảng viên muốn xóa trên bảng rồi ấn xóa hệ thống sẽ xóa giảng viên được chọn.
  + Nút sửa: Người quản trị chọn thông tin của giảng viên cần sửa, nhập lại đầy đủ thông tin giảng viên. Nếu vi phạm những điều kiện về dữ liệu thì hệ thống hiển thị thông báo. Nếu không vi phạm điều kiện dữ liệu nào thì hệ thống sẽ sửa lại thông tin giảng viên.
  + Nút quay lại: cho phép admin quay trở lại menu quản lý.

Code xử lý nút Thêm:

1. private void jButtonThemActionPerformed(java.awt.event.ActionEvent evt) {
2. // TODO add your handling code here:
3. String bacLuong = (String) cbxBacLuong.getSelectedItem();
4. if(jTextMaGV.getText().trim().equals("") ||jTextTenGV.getText().trim().equals("")){
5. JOptionPane.showMessageDialog(this,"Chua du thong tin de them");
6. }else{
7. if(kiemtra(jTextMaGV.getText())){
8. gv = new GiangVien(jTextMaGV.getText(), jTextTenGV.getText(),txtDiaChi.getText(),bacLuong,txtQueQuan.getText());
9. ds.add(gv);
10. nhapfile();
11. loadtb();
12. }
13. else{
14. JOptionPane.showMessageDialog(this,"Đã Tồn Tại Mã :"+jTextMaGV.getText());
15. }
16. }
17. }

Code xử lý nút Xóa:

1. private void jButtonXoaActionPerformed(java.awt.event.ActionEvent evt) {
2. // TODO add your handling code here:
3. dong= jTableGV.getSelectedRow();
4. if(dong != -1){
5. ds.remove(dong);
6. nhapfile();
7. loadtb();
8. }
9. }

Code xử lý nút Sửa:

1. private void jButtonSuaActionPerformed(java.awt.event.ActionEvent evt) {
2. // TODO add your handling code here:
3. dong = jTableGV.getSelectedRow();
4. if(dong !=-1){
5. if(jTextMaGV.getText().trim().equals("") ||jTextTenGV.getText().trim().equals("")||txtDiaChi.getText().equals("")||txtQueQuan.getText().equals("")){
6. JOptionPane.showMessageDialog(this,"Chua du thong tin de them");
7. }else{
8. if(kiemtra(jTextMaGV.getText())==false){
9. if(jTextMaGV.getText().trim().equals(ds.get(dong).getMaGiangVien().trim())){
10. gv = new GiangVien(jTextMaGV.getText(), jTextTenGV.getText(),txtDiaChi.getText(),(String)cbxBacLuong.getSelectedItem(),txtQueQuan.getText());
11. ds.set(dong,gv);
12. nhapfile();
13. loadtb();
14. }
15. else{
16. JOptionPane.showMessageDialog(this,"Đã có mã: "+jTextMaGV.getText()+" Ở một dòng khác");
17. }
18. }
19. else{
20. gv = new GiangVien(jTextMaGV.getText(), jTextTenGV.getText());
21. ds.set(dong,gv);
22. nhapfile();
23. loadtb();
24. }
25. }
26. }
27. }

Code xử lý nút Quay lại:

1. private void jButtonqlActionPerformed(java.awt.event.ActionEvent evt) {
2. // TODO add your handling code here:
3. MenuQuanLyAdmin MN= new MenuQuanLyAdmin();
4. MN.setVisible(true);
5. dispose();
6. }

**Hướng đối tượng**

Chia bài toán thành các đối tượng mỗi đối tượng sẽ có thuộc tính và phương thức, áp dụng các tính các tính chất hướng đối tượng vào bài toán như sau:

+ Tính đóng gói : chia các class có tính chất giống nhau vào các package , kết hợp với các access modifier(public,protected,private) để bảo vệ dữ liệu

+ Tính kế thừa : tạo ra class cha có thuộc tính và phương thức chung để các class con kế thùa và sử dụng chung .

+ Tính đa hình: Tính đa hình thể hiện ở runtime bằng cách override các method ở class cha, và tính đa hình thể hiện ở complie time bằng overloading(1 class có nhiều phương thức trùng tên nhưng khác số lượng tham số hoặc kiểu dữ liệu của tham số)

+ Tính trừu tượng: sử dụng các abstract class(AbstractTableModel) để áp dụng tính trừu tượng, các class extend class abstract sẽ override để cụ thể hóa các logic khác nhau tại mỗi class.

**Collection sử dụng trong code :**

Sử dụng các Arraylist để lưu trữ danh sách dữ liệu và sử dụng một số hành động sau

+ lấy dữ liệu đối tượng trong danh sách(get)

+ thêm dữ liệu đối tượng vào danh sách(add)

+ xóa dữ liệu đôi tượng khỏi danh sách(remove)

+ trả về int vị trí của đối tượng trong danh sách(indexof())

+ Kiểm tra đối tượng tồn tại trong array list không(contain).

static ArrayList<GiangVien> ds = new ArrayList<>();

static GiangVien gv = new GiangVien();

* Giao diện và file đã dùng :

**Giao diện login dùng dùng các chữ jlabel, các button ,… và sử dụng các file**

+ file GiangVien.txt

+file TaiKhoan.txt

**Bắt lỗi**

Bắt lỗi không có file

catch (FileNotFoundException e) {

JOptionPane.showMessageDialog(this, e);

}

Bắt lỗi ngoại lệ Exception

catch (Exception ex) {

JOptionPane.showMessageDialog(this, ex);

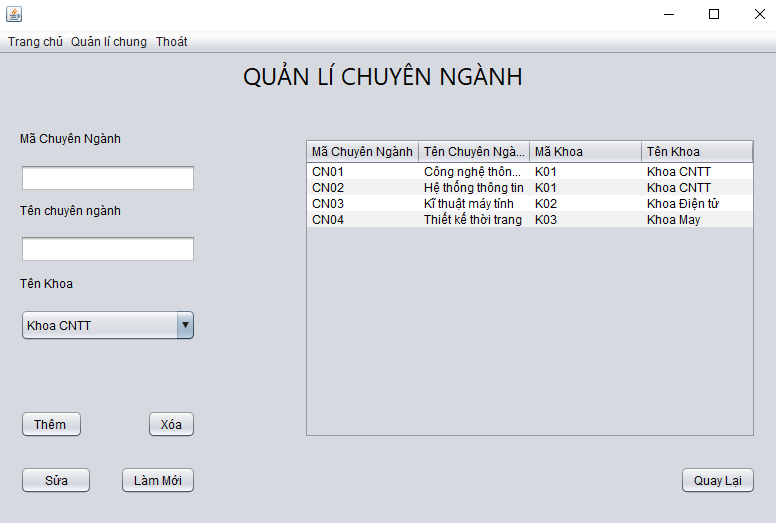
}

1. void loadfie(){
2. try {
3. ds=(ArrayList<GiangVien>) DBE.docFile(fileGV);
4. } catch (Exception ex) {
5. JOptionPane.showMessageDialog(this,ex);
6. System.err.println(ex.getMessage());
7. }
8. }
9. void nhapfile(){
10. try {
11. DBE.LuuFile(fileGV, ds);
12. }catch (FileNotFoundException e) {
13. JOptionPane.showMessageDialog(this, e);
14. } catch (Exception ex) {
15. JOptionPane.showMessageDialog(this,ex);
16. }
17. }

* Mã nguồn chính của chức năng

Code xử lí load data, đọc, ghi file

* Quản lý Chuyên Ngành
* Giao diện quản lý chuyên ngành

*Hình ảnh 2.4.2.2 Màn hình quản lí chuyên ngành*

* Mô tả các đối tượng trên màn hình
  + Các ô nhập thông tinchuyên ngành : nhập mã, tên chuyên ngành.
  + Nút thêm: Người dùng điền thông tincủa môn học. Sau đó nếu muốn them môn học vào danh sách thì ấn nút thêm.
  + Nút xóa: Người dùng nhấn chọn dòng thông tin chuyên ngành trong danh sách chuyên ngánh muốn xóa và nhấn “xóa”. Hệ thống sẽ xóa môn đó trong danh sách.
  + Nút sửa: Người dùng nhấn chọn dòng thông tin chuyên ngành trong danh sách chuyên ngánh muốn sửa. Thông tin môn học sẽ hiện ra ở dưới, người dung nhập thông tin mới của môn và nhấn “sửa”. Hệ thống sẽ xóa môn đó trong danh sách.
  + Nút làm mới: Người dùng ấn vào làm mới thì thông tin đang nhập sẽ tự xóa.
  + Nút quay lại: cho phép admin quay trở lại menu quản lý.

Code nút thêm:

1. private void jButtonThemActionPerformed(java.awt.event.ActionEvent evt) {
2. // TODO add your handling code here:
3. if(jTextMaCN.getText().trim().equals("") ||jTextTenCN.getText().trim().equals("")){
4. JOptionPane.showMessageDialog(this,"Chua du thong tin de them");
5. }else{
6. if(kiemtra(jTextMaCN.getText())){
7. cn = new ChuyenNganh(jTextMaCN.getText(), jTextTenCN.getText(),mak);
8. ds.add(cn);
9. nhapfile();
10. loadtb();
11. }
12. else{
13. JOptionPane.showMessageDialog(this,"Đã Tồn Tại Mã :"+jTextMaCN.getText());
14. }
15. }
16. }

Code nút sửa:

1. private void jButtonSuaActionPerformed(java.awt.event.ActionEvent evt) {
2. // TODO add your handling code here:
3. dong = jTableCN.getSelectedRow();
4. if(dong !=-1){
5. if(jTextMaCN.getText().trim().equals("") ||jTextTenCN.getText().trim().equals("")){
6. JOptionPane.showMessageDialog(this,"Chua du thong tin de them");
7. }else{
8. if(kiemtra(jTextMaCN.getText())==false){
9. if(jTextMaCN.getText().trim().equals(ds.get(dong).getMaCN().trim())){
10. cn = new ChuyenNganh(jTextMaCN.getText(), jTextTenCN.getText(),mak);
11. ds.set(dong,cn);
12. nhapfile();
13. loadtb();
14. }
15. else{
16. JOptionPane.showMessageDialog(this,"Đã có mã: "+jTextMaCN.getText()+" Ở một dòng khác");
17. }
18. }
19. else{
20. cn = new ChuyenNganh(jTextMaCN.getText(), jTextTenCN.getText(),mak);
21. ds.set(dong,cn);
22. nhapfile();
23. loadtb();
24. }
25. }
26. }
27. }

Code nút xóa:

1. private void jButtonXoaActionPerformed(java.awt.event.ActionEvent evt) {
2. // TODO add your handling code here:
3. dong= jTableCN.getSelectedRow();
4. if(dong != -1){
5. ds.remove(dong);
6. nhapfile();
7. loadtb();
8. }
9. }

Code xử lý nút làm mới:

1. private void jButtonLamMoiActionPerformed(java.awt.event.ActionEvent evt) {
2. // TODO add your handling code here:
3. jTextMaCN.setText("");
4. jTextTenCN.setText("");
5. loacb();
6. }

Code nút quay lại:

private void jButtonqlActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

MeNuQuanLy MN= new MeNuQuanLy();

MN.setVisible(true);

dispose();

}

**Collection sử dụng trong code :**

Sử dụng các Arraylist để lưu trữ danh sách dữ liệu và sử dụng một số hành động sau

+ Lấy dữ liệu đối tượng trong danh sách (get)

+ Thêm dữ liệu đối tượng vào danh sách (add)

+ Xóa dữ liệu đối tượng khỏi danh sách (remove)

+ Trả về int vị trí của đối tượng trong danh sách (indexof())

+ Kiểm tra đối tượng tồn tại trong array list không (contain).

1. static ArrayList<ChuyenNganh> ds= new ArrayList<>();
2. static ArrayList<Khoa> dsK= new ArrayList<>();

**Giao diện và file đã dùng :**

Giao diện login dùng dùng các chữ jlabel, các button ,… và sử dụng các file

+ file ChuyenNganh.txt

+file Khoa.txt

**Hướng đối tượng**

Chia bài toán thành các đối tượng mỗi đối tượng sẽ có thuộc tính và phương thức, áp dụng các tính các tính chất hướng đối tượng vào bài toán như sau:

+ Tính đóng gói : chia các class có tính chất giống nhau vào các package , kết hợp với các access modifier(public,protected,private) để bảo vệ dữ liệu

+ tính kế thừa : tạo ra class cha có thuộc tính và phương thức chung để các class con kế thùa và sử dụng chung .ví dụ class thu là class cha , và các class con là thu chi đầu năm ,thu bổ sung, thu khảo

+ Tính đa hình: Tính đa hình thể hiện ở runtime bằng cách override các method ở class cha, và tính đa hình thể hiện ở complie time bằng overloading(1 class có nhiều phương thức trùng tên nhưng khác số lượng tham số hoặc kiểu dữ liệu của tham số)

+ Tính trừu tượng: sử dụng các abstract class(AbstractTableModel) để áp dụng tính trừu tượng, các class extend class abstract sẽ override để cụ thể hóa các logic khác nhau tại mỗi class.

**Bắt lỗi**

Bắt lỗi không có file

catch (FileNotFoundException e) {

JOptionPane.showMessageDialog(this, e);

}

Bắt lỗi ngoại lệ Exception

catch (Exception ex) {

JOptionPane.showMessageDialog(this, ex);

}

1. void loadfieCN(){
2. try {
3. ds=(ArrayList<ChuyenNganh>) DBE.docFile(fileCN);
4. } catch (Exception ex) {
5. JOptionPane.showMessageDialog(this,ex);
6. }
7. }
8. void loadfieK(){
9. try {
10. dsK=(ArrayList<Khoa>) DBE.docFile(fileK);
11. } catch (Exception ex) {
12. JOptionPane.showMessageDialog(this,ex);
13. }
14. }
15. void nhapfile(){
16. try {
17. DBE.LuuFile(fileCN, ds);
18. } catch (Exception ex) {
19. JOptionPane.showMessageDialog(this,ex);
20. }
21. }

**Tập hợp**

Collection được sử dụng :

Lấy dữ liệu đối tượng (getter)

Gán dữ liệu đối tượng (setter)

1. static ArrayList<ChuyenNganh> ds= new ArrayList<>();
2. static ArrayList<Khoa> dsK= new ArrayList<>();ss

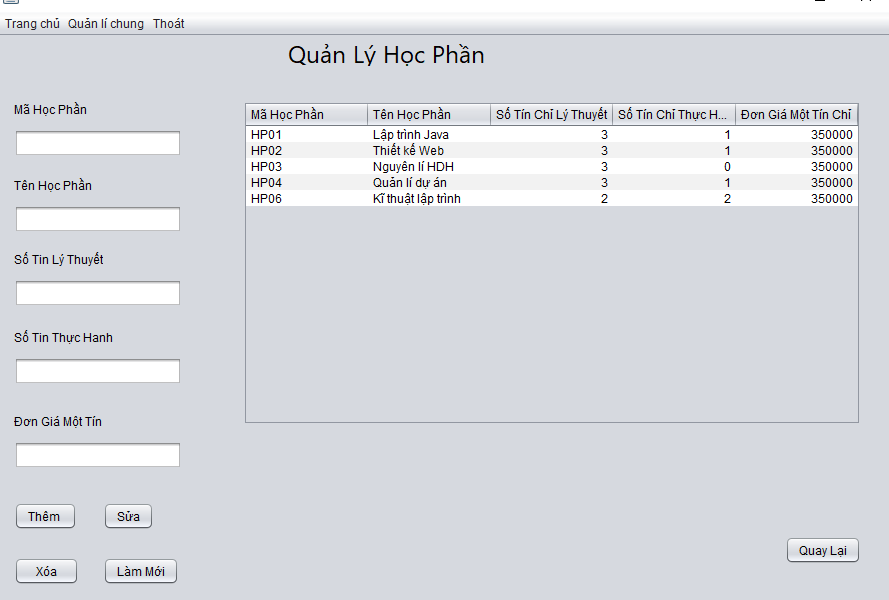
**Thao tác file**

Thao tác với file GiangVien.txt

* Mã nguồn chính của chức năng

Code xử lí load data, đọc, ghi file

* Quản lý học phần
* Giao diện quản lý chuyên ngành



*Hình ảnh 2.4.2.3 Màn hình quản lí học phần*

**Hướng đối tượng**

Chia bài toán thành các đối tượng mỗi đối tượng sẽ có thuộc tính và phương thức, áp dụng các tính các tính chất hướng đối tượng vào bài toán như sau:

+ Tính đóng gói: chia các class có tính chất giống nhau vào các package , kết hợp với các access modifier(public,protected,private) để bảo vệ dữ liệu

+ Tính kế thừa: tạo ra class cha có thuộc tính và phương thức chung để các class con kế thùa và sử dụng chung .

+ Tính đa hình: Tính đa hình thể hiện ở runtime bằng cách override các method ở class cha, và tính đa hình thể hiện ở complie time bằng overloading (1 class có nhiều phương thức trùng tên nhưng khác số lượng tham số hoặc kiểu dữ liệu của tham số)

+ Tính trừu tượng: sử dụng các abstract class (AbstractTableModel) để áp dụng tính trừu tượng, các class extend class abstract sẽ override để cụ thể hóa các logic khác nhau tại mỗi class.

Mô tả các đối tượng trên màn hình

* Các ô nhập dữ liệu của sinh viên: mã sinh viên, họ tên, ngày sinh, giới tính, địa chỉ, hệ đào tạo

Bảng hiển thị danh sách sinh viên

* Nút Thêm: Người quản trị nhập thông tin sinh viên muốn thêm vào. Hệ thống sẽ kiểm tra thông tin nhập nếu hợp lệ thì thêm thành công. Hệ thống sẽ thêm một sinh viên và thêm một tài khoản sinh viên lưu vào cơ sở dữ liệu. Nếu không hợp lệ hoặc trùng mã sinh viên thì hệ thống sẽ báo lỗi.

Code xử lý nút Thêm:

1. private void jButtonThemActionPerformed(java.awt.event.ActionEvent evt) {
2. // TODO add your handling code here:
3. // TODO add your handling code here:
4. if (jTextTenHP.getText().trim().equals("") || jTextMaHP.getText().trim().equals("")
5. || jTextSTHH.getText().trim().equals("") || jTextSoTCLT.getText().trim().equals("") || jTextDG.getText().trim().equals("")) {
6. JOptionPane.showMessageDialog(this, "Chua du thong tin de them");
7. } else {
8. if (kiemtra(jTextMaHP.getText())) {
9. try {
10. hp = new HocPhan(jTextMaHP.getText(), jTextTenHP.getText(), Integer.parseInt(jTextSoTCLT.getText()),
11. Integer.parseInt(jTextSTHH.getText()), Integer.parseInt(jTextDG.getText()));
12. ds.add(hp);
13. nhapfile();
14. loadtb();
15. } catch (NumberFormatException e) {
16. JOptionPane.showMessageDialog(this, "Số Tin Chỉ Và Giá Tiền Là Số Nguyên!!!");
17. } catch (Exception e) {
18. System.out.println(e);
19. }
20. } else {
21. JOptionPane.showMessageDialog(this, "Đã Tồn Tại Mã :" + jTextMaHP.getText());
22. }
23. }
24. }

* Nút Sửa: Người quản trị chọn thông tin của sinh viên cần sửa, nhập lại đầy đủ thông tin sinh viên. Nếu vi phạm những điều kiện về dữ liệu thì hệ thống hiển thị thông báo. Nếu không vi phạm điều kiện dữ liệu nào thì hệ thống sẽ sửa lại thông tin sinh viên. Nếu sửa với một mã sinh viên mới, thì cập nhập lại tài khoản của sinh viên đó.

Code xử lý nút Sửa:

1. private void jButtonSuaActionPerformed(java.awt.event.ActionEvent evt) {
2. // TODO add your handling code here:
3. // TODO add your handling code here:
4. dong = jTableHP.getSelectedRow();
5. if (dong != -1) {
6. if (jTextTenHP.getText().trim().equals("") || jTextMaHP.getText().trim().equals("")
7. || jTextSTHH.getText().trim().equals("") || jTextSoTCLT.getText().trim().equals("") || jTextDG.getText().trim().equals("")) {
8. JOptionPane.showMessageDialog(this, "Chua du thong tin de them");
9. } else {
10. if (kiemtra(jTextMaHP.getText()) == false) {
11. if (jTextMaHP.getText().trim().equals(ds.get(dong).getMaHocPhan().trim())) {
12. try {
13. hp = new HocPhan(jTextMaHP.getText(), jTextTenHP.getText(),
14. Integer.parseInt(jTextSoTCLT.getText()), Integer.parseInt(jTextSTHH.getText()), Integer.parseInt(jTextDG.getText()));
15. ds.set(dong, hp);
16. nhapfile();
17. loadtb();
18. } catch (NumberFormatException e) {
19. JOptionPane.showMessageDialog(this, "Số Tin Chỉ Và Giá Tiền Là Số Nguyên!!!");
20. } catch (Exception e) {
21. System.out.println(e);
22. }
23. } else {
24. JOptionPane.showMessageDialog(this, "Đã có mã: " + jTextMaHP.getText() + " Ở một dòng khác");
25. }
26. } else {
27. try {
28. hp = new HocPhan(jTextMaHP.getText(), jTextTenHP.getText(),
29. Integer.parseInt(jTextSoTCLT.getText()), Integer.parseInt(jTextSTHH.getText()), Integer.parseInt(jTextDG.getText()));
30. ds.set(dong, hp);
31. nhapfile();
32. loadtb();
33. } catch (NumberFormatException e) {
34. JOptionPane.showMessageDialog(this, "Số Tin Chỉ Và Giá Tiền Là Số Nguyên!!!");
35. } catch (Exception e) {
36. JOptionPane.showMessageDialog(this, "Số Tin Chỉ Và Giá Tiền Là Số Nguyên!!!");
37. }
38. }
39. }
40. }
41. }

* Nút Xoá: Người quản trị chọn sinh viên muốn xóa trên bảng rồi ấn xóa hệ thống sẽ xóa sinh viên đó và xoá tài khoản của sinh viên đó.

Code xử lý nút Xoá:

1. private void jButtonXoaActionPerformed(java.awt.event.ActionEvent evt) {
2. // TODO add your handling code here:
3. dong = jTableHP.getSelectedRow();
4. if (dong != -1) {
5. ds.remove(dong);
6. nhapfile();
7. loadtb();
8. }
9. }

* Nút Quay lại: Quay lại menu của admin

Code xử lý nút Quay lại:

1. private void jButtonqlActionPerformed(java.awt.event.ActionEvent evt) {
2. // TODO add your handling code here:
3. MenuQuanLyAdmin MN= new MenuQuanLyAdmin();
4. MN.setVisible(true);
5. dispose();
6. }

* Bắt lỗi

Bắt lỗi không có file

catch (FileNotFoundException e) {

JOptionPane.showMessageDialog(this, e);

}

Bắt lỗi ngoại lệ Exception

catch (Exception ex) {

JOptionPane.showMessageDialog(this, ex);

}

Bắt lỗi NumberFormatException

catch (NumberFormatException e) {

JOptionPane.showMessageDialog(this, e);

}

1. private void jButtonThemActionPerformed(java.awt.event.ActionEvent evt) {
2. // TODO add your handling code here:
3. // TODO add your handling code here:
4. if (jTextTenHP.getText().trim().equals("") || jTextMaHP.getText().trim().equals("")
5. || jTextSTHH.getText().trim().equals("") || jTextSoTCLT.getText().trim().equals("") || jTextDG.getText().trim().equals("")) {
6. JOptionPane.showMessageDialog(this, "Chua du thong tin de them");
7. } else {
8. if (kiemtra(jTextMaHP.getText())) {
9. try {
10. hp = new HocPhan(jTextMaHP.getText(), jTextTenHP.getText(), Integer.parseInt(jTextSoTCLT.getText()),
11. Integer.parseInt(jTextSTHH.getText()), Integer.parseInt(jTextDG.getText()));
12. ds.add(hp);
13. nhapfile();
14. loadtb();
15. } catch (NumberFormatException e) {
16. JOptionPane.showMessageDialog(this, "Số Tin Chỉ Và Giá Tiền Là Số Nguyên!!!");
17. } catch (Exception e) {
18. System.out.println(e);
19. }
20. } else {
21. JOptionPane.showMessageDialog(this, "Đã Tồn Tại Mã :" + jTextMaHP.getText());
22. }
23. }
24. }

* Tập hợp

Collection được sử dụng:

Lấy dữ liệu đối tượng (getter)

Gán dữ liệu đối tượng (setter)



* Thao tác file

Thao tác với file HocPhan.txt

* Mã nguồn chính của chức năng

Code xử lí load data, đọc, ghi file

### Vy Văn Hùng – Quản lí đăng kí học phần – Xem lịch học- Xem thời khóa biểu – Thay đổi mật khẩu

* **Chức năng quản lí đăng kí học phần**Graphical user interface

  Description automatically generated with medium confidence

*Hình 2.4.3.1: Giao diện quản lý học phần*

* Mô tả các đối tượng trên màn hình
* Sử dụng thanh menubar để điều hướng giữa các màn hình.
* Bảng hiển thị danh sách học phần bao gồm các cột thông tin cần thiết như mã lớp học phần, mã học phần, tên môn học, giảng viên, tiết, thứ, ngày bắt đầu, địa điểm, học phí.
* Nút đăng kí học phần: Sinh viên sẽ là người sử dụng để tiến hành đăng kí học phần mong muốn sau khi đã chọn học phần trên màn hình. Hành động đăng kí có xử lí các trường hợp như trùng ngày học, trùng môn học phần đã đăng kí, tiến hành cập nhật số tiền nợ sau khi đăng kí thành công.

1. private void btn\_dangKyHocActionPerformed(java.awt.event.ActionEvent evt) {
2. // TODO add your handling code here:
3. dong = table\_DangKyHoc.getSelectedRow(); // lay dong chon ve
4. try {
5. if (dong != -1) {
6. String malhp = (table\_DangKyHoc.getModel().getValueAt(dong, 0).toString()); // lay gia tri cot ma lop hoc phan
7. String mahp = getMaHocPhanByMaLopHocPhan(malhp);
8. if (checkTrungNgayHoc(malhp)) { // kiem tra trung ngay hoc bao loi
9. if (kiemTraDangKy(mahp)) { // kiem tra trung mon dang ki bao loi
10. dk = new DangKy(maSV, malhp, mahp); // tao doi tuong dang ki
11. dsdk.add(dk);
12. luuFileDSDK(); // luu vao file
13. showDialog("Bạn Đã Đang Ký Thành Công Lớp " + malhp);
14. } else {
15. String tenhp = (table\_DangKyHoc.getModel().getValueAt(dong, 1).toString());
16. showDialog("Bạn Đã Đang Ký Học Phần: '" + tenhp + "' Rồi");
17. }
18. } else {
19. showDialog("Trùng lịch học vui lòng đăng kí lại");
20. }
21. }
22. setViewTongTien();
23. } catch (Exception e) {
24. showDialog(e.getMessage());
25. }
26. }

* Nút làm mới: Sinh viên sẽ kích vào nút làm mới khi cần cập nhật thông tin mới nhất từ bảng đăng kí hoặc cần cập nhật lại số tiền công nợ.

1. private void btn\_LamMoiActionPerformed(java.awt.event.ActionEvent evt) {
2. // TODO add your handling code here:
3. txt\_timKiem.setText("");
4. loadTable();
5. setViewTongTien();
6. }

* Nút tìm kiếm: Nút có chức năng tìm kiếm theo lựa chọn từ combobox (tìm kiếm theo tên lớp học, mã học phần).

1. private void btn\_timKiemActionPerformed(java.awt.event.ActionEvent evt) {
2. // TODO add your handling code here:
3. String timkiemBox = txt\_timKiem.getText().trim().toUpperCase();
4. ArrayList<LopHocPhan> dsTK = new ArrayList<>();
5. if (cbbTimKiem.getSelectedIndex() == 1) {
6. for (HocPhan it : dshp) {
7. if (it.getMaHocPhan().trim().toUpperCase().contains(timkiemBox)) {
8. for (LopHocPhan itL : dsLhp) {
9. if (it.getMaHocPhan().equals(itL.getMaHocPhan())) {
10. dsTK.add(itL);
11. }
12. }
13. }
14. }
15. } else {
16. for (HocPhan it : dshp) {
17. if (it.getTenHocPhan().trim().toUpperCase().contains(timkiemBox)) {
18. for (LopHocPhan itL : dsLhp) {
19. if (it.getMaHocPhan().equals(itL.getMaHocPhan())) {
20. dsTK.add(itL);
21. }
22. }
23. }
24. }
25. }
26. table\_DangKyHoc.setModel(new DKHocPhanDAO(dsTK, dshp, dsGv));
27. }

* Ô lựa chọn tìm kiếm theo: Là ô lựa chọn cho phép người dùng lựa chọn trường phù hợp tìm kiếm thấy môn học đăng kí theo 2 trường là tên môn học và mã môn học
* Ô nhập: Lấy dữ liệu tìm kiếm để tiến hành tìm kiếm
* Nhãn tổng số tiền: Hiển thị tổng số tiền nợ sau khi đăng kí môn.

Code đọc ghi file dữ liệu:

1. void loadFileGV() {
2. try {
3. dsGv = (ArrayList<GiangVien>) DBE.docFile(fileGV);
4. } catch (FileNotFoundException e) {
5. showDialog("Không tìm thấy file lưu trữ");
6. } catch (Exception ex) {
7. showDialog(ex.getMessage());
8. }
9. }
10. void loadFiledk() {
11. try {
12. dsdk = (ArrayList<DangKy>) DBE.docFile(fileDK);
13. } catch (FileNotFoundException e) {
14. showDialog("Không tìm thấy file lưu trữ");
15. } catch (Exception ex) {
16. showDialog(ex.getMessage());
17. }
18. }

**Hướng đối tượng:** Chia bài toán thành các đối tượng mỗi đối tượng sẽ có thuộc tính và phương thức, áp dụng các tính các tính chất hướng đối tượng vào bài toán như sau:

+ Tính đóng gói : chia các class có tính chất giống nhau vào các package , kết hợp với các access modifier(public,protected,private) để bảo vệ dữ liệu

+ Tính đa hình: Tính đa hình thể hiện ở runtime bằng cách override các method và tính đa hình thể hiện ở complie time bằng overloading(1 class có nhiều phương thức trùng tên nhưng khác số lượng tham số hoặc kiểu dữ liệu của tham số thể hiện ở hàm khởi tạo các frame màn hình )

+ Tính trừu tượng: sử dụng các abstract class(AbstractTableModel) để áp dụng tính trừu tượng, các class extend class abstract sẽ override để cụ thể hóa các logic khác nhau tại mỗi class.

**Bắt lỗi :**

+ Bắt lỗi cần chọn dòng để thao tác đăng kí

+ Bắt lỗi đọc file not found exception khi không tìm thấy file lưu trữ

+ Bắt lỗi con ép kiểu chuyển đối trong quá trình đọc file

**Collection** sử dụng trong code :

Sử dụng các Arraylist để lưu trữ danh sách dữ liệu và sử dụng một số hành động sau

+ Lấy dữ liệu đối tượng trong danh sách (get)

+ Sửa dữ liệu đối tượng trong danh sách (set)

+ Thêm dữ liệu đối tượng trong danh sách (add)

+ Kiểm tra đối tượng tồn tại trong array list không (contain).

**Thao tác file**:

+ DangKy.txt

+ HocPhan.txt

+ GiangVien.txt

+ LopHocPhan.txt

* **Chức năng xem lịch học** Graphical user interface, text, email

  Description automatically generated

*Hình 2.4.3.2 Màn hình xem lịch học*

* Mô tả các đối tượng trên màn hình
* Nút đến đăng kí học phần: là nút cho người sử dụng có thể đến màn hình đăng kí học phần ngay lập tức .

1. private void btn\_DangKyHocPhanActionPerformed(java.awt.event.ActionEvent evt) {
2. // TODO add your handling code here:
3. DangKyHocPhan dkhp = new DangKyHocPhan(maSV, "", "");
4. dkhp.setVisible(true);
5. dispose();
6. }

* Nút đổi chế độ xem tiến hành đổi chế độ xem về chế độ thời khóa biểu được thiết kế thông minh hơn

1. private void btn\_doiCheDoXemActionPerformed(java.awt.event.ActionEvent evt) {
2. // TODO add your handling code here:
3. DangKiHocPhanAnotherView dKHocPhanAnotherView = new DangKiHocPhanAnotherView(maSV);
4. dKHocPhanAnotherView.setVisible(true);
5. }

* Nút hủy đăng kí là nút cho phép sinh viên hủy môn đã đăng kí , Nút có hiển thị DialogConfirm chắc chắn húy hay không sau khi bấm

1. private void btn\_HuyDangKyActionPerformed(java.awt.event.ActionEvent evt) {
2. // TODO add your handling code here:
3. dong = table\_LichHoc.getSelectedRow();
4. try
5. {
6. if (dong != -1) {
7. int answer = JOptionPane.showConfirmDialog(this,
8. "Bạn có chắc chắn muốn hủy học phần này", "Thông báo",
9. JOptionPane.YES\_NO\_OPTION);
10. if (answer == JOptionPane.YES\_OPTION) {
11. dsdkall.remove(dsdk.get(dong));
12. nhapfile();
13. loadFiledk();
14. loadTable();
15. } else if (answer == JOptionPane.NO\_OPTION) {
16. // do something else
17. }
18. } else {
19. throw new Exception("Cần chọn dòng để thao tác");
20. }
21. }catch(Exception e){
22. showDialog(e.getMessage());
23. }
24. }

* Nút Quay lại : tiến hành trở về màn hình chính

**Hướng đối tượng:** Chia bài toán thành các đối tượng mỗi đối tượng sẽ có thuộc tính và phương thức, áp dụng các tính các tính chất hướng đối tượng vào bài toán như sau:

+ Tính đóng gói : chia các class có tính chất giống nhau vào các package , kết hợp với các access modifier(public,protected,private) để bảo vệ dữ liệu

+ Tính đa hình: Tính đa hình thể hiện ở runtime bằng cách override các method và tính đa hình thể hiện ở complie time bằng overloading (1 class có nhiều phương thức trùng tên nhưng khác số lượng tham số hoặc kiểu dữ liệu của tham số thể hiện ở hàm khởi tạo các frame màn hình)

+ Tính trừu tượng: sử dụng các abstract class (AbstractTableModel) để áp dụng tính trừu tượng, các class extend class abstract sẽ override để cụ thể hóa các logic khác nhau tại mỗi class.

**Bắt lỗi :**

+ Bắt lỗi cần chọn dòng để thao tác hủy đăng đăng kí

+ Bắt lỗi đọc file not found exception khi không tìm thấy file lưu trữ

+ Bắt lỗi con ép kiểu chuyển đối trong quá trình đọc file

**Collection** sử dụng trong code :

Sử dụng các Arraylist để lưu trữ danh sách dữ liệu và sử dụng một số hành động sau

+ Lấy dữ liệu đối tượng trong danh sách(get)

+ Sửa dữ liệu đối tượng trong danh sách(set)

+ Thêm dữ liệu đối tượng trong danh sách(add)

+ Kiểm tra đối tượng tồn tại trong array list không(contain).

**Thao tác file**:

+ DangKy.txt

+ HocPhan.txt

+ LopHocPhan.txt

* **Chức năng hiển thị các môn học dưới màn hình thời khóa biểu**

A picture containing text, indoor, cabinet, screenshot

Description automatically generated

*Hình 2.4.3.3 Màn hình xem thời khóa biểu*

* Nút đổi chế độ xem: Cho phép quay lại màn hình xem lịch học
* Nút làm mới: đảm bảo cập nhật các môn học sau khi đăng kí
* Nút [chọn]: cho phép mở cửa số hiển thị các môn đăng kí theo từng buổi , từng thứ trong tuần hỗ trợ đăng kí nhanh hơn

1. private void lb\_st2MouseClicked(java.awt.event.MouseEvent evt) {
2. // TODO add your handling code here:
3. DangKyHocPhan dkhp = new DangKyHocPhan(this.maSV, "2", "S");
4. dkhp.setVisible(true);
5. //dispose();
6. }

**Hướng đối tượng:** Chia bài toán thành các đối tượng mỗi đối tượng sẽ có thuộc tính và phương thức, áp dụng các tính các tính chất hướng đối tượng vào bài toán như sau:

+ Tính đóng gói : chia các class có tính chất giống nhau vào các package , kết hợp với các access modifier(public,protected,private) để bảo vệ dữ liệu

+ Tính đa hình: Tính đa hình thể hiện ở runtime bằng cách override các method và tính đa hình thể hiện ở complie time bằng overloading(1 class có nhiều phương thức trùng tên nhưng khác số lượng tham số hoặc kiểu dữ liệu của tham số thể hiện ở hàm khởi tạo các frame màn hình )

+ Tính trừu tượng: sử dụng các abstract class(AbstractTableModel) để áp dụng tính trừu tượng, các class extend class abstract sẽ override để cụ thể hóa các logic khác nhau tại mỗi class.

**Bắt lỗi :**

+ Bắt lỗi đọc file not found exception khi không tìm thấy file lưu trữ

+ Bắt lỗi con ép kiểu chuyển đối trong quá trình đọc file

**Collection** sử dụng trong code :

Sử dụng các Arraylist để lưu trữ danh sách dữ liệu và sử dụng một số hành động sau

+ Lấy dữ liệu đối tượng trong danh sách(get)

+ Sửa dữ liệu đối tượng trong danh sách(set)

+ Thêm dữ liệu đối tượng trong danh sách(add)

+ Kiểm tra đối tượng tồn tại trong array list không(contain).

**Thao tác file**:

+ DangKy.txt

+ HocPhan.txt

+ GiangVien.txt

+ LopHocPhan.txt

* **Màn hình đổi mật khẩu**

Graphical user interface, application

Description automatically generated

*Hình ảnh 2.4.3.4 Màn hình đổi mật khẩu*

Mô tả đối tượng :

* Nút đổi mật khẩu cho phép người dùng đổi mật khẩu
* Nút đóng cho phép người dùng đóng cửa số đổi mật khẩu
* Textbox mật khẩu cũ : lấy mật khẩu cũ người dùng
* Textbox mật khẩu mới : lấy mật khẩu mới người dùng
* Textbox xác nhận mật khẩu : xác nhận lại mật khẩu mới

1. private void btn\_DoiMatKhauActionPerformed(java.awt.event.ActionEvent evt) {
2. // TODO add your handling code here:
3. String matKhauCu = txt\_matKhauCu.getText().toString().trim();
4. String matKhauMoi = txt\_matKhauMoi.getText().toString().trim();
5. String matKhauXacNhan = txt\_matKhauXacNhan.getText().toString().trim();
6. checkLogin(); // keim tra dang nhap
7. try {
8. if (matKhauMoi.equals(matKhauXacNhan)) {
9. if (!matKhauCu.equals("") && !matKhauMoi.equals("")) {
10. if (matKhauCu.equals(matKhauMoi)) {
11. throw new Exception("Mật khẩu mới phải khác mật khẩu cũ");
12. } else {
13. if (Login(maSV, matKhauCu)) {
14. //JOptionPane.showMessageDialog(this,"Thanh cong");
15. for (int i = 0; i < danhSachTaiKhoan.size(); i++) {
16. if (danhSachTaiKhoan.get(i).getTK().equals(maSV) && danhSachTaiKhoan.get(i).getMK().equals(matKhauCu)) {
17. TaiKhoan taiKhoan = danhSachTaiKhoan.get(i);
18. taiKhoan.setMK(matKhauMoi);
19. danhSachTaiKhoan.set(i, taiKhoan);
20. writeUsers();
21. showDialog("Đổi mật khẩu thành công");
22. break;
23. }
24. }
25. } else {
26. throw new Exception("Mật khẩu cũ không chính xác !");
27. }
28. }
29. } else {
30. throw new Exception("mật khẩu cũ và mật khẩu mới !");
31. }
32. } else {
33. throw new Exception("Mật khẩu mới không khớp với mật khẩu xác nhận lại");
34. }
35. } catch (Exception exception) {
36. showDialog(exception.getMessage());
37. }
38. }

**Hướng đối tượng:** Chia bài toán thành các đối tượng mỗi đối tượng sẽ có thuộc tính và phương thức, áp dụng các tính các tính chất hướng đối tượng vào bài toán như sau:

+ Tính đóng gói : chia các class có tính chất giống nhau vào các package , kết hợp với các access modifier(public,protected,private) để bảo vệ dữ liệu

+ Tính đa hình: Tính đa hình thể hiện ở runtime bằng cách override các method và tính đa hình thể hiện ở complie time bằng overloading(1 class có nhiều phương thức trùng tên nhưng khác số lượng tham số hoặc kiểu dữ liệu của tham số thể hiện ở hàm khởi tạo các frame màn hình )

+ Tính trừu tượng: sử dụng các abstract class(AbstractTableModel) để áp dụng tính trừu tượng, các class extend class abstract sẽ override để cụ thể hóa các logic khác nhau tại mỗi class.

**Bắt lỗi :**

+ Bắt lỗi các trường mật khẩu không hợp lệ.

+ Bắt lỗi các mật khẩu cũ không trùng ,mật khẩu cũ giống mật khẩu mới

+ Mật khẩu xác nhận khác mật khẫu mới.

+ Bắt lỗi đọc file not found exception khi không tìm thấy file lưu trữ .

+ Bắt lỗi con ép kiểu chuyển đối trong quá trình đọc file .

**Collection** sử dụng trong code :

Sử dụng các Arraylist để lưu trữ danh sách dữ liệu và sử dụng một số hành động sau:

+ Lấy dữ liệu đối tượng trong danh sách(get)

+ Sửa dữ liệu đối tượng trong danh sách(set)

+ Thêm dữ liệu đối tượng trong danh sách(add)

+ Kiểm tra đối tượng tồn tại trong array list không(contain).

**Thao tác file**:

+TaiKhoan.txt

5) Một số màn hình khác màn hình Trang chủ menu sinh viên

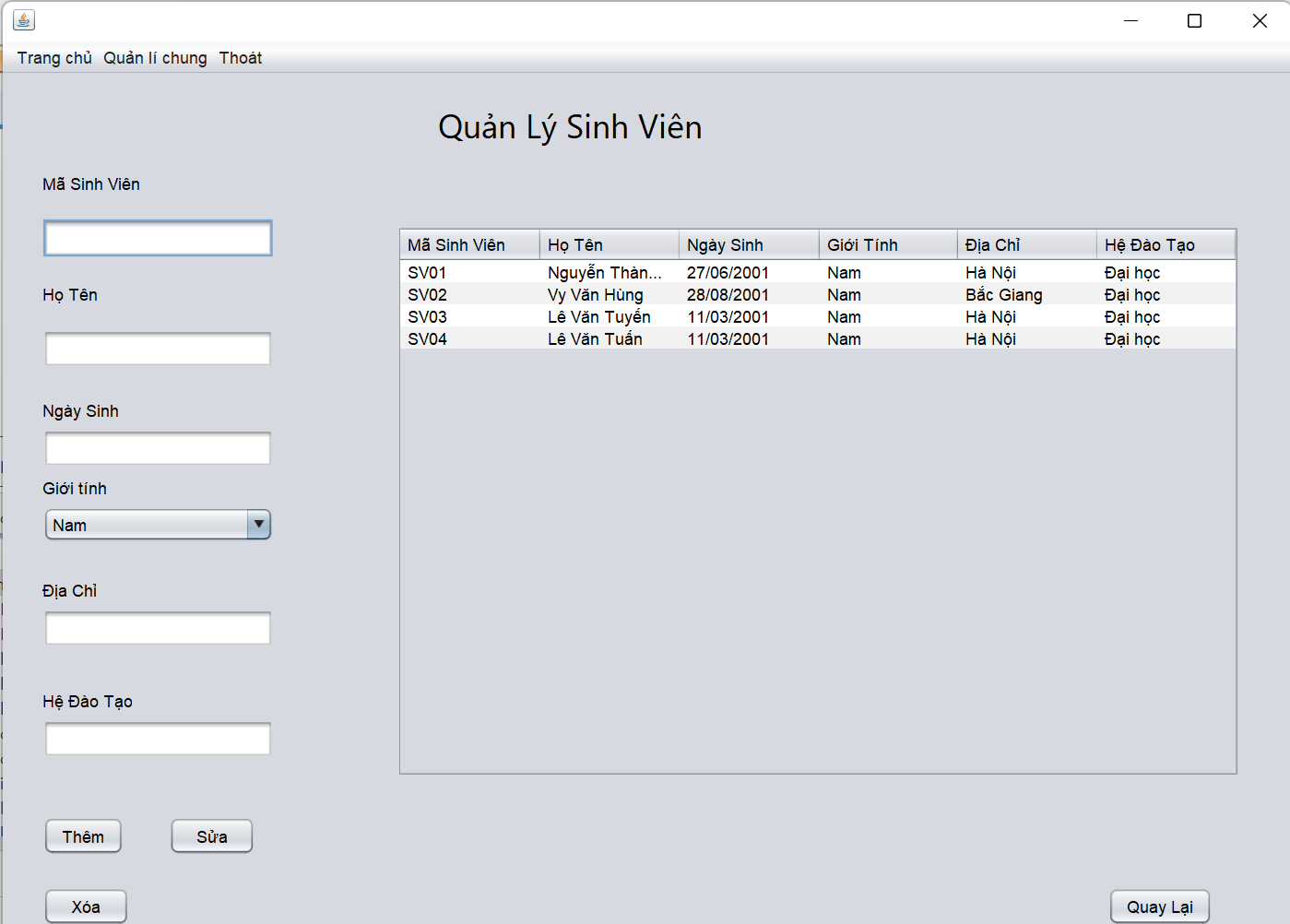
Graphical user interface

Description automatically generated with medium confidence

*Hình 2.4.3.5 Màn hình trang chủ sinh viên*

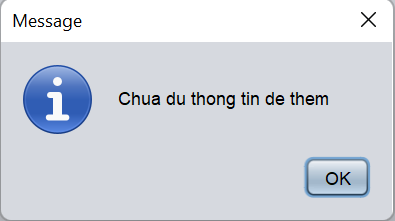
### Lê Văn Tuyến – Chức năng quản lý sinh viên, quản lý lớp, quản lý đăng nhập

* **Quản lý sinh viên**

****

*Hình 2.4.4.1: Giao diện quản lý Sinh Viên*

Danh sách các Sinh viên được load từ file SinhVien.txt

* Khi người quản trị Click và nút “Thêm” và nhập đầy đủ thông tin nếu nhập thiếu thông tin thì sẽ thông báo:
* Source code:
* Xử lý khi người quản trị click vào nút thêm:

1. private void jButtonThemActionPerformed(java.awt.event.ActionEvent evt) {
2. // TODO add your handling code here:
3. if(jTextMaSV.getText().trim().equals("") ||jTextNS.getText().trim().equals("") ||jTextHT.getText().trim().equals("") ||jTextDC.getText().trim().equals("") || jTextDT.getText().trim().equals("")){
4. JOptionPane.showMessageDialog(this,"Chua du thong tin de them");
5. }else{
6. if(kiemtra(jTextMaSV.getText())){
7. sv = new SinhVien(jTextMaSV.getText().toUpperCase(),jTextHT.getText(), jTextNS.getText(),GT,jTextDC.getText(),jTextDT.getText());
8. tk=new TaiKhoan(jTextMaSV.getText(),"1234");
9. ds.add(sv);
10. dsTK.add(tk);
11. nhapfile();
12. nhapfileTK();
13. loadtb();
14. }
15. else{
16. JOptionPane.showMessageDialog(this,"Đã Tồn Tại Mã :"+jTextMaSV.getText());
17. }
18. }
19. }

* Xử lý thông tin khi người dùng nhấn nút sửa:

1. private void jButtonSuaActionPerformed(java.awt.event.ActionEvent evt) {
2. // TODO add your handling code here:
3. dong = JTableSV.getSelectedRow();
4. if(dong !=-1){
5. if(jTextMaSV.getText().trim().equals("") ||jTextNS.getText().trim().equals("") ||jTextHT.getText().trim().equals("") ||jTextDC.getText().trim().equals("") || jTextDT.getText().trim().equals("")){
6. JOptionPane.showMessageDialog(this,"Chua du thong tin de them");
7. }else{
8. if(kiemtra(jTextMaSV.getText())==false){
9. if(jTextMaSV.getText().trim().equals(ds.get(dong).getMaSinhVien().trim())){
10. sv = new SinhVien(jTextMaSV.getText().toUpperCase(),jTextHT.getText(), jTextNS.getText(),GT,jTextDC.getText(),jTextDT.getText());
11. ds.set(dong,sv);
12. nhapfile();
13. loadtb();
14. }
15. else{
16. JOptionPane.showMessageDialog(this,"Đã có mã: "+jTextMaSV.getText()+" Ở một dòng khác");
17. }
18. }
19. else{
20. sv = new SinhVien(jTextMaSV.getText().toUpperCase(),jTextHT.getText(), jTextNS.getText(),GT,jTextDC.getText(),jTextDT.getText());
21. ds.set(dong,sv);
22. dsTK.get(dong+1).setTK(jTextMaSV.getText());
23. nhapfile();
24. nhapfileTK();
25. loadtb();
26. }
27. }
28. }
29. }

* Xử lý thông tin khi người dùng nhấn nút xóa:

1. private void jButtonXoaActionPerformed(java.awt.event.ActionEvent evt) {
2. // TODO add your handling code here:
3. dong= JTableSV.getSelectedRow();
4. if(dong != -1){
5. ds.remove(dong);
6. dsTK.remove(dong+1);
7. nhapfileTK();
8. nhapfile();
9. loadtb();
10. }
11. }

**Hướng đối tượng** :

+ Chia bài toán thành các đối tượng mỗi đối tượng sẽ có thuộc tính và phương thức, áp dụng các tính các tính chất hướng đối tượng vào bài toán như sau:

+ Tính đóng gói: chia các class có tính chất giống nhau vào các package , kết hợp với các access modifier(public,protected,private) để bảo vệ dữ liệu

+ Tính kế thừa: tạo ra class cha có thuộc tính và phương thức chung để các class con kế thừa và sử dụng chung

**Bắt lỗi :**

+ bắt lỗi khi đọc ghi file.

**Collection sử dụng trong code :**

Sử dụng các Arraylist để lưu trữ danh sách dữ liệu và sử dụng một số hành động sau

+ Lấy dữ liệu đối tượng trong danh sách (get)

+ Thêm dữ liệu đối tượng vào danh sách (add)

+ Xóa dữ liệu đôi tượng khỏi danh sách (remove)

+ Trả về int vị trí của đối tượng trong danh sách (indexof())

+ Kiểm tra đối tượng tồn tại trong array list không (contain).

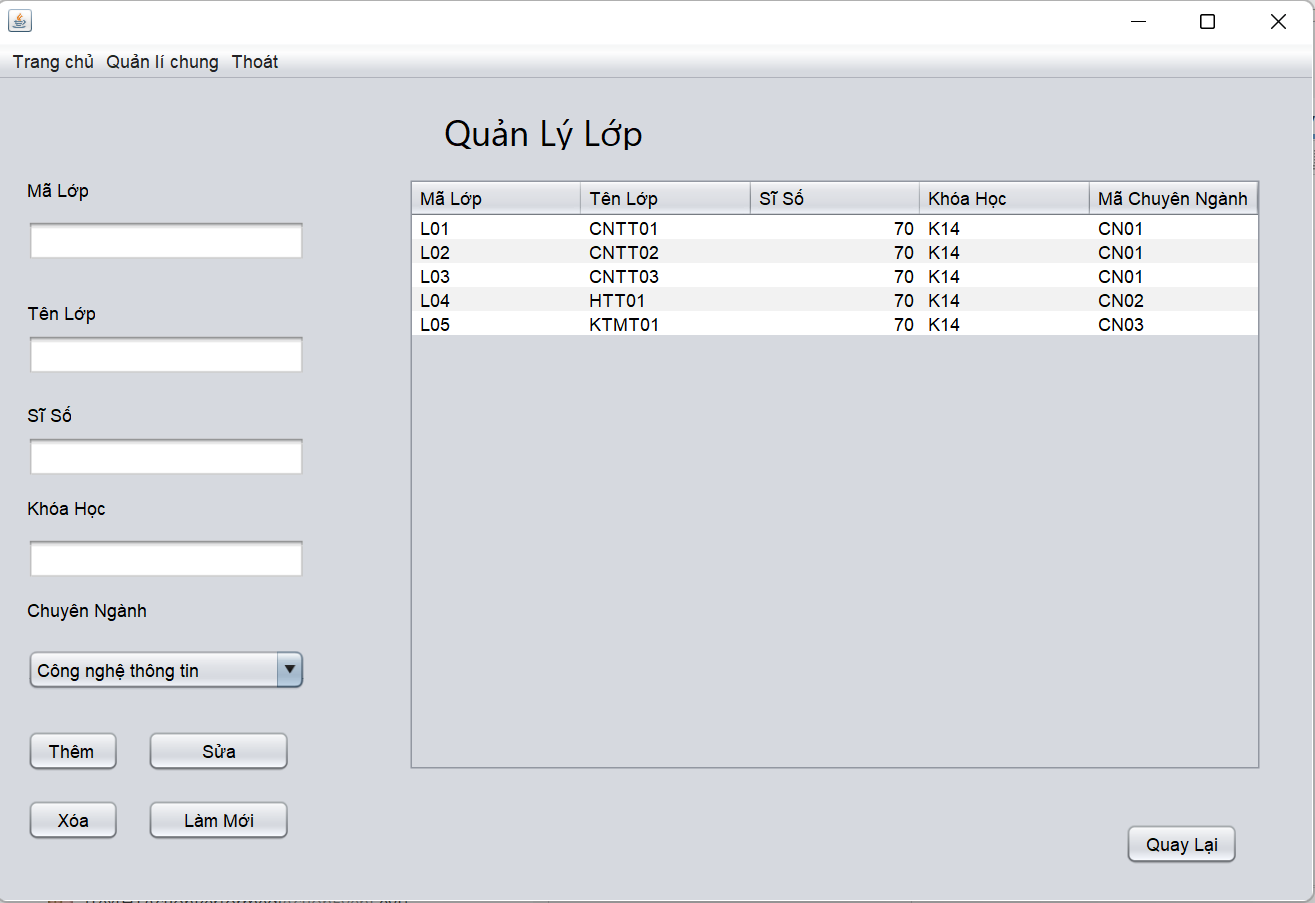
* Giao diện và file đã dùng:

Giao diện quản lý Sinh viên dùng dùng các chữ jlabel, các button,… và sử dụng **Các file**

+ file SinhVien.txt

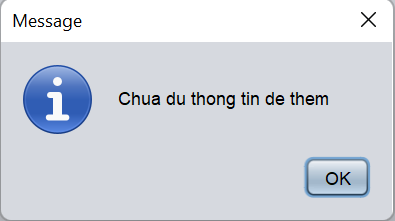
+ file TaiKhoan.txt

* **Quản lý Lớp**

****

*Hình 2.4.4.2: Giao diện quản lý lớp*

Danh sách các Lớp được load từ file Lop.txt

* Khi người quản trị Click và nút “Thêm” và nhập đầy đủ thông tin nếu nhập thiếu thông tin thì sẽ thông báo:
* Source code:
* Xử lý khi người quản trị click vào nút thêm:

1. private void jButtonThemActionPerformed(java.awt.event.ActionEvent evt) {
2. // TODO add your handling code here:
3. if(jTextMaL.getText().trim().equals("") ||jTextTenL.getText().trim().equals("")||jTextSiso.getText().trim().equals("")|| jTextKhoaHoc.getText().trim().equals("")){
4. JOptionPane.showMessageDialog(this,"Chua du thong tin de them");
5. }else{
6. if(kiemtra(jTextMaL.getText())){
7. lp= new Lop(jTextMaL.getText(), jTextTenL.getText(), Integer.parseInt(jTextSiso.getText()), jTextKhoaHoc.getText(),maCN);
8. ds.add(lp);
9. nhapfile();
10. loadtb();
11. }
12. else{
13. JOptionPane.showMessageDialog(this,"Đã Tồn Tại Mã :"+jTextMaL.getText());
14. }
15. }
16. }

* Xử lý thông tin khi người dùng nhấn nút sửa:

1. private void jButtonSua2ActionPerformed(java.awt.event.ActionEvent evt) {
2. // TODO add your handling code here:
3. dong = jTableLop.getSelectedRow();
4. if(dong !=-1){
5. if(jTextMaL.getText().trim().equals("") ||jTextTenL.getText().trim().equals("")){
6. JOptionPane.showMessageDialog(this,"Chua du thong tin de them");
7. }else{
8. if(kiemtra(jTextMaL.getText())==false){
9. if(jTextMaL.getText().trim().equals(ds.get(dong).getMaLop().trim())){
10. lp= new Lop(jTextMaL.getText(), jTextTenL.getText(), Integer.parseInt(jTextSiso.getText()), jTextKhoaHoc.getText(), maCN);
11. ds.set(dong,lp);
12. nhapfile();
13. loadtb();
14. }
15. else{
16. JOptionPane.showMessageDialog(this,"Đã có mã: "+jTextMaL.getText()+" Ở một dòng khác");
17. }
18. }
19. else{
20. lp= new Lop(jTextMaL.getText(), jTextTenL.getText(), Integer.parseInt(jTextSiso.getText()), jTextKhoaHoc.getText(), maCN);
21. ds.set(dong,lp);
22. nhapfile();
23. loadtb();
24. }
25. }
26. }
27. }

* Xử lý thông tin khi người dùng nhấn nút xóa:

1. private void jButtonXoaActionPerformed(java.awt.event.ActionEvent evt) {
2. // TODO add your handling code here:
3. dong= jTableLop.getSelectedRow();
4. if(dong != -1){
5. ds.remove(dong);
6. nhapfile();
7. loadtb();
8. }
9. }

**Hướng đối tượng :**

+ Chia bài toán thành các đối tượng mỗi đối tượng sẽ có thuộc tính và phương thức, áp dụng các tính các tính chất hướng đối tượng vào bài toán như sau:

+ Tính đóng gói: chia các class có tính chất giống nhau vào các package , kết hợp với các access modifier (public,protected,private) để bảo vệ dữ liệu

+ Tính kế thừa: tạo ra class cha có thuộc tính và phương thức chung để các class con kế thừa và sử dụng chung

**Bắt lỗi :**

+ Bắt lỗi khi đọc ghi file.

**Collection sử dụng trong code :**

Sử dụng các Arraylist để lưu trữ danh sách dữ liệu và sử dụng một số hành động sau

+ Lấy dữ liệu đối tượng trong danh sách(get)

+ Thêm dữ liệu đối tượng vào danh sách(add)

+ Xóa dữ liệu đôi tượng khỏi danh sách(remove)

+ Trả về int vị trí của đối tượng trong danh sách(indexof())

+ Kiểm tra đối tượng tồn tại trong array list không(contain).

**Giao diện và file đã dùng :**

Giao diện Quản lý lớp dùng dùng các chữ jlabel, các button ,… và sử dụng các file

+ file Lop.txt

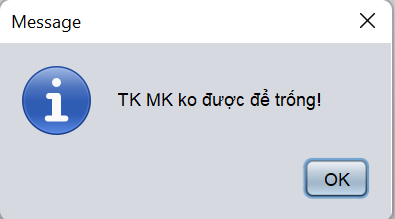
+file ChuyenNganh.txt

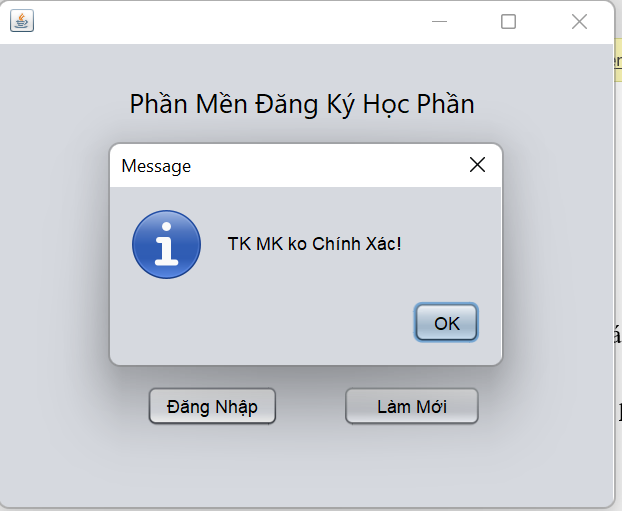
#### Quản lý Đăng nhập

*Hình 2.4.4.3. Giao diện đăng nhập*

* Mô tả các đối tượng trên màn hình
* Các ô nhập thông tin tài khoản: tên tài khoản và mật khẩu.
* Nút đăng nhập: Người dùng điền thông tin vào đăng nhập gồm username và password.

Danh sách các Tài khoản đăng nhập được load từ file TaiKhoan.txt

* Khi người dùng nhập đầy đủ tài khoản và mật khẩu để tiên hành đăng nhập nếu nhập thiếu thông tin thì sẽ thông báo:
* Nếu người dùng nhập sai tài khoản hoặc mật khẩu thì sẽ thông báo:



* Nút làm mới: Người dùng ấn vào nút làm mới sẽ làm mới ô input tài khoản và mật khẩu.

Code xử lý làm mới:

1. private void jButtonLMActionPerformed(java.awt.event.ActionEvent evt) {
2. // TODO add your handling code here:
3. jTextMK.setText("");
4. jTextTK.setText("");
5. }

Code xử lý nút Đăng nhập

1. private void jBtDangNhapActionPerformed(java.awt.event.ActionEvent evt) {
2. if(jTextTK.getText().trim().equals("") == false && jTextMK.getText().trim().equals("")==false){
3. String Tk=null;
4. for( var it :dstk){
5. if(it.getTK().equals(jTextTK.getText().trim()) && it.getMK().equals(jTextMK.getText().trim())){
6. Tk=it.getTK();
7. break;
8. }
9. }
10. if(Tk!=null){
11. if(Tk.trim().toUpperCase().contains("SV")){
12. MenuSinhVien menusv=new MenuSinhVien(Tk.trim().toUpperCase());
13. //menusv.setMaSV();
14. menusv.setVisible(true);
15. dispose();
16. }
17. else{
18. MenuQuanLyAdmin MN= new MenuQuanLyAdmin();
19. MN.setVisible(true);
20. dispose();
21. }
22. }
23. else{
24. JOptionPane.showMessageDialog(this,"TK MK ko Chính Xác!");
25. }
26. }
27. else{
28. JOptionPane.showMessageDialog(this,"TK MK ko được để trống!");
29. }
30. }

**Hướng đối tượng :**

+ Chia bài toán thành các đối tượng mỗi đối tượng sẽ có thuộc tính và phương thức, áp dụng các tính các tính chất hướng đối tượng vào bài toán như sau:

+ Tính đóng gói : chia các class có tính chất giống nhau vào các package , kết hợp với các access modifier(public,protected,private) để bảo vệ dữ liệu

+ Tính kế thừa : tạo ra class cha có thuộc tính và phương thức chung để các class con kế thùa và sử dụng chung .

**Bắt lỗi :**

+ Bắt lỗi khi đọc ghi file.

**Collection sử dụng trong code :**

Sử dụng các Arraylist để lưu trữ danh sách dữ liệu và sử dụng một số hành động sau:

+ Lấy dữ liệu đối tượng trong danh sách (get)

+ Trả về int vị trí của đối tượng trong danh sách (indexof())

* Giao diện và file đã dùng :

**Giao diện login dùng dùng các chữ jlabel, các button ,… và sử dụng các file**

+ File TaiKhoan.txt

# PHẦN 3. KẾT LUẬN VÀ BÀI HỌC KINH NGHIỆM

## 3.1. Nội dung đã thực hiện.

Trong quá trình xây dựng hệ thống đăng ký học phần của trường Đại học Công nghiệp Hà Nội, các thành viên trong nhóm đã tổng hợp được rất nhiều kiến thức của môn lập trình Java và một số môn học khác như phân tích thiết kế hệ thống, thiết kế cơ sở dữ liệu, …để áp dụng vào xây dựng một hệ thống thực tế. Thông qua quá trình làm việc nhóm các thành viên trong trong nhóm đã đoàn kết với nhau hơn, tinh thần làm việc tập thể được nâng cao hơn, rèn luyện kỹ năng làm việc nhóm. Nhưng do kiến thức và kinh nghiệm còn hạn chế nên nhóm rất mong nhận được sự đóng góp ý kiến của cô giáo và các bạn để hệ thống của nhóm xây dựng được hoàn thiện hơn. Nhóm xin chân thành cảm ơn cô giáo Vũ Thị Dương đã nhiệt tình hướng dẫn nhóm trong suốt quá trình thực hiện đề tài này, được sự nhiệt tình hướng dẫn của cô nhóm đã hoàn thành được đề tài và hiểu hơn về môn học.

* Thực hành lập trình giao diện cơ bản bằng Java Swing cho phần mềm đăng ký môn học – Đại học công nghiệp Hà Nội.
* Phân tách phần xử lý giao diện, xử lý logic và lưu trữ bằng đọc ghi file.
* Phân tích thiết kế một phần mềm đơn giản với các công cụ UML.
* Rút ra được kinh nghiệm trong phân tích nghiệp vụ

## 3.2. Hướng phát triển.

Hoàn thiện hệ thống hơn về cả chức năng và giao diện để hệ thống có thể áp dụng vào thực tế quản lý công tác đăng ký học phần của các trường đại học khác.