|  |
| --- |
|  |
| **TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**  **KHOA CÔNG NGHỆ THÔNG TIN**  **---------------------------------------** |
|  |
| BÁO CÁO BÀI TẬP LỚN HỌC PHẦN  LẬP TRÌNH JAVA |
| Đề Tài: XÂY DỰNG CHƯƠNG TRÌNH QUẢN LÝ NHÓM HỌC MÔN LẬP TRÌNH JAVA |
|  |
|  |
|  |
| **GVHD: *ThS. Vũ Thị Dương*** |
| **Sinh viên: Phảm Ích Quảng**  **Mẫn Văn Quân**  **Lê Đăng Quang**  **Hồ Hữu Quang**  **Phạm Thanh Quý** |
| **Nhóm: 12**  **Lớp:** 202020503175006 **Khóa: 13** |
|  |
| **Hà Nội – Năm 2021** |

[PHẦN 1. MỞ ĐẦU 3](#_Toc73067890)

[1.1. Ý tưởng và chủ đề 3](#_Toc73067891)

[1.2. Mục tiêu 3](#_Toc73067892)

[1.3. Công nghệ sử dụng 4](#_Toc73067893)

[1.3.1. Kế thứa 4](#_Toc73067894)

[1.3.2. Bắt sự kiện trong Java AWT 4](#_Toc73067895)

[1.3.3. Giao diện tương tác 4](#_Toc73067896)

[PHẦN 2. KẾT QUẢ NGHIÊN CỨU 6](#_Toc73067897)

[2.1. Khảo sát hệ thống 7](#_Toc73067898)

[2.1.1. Mục tiêu 7](#_Toc73067899)

[2.1.2. Tổng quan về các vấn đề bài toán giải quyết 7](#_Toc73067900)

[2.1.3. Khảo sát sơ bộ 7](#_Toc73067901)

[2.1.4. Tài liệu đặc tả yêu cầu người dùng 10](#_Toc73067902)

[2.2. Phân tích hệ thống 12](#_Toc73067903)

[2.2.1. Mô hình hóa chức năng hệ thống 12](#_Toc73067904)

[2.2.2. Mô hình hóa dữ liệu của hệ thống. 26](#_Toc73067905)

[2.3. Thiết kế hệ thống 27](#_Toc73067906)

[2.3.1. Thiết kế giao diện 27](#_Toc73067907)

[2.3.2. Thiết kế dữ liệu- Ánh xạ lớp sang bảng 42](#_Toc73067908)

[2.4. Thực hiện bài toán 45](#_Toc73067909)

[2.4.1. Hồ Hữu Quang- Quản lý lớp 45](#_Toc73067910)

[2.4.2. Lê Đăng Quang- Chọn đề tài 57](#_Toc73067911)

[2.4.3. Lê Đăng Quang – Thông báo leader 57](#_Toc73067912)

[2.4.4. Lê Đăng Quang – Xem đánh giá 57](#_Toc73067913)

[2.4.5. Phạm Ích Quảng – Đăng nhập 57](#_Toc73067914)

[2.4.6. Phạm Ích Quảng – Đánh giá 57](#_Toc73067915)

[2.4.7. Phạm Thanh Quý – Công việc 57](#_Toc73067916)

[2.4.8. Phạm Thanh Quý – Đăng xuất 57](#_Toc73067917)

[2.4.9. Mẫn Văn Quân – Nộp bài 57](#_Toc73067918)

[2.4.10. Mẫn Văn Quân – Xem thông báo 57](#_Toc73067919)

[2.4.11. Mẫn Văn Quân – Xuất file 57](#_Toc73067920)

[PHẦN 3. KẾT LUẬN VÀ BÀI HỌC KINH NGHIỆM 57](#_Toc73067921)

[3.1. Nội dung đã thực hiện 57](#_Toc73067922)

[3.2. Hướng phát triển. 57](#_Toc73067923)

# MỞ ĐẦU

## Ý tưởng và chủ đề

Công nghệ thông tin hiện nay đang phát triển rất mạnh mẽ và có vai trò rất quan trọng trong cuộc sống hàng ngày của chúng ta. Việc áp dụng công nghệ thông tin vào cuộc sống giúp cho các công việc được tiến hành nhanh chóng và hiểu quả hơn. Để tối ưu việc quản lý, đánh giá, họ tập và làm việc nhóm một cách có hiệu quả. Chúng em chọn đề tài “Xây dựng chương trình quản lý các nhóm học môn Lập trình Java” nhằm tìm hiểu về ngôn ngữ Java trong lập trình hướng đối tượng. Từ đó viết viết một ứng dụng cụ thể thử nghiệm để củng cố kiến thức và định hướng kế hoạch phát triển các mô hình khác trong tương lại. Bài tập lớn này nhằm mục đích giúp sinh viên tìm hiểu và bước đầu làm quen với lập trình Java, lập trình những ứng dụng Java đơn giản có mục đích và tính hữu dụng trong thực tế.

## Mục tiêu

Hiểu được cách phân tích và giải quyết bài toán, từ đó nắm vững được kiến thức về ngôn ngữ Java, các câu lệnh và một số công nghệ hướng đối tượng.

Ngoài ra mục đích của ứng dụng chính là quản lý và đánh giá cách làm việc nhóm của các nhóm học trong môn lập trình Java.

## Công nghệ sử dụng

Bài tập lớn này được thiết kế theo phương pháp lập trình hướng đối tượng do SunMircoSystem đưa ra vào năm 1991. Chính vì vậy mà nó giải quyết được các vướng mắc gặp phải khi thiết kế theo lập trình hướng thủ tục.

### Kế thứa

Tính kế thừa trong java là một kỹ thuật mà trong đó đó một đối tượng thu được tất cả thuộc tính và hành vi của đối tượng cha. Ý tưởng đằng sau tính kế thừa trong Java là bạn có thể tạo các lớp mới mà được xây dựng dựa trên các lớp đang tồn tại. Khi bạn kế thừa từ một lớp đang tồn tại, bạn có thể tái sử dụng các phương thức và các trường của lớp cha, và bạn cũng có thể bổ sung thêm các phương thức và các trường khác. Tính kế thừa biểu diễn mối quan hệ IS­A, còn được gọi là mối quan hệ cha­con.

Khi chúng ta nói về tính kế thừa, từ khóa thường xuyên nhất được sử dụng là extends và implements. Những từ khóa này có thể định nghĩa một kiểu là loại IS­A của loại khác. Sử dụng những từ khóa, chúng ta có thể tạo một đối tượng sử dụng thuộc tính của đối tượng khác. Chúng ta sử dụng từ khóa extends của lớp con để có thể kế thừa các thuộc tính của lớp cha trừ các thuộc tính private của lớp cha.

### Bắt sự kiện trong Java AWT

Các hệ thống GUI xử lý các tương tác người dùng với sự trợ giúp của mô hình hướng sự kiện. Sự kiện trong Java:

* Sự kiện có thể được định nghĩa như là một kiểu tín hiệu tới chương trình thông báo một việc gì đó đã xảy ra.
* Sự kiện được sinh ra bởi các hành động của người dùng, ví dụ như di chuyển, click chuột, hay nhấn phím....

### Giao diện tương tác

Các ứng dụng phần mềm hiện nay rất thân thiện vì được trình bày nhiều màn hình giao diện đồ họa đẹp mắt. Các ngôn ngữ lập trình hiện nay cung cấp các đối tượng đồ họa, chúng có thể được điều khiển bởi người lập trình, hay bởi người sử dụng. Một trong số những kết quả quan trọng nhất chính là các ngôn ngữ hiện nay được dựa trên Giao diện người dùng đồ họa (Graphical UserInterface ­ GUI).

GUI cung cấp chức năng nhập liệu theo cách thân thiện với

người dùng. GUI đa dạng từ ứng dụng đến ứng dụng và có thể

chứa nhiều điều khiển như hộp văn bản, nhãn, hộp danh sách hay các điều khiển khác. Các ngôn ngữ lập trình khác nhau cung cấp

nhiều cách khác nhau để tạo GUI. Ngôn ngữ Java có thể cung cấp chức năng kéo và thả trong khi đó phần mềm giống như C++ yêu cầu người lập trình phải viết toàn bộ mã để xây dựng GUI.

Một phần tử (element) GUI được thiết lập bằng cách sử dụng thủ tục sau:

* Tạo đối tượng
* Xác định sự xuất hiện ban đầu của đối tượng
* Chỉ ra nó nằm ở đâu
* Thêm phần tử vào giao diện trên màn hình

Một thành phần (component) GUI là một đối tượng trực quan. Người dùng tương tác với đối tượng này thông qua con trỏ chuột hay bàn phím. Các thành phần như là button, label v.v… có thể được nhìn thấy trên màn hình. Bất kỳncái gì chung cho tất cả các thành phần GUI đều được tìm thấy trong lớp Component. Để tạo các đối tượng GUI chúng ta cần nhập gói java.awt hoặc gói javax.swing.

1.4 Mô hình xây dựng sản phẩm

|  |  |  |
| --- | --- | --- |
|  | **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) |

# KẾT QUẢ NGHIÊN CỨU

Hiện nay việc quản lý các nhóm sinh viên theo cách truyền thống dần không còn hiệu quả nữa do số lượng sinh viên ngày càng nhiều. Với sự phát triển của công nghệ thông tin việc xây dựng 1 phần mềm quản lý nhóm sẽ dễ dàng hơn cho giảng viên trong việc quản lý sinh viên. Để xây dựng chương trình nhóm chúng em đã chọn mô hình thác nước vì nó dễ quản lý, dễ hiểu và có nhiều tài liệu để cung cấp thông tin cho người đọc. Bước đầu tiên của quy trình là phân tích và xác định yêu cầu của phần mềm, sau khi thực hiện xong bước này thì sẽ thu được một tài liệu mô tả yêu cầu. Bước tiêp theo là thiết kế hệ thống để thu được các bản thiết kế giao diện, thiết kế thành phần và cấu trúc dữ liệu của chương trình. Sau khi thu được tài liệu mô tả yêu cầu và các bản thiết kế thì chúng em bắt đầu xây dựng từng module của chương trình sau đó tích hợp và kiểm thử toàn bộ hệ thống. Sau khi thực hiện các bước trên thì sản phẩm nhóm em thu về được là một ứng dụng chạy được. Dưới đâu là cụ thể từng bước trong quá trình xây dựng ứng dụng của nhóm chúng em.

## Khảo sát hệ thống

### Mục tiêu

Xây dựng chương trình quản lý các nhóm học môn lập trình Java của trường ĐHCNHN

### Tổng quan về các vấn đề bài toán giải quyết

Để xây dựng được chương trình thì cần phải giải quyết một số vấn đề sau:

* Quy mô của dự án
* Những tính năng cần có

Khảo sát sơ bộ

Phương pháp phỏng vấn:

|  |  |
| --- | --- |
| **Kế hoạch phỏng vấn** | |
| Người được hỏi : ThS. Vũ Thị Dương | Người phỏng vấn: Phạm Thanh Quý, Lê Đăng Quang |
| Địa chỉ: Trường đại học công nghiệp  Hà Nội | Thời gian hạn: Thứ bảy ngày 4 tháng 5 năm 2021.  Thời điểm bắt đầu: 8 giờ.  Thời điểm kết thúc : 9 giờ |
| Đối tượng được hỏi : Giáo viên bộ môn lập trình Java   * Dữ liệu cần thu thập: Dự án xây dựng chương trình quản lý nhóm học môn lập trình Java * Thỏa thuận về: Trao đổi thông tin về dự án xây dựng chương trình quản lý nhóm học môn lập trình Java. | Các yêu cầu đối với người được phỏng vấn:   * Vai trò: giáo viên bộ môn * Trình độ: Thạc sĩ * Kinh nghiệm: nhiều năm giảng dạy tại trường Đại học công nghiệp Hà Nội |
| Chương trình:   * Giới thiệu: Em là Lê Đăng Quang, đến từ nhóm được giao nhiệm vụ tìm hiểu dự án quản lý các nhóm học môn lập trình Java. * Tổng quan về dự án: Xây dựng chương trình quản lý nhóm học môn lập trình Java. * Tổng quan về phỏng vấn: Thu thập thông tin về hệ thống quản lý nhóm học môn lập trình Java * Chủ đề 1: Quản lý thành viên * Chủ đề 2: Quản lý hoạt động * Chủ đề 3: Tiêu chi đánh giá * Chủ đề 4: Đánh giá hoạt động * Tổng hợp các nội dung ý kiến của người được hỏi:   Kết thúc: Cảm ơn thông tin mà cô đã cung cấp. Thông tin sẽ được chúng em sử dụng để xây dựng chương trình quản lý nhóm học môn lập trinh java . | Ước lượng thời gian:   * -1 phút * -1 phút * -1 phút * -20 phút * -15 phút   - 5 phút   * - 5 phút * -2 phút |
|  | Dự kiến tổng thời gian: 50 phút |

|  |  |
| --- | --- |
| **PHIẾU TRẢ LỜI PHỎNG VẤN** | |
| Dự án: Xây đựng chương trình quản lý các nhóm học môn lập trình Java của trường ĐHCNHN | |
| Người được hỏi: Ths. Vũ Thị Dương | Người hỏi: Lê Đăng Quang  Ngày hỏi: 04/05/2021 |
| **Câu hỏi** | **Ghi chú** |
| Câu hỏi 1: Chương trình sẽ quản lý một hay nhiều lớp? Nếu nhiều thì khoảng bao nhiểu ? | Trả lời: Nhiều lớp.  Khoảng 4 đến 5 lớp |
| Câu hỏi 2: Giáo viên sẽ tạo tự động các tài khoản cho sinh viên hay sinh viên phải tự đăng kí ? | Trả lời:Giáo viên tạo tài khoản và cấp cho sinh viên |
| Câu hỏi 3: Những chức năng quản lý mà giáo viên muốn có trong phần mềm? | Trả lời:Quản lý lớp(tạo,sửa, xóa) giao việc và đánh giá. |
| Câu hỏi 4: Dữ liệu thì làm việc với Excel hay Sql sever ? | Trả lời: Cả hai |
| Câu hỏi 5: Hệ thống cần lưu những thôn tin gì để quản lí thông tin sinh viên? | Trả lời: Họ tên, mã sinh viên, số nhóm |
| Câu hỏi 6: Các tiêu chi mà giáo viên muốn đánh giá các nhóm ? | Trả lời: Đánh giá theo thời gian hoàn thành bài tập, thái độ |
| Câu hỏi 7: Nhóm trưởng của các nhóm có cần chức năng giao công việc cho từng thành viên trong nhóm không? | Trả lời: Có |
| Câu hỏi 8: Sinh viên có thể xem được kết quả học tập của bản thân không? | Trả lời: Có |
| Câu hỏi 9: Có cần phải sắp xếp các nhóm hay thứ tự các thành viên không ? nếu có thì theo tiêu chí nào ? | Trả lời:Có và sắp xếp theo số thứ tự của đề tài hoặc alpha beta. |
| Câu hỏi 10: Sinh viên thể xem lịch học trong chương trình không? | Trả lời:Không. Sinh viên có thể xem lịch học trên trang trường. |
| Đánh giá chung:   * Người được hỏi có thái độ rất thân thiện, nhiệt tình, trả lời đúng trọng tâm câu hỏi, đưa ra thông tin chính xác, đầy đủ. * Đã thu thập được đầy đủ thông tin về chủ đề phỏng vấn. | |

### Tài liệu đặc tả yêu cầu người dùng

#### Hoạt động chính của hệ thống

* Ở trang chính của hệ thống, cho phép giáo viên và sinh viên **đăng nhập** vào hệ thống.
* Đối với **giáo viên** :

+ Sau khi đăng nhập thành công sẽ hiện thị **Menu 1** chức năng bao gồm : **Quản lý lớp học , Đánh giá sinh viên, Giao việc( bài tập), Đăng xuất.**

**∙ Quản lý lớp học**

Khi giáo viên chọn chức năng **Quản lý lớp học** hệ thống sẽ hiện thị **Menu 1\_1** chức năng bao gồm : **Tạo lớp, xóa lớp, sửa lớp, tìm lớp, quản lý thành viên lớp, xuất file, quay lại.**

**∙ Quản lý thành viên lớp**

Khi giáo viên chọn chức năng **Quản lý thành viên lớp** hệ thống sẽ hiện thi danh sách lớp được chọn và **Menu 1\_1\_1** chức năng bao gồm : **Thêm sinh viên, sửa thông tin sinh viên, xóa sinh viên, sắp xếp theo tên sinh viên, sắp xếp theo mã sinh viên, tạo thêm sửa nhóm, quay lại.**

**∙ Đánh giá sinh viên**

Khi giáo viên chọn chức năng **Đánh giá sinh viên** hệ thống sẽ hiện thị **Menu 1\_2** chức năng bao gồm : **Đánh giá điểm, Quay lại** và hiện mục yêu cầu giáo viên chọn lớp muốn đánh giá**.**

**∙ Giao việc**

Khi giáo viên chọn chức năng Giao việc hệ thống sẽ hiện thị **Menu 1\_3** chức năng bao gồm : **Thông báo, Gửi đề tài, Quay lại** và hiện mục yêu cầu giáo viên chọn lớp muốn giao việc**.**

* Đối với Sinh viên :

+ Khi sinh viên đăng nhập thành công, hệ thống sẽ hiện thi **Menu2** chức năng bao gồm : **Nộp bài, Xem đánh giá, Xem thông báo, Đăng xuất.**

* Đói với Snh viên –nhóm trưởng :

+ Khi sinh viên đăng nhập thành công, hệ thống sẽ hiện thi **Menu2** chức năng bao gồm :  **Chọn đề tài, Thông báo, Nộp bài, Xem đánh giá, Xem thông báo, Đăng xuất.**

#### Yêu cầu chức năng

* Đăng nhập.
* Đăng xuất.
* Quay lại.
* Tạo lớp, sửa lớp.
* Thêm sinh viên, sửa thông tin sinh viên, xóa sinh viên.
* Sắp xếp sinh viên theo tên, sắp xếp sinh viên theo mã.
* Đánh giá sinh viên theo điểm.
* Thông báo
* Gửi đề tài.
* Chọn đề tài
* Nộp bài
* Xem thông báo
* Xem đánh giá

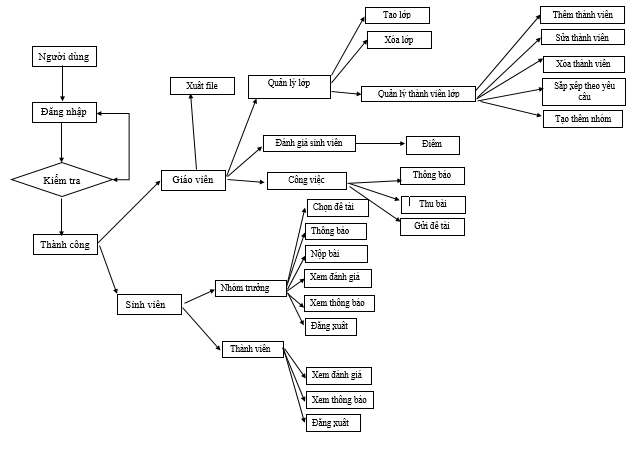
#### Yêu cầu phi chức năng

* **Yêu cầu về giao diện**:  Giao diện thân thiện với người dùng trong các tương tác giữa hệ thống với các đối tượng sử dụng. Yêu cầu này làm tăng mức độ dễ sử dụng của người dùng.
* **Yêu cầu hệ thống**: Hoạt động tốt, thời gian phản hồi không quá 10s.
* **Khả năng thích nghi**: Phần mềm có thể chạy được trên tất cả các phiên bản windows
* **Tính tái sử dụng**: Phần mềm có thể dễ dàng cài đặt nâng cấp bảo trì.

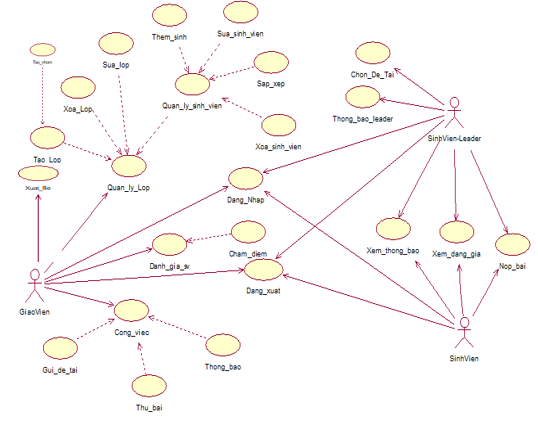
## Phân tích hệ thống

### Mô hình hóa chức năng hệ thống

* Mô tả cơ bản về hoạt động của chương trình



Biểu đồ Use Case :



#### Mô tả use case <Quản lý lớp > (Hồ Hữu Quang)

* **Tên use case :** Quản lý lớp.
* **Mô tả vắn tắt :** Use case cho phép giáo viên quản lý được các lớp học và thành viên của lớp học.
* **Luồng sự kiện**
  + **Luồng cơ sở**
    - * Use case này bắt đầu khi giáo viên bấm vào nút **Quản lý lớp học** ở trên Menu. Hệ thống sẽ hiện thị các chức năng **tạo lớp, xóa lớp, quản lý sinh viên.**
        + **Tạo lớp:**

Khi giáo viên chọn chức năng **tạo lớp** , hệ thống sẽ yêu cầu giao viên chọn file excel thông tin lớp học.

Sau khi giáo viên chọn file excel chưa thông tin lớp và chọn **tạo lớp**. hệ thống sẽ thu thập thông tin từ excel và yêu cầu chọn tạo nhóm theo **ramdom** hoặc theo **số thứ tự**.

Sau khi chọn tiêu chí tạo nhóm hệ thống sẽ tự động tạo nhóm và hệ thống sẽ yêu cầu giáo viên chọn sinh nhóm trưởng cho từng nhóm.

Sau khi chọn nhóm trưởng cho từng nhóm và click xác nhận. Hệ thống sẽ tự động tạo tài khoản cho sinh viên với tài khoản là mã sinh viên, mật khẩu mặc định và lưu thông tin vào bảng TAIKHOAN, NHOM, SINHVIEN và LOPHOC.

* + - * + **Xóa lớp:**

Để thực hiện chức năng xóa lớp, giáo viên cần phải chọn vào lớp cần xóa và click vào **xóa lớp.** Hệ thống sẽ hiện thị yêu cầu xác nhận có chắc chắn xóa lớp hay không.

Sau giáo viên khi xác nhận chắc chắn xóa lớp . Hệ thống sẽ xóa lớp được chọn khỏi CSDL.

* + - * + **Quản lý sinh viên lớp:**

Khi giáo viên chọn chức năng **quản lý sinh viên lớp,** hệ thống sẽ lấy tên lớp từ bảng LOPHOC và yêu cầu giáo viên chọn lớp cần quản lý.

Sau khi giáo viên chọn lớp học, hệ thống sẽ lấy thông tin từ bảng SINHVIEN và hiện thi ra bảng danh sách sinh viên của lớp bao gồm các chức năng **thêm sinh viên, sửa thông tin sinh viên, xóa sinh viên, sắp xếp sinh viên.**

**Thêm sinh viên:**

Giáo viên sẽ nhập các thông tin sau đó click vào **thêm sinh viên,** hệ thống sẽ lưu thông tin vào bảng SINHVIEN và tự động tạo tài khoản cho sinh viên và lưu vào bảng TAIKHOAN sau đó hiên thị lại danh sách sinh viên.

**Xóa sinh viên:**

Sự kiện xảy ra khi giáo viên chọn sinh viên cần xóa và click vào **xóa sinh viên,** hệ thống sẽ yêu cầu xác nhận chắc chắn muốn xoá sinh viên không.

Sau khi giáo viên xác nhận xóa sinh viên, hệ thống sẽ xóa toàn bộ dữ liệu của sinh viên khỏi CSDL và hiên thị lại danh sách sinh viên.

**Sửa thông tin sinh viên:**

Sự kiện xảy ra khi giáo viên chọn vào sinh viên muốn sửa thông tin danh sách, sửa các thông tin trong các của sinh và click vào **Sửa thông tin sinh viên,** Hệ thống sẽ yêu cầu xác nhận chắc chắn muốn sửa thông tin sinh viên không.

Sau khi giáo viên xác nhận sửa thông sinh viên, hệ thống sẽ cập nhật lại thông tin sinh viên trong bảng SINHVIEN và hiên thị lại danh sách sinh viên.

**Sắp xếp sinh viên:**

Sự kiện xảy ra khi giáo viên chọn tiêu chcần sắp xếp trong p và click vào **sắp xếp,** hệ thống sẽ sắp xếp sinh viên theo tiêu chi đã chọn và hiện thị lại danh sách sinh viên.

* + **Luồng rẽ nhánh**
    - 1. Tại bước 1 hoặc bước 2 của **tạo lớp** khi giáo viên chọn **quay lại .** sự kiện sẽ kết thúc.
      2. Tại bước 1 của **xóa lớp** khi giáo viên chọn **không** . sự kiện sẽ kết thúc .
      3. Tại bước 1 của **thêm sinh viên** nếu mã sinh viên đã tồn tại trong CSDL thì sẽ thông báo “Mã sinh viên đã tồn tại yêu cầu nhập lại”. Ngược lại vẫn tiếp tục.
      4. Tại bước 1 của **xóa sinh viên** nếu giáo viên không chọn sinh viên trong bảng, hệ thống sẽ thông báo “Bạn hãy chọn sinh viên cần xóa”.
      5. Tại bước 2 của **xóa sinh viên** khi giáo viên chọn **không** . sự kiện sẽ kết thúc .
      6. Tại bước 2 của **sửa thông tin sinh viên** nếu mã sinh viên không tồn tại trong CSDL hệ thống sẽ thông báo “Mã sinh viên đã tồn tại yêu cầu nhập lại”.
      7. Tại bất cứ thời điểm nào trong quá trình thực hiện use case, nếu không kết nối được với CSDL thì hệ thống sẽ hiện thị ra một bảng báo lỗi và use case kết thúc.
* **Các yêu cầu đặc biệt**
  + Không có.
* **Tiền điều kiện**
  + Không có.
* **Hậu điều kiện**
  + Không có.
* **Điểm mở rộng**
  + Không có.

#### Mô tả use case <Chọn đề tài> (Lê Đăng Quang)

Use case này cho phép nhóm trưởng chọn đề tài cho nhóm

* **Luồng sự kiện**
* **Luồng cơ bản**

1. Use Case này bắt đầu khi trưởng nhóm chọn chức năng “Chọn đề tài” trên menu
2. Hệ thống sẽ truy cập vào bảng DETAI trong CSDLvà hiển thị mã đề tài, tên đề tài lên màn hình
3. Trưởng nhóm click vào đề tài muốn chọn và click vào nút “Chọn”
4. Hệ thống sẽ hiện lên 2 lựa chọn “Xác nhận” và “Hủy”
5. Sinh viên nhấn vào “Xác nhận”
6. Hệ thống thông báo “Thành công”, đánh dấu đề tài đã chọn, ẩn đề tài đó khỏi danh sách đề tài và cập nhật lại CSDL

* **Luồng rẽ nhánh**

1. Tại bước 4 ở luồng cơ bản. Nếu trưởng nhóm click vào “Hủy” use case kết thúc.
2. Tại bất kì thời điểm nào trong quá trình thực hiện use case nếu không kết nối được tới CSDL thì hệ thống sẽ hiện ra một bảng báo lỗi và use case kết thúc.

* **Các yêu cầu đặc biệt**

Không có

* **Tiền điều kiện**

Sinh viên phải đăng nhập vào hệ thống bằng tài khoản nhóm trưởng

* **Hậu điều kiện**

Không có

* **Điểm mở rộng**

Không có

#### Mô tả use case < Thông báo> (Lê Đăng Quang)

Use case này cho phép nhóm trưởng gửi thông báo tới các thành viên trong nhóm

* **Luồng sự kiện**
  + **Luồng cơ bản**
    1. Use case bắt đầu khi nhóm trưởng chọn chức năng “Thông báo”
    2. Hệ thống sẽ hiển thị form nhập thông báo
    3. Nhóm trưởng nhập thông báo vào form và nhấn “Gửi”
    4. Hệ thống sẽ lưu thông báo vào bảng THONGBAO, hiển thị thông báo “Gửi thông báo thành công” và cập nhật lại cơ sở dữ liệu
  + **Luồng rẽ nhánh** 
    1. Tại bước 4 ở luồng cơ bản, nếu nhóm trưởng nhấn vào “Hủy” use case kết thúc.
    2. Tại bất kì thời điểm nào trong quá trình thực hiện use case nếu không kết nối được tới CSDL thì hệ thống sẽ hiện ra một bảng báo lỗi và use case sẽ kết thúc
* **Các yêu cầu đặc biệt**

Không có

* **Tiền điều kiện**

Sinh viên phải đăng nhập vào hệ thống bằng tài khoản nhóm trưởng

* **Hậu điều kiện**

Không có

* **Điểm mở rộng**

Không có

#### Mô tả use case <Xem đánh giá> (Lê Đăng Quang)

Use case này cho phép sinh viên xem đánh giá từ giáo viên

* **Luồng sự kiện**
  + **Luồng cơ bản**
    1. Use case bắt đầu khi Sinh viện chọn chức năng “Xem đánh giá”
    2. Hệ thống sẽ truy cập vào bảng DANHGIA và hiển thị tên lớp, mã nhóm, mã sinh viên, tên sinh viên và đánh giá lên màn hình
    3. Sinh viên click vào “Thoát”, use case kết thúc
  + **Luồng rẽ nhánh** 
    1. Tại bất kì thời điểm nào trong quá trình thực hiện use case nếu không kết nối được tới CSDL thì hệ thống sẽ hiện ra một bảng báo lỗi và use case sẽ kết thúc
* **Các yêu cầu đặc biệt**

Không có

* **Tiền điều kiện**

Sinh viên phải đăng nhập vào hệ thống bằng tài khoản sinh viên

* **Hậu điều kiện**

Không có

* **Điểm mở rộng**

Không có

#### Mô tả use case <Đăng nhập>( Phạm Ích Quảng)

Use case này cho phép người dùng đăng nhập vào hệ thống

* Luồng sự kiện:
* Luồng cơ bản:

1. Use case này bắt đầu khi người dùng ấn vào nút “Đăng nhập”.
2. Xuất hiện Form đăng nhập và 2 nút “Xác nhận” và “Thoát”.

Người dùng nhập tài khoản mật khẩu rồi click “Xác nhận”.

1. Hệ thống sẽ truy cập vào bảng TAIKHOAN để kiểm tra quyền của tài khoản và hiển thị ra màn hình:

* Giáo viên: Hiển thị bảng quản lý các lớp, giao việc và đánh giá sinh viên.
* Nhóm trưởng:HIển thị chọn đề tài, thành viên trong nhóm, thông báo, nộp bài, xem đánh giá, xem thông báo.
* Sinh viên:Hiển thị thành viên trong nhóm, xem thông báo, nộp bài, xem đánh giá.
* Luồng rẽ nhánh:

1. Tại bất kì thời điểm nào trong quá trình thực hiện use case, khi hệ thống không thể kết nối với CSDL thì hệ thống sẽ hiển thị bảng thông báo lỗi và use case kết thúc.
2. Tại bước 3 trong luồng cơ bản, nếu người dùng nhập sai tài khoản hệ thống sẽ thông báo lỗi và use case kết thúc.
3. Tại bước 2 trong luồng cơ bản, nếu người dùng nhấn nút “Thoát” use case kết thúc.

* Các yêu cầu đặc biệt: Không có
* Tiền điều kiện: Người dùng muốn đăng nhập để sử dụng ứng dụng
* Hậu điều kiện: Không có
* Điểm mở rộng: Không có.

#### Mô tả use case <Đánh giá>( Phạm Ích Quảng)

Use case này cho phép giáo viên đánh giá và cho điểm bài làm của sinh viên.

* Luồng sự kiện:
* Luồng cơ bản:

Use case bắt đầu khi giáo viên nhấn vào nút “Đánh giá”

Hệ thống sẽ truy cập vào bảng LOPHOC và hiển thị lớp học trong 1 ComboBox để chọn lớp để đánh giá và cho điểm.

Sau khi giáo Hệ thống hiển thị ra danh sách nhóm và các đề tài.

Sau khi nhấn vào nhóm bất kì,hệ thống hiển thị ra Form gồm mã lớp và điểm để giáo viên nhập đánh giá và hiển thị nút “Thêm”, ”Sửa” và “Hủy” . Sau khi thêm, điểm sẽ được lưu vào bảng DANHGIA và cập nhập lại CSDL.

* Luồng rẽ nhánh:

1. Tại bất kì thời điểm nào trong quá trình thực hiện use case, nếu không kết nối được với CSDL thì hệ thống sẽ hiển thị một bảng báo lỗi và use case kết thúc
2. Tại bước 2 ở đánh giá, nếu giáo viên ấn “Hủy” thì use case sẽ kết thúc.
3. Tại bước 3 của luồng sự kiện cơ bản, nếu gáo viên nhấn “Hủy” thì use case kết thúc

* Các yêu cầu đặc biệt: Không có.
* Tiền điều kiện**:** Giáo viên phải đăng nhập vào hệ thống.
* Hậu điều kiện:Không có.
* Điểm mở rộng:Không có.

#### Mô tả use case <Công việc>(Phạm Thanh Quý)

Use case này cho phép giáo viên tùy chọn các thao tác như “gửi đề tài”, “thông báo”, “thu bài” cho sinh viên.

* Luồng sự kiện:
* Luồng cơ bản:
  + - * 1. Use case này sẽ bắt đầu khi giáo viên click vào “Công việc” . Hệ thống sẽ hiển thị các chức năng của công việc lên màn hình.

2. Thông báo :

Khi giáo viên click vào ô “Thông báo”, hệ thống sẽ hiển thị form để nhập thông báo.

Sau khi nhập thông báo và đính kèm file, giáo viên ấn “Gửi”. Hệ thống sẽ kiểm tra thông tin và hiển thị thông báo xác nhận

Giáo viên click vào “Yes”, thông báo sẽ được cập nhật vào bảng THONGBAO và hệ thống sẽ hiển thị thông báo thành công.

Hệ thống cập nhật thông tin vào CSDL

Khi giáo viên click vào “Quay lại” thì sẽ quay về màn hình chính.

3. Thu bài:

Khi giáo viên click vào nút “Thu bài”, hệ thống sẽ truy cập vào bảng BAITAP(mã bài tập, ghi chú, link bài tập, mã nhóm, mã sinh viên)

Giáo viên chọn lớp và click vào xuất file, hệ thống sẽ lấy thông tin từ bảng BAITAP(mã bài tập, ghi chú, link bài tập, mã nhóm) rồi tạo ra 1 file excel và hiển thị lên màn hình.

4. Gửi đề tài:

Khi giáo viên click vào nút “Gửi đề tài”, hệ thống sẽ hiển thị form nhập đề tài

Giáo viên click vào “open file” để mở file đề tài, sau đó click vào “upload” để gửi đề tài.

Hệ thống sẽ kiểm tra thông tin và thông báo xác nhận

Giáo viên click vào “Xác nhận”, file đề tài sẽ được cập nhật vào bảng DETAI. Hệ thống cập nhật lại CSDL

Khi giáo viên bấm “Quay lại” sẽ quay về màn chính.

1. Use case sẽ kết thúc khi nhân viên click nút “Quay về màn hình chính”.
   * + Luồng rẽ nhánh:
       1. Tại bước 2.c ở luồng cơ bản nếu giáo viên click vào nút “Hủy” thì Usecase sẽ kết thúc.
       2. Tại bước 4.d ở luồng cơ bản nếu giáo viên click vào nút “Hủy” thì use case sẽ kết thúc.
   * Các yêu cầu đặc biệt:
     + Use case chỉ được thực hiện bởi giáo viên.
   * Tiền điều kiện:

Không có

* + Hậu điều kiện:

Không có

* + Điểm mở rộng:

Không có

#### Mô tả use case <Đăng xuất> (Phạm Thanh Quý)

* **Luồng sự kiện**
  + **Luồng cơ sở**

1. Use case này bắt đầu khi giáo viên bấm vào nút “Đăng xuất” ở trên màn hình.

2. Hệ thống sẽ hiển thị yêu cầu xác nhận từ giáo viên

3. Giáo viên xác nhận đăng xuất

4. Hệ thống đăng xuất tài khoản giáo khỏi hệ thống.

* + **Luồng rẽ nhánh**

1. Tại bất cứ thời điểm nào trong quá trình thực hiện use case, nếu không kết nối được với CSDL thì hệ thống sẽ hiện thị ra một bảng báo lỗi và use case kết thúc.

2. Nếu giáo viên click “Huy” thì hệ thống sẽ giữ nguyên hiện trạng.

* **Các yêu cầu đặc biệt**

Không có.

* **Tiền điều kiện**

Không có.

* **Hậu điều kiện**

Không có.

* **Điểm mở rộng**

Không có.

#### Mô tả Use case <Nộp bài> (Mẫn Văn Quân)

* **Mô tả vắn tắt:** Use case này cho phép sinh viên nộp bài cho giáo viên
* **Luồng sự kiện:**
* **Luồng cơ bản:**

1. Use case này bắt đầu khi sinh viên click chọn “Nộp bài” trên thanh menu chính. Sau đó hệ thống sẽ truy cập vào bảng BAITAP để hiển thị một form nộp bài.
2. Sinh viên sẽ gửi link bài tập lên form(+ ghi chú) rồi click vào “Gửi bài”.
3. Sinh viên click vào “Yes”, hệ thống sẽ kiểm tra tính hợp lệ của thông tin và hiển thị thông báo. Nếu hợp lệ thì sẽ được cập nhập vào bảng BAITAP và hệ thống sẽ hiển thị thông báo “Nộp bài thành công”.
4. Hệ thống sẽ cập nhập thông tin vào CSDL.
5. Khi sinh viên click chọn “Quay lại” thì sẽ quay lại màn hình chính và Use Case kết thúc.

* **Luồng rẽ nhánh:**

1. Tại bước 3 trong luồng cơ bản, nếu sinh viên click vào “No” thì UC sẽ kết thúc
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 thông báo lỗi và Use Case kết thúc.

* **Các yêu cầu đặc biệt:** Không có
* **Tiền điều kiện:** Sinh viên cần phải đăng nhập vào hệ thống bằng tài khoản sinh viên
* **Hậu điều kiện:**Không có.
* **Điểm mở rộng:** Không có

#### Mô tả Use case <Xem thông báo> (Mẫn Văn Quân)

* **Mô tả vắn tắt:** Use case này cho phép sinh viên xem thông báo của giáo viên và nhóm trưởng.
* **Luồng sự kiện:**
* **Luồng cơ bản:** 
  1. Use case này bắt đầu khi sinh viên click vào “Xem thông báo” trên thanh menu chính.
  2. Hệ thống sẽ lấy thông tin từ bảng THONGBAO và hiển thị lên màn hình thành 1 bảng danh sách các thông báo.
  3. Khi sinh viên click chọn “Quay lại” thì sẽ quay lại màn hình chính và Use Case kết thúc.
* **Luồng rẽ nhánh:**

1. Tại bất kỳ thời điểm nào trong quá trình thực hiện UC nếu không kết nối được với CSDL thì hệ thống sẽ thông báo lỗi và Use Case kết thúc.

* **Các yêu cầu đặc biệt:** Không có.
* **Tiền điều kiện:** Sinh viên cần phải đăng nhập vào hệ thống bằng tài khoản sinh viên.
* **Hậu điều kiện:** Không có.
* **Điểm mở rộng:** Không có.

#### Mô tả Use case <Xuất file> (Mẫn Văn Quân)

* **Mô tả vắn tắt:** Use case này cho phép giáo viên xuất điểm va đánh giá của các nhóm ra file Excel.
* **Luồng sự kiện:**
* **Luồng cơ bản:** 
  1. Use case này bắt đầu khi giáo viên click vào “Xuất file” trên thanh menu chính.
  2. Hệ thống sẽ lấy thông tin từ bảng LOPHOC (tên lớp) để hiển thị ra màn hình danh sách các lớp mà giáo viên dạy.
  3. Giáo viên chọn lớp muốn xuất file và click vào Xuất File, hệ thống sẽ lấy thông tin từ bảng DANHGIA (bao gồm mã đánh giá, mã nhóm và điểm) rồi tạo ra 1 file Excel và hiển thị lên màn hình.
* **Luồng rẽ nhánh:**
  1. Tại bất kỳ thời điểm nào trong quá trình thực hiện UC nếu không kết nối được với CSDL thì hệ thống sẽ thông báo lỗi và Use Case kết thúc.
* **Các yêu cầu đặc biệt:** Không có.
* **Tiền điều kiện:** Giáo viên cần phải đăng nhập vào hệ thống bằng tài khoản giáo viên.
* **Hậu điều kiện:** Không có.
* **Điểm mở rộng:** Không có.

### Mô hình hóa dữ liệu của hệ thống.

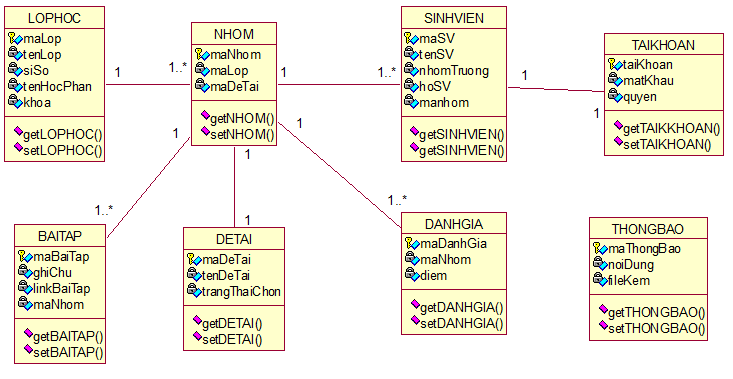
#### Mô tả thực thể

* LOPHOC : lưu trữ các thông tin lớp học gồm :mã lớp, tên lớp, sĩ số, khóa, tên học phần.
* TAIKHOAN : lưu trữ các thông tin tài khoản của sinh viên và giáo viên : tài khoản, mật khẩu, quyền.
* NHOM : lưu trữ các thông tin của nhóm bao gồm : mã nhóm, mã lớp, mã đề tài.
* SINHVIEN : lưu trữ các thông tin của sinh viên bao gồm : mã sinh viên,họ sinh viên, tên sinh viên, mã nhóm.
* THONGBAO : lưu trữ các thông báo của nhóm trưởng hoặc giáo viên bao gồm : mã thông báo, nội dung, file đính kèm .
* BAITAP : lưu trữ các thông tin liên quan đến bài tập lớn bao gồm : mã bài tập, ghi chú, file bài tập, mã nhóm.
* DETAI : lưu trữ các đề tài của bài tập lớn bao gồm : mã đề tài, tên đề tài, trang thái chọn.
* DANHGIA : lưu trữ các đánh giá của giáo viên bao gồm : mã đánh giá, mã nhóm, điểm.

#### Mô tả các quy tắc

* Mỗi lớp có nhiều nhóm. Mỗi nhóm có thể có 1 hoặc nhiều sinh viên.Mỗi sinh viên có một tài khoản.
* Mỗi sinh viên có 1 nhóm. Mỗi nhóm có 1 đề tài. Mỗi nhóm có một hoặc nhiều lân gửi bài tập.
* Mỗi nhóm có một hoặc nhiều đánh giá.

#### Quan hệ giữa các lớp

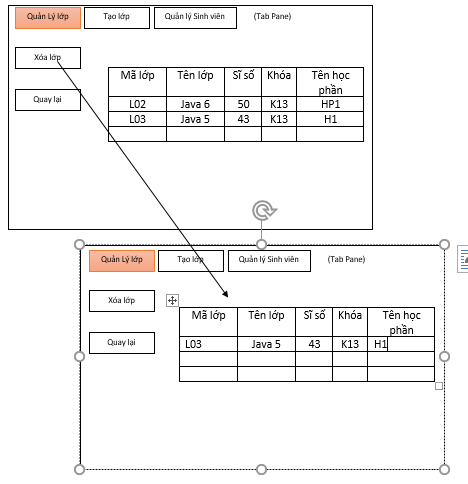


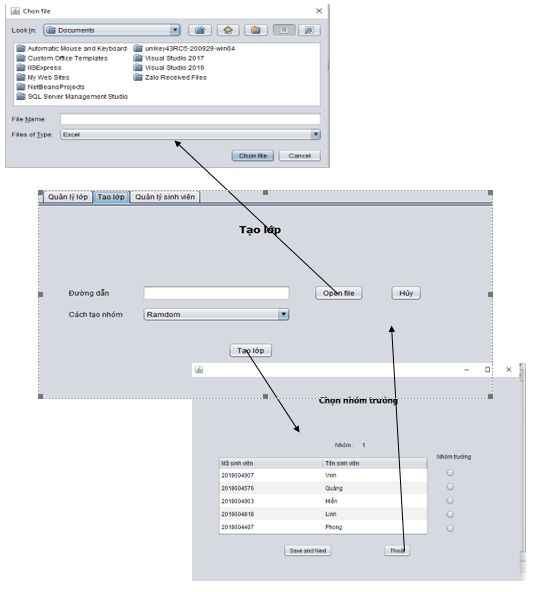
## Thiết kế hệ thống

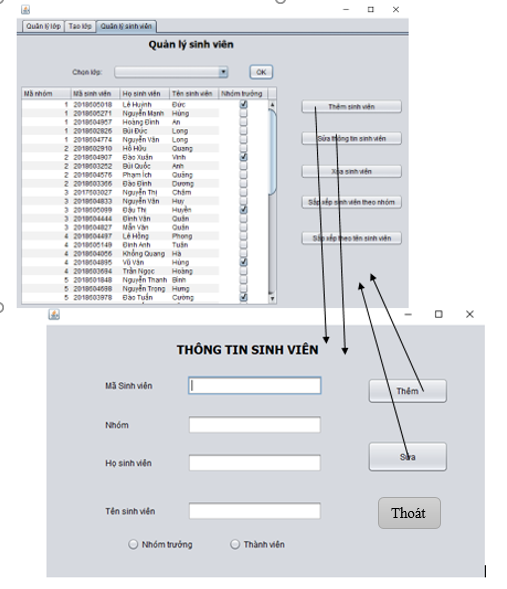
### Thiết kế giao diện

#### Quản lý lớp <Hồ Hữu Quang>

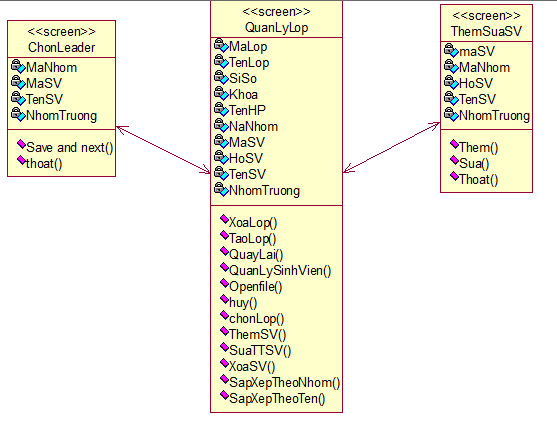
Hình dung màn hình





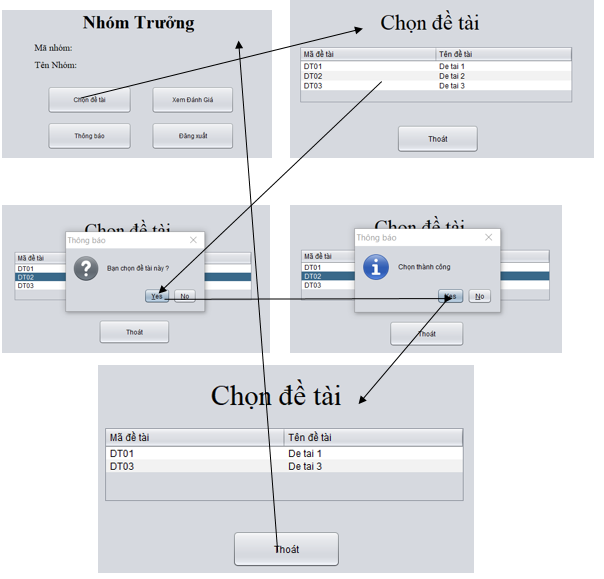


Biểu đồ lớp màn hình :

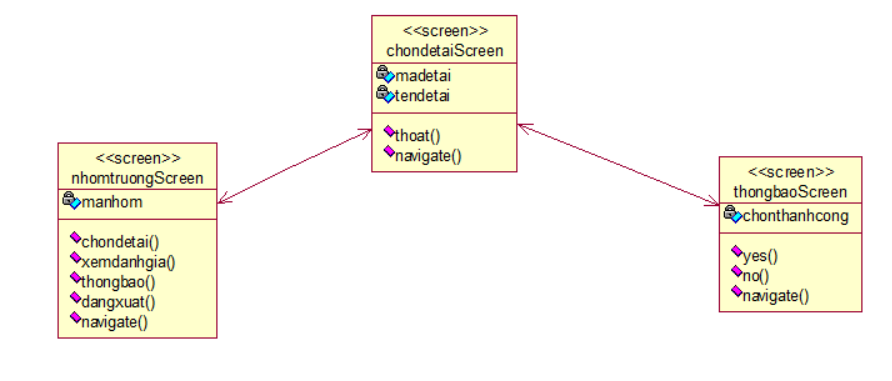


#### Chọn đề tài<Lê Đăng Quang>

Hình dung màn hình :

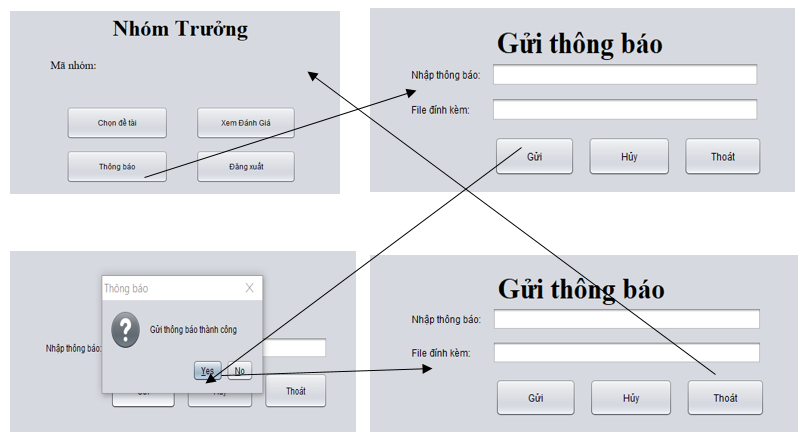


Biểu đồ lớp màn hình :

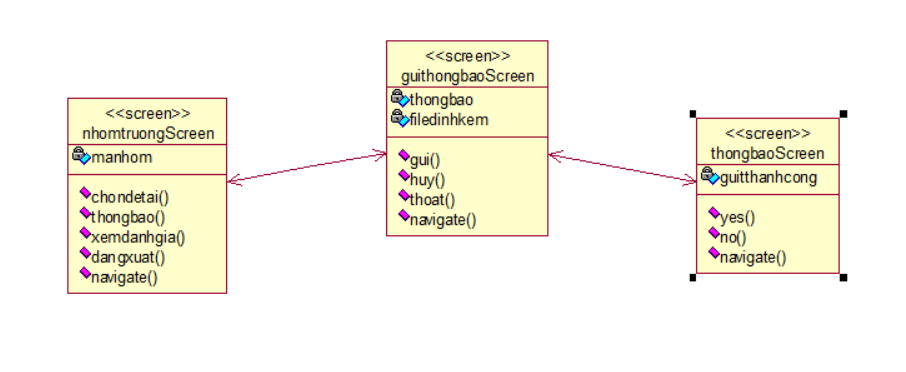


#### Thông báo <Lê Đăng Quang>

Hình dung màn hình :

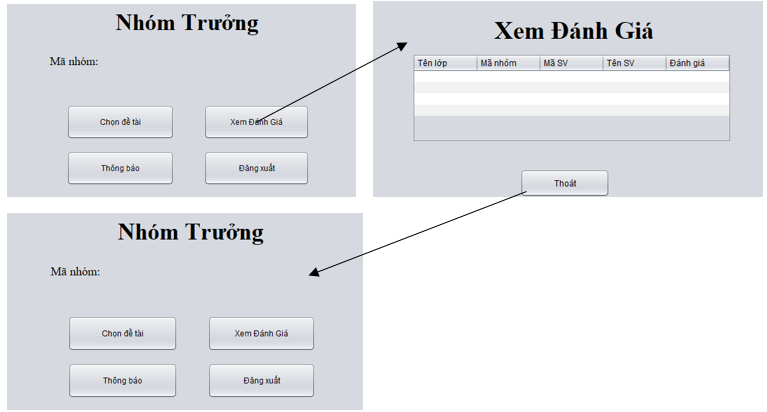


Biểu đồ lớp màn hình :



#### Xem đánh giá<Lê Đăng Quang>

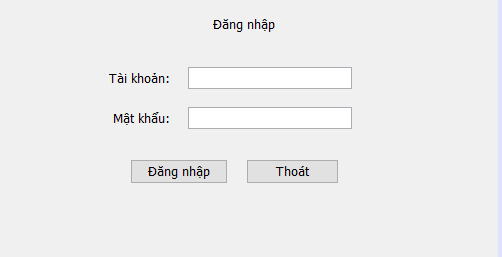
Hình dung màn hình :

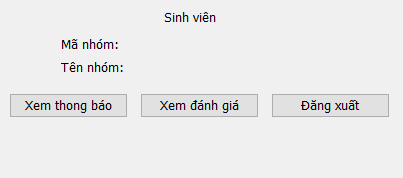
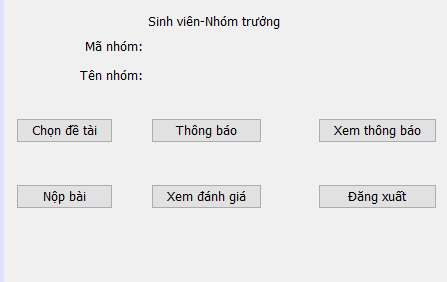
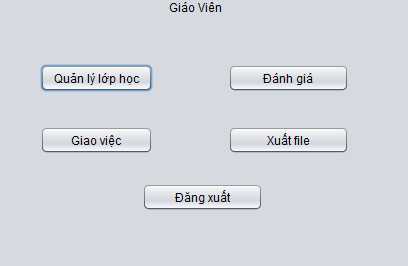


Biểu đồ lớp màn hình :



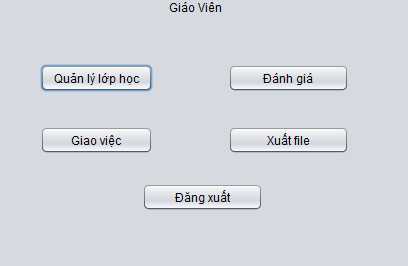
#### Đăng nhập<Phạm Ích Quảng>

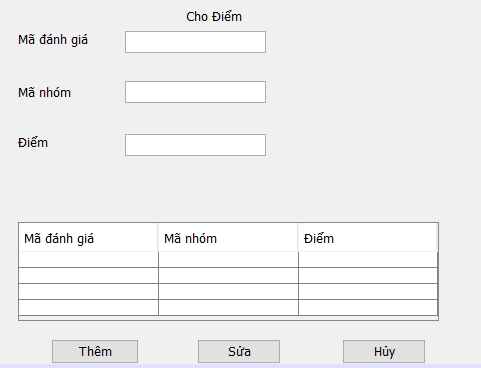


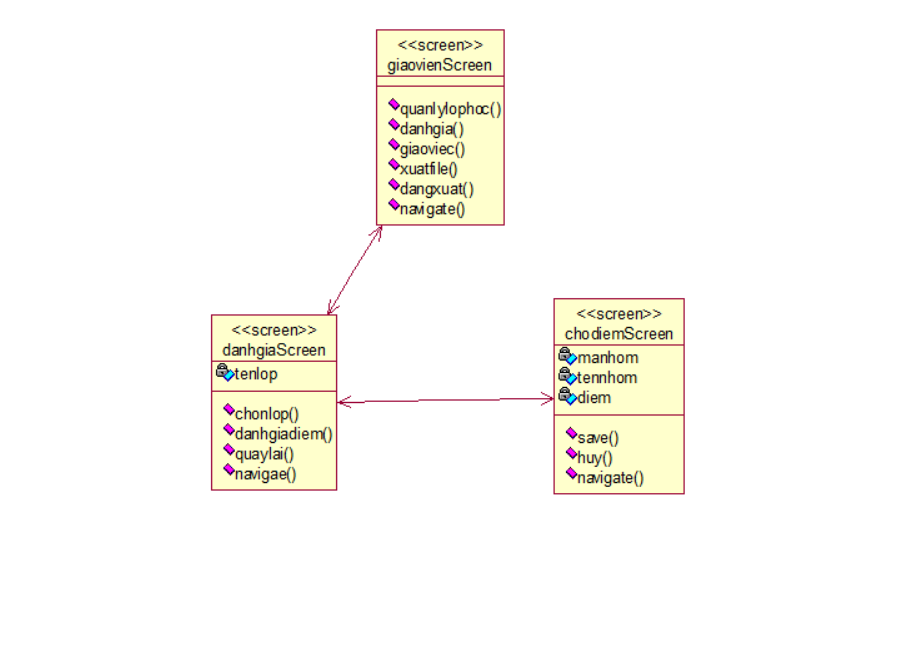




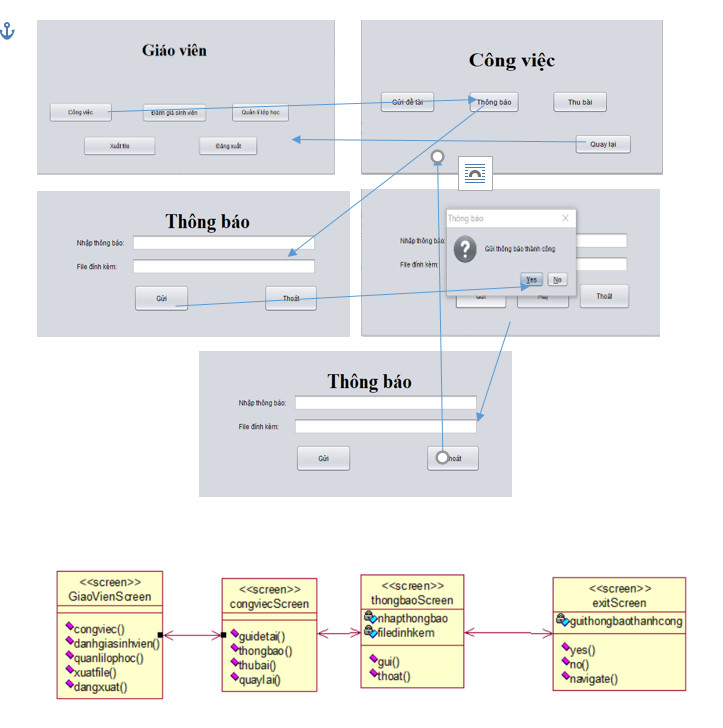
#### Đánh giá<Phạm Ích Quảng>

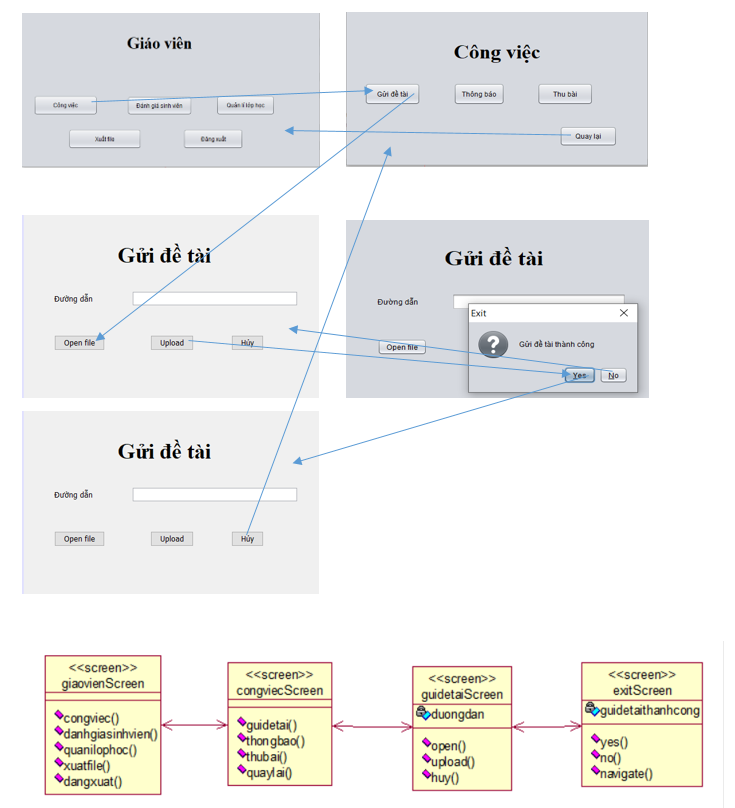


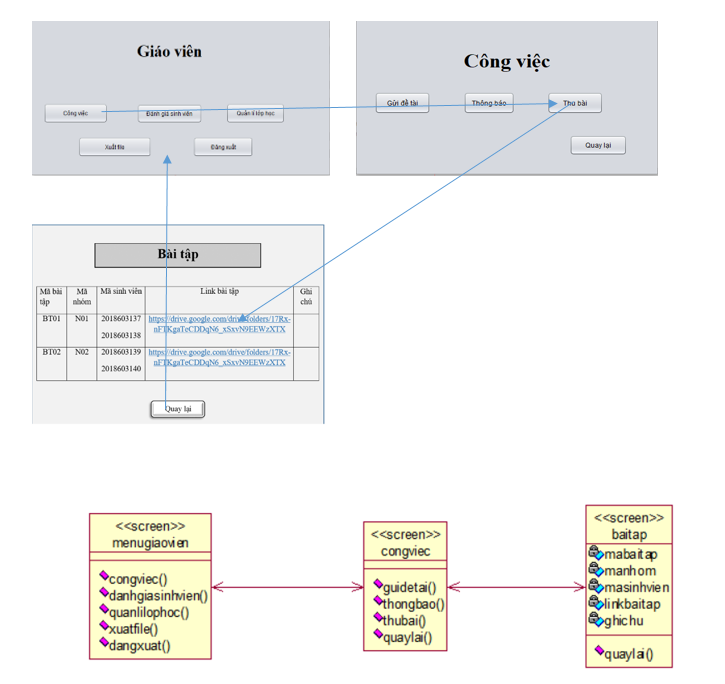




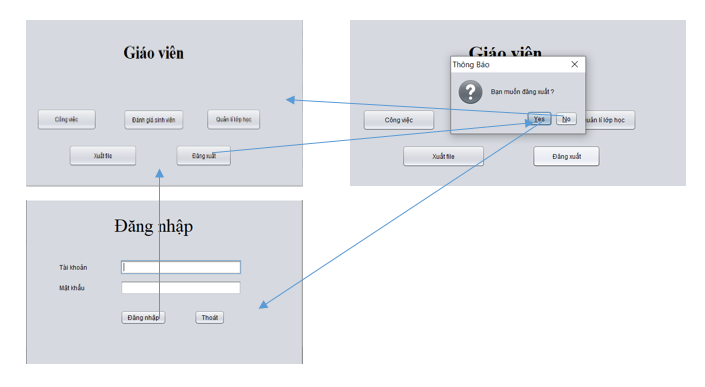
#### Công Việc<Phạm Thanh Quý>

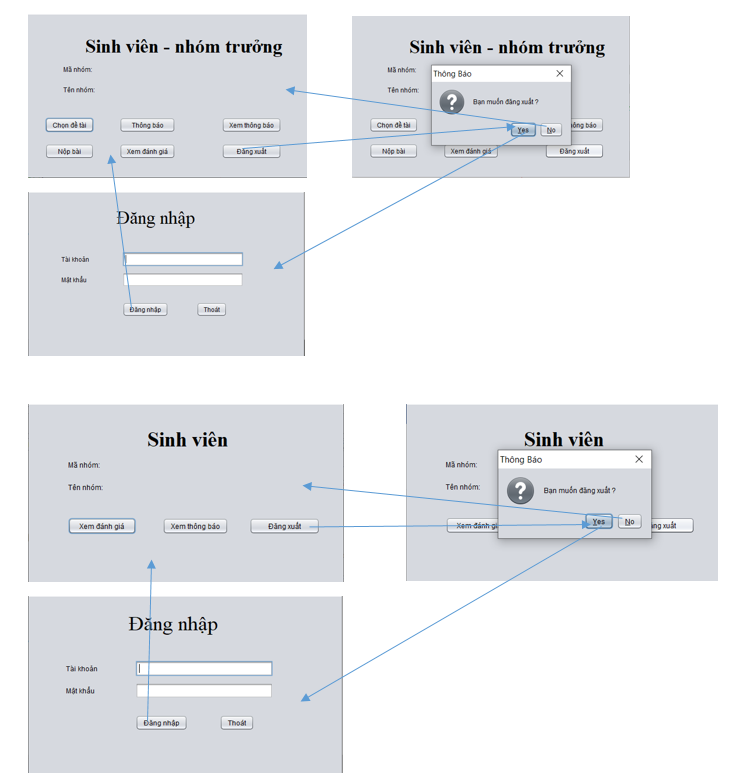


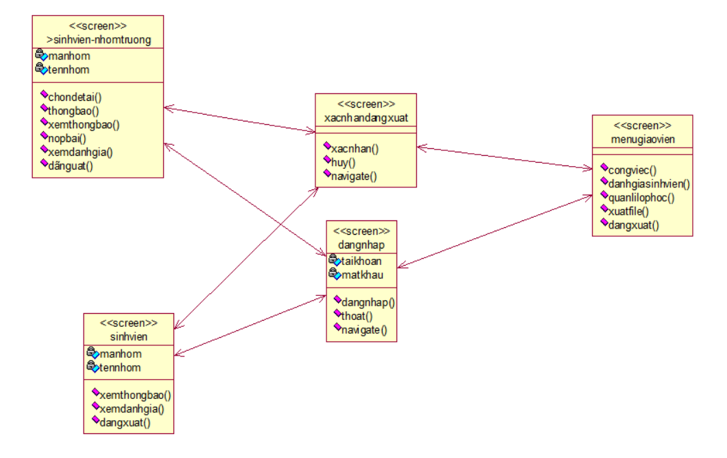




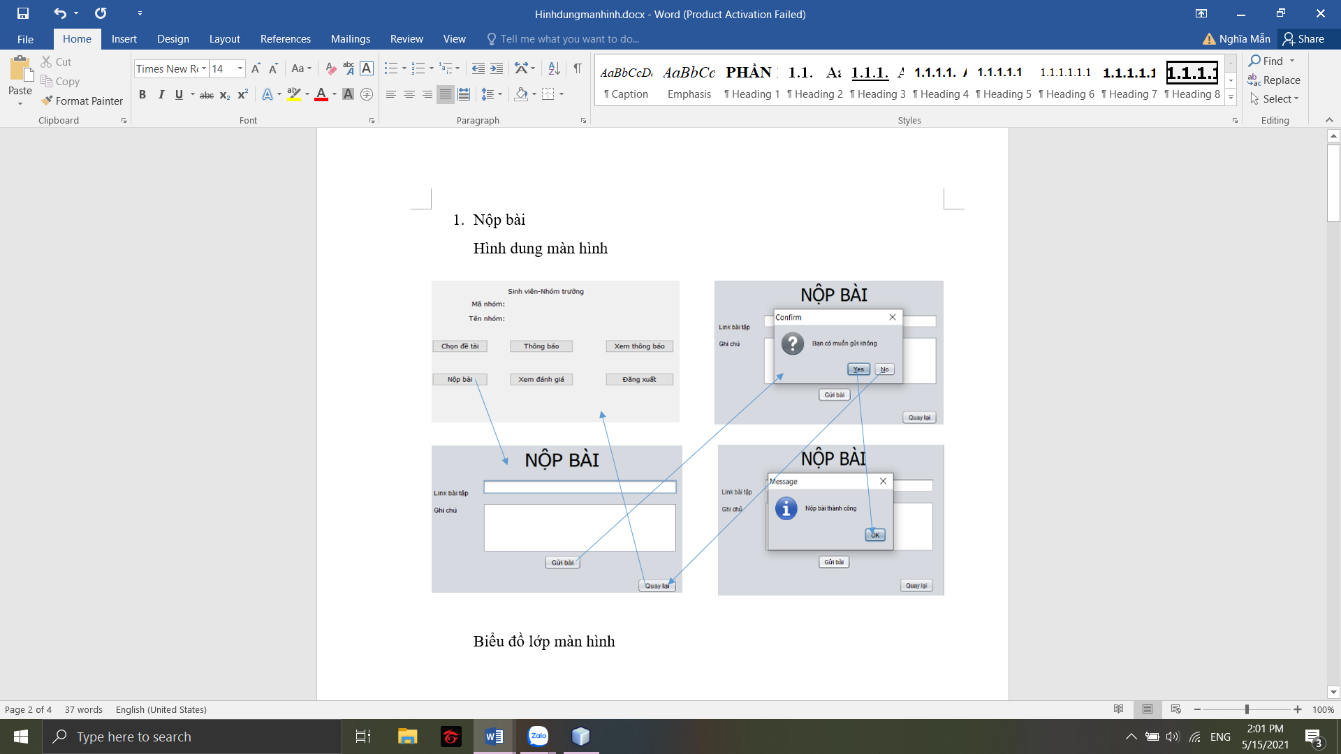
#### Đăng xuất<Phạm Thanh Quý>



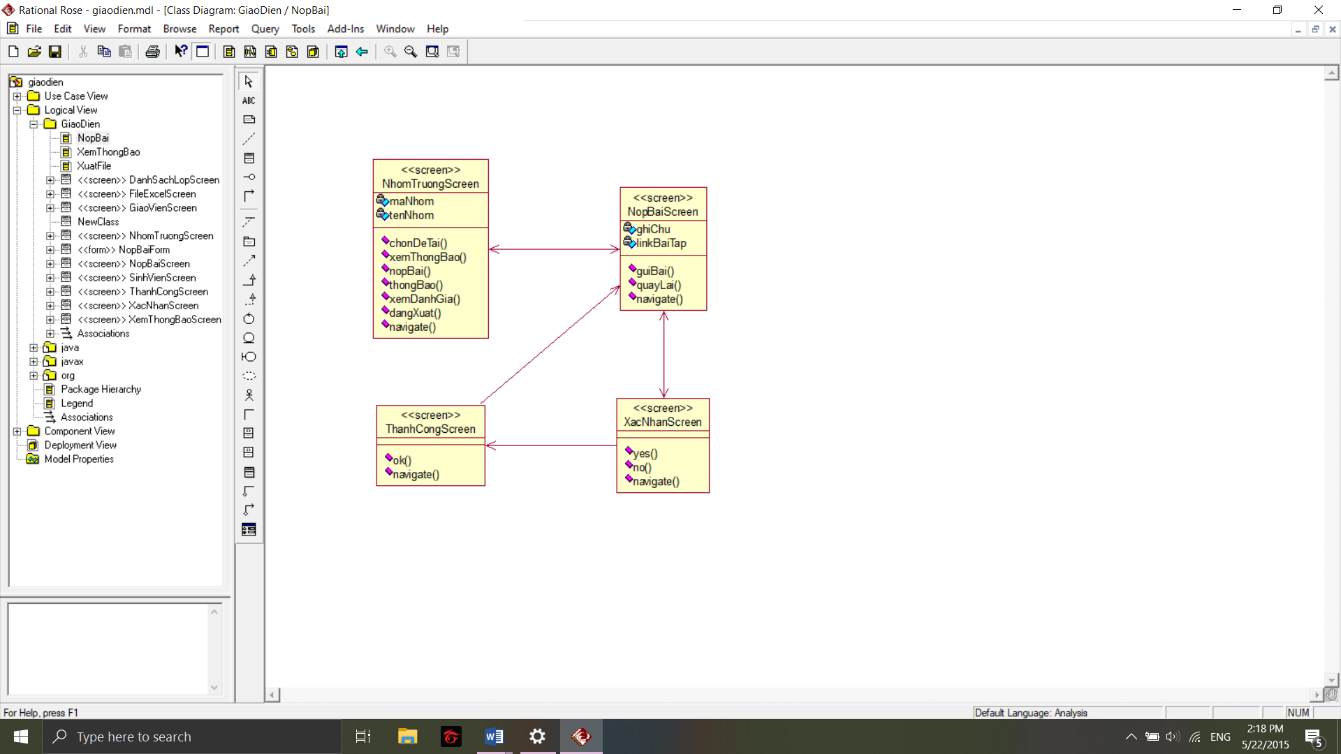




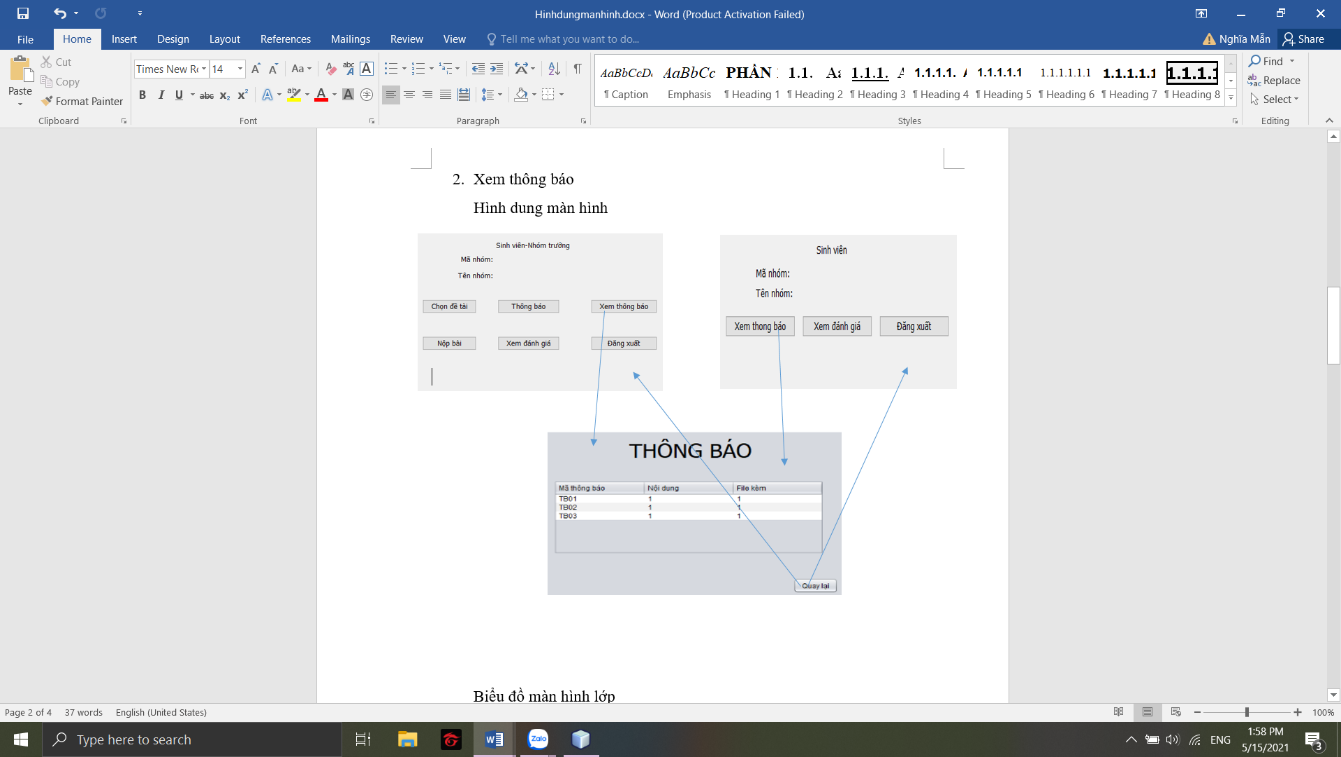
#### Nộp Bài<Mẫn Văn Quân>

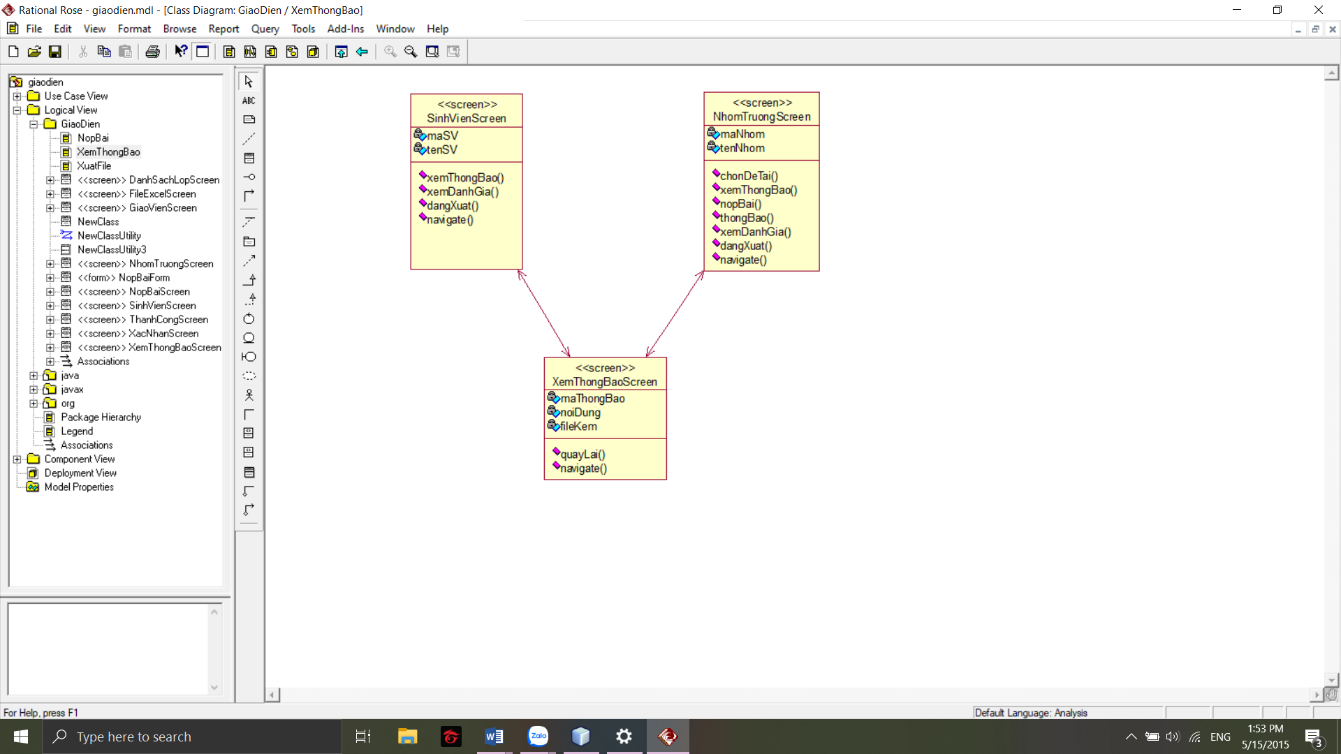
Hình dung màn hình :

Biểu đồ lớp màn hình :



#### Xem Thông báo<Mẫn Văn Quân>



Biểu đồ lớp màn hình :

#### Xuất file<Mẫn Văn Quân

Biểu đồ lớp màn hình :

### Thiết kế dữ liệu- Ánh xạ lớp sang bảng

* Nguyên tắc ánh xạ lớp sang bảng:

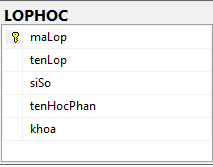
Khi ánh xạ mô hình đối tượng thành bảng, ta có thể bắt đầu với biểu đồ lớp phân tích hoặc biểu đồ lớp thiết kế. Biểu đồ lớp phân tích gần gũi mới mô hình quan hệ hơn vì nó không chỉ ra hướng liên kết. Tuy nhiên, biểu đồ lớp thiết kế có kiểu gán cho các trường nên tiện lợi khi thiết kế bảng.

Vì vậy, chúng ta sẽ sử dụng mô hình phân tích để thiết kế các bảng và lấy kiểu từ mô hình thiết kế.

Mỗi lớp thực thể sẽ tương ứng với một bảng:

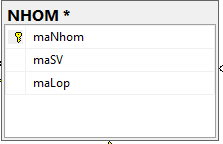
* Liên kết 1-1: đưa khóa chính của một bảng vào bảng kia để nó trở thành khóa ngoài.
* Liên kết 1-n: đưa khóa chính của bảng bên 1 vào bảng bên nhiều để nó trở thành khóa ngoài.
* Liên kết n-n: tạo ra một bảng kết nối, khóa chính của bảng kết nối là khối bội hợp từ 2 khóa chính của 2 bảng.
* Sơ đồ quan hệ các bảng sau khi ánh xạ:
  + Bảng LOPHOC

|  |  |  |  |
| --- | --- | --- | --- |
| Tên thuộc tín | Kiểu dữ liệu | Ràng buộc | Ý nghĩa |
| MaLop | Varchar(30) | Khóa chính | Mã lớp |
| TenLop | Varchar(30) |  | Tên lớp |
| SiSo | Int |  | Sĩ số |
| Khoa | Varchar(10) |  | Khóa |
| TenHP | Varchar(30) |  | Tên học phần |



* + NHOM

|  |  |  |  |
| --- | --- | --- | --- |
| Tên thuộc tín | Kiểu dữ liệu | Ràng buộc | Ý nghĩa |
| MaNhom | Int | Khóa chính | Mã nhóm |
| MaLop | Varchar(30) | Khóa phụ | Mã lớp |
| MaDeTai | Varchar(30) | Khóa phụ | Mã đề tài |
|  |  |  |  |



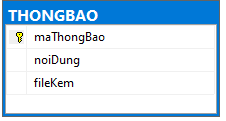
* + SinhVien

|  |  |  |  |
| --- | --- | --- | --- |
| Tên thuộc tín | Kiểu dữ liệu | Ràng buộc | Ý nghĩa |
| MaSV | Varchar(30) | Khóa chính | Mã sinh viên |
| MaNhom | Int | Khóa phụ | Mã nhóm |
| HoSV | Varchar(30) |  | Họ sinh viên |
| TenSV | Varchar(10) |  | Tên sinh viên |
| NhomTruong | Boolean |  | Nhóm trưởng |



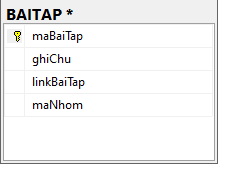
* + THONGBAO

|  |  |  |  |
| --- | --- | --- | --- |
| Tên thuộc tín | Kiểu dữ liệu | Ràng buộc | Ý nghĩa |
| MaThongBao | Varchar(30) | Khóa chính | Mã thông báo |
| NoiDung | Varchar(2000) |  | Nội dung |
| FileKem | Varchar(40) |  | File đính kèm |



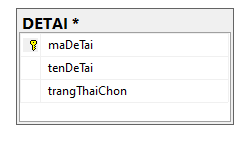
* + BAITAP

|  |  |  |  |
| --- | --- | --- | --- |
| Tên thuộc tín | Kiểu dữ liệu | Ràng buộc | Ý nghĩa |
| MaBaiTap | Varchar(30) | Khóa chính | Mã bài tập |
| GhiChu | Varchar(100) |  | Ghi chú |
| LinkBaiTap | Varchar(50) |  | Link bài tập |
| MaNhom | Varchar(30) | Khóa phụ | Mã nhóm |



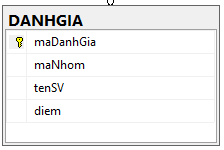
* + DETAI

|  |  |  |  |
| --- | --- | --- | --- |
| Tên thuộc tín | Kiểu dữ liệu | Ràng buộc | Ý nghĩa |
| MaDeTai | Varchar(30) | Khóa chính | Mã đề tài |
| TenDeTai | Varchar(30) |  | Tên đề tài |
| TrangThaiChon | Boolean |  | Trạng thái chọn |



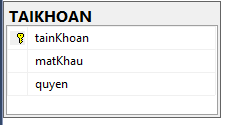
* + DANHGIA

|  |  |  |  |
| --- | --- | --- | --- |
| Tên thuộc tín | Kiểu dữ liệu | Ràng buộc | Ý nghĩa |
| MaDanhGia | Varchar(30) | Khóa chính | Mã đánh giá |
| MaNhom | Varchar(30) | Khóa phụ | Mã nhóm |
| Diem | Double |  | Điểm |



* TAIKHOAN

|  |  |  |  |
| --- | --- | --- | --- |
| Tên thuộc tín | Kiểu dữ liệu | Ràng buộc | Ý nghĩa |
| TaiKhoan | Varchar(30) | Khóa chính | Tài khoản |
| MatKhau | Varchar(30) |  | Mật khẩu |
| Quyen | Varchar(10) |  | Quyền |

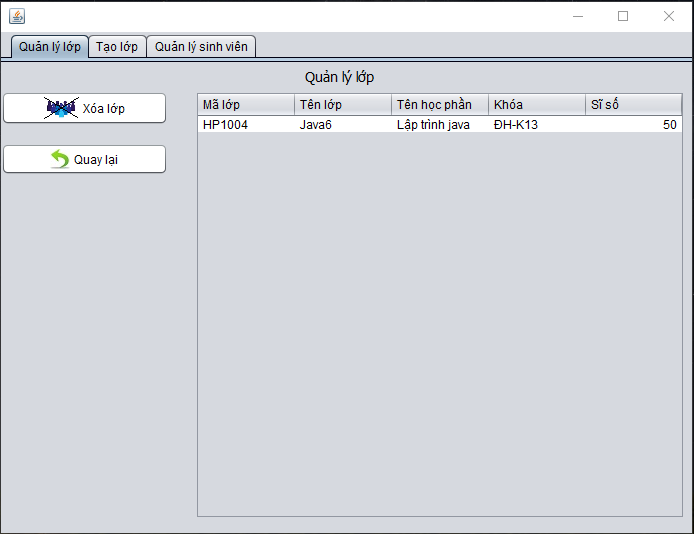


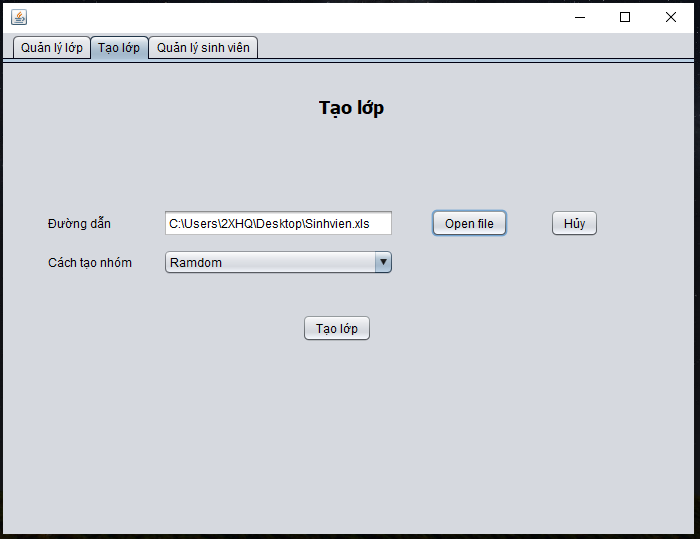
## Thực hiện bài toán

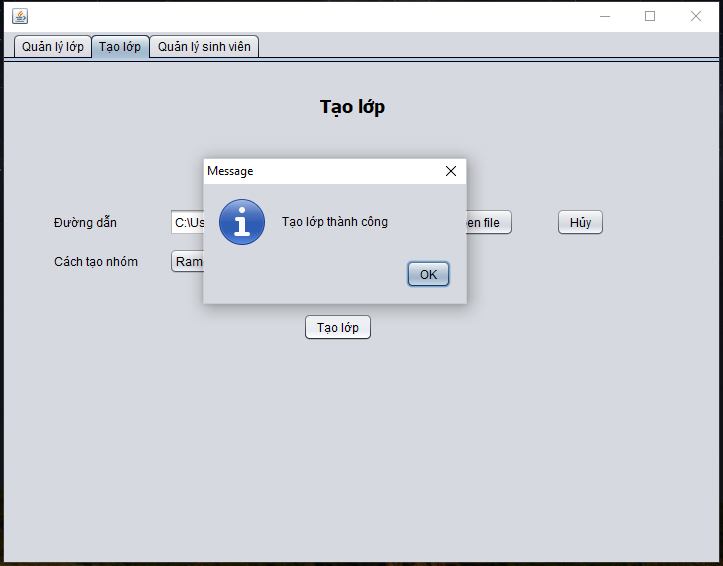
### Hồ Hữu Quang- Quản lý lớp

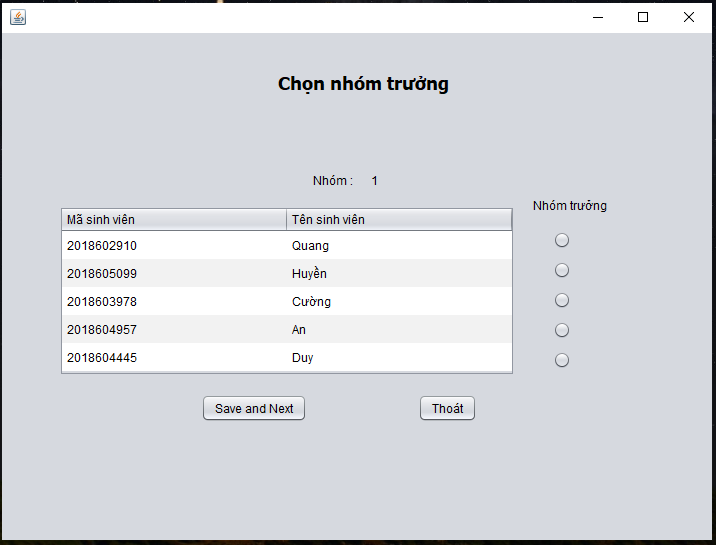
* Thiết kế giao diện vào/ra

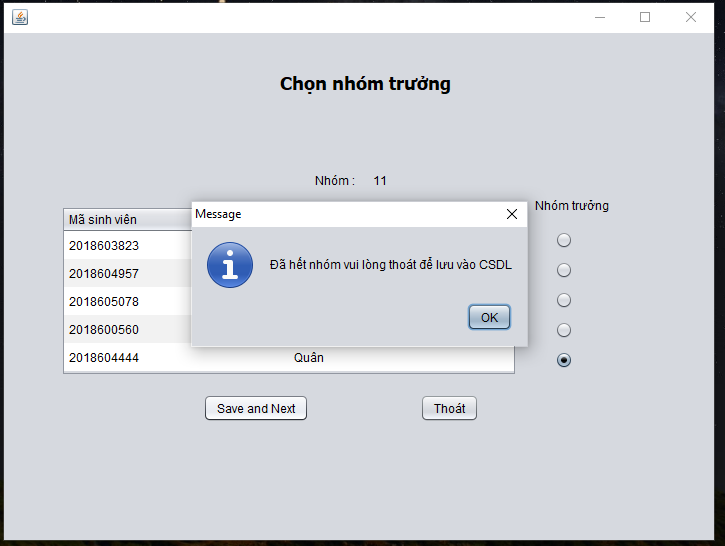
Sau khi từ menu chính chọn chức năng **Quản lý lớp :**

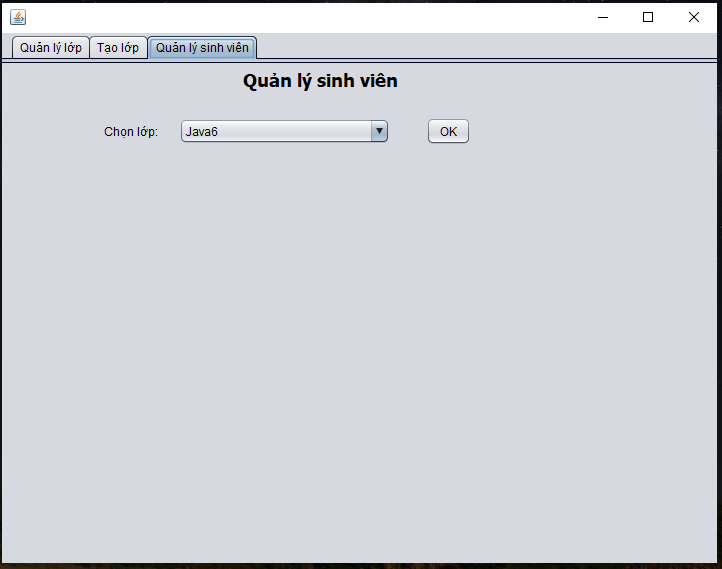


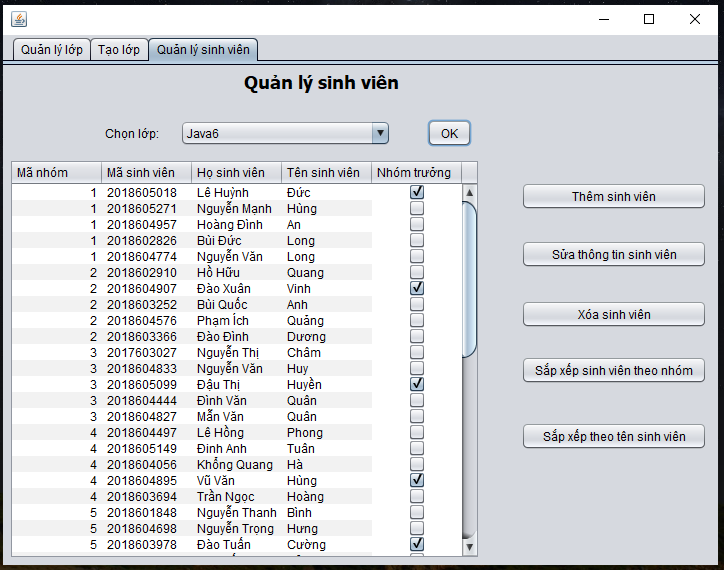


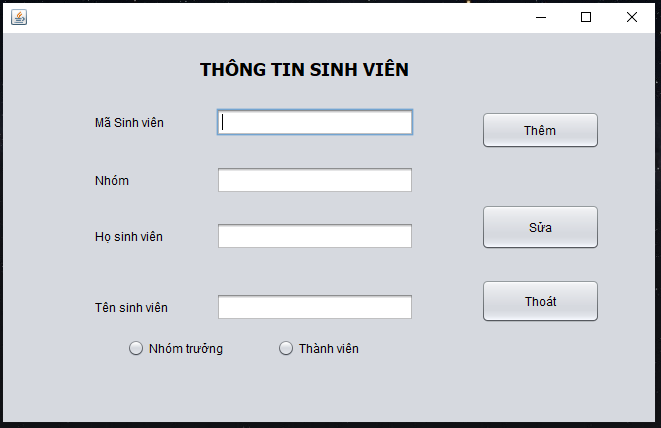












* Mô tả các đối tượng trên màn hình. Mô tả thuật toán kiểm tra tính hợp lệ dữ liệu đầu vào .
* Ở chức năng quản lý lớp gồm có 3 Tab Pane chính :
  + Quản lý lớp :

Gồm có 1 table danh sách lớp và 2 button xóa lớp và quay lại.

* + - Mô tả thuật toán :

+ Khi screen được gọi hệ thống sẽ lấy dữ liệu bảng LOPHOC và truyền vào bảng sinh viên qua class CUSTOMTABLE.

+ Khi thực hiện thao tác xóa lớp, giáo viên cần phải chọn vào lớp muốn xóa trong bảng dữ liệu. Hệ thống thông báo xác nhận xóa lớp sau đó nếu đồng ý sẽ truy vấn bằng câu lệnh SQL để xóa lớp đấy trên cơ sở dữ liệu. Nếu giáo viên không chọn lớp thì hệ thống sẽ báo lỗi yêu cầu giáo viên chọn lớp cần xóa.

+ Khi thực hiện thao tác quay lại, hệ thống sẽ gọi lại screen trang chính của giáo viên.

* + Tạo lớp

Gồm có 1 textbox để hiển thị đường dẫn file, 1 combobox để chọn các tạo nhóm và 3 button thao tác tạo lớp.

* + - Mô tả thuật toán :

+ Khi click vào open file, hệ thống sẽ gọi 1 dialog để chọn file lớp. Sau khi chọn hệ thống sẽ lấy đường dẫn lưu vào text box.

+ Click vào combobox để chọn cách tạo nhóm cho lớp, có 2 lựa chọn là ngẫu nhiên và theo thứ tự. Sau khi chọn cách tạo nhóm và click vào tạo button tạo lớp. hệ thống sẽ thực hiện lấy dữ liệu từ file excel và nạp vào các arrayList, sau đó lưu dữ liệu lớp vào bảng LOPHOC và chuyển màn hình vào form chọn nhóm trưởng.

+ Tại form nhóm trưởng giáo viên click vào thành viên muốn chọn làm nhóm trưởng và click vào save and next. Nếu giáo viên không chọn nhóm trưởng thì hệ thống sẽ báo lỗi yêu cầu chọn phải chọn nhóm trưởng. Sau khi chọn hết danh sách lớp. Hệ thống sẽ yêu cầu thoát và lưu vào cơ sở dữ liệu. Khi click vào thoát hệ thống sẽ thực hiện câu lệnh SQL để lưu vào cơ sở dữ liệu.

* + Quản lý sinh viên

Gồm có 1 textbox, 1 combobx, 1 table và 6 button. Combobox để chọn lớp, table để hiển thị danh sách các sinh viên.

* + - Mô tả thuật toán :

+ Khi screen được gọi hệ thống sẽ hiển thị một combobox để chọn lớp. Sau khi chọn lớp click vào button ok. Hệ thống sẽ hiển thị panel bao gồm table danh sách sinh viên lớp đã chọn và các chức năng thêm, sửa xóa, sắp xếp.

+ Khi click vào button thêm, hệ thống sẽ hiển thị một form mới cho phép thêm sinh viên vào CSDL.

+ Khi sửa thông tin sinh viên, hệ thống yêu cầu chọn sinh viên cần sửa trên table. Sau khi click vào button sửa, nếu giáo viên không chọn sinh viên trong table hệ thống sẽ báo lỗi, nếu đã chọn hệ thống sẽ hiển thị form mới cho phép giáo viên sửa thông tin của sinh viên và lưu CSDL.

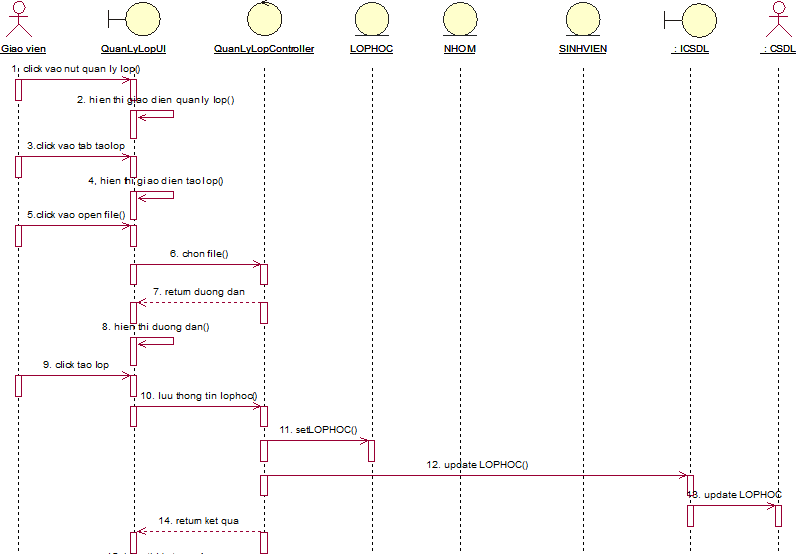
+ Khi xóa thông tin sinh viên, hệ thống yêu cầu chọn sinh viên cần xóa. Sau khi click vào button xóa, nếu giáo viên không chọn sinh viên trong table hệ thống sẽ báo lỗi. Ngược lại, hệ thống sẽ hiển thị xác nhận có chắc chắn xóa sinh viên không. Nếu chọn yes, hệ thống sẽ xóa sinh viên được chọn khỏi CSDL.

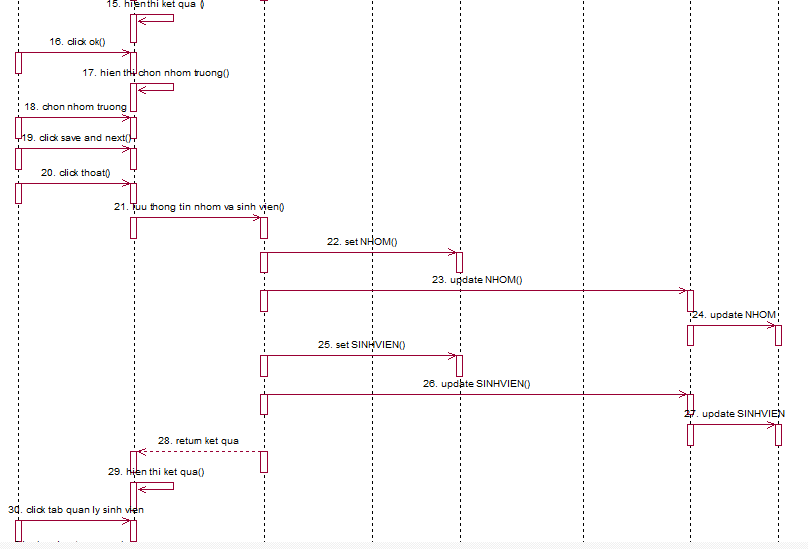
+ Khi giáo viên click sắp xếp theo nhóm, hệ thống sẽ sắp xếp lại danh sách sinh viên và truyền lại vào table.

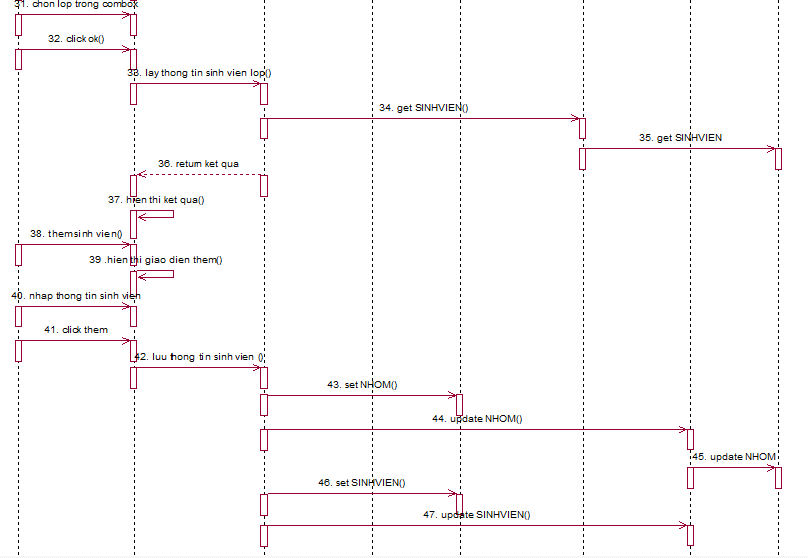
+ Khi giáo viên click sắp xếp theo tên, hệ thống sẽ sắp xếp lại danh sách sinh viên và truyền lại vào table.

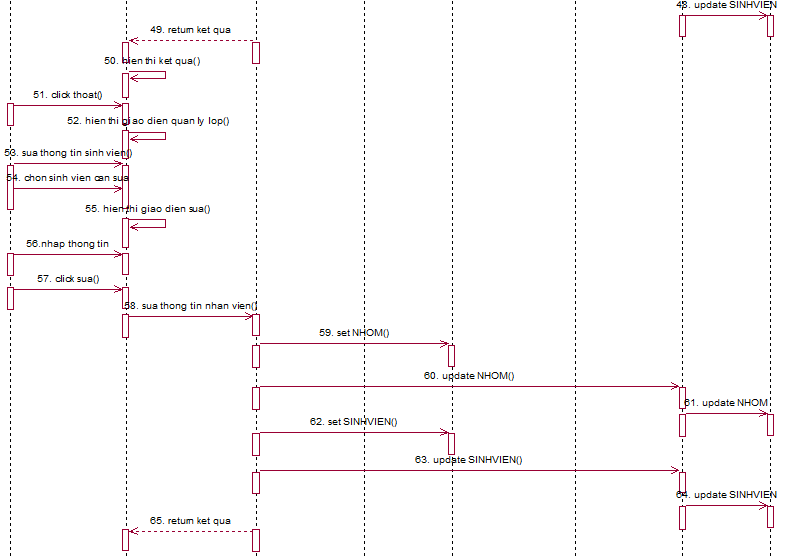
* Mô tả thuật toán xử lý tương ứng với các tác động của người dùng :

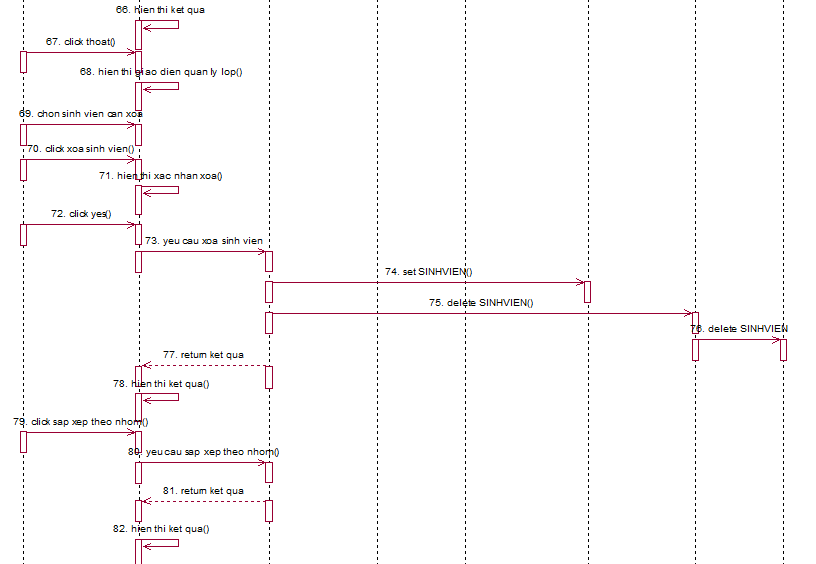
Quản lý lớp học :

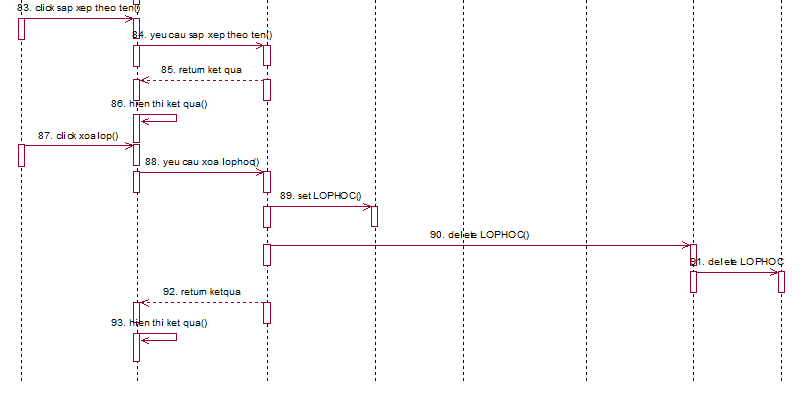












* Các thư viện và đoạn code mẫu
* Để đọc file excel vào phải sử dụng Apche Poi và cần import các thư viện.

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.IOException;

import java.util.ArrayList;

import java.util.Collections;

import java.util.Iterator;

import org.apache.poi.hssf.usermodel.HSSFSheet;

import org.apache.poi.hssf.usermodel.HSSFWorkbook;

import org.apache.poi.ss.usermodel.Row;

import org.apache.poi.ss.usermodel.Sheet;

import org.apache.poi.ss.usermodel.Workbook;

public static ArrayList<SinhVien> list = new ArrayList<>();

static Workbook wb;

static Sheet sheet;

static SinhVien sv;

public static int sonhom;

public static Lop lop = new Lop();

String filename = "C:\\Users\\2XHQ\\Desktop\\Sinhvien.xls";

excuteSQL exsql = new excuteSQL();

public TaoLop(){

}

public void DocDL() throws FileNotFoundException, IOException, Exception {

//doc file

FileInputStream file = new FileInputStream(filename);

HSSFWorkbook workbook = new HSSFWorkbook(file);

HSSFSheet sheet = workbook.getSheetAt(0);

Iterator<Row> rowIterator = sheet.iterator();

//lay du lieu lop

Row r = rowIterator.next();

lop.setMaLop(r.getCell(1).getStringCellValue());

lop.setTenLop(r.getCell(3).getStringCellValue());

r = rowIterator.next();

lop.setKhoa(r.getCell(1).getStringCellValue());

r = rowIterator.next();

lop.setTenHP(r.getCell(1).getStringCellValue());

double a = r.getCell(3).getNumericCellValue();

lop.setSiso((int)a);

//tim so nhom

if(lop.getSiso() % 5 == 0) sonhom = lop.getSiso() /5;

else sonhom = lop.getSiso() /5 + 1;

//lay du leu sinh vien

rowIterator.next();

while (rowIterator.hasNext()) {

Row row = rowIterator.next();

sv = new SinhVien();

double ma = row.getCell(1).getNumericCellValue();

Integer k = (int) ma;

sv.setMaSV(k.toString());

sv.setHoSV(row.getCell(2).getStringCellValue());

sv.setTenSV(row.getCell(3).getStringCellValue());

list.add(sv);

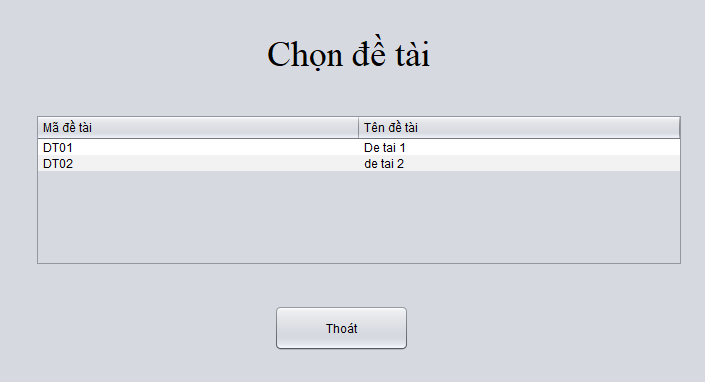
}

file.close();

}

### Lê Đăng Quang- Chọn đề tài

* Thiế kế giao diện vào/ra



* Mô tả chi tiết

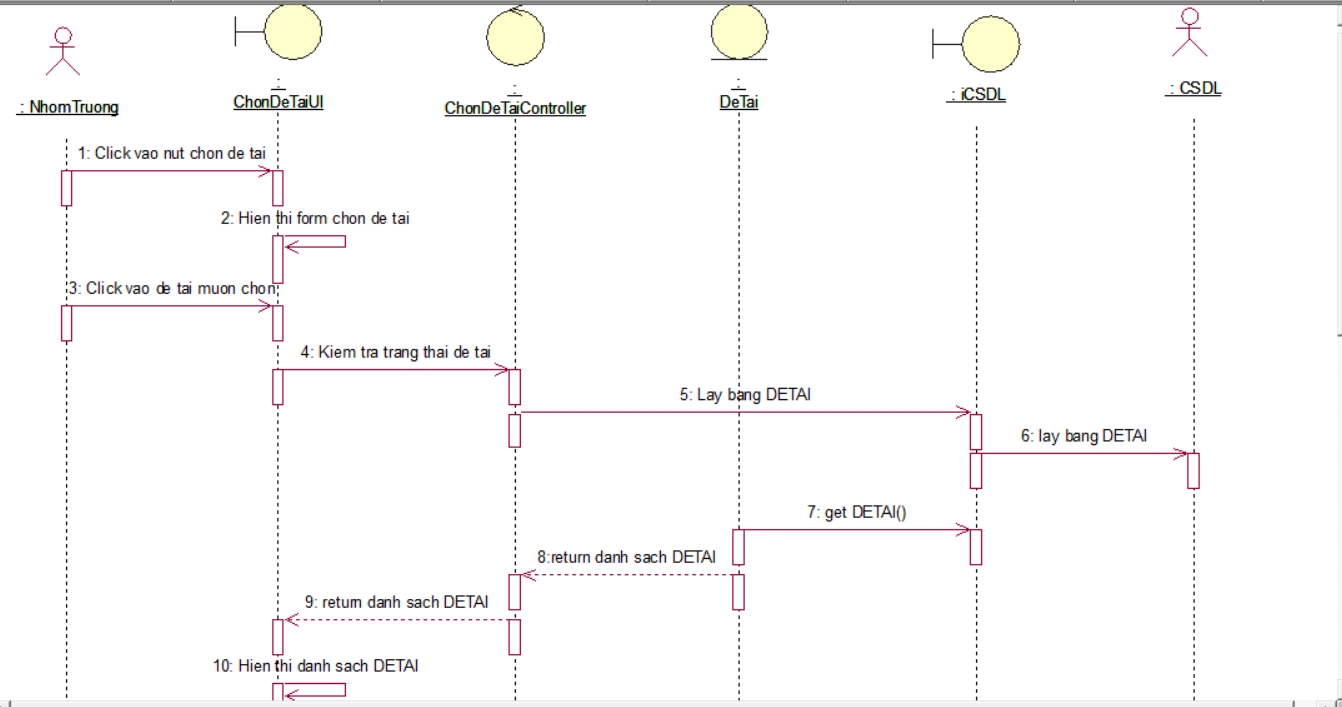
- Mô tả chi tiết đối tượng trên màn hình:

Form “Chọn đề tài” gồm một bảng chứa dữ liệu và một button để bắt sự kiện.

- Mô tả thuật toán kiểm tra tính hợp lệ dữ liệu đầu vào”

Người dùng có thể click vào mộ dòng trong bảng dữ liệu. Một thông báo xác nhận người dùng mốn chọn đề tài này sẽ hiện ra. Khi người dùng nhấn “Yes” sẽ hiện thông báo “Chọn đề tài thành công” và ẩn để tài khỏi bảng.

* Mô tả thuật toán xử lý tương ứng với các tác động của người dùng:





-Hướng dẫn sử dụng các thư viện mẫu

import java.sql.Connection: Dùng để liên kết với cơ sở dữ liệu

import java.sql.DriverManager: Dùng để quản lý các Driver

import java.sql.PreparedStatement: Dùng để thực hiện truy vấn tham số

import java.sql.ResultSet: Biểu diễn một tập kết quả trong cơ sở dữ liệu tạo ra bởi việc sử dụng một câu lệnh SQL là select

import java.sql.SQLException: Dùng để bắt các lỗi liên quan đến thực thi câu lệnh SQL

import java.sql.Statement: Cung cấp các phương thức để thực thi các câu lện truy vấn cơ sở dữ liệu. Nó cung cấp phương thức để tạo ra đối tượng ResultSet

import java.util.ArrayList: Dùng để khai một mảng động được đồng bộ

import javax.swing.JoptionPane: Dùng để hiển thị các thông báo lên màn hình khi cần thiết

-Code minh họa

public void getDetai() {

db.getConnect();

String sql = "select \* from DETAI where TRANGTHAICHON = " + false + "";

dsdt = db.getdsDeTaiQ2(sql);

}

private void jTableDetaiMouseClicked(java.awt.event.MouseEvent evt) {

// TODO add your handling code here:

selectedRow = jTableDetai.getSelectedRow();

int ret = JOptionPane.showConfirmDialog(null, "Bạn chọn đề tài này ?", "Thông báo", JOptionPane.YES\_NO\_OPTION);

if (ret == JOptionPane.YES\_OPTION) {

JOptionPane.showConfirmDialog(null, "Chọn thành công", "Thông báo", JOptionPane.OK\_OPTION, JOptionPane.INFORMATION\_MESSAGE);

try {

ex.setTrangthaidt(listdt.get(selectedRow).getMaDeTai());

} catch (Exception ex) {

Logger.getLogger(ChondetaiUI.class.getName()).log(Level.SEVERE, null, ex);

}

try {

ex.setmaDeTai(listdt.get(selectedRow).getMaDeTai());

} catch (Exception ex) {

Logger.getLogger(ChondetaiUI.class.getName()).log(Level.SEVERE, null, ex);

}

listdt.remove(selectedRow);

jTableDetai.setModel(new CustomTableDeTai(listdt));

}

}

### Lê Đăng Quang – Thông báo leader

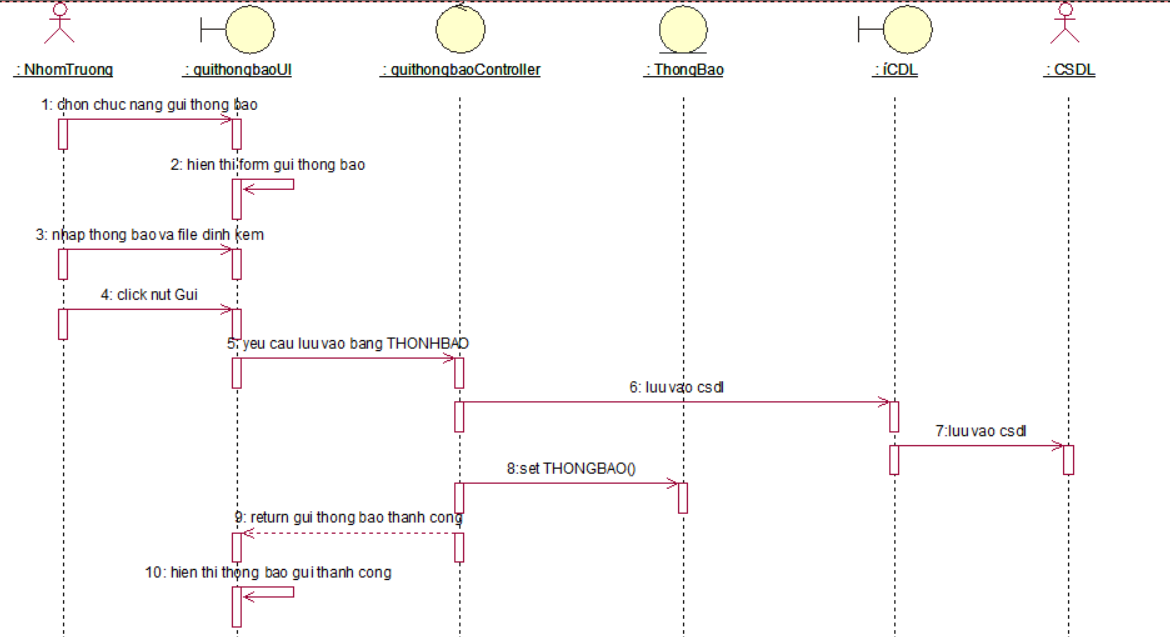
* Thiết kế giao diện vào ra

****

* Mô tả chi tiết
* Mô tả các đối tượng trên màn hình:

Form Gửi thông báo gồm 2 ô text box để nhập dữ liệu và 4 button dùng để thực hiện các chức năng cùng tên

* Mô tả thuật toán xử lý tương ứng với tác động của người dùng:



* Hướng dẫn sử dụng các thư viện mẫu

import java.sql.Connection: Dùng để liên kết với cơ sở dữ liệu

import java.sql.DriverManager: Dùng để quản lý các Driver

import java.sql.PreparedStatement: Dùng để thực hiện truy vấn tham số

import java.sql.ResultSet: Biểu diễn một tập kết quả trong cơ sở dữ liệu tạo ra bởi việc sử dụng một câu lệnh SQL là select.

import java.sql.SQLException: Dùng để bắt các lỗi liên quan đến thực thi câu lệnh SQL.

import java.sql.Statement: Cung cấp các phương thức để thực thi các câu lện truy vấn cơ sở dữ liệu. Nó cung cấp phương thức để tạo ra đối tượng ResultSet.

import java.util.ArrayList: Dùng để khai một mảng động được đồng bộ

import javax.swing.JoptionPane: Dùng để hiển thị các thông báo lên màn hình khi cần thiết.

* Code minh họa

public void guiThongBao(String tb, String filedk) throws Exception {

maTB++;

String matt = STT + maTB;

db.getConnect();

String sql = "Insert into THONGBAO values('" + matt + "','" + tb + "','" + filedk + "')";

db.doSQL(sql);

}

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

try {

ex.guiThongBao(jtxtthongbao.getText(),jtxtfiledinhkem.getText());

} catch (Exception ex) {

Logger.getLogger(ThongBaoLeaderUI.class.getName()).log(Level.SEVERE, null, ex);

}

JOptionPane.showConfirmDialog(null, "Gửi thông báo thành công", "Thông báo", JOptionPane.OK\_OPTION);

}

### Lê Đăng Quang – Xem đánh giá

* Mô hình giao diện vào/ra

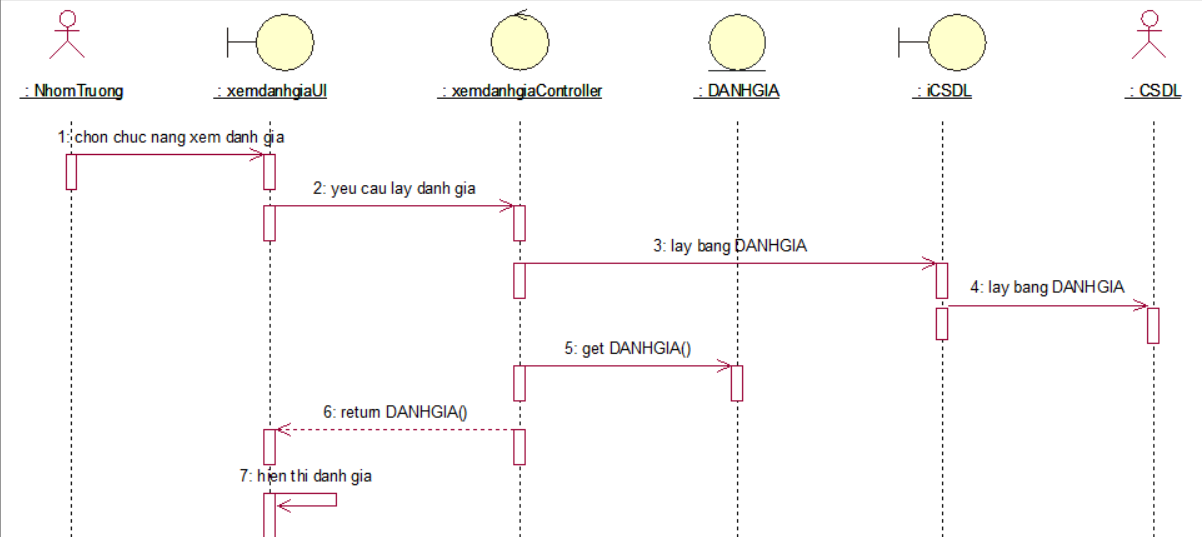
****

* Mô tả chi tiết

- Mô tả các đối tượng trên màn hình:

Form xem đánh giá gồm 1 table lưu các đánh giá, một button để thoát khỏi form đánh giá.

* Mô tả thuật toán xử lý tương ứng với các tác động của người dùng



* Hướng dẫn sử dụng các thư viện mẫu

import java.sql.Connection: Dùng để liên kết với cơ sở dữ liệu

import java.sql.DriverManager: Dùng để quản lý các Driver

import java.sql.PreparedStatement: Dùng để thực hiện truy vấn t ham số

import java.sql.ResultSet: Biểu diễn một tập kết quả trong cơ sở dữ liệu tạo ra bởi việc sử dụng một câu lệnh SQL là select

import java.sql.SQLException: Dùng để bắt các lỗi liên quan đến thực thi câu lệnh SQL

import java.sql.Statement: Cung cấp các phương thức để thực thi các câu lện truy vấn cơ sở dữ liệu. Nó cung cấp phương thức để tạo ra đối tượng ResultSet

import java.util.ArrayList: Dùng để khai một mảng động được đồng bộ

import javax.swing.JoptionPane: Dùng để hiển thị các thông báo lên màn hình khi cần thiết

* Code Minh họa

public int getmanhomt() {

db.getConnect();

msv = ManHinhDangNhap.taikhoan.getUsername();

String sql = "select MANHOM from SINHVIEN where MASV ='" + msv + "'";

return db.getmanhom(sql);

}

public void getDanhGia() {

db.getConnect();

String sql = "select \* from DANHGIA where MANHOM = " + getmanhomt() + "";

dsdg = db.getDataDG(sql);

}

ArrayList<DanhGia> listdg = new ArrayList<DanhGia>();

executeSQL ex = new executeSQL();

public XemDanhGiaUI() {

initComponents();

setLocationRelativeTo(null);

ex.getDanhGia();

listdg = ex.getDsdg();

hienthibang();

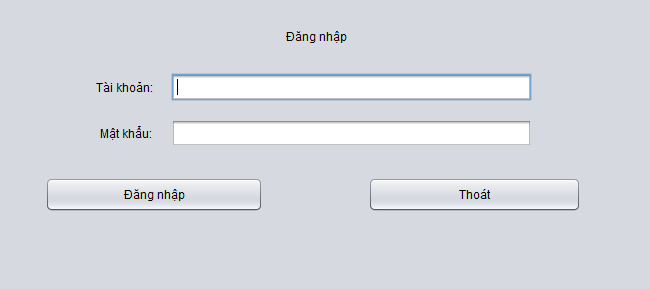
}

public void hienthibang() {

jTableDanhgia.setModel(new CustomTableDG(listdg));

}

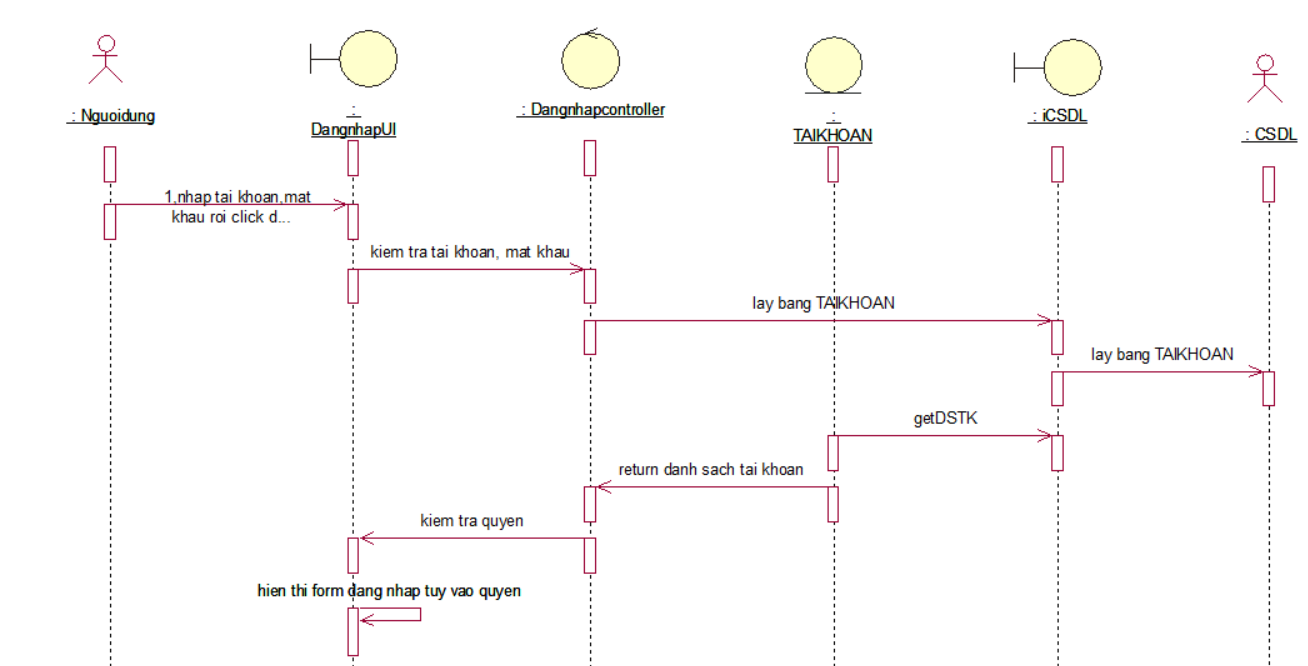
### Phạm Ích Quảng – Đăng nhập

* Thiết kế giao diện vào ra
* Mô tả chi tiết
* Đối tượng trên màn hình

Form đăng nhập gồm 2 ô textbox để nhập tài khoản, mật khẩu và 2 button để đăng nhập hoặc thoát khoi chương trình.

* Mô tả chi tiết thuật toán

Sau khi người dùng nhập tài khoản, mật khẩu thì hệ thống sẽ kiểm tra quyền của người dùng sau đó hiện ra form phù hợp

* ­Mô tả thuật toán xử lý tương ứng với các tác động của người dùng:
* Hướng dẫn sử dụng các thư viện mẫu

import java.sql.Connection: Dùng để liên kết với cơ sở dữ liệu

import java.sql.DriverManager: Dùng để quản lý các Driver

import java.sql.PreparedStatement: Dùng để thực hiện truy vấn tham số

import java.sql.ResultSet: Biểu diễn một tập kết quả trong cơ sở dữ liệu tạo ra bởi việc sử dụng một câu lệnh SQL là select

import java.sql.SQLException: Dùng để bắt các lỗi liên quan đến thực thi câu lệnh SQL

import java.sql.Statement: Cung cấp các phương thức để thực thi các câu lện truy vấn cơ sở dữ liệu. Nó cung cấp phương thức để tạo ra đối tượng ResultSet

import java.util.ArrayList: Dùng để khai một mảng động được đồng bộ

import javax.swing.JoptionPane: Dùng để hiển thị các thông báo lên màn hình khi cần thiết

* Code minh họa

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

String tk = txtTaikhoan.getText();

if (tk.equals("")) {

JOptionPane.showMessageDialog(this, "Vui lòng điền tài khoản");

} else if (mk.equals("")) {

JOptionPane.showMessageDialog(this, "Vui lòng điền mật khẩu");

} else {

try {

ex.selectDSTK();

for (DangNhap dn : ex.getDstk()) {

if (tk.equals(dn.getUsername()) && mk.equals(dn.getPassword())) {

if (dn.getQuyen().equals("GV")) {

this.setVisible(false);

new DangNhapGiaoVienForm().setVisible(true);

return;

} else if (dn.getQuyen().equals("NT")) {

taikhoan = dn;

this.setVisible(false);

new DangNhapNhomTruongForm().setVisible(true);

return;

} else if (dn.getQuyen().equals("TV")) {

taikhoan = dn;

this.setVisible(false);

new DangNhapSinhVienForm().setVisible(true);

return;

}

}

}

JOptionPane.showMessageDialog(null, "Sai tài khoản hoặc mật khẩu!!!", "Eror", EXIT\_ON\_CLOSE);

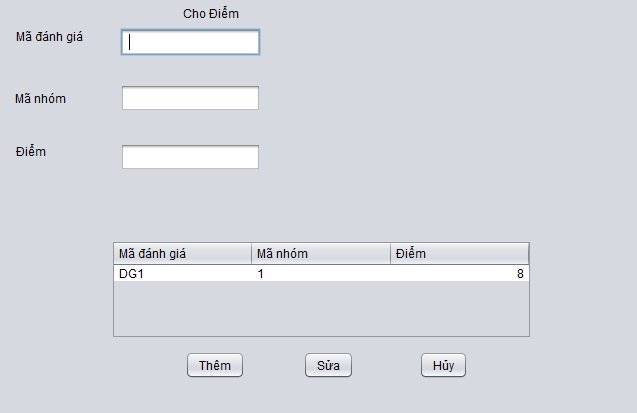
} catch (Exception ex) {

JOptionPane.showMessageDialog(null, ex);

}

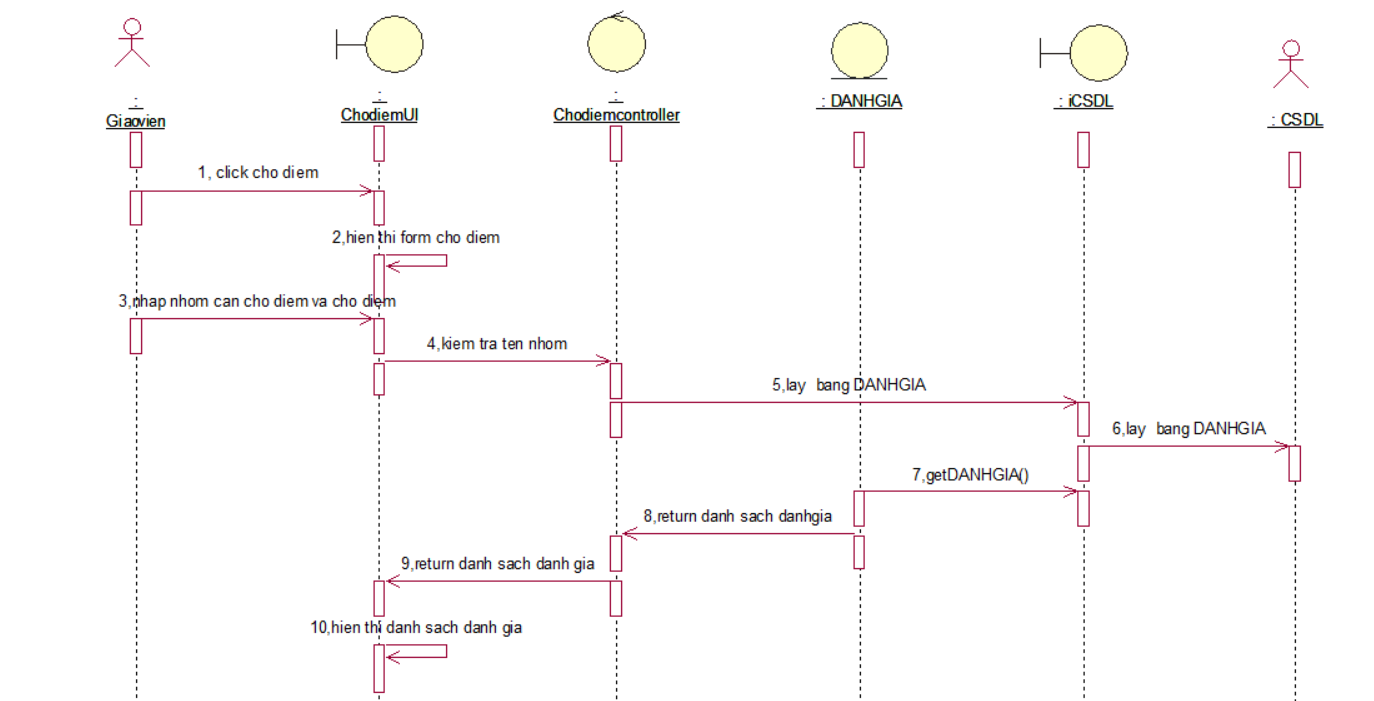
}

### Phạm Ích Quảng – Đánh giá

* Thiết kế giao diện
* Mô tả chi tiết:
* Các đối tượng trên màn hình

Form cho điểm gồm 3 textbox để nhập dữ liệu và 1 table lưu các đánh giá, 3 button để thêm, sửa và thoát khỏi form cho điểm

* Mô tả thuật toán xử lý tương ứng với các tác động của người dùng



* Hướng dẫn sử dụng các thư viện mẫu

import java.sql.Connection: Dùng để liên kết với cơ sở dữ liệu

import java.sql.DriverManager: Dùng để quản lý các Driver

import java.sql.PreparedStatement: Dùng để thực hiện truy vấn tham số

import java.sql.ResultSet: Biểu diễn một tập kết quả trong cơ sở dữ liệu tạo ra bởi việc sử dụng một câu lệnh SQL là select

import java.sql.SQLException: Dùng để bắt các lỗi liên quan đến thực thi câu lệnh SQL

import java.sql.Statement: Cung cấp các phương thức để thực thi các câu lện truy vấn cơ sở dữ liệu. Nó cung cấp phương thức để tạo ra đối tượng ResultSet

import java.util.ArrayList: Dùng để khai một mảng động được đồng bộ

import javax.swing.JoptionPane: Dùng để hiển thị các thông báo lên màn hình khi cần thiết

* Code minh họa

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

try {

ex.insertDSDanhgia(Integer.valueOf(txtManhom.getText()), Double.valueOf(txtDiem.getText()));

} catch (Exception ex) {

Logger.getLogger(ChodiemForm.class.getName()).log(Level.SEVERE, null, ex);

}

}

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

try {

ex.suaDSDanhgia(txtMadanhgia.getText(),Double.valueOf(txtDiem.getText()));

} catch (Exception ex) {

Logger.getLogger(ChodiemForm.class.getName()).log(Level.SEVERE, null, ex);

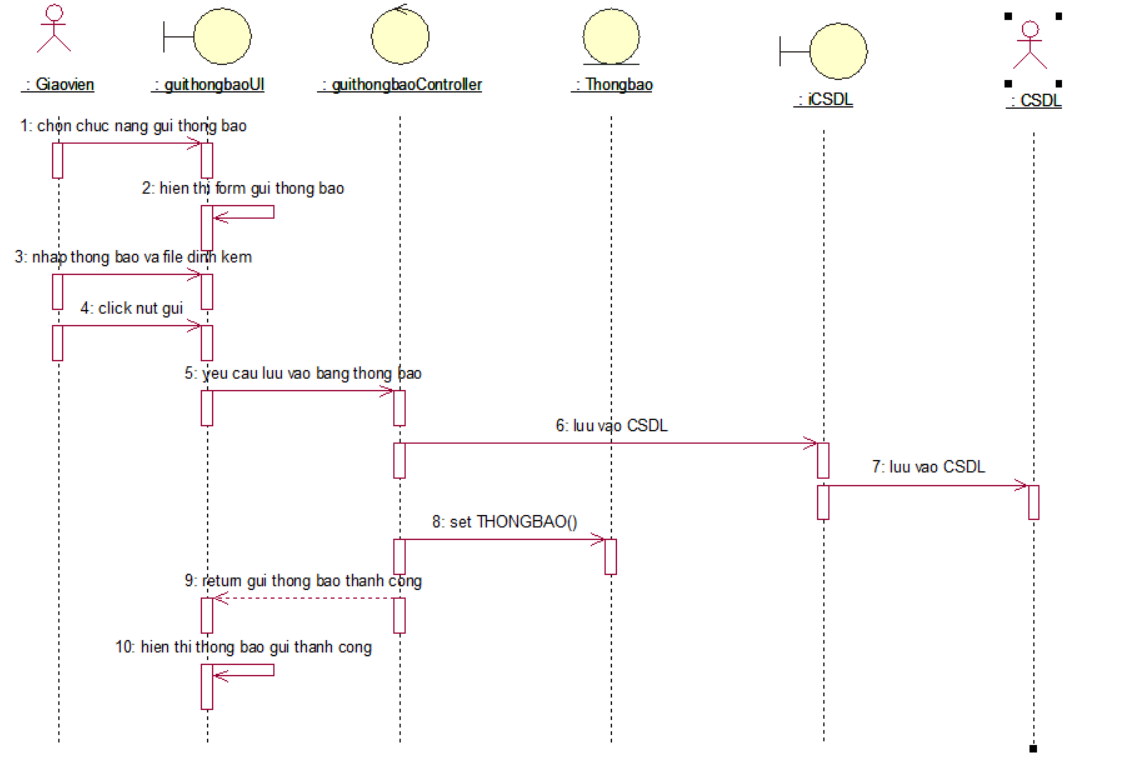
}

### Phạm Thanh Quý – Công việc

* + - 1. **Gửi thông báo**
* Thiết kế giao diện vào/ra



* Mô tả các đối tượng trên màn hình
* 2 textfield: nhập thông báo và nhập file đính kèm
* 1 button gửi: cho phép gửi thông báo
* 1 button thoát: cho phép quay lại màn hình giáo viên
* Mô tả thuật toán kiểm tra tính hợp lệ dữ liệu đầu vào:
* Nếu không nhập thông báo sẽ báo lỗi
* Mô tả thuật toán xử lý tương ứng với các tác động của người dùng



* Code mẫu:

import java.sql.SQLException;

static int maGV = 0;

final String stt = "GVTB";

public void guiThongBaoGV(String nd, String filedk) throws Exception {

maGV++;

String matt = stt + maGV;

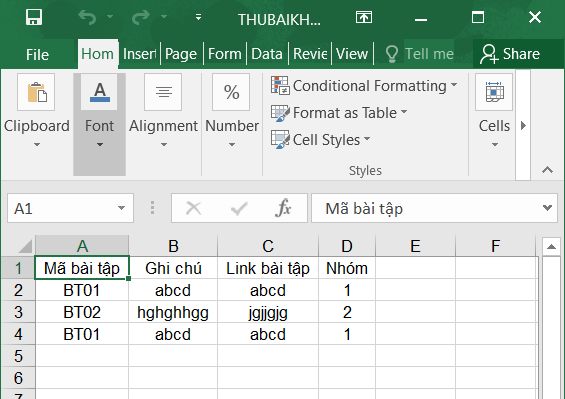
db.getConnect();

String sql = "Insert into THONGBAO values('" + matt + "','" + nd + "','" + filedk + "')";

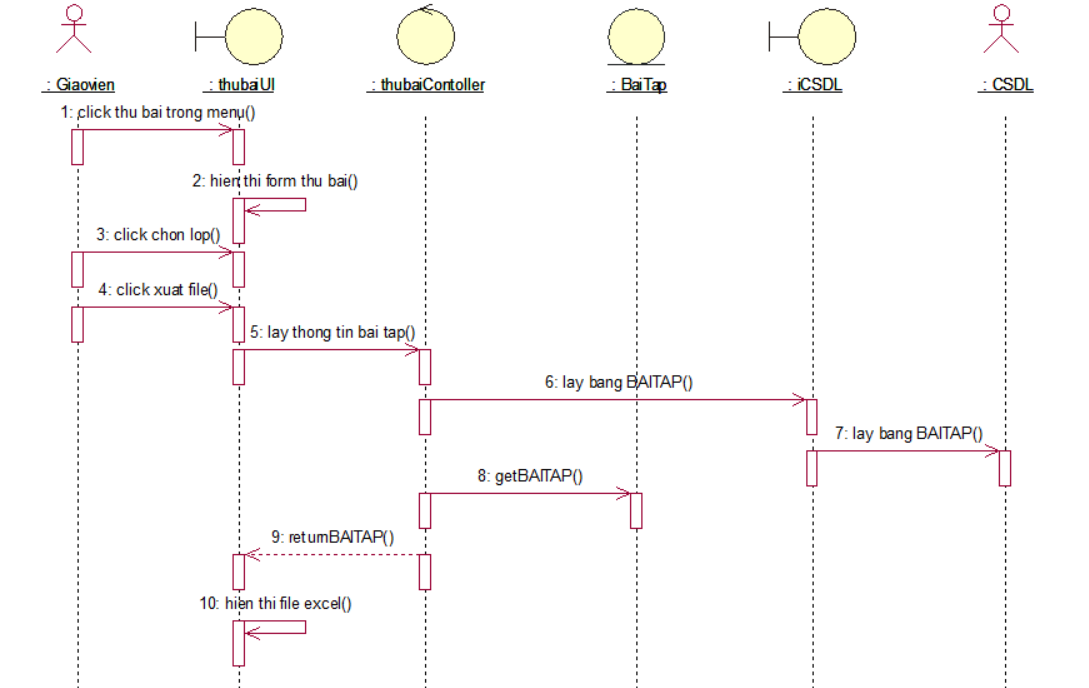
db.doSQL(sql);

}

* + - 1. **Thu bài:**
* Thiết kế giao diện vào/ra



* Mô tả các đối tượng trên màn hình:
* Combobox: hiển thị danh sách các lớp học
* Button Xuất file: Xuất file Excel
* Table: hiển thị danh sách BAITAP
* File Excel bảng tính: hiển thị danh sách bài tập của các nhóm
* Button quay lại: quay lại công việc
* Mô tả thuật toán kiểm tra tính hợp lệ dữ liệu đầu vào
* Mô tả thuật toán xử lý tương ứng với các tác động của người



* Code mẫu:

import Model.BaiTap;

import Model.DanhGia;

import Model.Lop;

import View.BaitapUI;

import java.awt.Desktop;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.IOException;

import java.util.ArrayList;

import java.util.Iterator;

import javax.swing.DefaultComboBoxModel;

import javax.swing.JComboBox;

import static javax.swing.JFrame.EXIT\_ON\_CLOSE;

import javax.swing.JOptionPane;

import org.apache.poi.hssf.usermodel.HSSFRow;

import org.apache.poi.hssf.usermodel.HSSFSheet;

import org.apache.poi.hssf.usermodel.HSSFWorkbook;

import org.apache.poi.ss.usermodel.Cell;

import org.apache.poi.ss.usermodel.CellStyle;

import org.apache.poi.ss.usermodel.HorizontalAlignment;

import org.apache.poi.ss.usermodel.VerticalAlignment;

public class XuatfileThuBai {

private final ConnectDB db = new ConnectDB();

public void selectDSLop(ArrayList<Lop> dsLop)

{

db.getDataLopQuan(dsLop);

}

public void selectDanhGia(ArrayList<BaiTap> dsbt)

{

db.getDataBaiTap(dsbt);

}

public void xuatFileAction(HSSFWorkbook workbook, ArrayList<BaiTap> dsbt, String tenLop) {

db.selectThuBaiByTenLop(dsbt, tenLop);

initExcelFile(workbook, dsbt, tenLop);

}

public void init(BaitapUI jFrameXuatFile, JComboBox jComboBoxchonlop, ArrayList<Lop> dsLop) {

jFrameXuatFile.setTitle("Xuất file Excel");

jFrameXuatFile.setLocationRelativeTo(null);

jFrameXuatFile.setDefaultCloseOperation(EXIT\_ON\_CLOSE); initCbo(jComboBoxchonlop, dsLop);

}

public void initCbo(JComboBox jComboBoxchonlop, ArrayList<Lop> dsLop) {

jComboBoxchonlop.setModel(new DefaultComboBoxModel());

db.getDataLopQuan(dsLop);

Iterator<Lop> itLopHoc = dsLop.iterator();

while (itLopHoc.hasNext()) {

jComboBoxchonlop.addItem(itLopHoc.next().getTenLop());

}

jComboBoxchonlop.setSelectedIndex(0);

}

public void initExcelFile(HSSFWorkbook workbook, ArrayList<BaiTap> dsbt, String tenLop) {

//Create a cellstyle

CellStyle cellStyle = workbook.createCellStyle();

cellStyle.setAlignment(HorizontalAlignment.CENTER);

cellStyle.setVerticalAlignment(VerticalAlignment.CENTER);

//Create a spreadsheet

HSSFSheet sheet = workbook.createSheet("Thu Bai " + tenLop);

//Create a row object

HSSFRow row;

//Create cell & setvalue

row = sheet.createRow(0);

Cell cell0 = row.createCell(0);

Cell cell1 = row.createCell(1);

Cell cell2 = row.createCell(2);

Cell cell3 = row.createCell(3);

cell0.setCellStyle(cellStyle);

cell1.setCellStyle(cellStyle);

cell2.setCellStyle(cellStyle);

cell3.setCellStyle(cellStyle);

cell0.setCellValue("Mã bài tập");

cell1.setCellValue("Ghi chú");

cell2.setCellValue("Link bài tập");

cell3.setCellValue("Nhóm");

//Create cells & rows for the data

for (int i = 0; i < dsbt.size(); i++) {

row = sheet.createRow(i + 1);

for (int j = 0; j < 4; j++) {

Cell cell = row.createCell(j);

cell.setCellStyle(cellStyle);

if (cell.getColumnIndex() == 0) {

cell.setCellValue(dsbt.get(i).getMaBaiTap());

}

if (cell.getColumnIndex() == 1) {

cell.setCellValue(dsbt.get(i).getGhiChu());

}

if (cell.getColumnIndex() == 2) {

cell.setCellValue(dsbt.get(i).getLinkBaiTap());

}

if (cell.getColumnIndex() == 3) {

cell.setCellValue(dsbt.get(i).getMaNhom());

}

}

}

//Auto resize column

for (int i = 0; i < 4; i++) {

sheet.autoSizeColumn(i);

}

//Writing excel file

writingExcelFile(workbook, tenLop);

}

public void writingExcelFile(HSSFWorkbook workbook, String tenLop) {

FileNameExtensionFilter("xlsx (\*.xlsx)", "xlsx");

JFileChooser.APPROVE\_OPTION) {

// File output = saveFile.getSelectedFile();

File a = new File("THUBAI" + tenLop + ".xls");

try (FileOutputStream out = new FileOutputStream(a)) {

workbook.write(out);

out.close();

File f = a.getAbsoluteFile();

if (!f.exists()) {

JOptionPane.showMessageDialog(null, "File không tồn tại.", "Message", JOptionPane.OK\_OPTION);

} else if (!Desktop.isDesktopSupported()) {

JOptionPane.showMessageDialog(null, "Desktop không hỗ trợ file này.", "Warning", JOptionPane.OK\_OPTION);

} else {

Desktop.getDesktop().open(f);

}

} catch (FileNotFoundException ex) {

JOptionPane.showMessageDialog(null, ex, "Message", JOptionPane.CLOSED\_OPTION);

} catch (IOException ex) {

JOptionPane.showMessageDialog(null, ex, "Message", JOptionPane.CLOSED\_OPTION);

}

// }

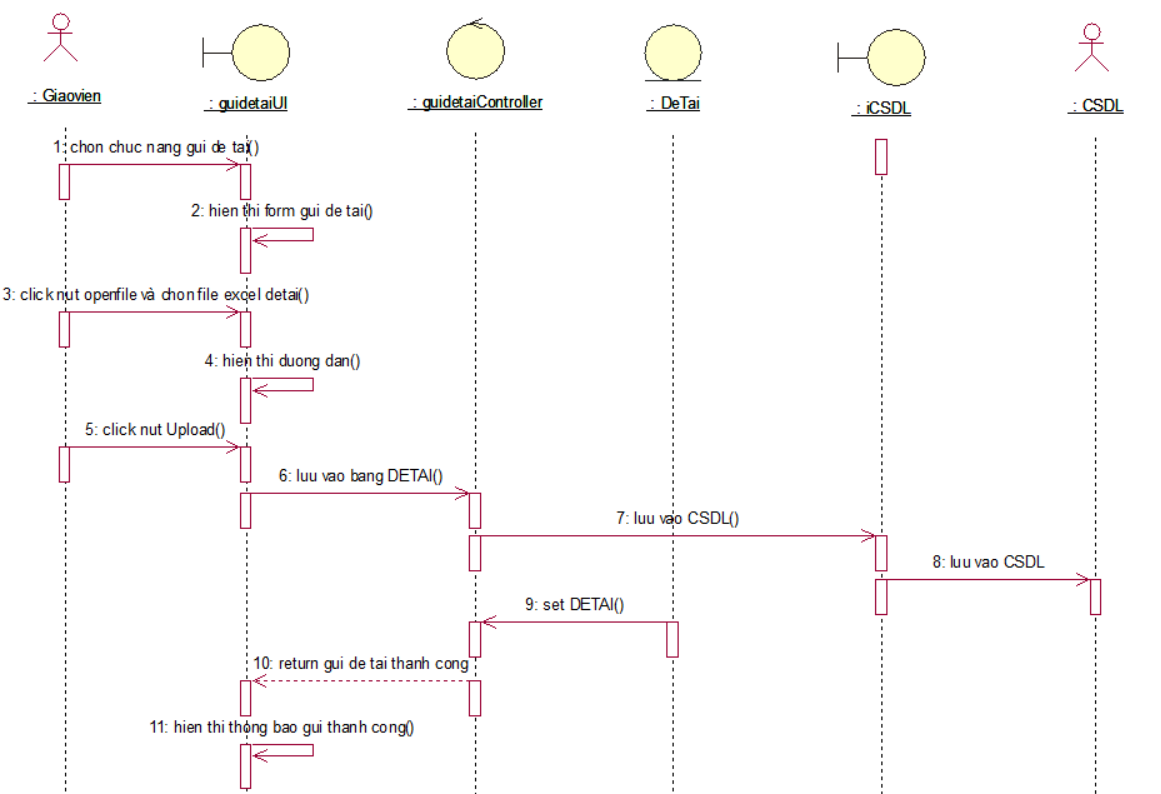
}

}

* + - 1. **Gửi đề tài:**



* Mô tả các đối tượng trên màn hình:
* TextField: hiển thị đường dẫn
* Button openfile: chọn file excel đề tài
* Button Upload: nạp đề tài vào CSDL
* Button Hủy: quay về công việc
* Mô tả thuật toán kiểm tra tính hợp lệ của dữ liệu đầu vào
* Nếu không chọn file hệ thống báo lỗi
* Mô tả thuật toán xử lý tương ứng với các tác động của người



* Code mẫu:

import Model.BaiTap;

import Model.DanhGia;

import Model.Lop;

import View.BaitapUI;

import java.awt.Desktop;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.IOException;

import java.util.ArrayList;

import java.util.Iterator;

import javax.swing.DefaultComboBoxModel;

import javax.swing.JComboBox;

import static javax.swing.JFrame.EXIT\_ON\_CLOSE;

import javax.swing.JOptionPane;

import org.apache.poi.hssf.usermodel.HSSFRow;

import org.apache.poi.hssf.usermodel.HSSFSheet;

import org.apache.poi.hssf.usermodel.HSSFWorkbook;

import org.apache.poi.ss.usermodel.Cell;

import org.apache.poi.ss.usermodel.CellStyle;

import org.apache.poi.ss.usermodel.HorizontalAlignment;

import org.apache.poi.ss.usermodel.VerticalAlignment;

public class XuatfileThuBai {

private final ConnectDB db = new ConnectDB();

public void selectDSLop(ArrayList<Lop> dsLop) {

db.getDataLopQuan(dsLop);

}

public void selectDanhGia(ArrayList<BaiTap> dsbt) {

db.getDataBaiTap(dsbt);

}

public void xuatFileAction(HSSFWorkbook workbook, ArrayList<BaiTap> dsbt, String tenLop) {

db.selectThuBaiByTenLop(dsbt, tenLop);

initExcelFile(workbook, dsbt, tenLop);

}

public void init(BaitapUI jFrameXuatFile, JComboBox jComboBoxchonlop, ArrayList<Lop> dsLop) {

jFrameXuatFile.setTitle("Xuất file Excel");

jFrameXuatFile.setLocationRelativeTo(null);

jFrameXuatFile.setDefaultCloseOperation(EXIT\_ON\_CLOSE);

initCbo(jComboBoxchonlop, dsLop);

}

public void initCbo(JComboBox jComboBoxchonlop, ArrayList<Lop> dsLop) {

jComboBoxchonlop.setModel(new DefaultComboBoxModel());

db.getDataLopQuan(dsLop);

Iterator<Lop> itLopHoc = dsLop.iterator();

while (itLopHoc.hasNext()) {

jComboBoxchonlop.addItem(itLopHoc.next().getTenLop());

}

jComboBoxchonlop.setSelectedIndex(0);

}

public void initExcelFile(HSSFWorkbook workbook, ArrayList<BaiTap> dsbt, String tenLop) {

//Create a cellstyle

CellStyle cellStyle = workbook.createCellStyle();

cellStyle.setAlignment(HorizontalAlignment.CENTER);

cellStyle.setVerticalAlignment(VerticalAlignment.CENTER);

//Create a spreadsheet

HSSFSheet sheet = workbook.createSheet("Thu Bai " + tenLop);

//Create a row object

HSSFRow row;

//Create cell & setvalue

row = sheet.createRow(0);

Cell cell0 = row.createCell(0);

Cell cell1 = row.createCell(1);

Cell cell2 = row.createCell(2);

Cell cell3 = row.createCell(3);

cell0.setCellStyle(cellStyle);

cell1.setCellStyle(cellStyle);

cell2.setCellStyle(cellStyle);

cell3.setCellStyle(cellStyle);

cell0.setCellValue("Mã bài tập");

cell1.setCellValue("Ghi chú");

cell2.setCellValue("Link bài tập");

cell3.setCellValue("Nhóm");

//Create cells & rows for the data

for (int i = 0; i < dsbt.size(); i++) {

row = sheet.createRow(i + 1);

for (int j = 0; j < 4; j++) {

Cell cell = row.createCell(j);

cell.setCellStyle(cellStyle);

if (cell.getColumnIndex() == 0) {

cell.setCellValue(dsbt.get(i).getMaBaiTap());

}

if (cell.getColumnIndex() == 1) {

cell.setCellValue(dsbt.get(i).getGhiChu());

}

if (cell.getColumnIndex() == 2) {

cell.setCellValue(dsbt.get(i).getLinkBaiTap());

}

if (cell.getColumnIndex() == 3) {

cell.setCellValue(dsbt.get(i).getMaNhom());

}

}

}

public void writingExcelFile(HSSFWorkbook workbook, String tenLop) {

File a = new File("THUBAI" + tenLop + ".xls");

try (FileOutputStream out = new FileOutputStream(a)) {

workbook.write(out);

out.close();

File f = a.getAbsoluteFile();

if (!f.exists()) {

JOptionPane.showMessageDialog(null, "File không tồn tại.", "Message", JOptionPane.OK\_OPTION);

} else if (!Desktop.isDesktopSupported()) {

JOptionPane.showMessageDialog(null, "Desktop không hỗ trợ file này.", "Warning", JOptionPane.OK\_OPTION);

} else {

Desktop.getDesktop().open(f);

}

} catch (FileNotFoundException ex) {

JOptionPane.showMessageDialog(null, ex, "Message", JOptionPane.CLOSED\_OPTION);

} catch (IOException ex) {

JOptionPane.showMessageDialog(null, ex, "Message", JOptionPane.CLOSED\_OPTION);

}

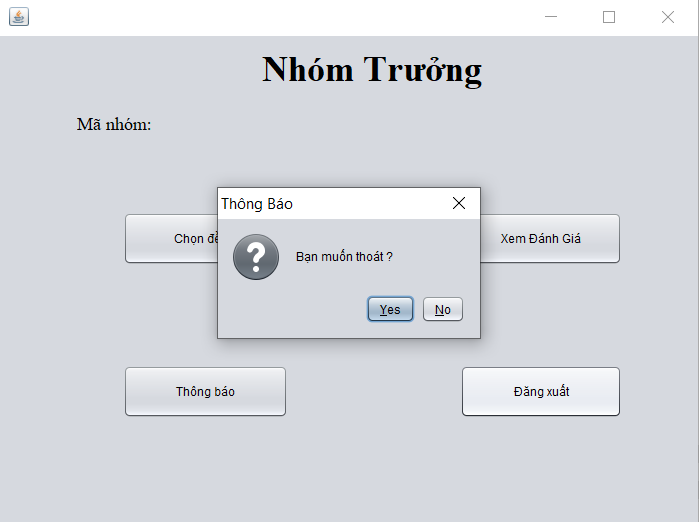
// }

}

}

### Phạm Thanh Quý – Đăng xuất

* Thiết kế giao diện vào ra



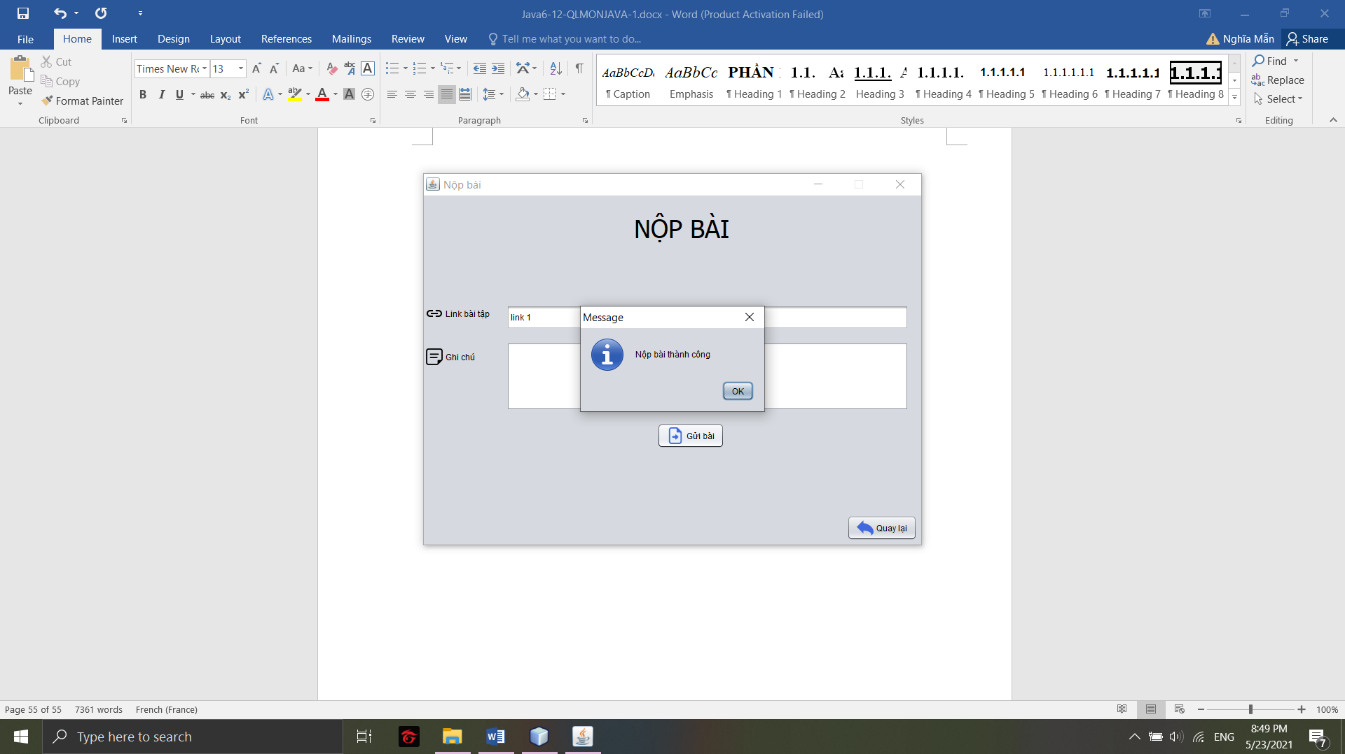
* Mô tả đối tượng trên màn hình
* 1 button đăng xuất : khi nguoif dùng click vào đăng xuất, hệ thông sẽ hỏi bạn có muốn đăng xuất không. Bấm yes sẽ đăng xuất và hiển thị form đăng nhập.
* Mô tả thuật toán để kiểm tra tính hợp lệ của đầu vào
* Mô tả thuật toán xử lý tương ứng với các tác động của người dùng
* Code mẫu :

new Manhinhdangnhap().setVisible(true) ;

this.setVisible(false) ;

### Mẫn Văn Quân – Nộp bài

* Thiết kế giao diện vào/ra



* Mô tả đối tượng trên màn hình :
  + Text Field : nhập link bài tập
  + Text Area : nhập ghi chú
  + Button Gửi bài : gửi bài lên CSDL
  + Button Quay lại : quay lại màn hình chính của nhóm trưởng
* Mô tả thuật toán kiểm tra tính hợp lệ dữ liệu đầu vào
  + Nếu link bài tập để trống thì sẽ có thông báo lỗi
* Mô tả thuật toán xử lý tương ứng với các tác động của người dùng



* Các thư viện và đoạn code mẫu

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import javax.swing.JOptionPane;

import java.util.ArrayList;

public void getMaBaiTap(ArrayList maBT) {

String sql = "select mabaitap from baitap where mabaitap = (select max(mabaitap) from baitap)";

try {

getConnect();

stm = cnn.createStatement();

rs = stm.executeQuery(sql);

while (rs.next()) {

maBT.add(new BaiTap(rs.getString(1)));

}

closeConnect();

} catch (SQLException ex) {

JOptionPane.showMessageDialog(null, "Loi lay du lieu\n" + ex);

}

}

public void insertBaiTap(String linkBaiTap, String noiDung, int maNhom) {

ArrayList<BaiTap> m = new ArrayList<>();

db.getMaBaiTap(m);

String a = null;

if (m.isEmpty()) {

a = "BT001";

} else {

String b[] = m.get(0).getMaBaiTap().split("BT");

int maBT = Integer.parseInt(b[1]) + 1;

switch (dem(maBT)) {

case 1:

a = "BT00" + maBT;

break;

case 2:

a = "BT0" + maBT;

break;

case 3:

a = "BT" + maBT;

break;

}

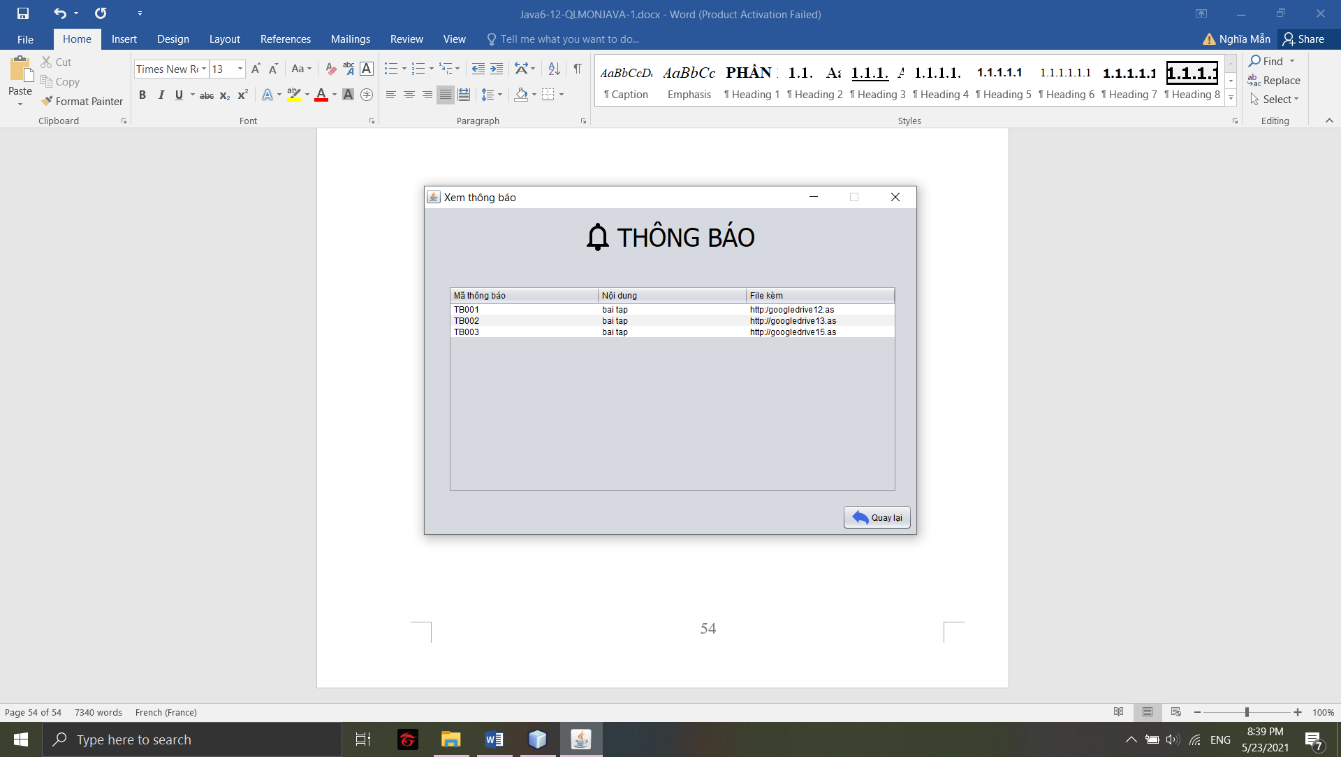
}

db.insertBaiTap(a, noiDung, linkBaiTap, maNhom);

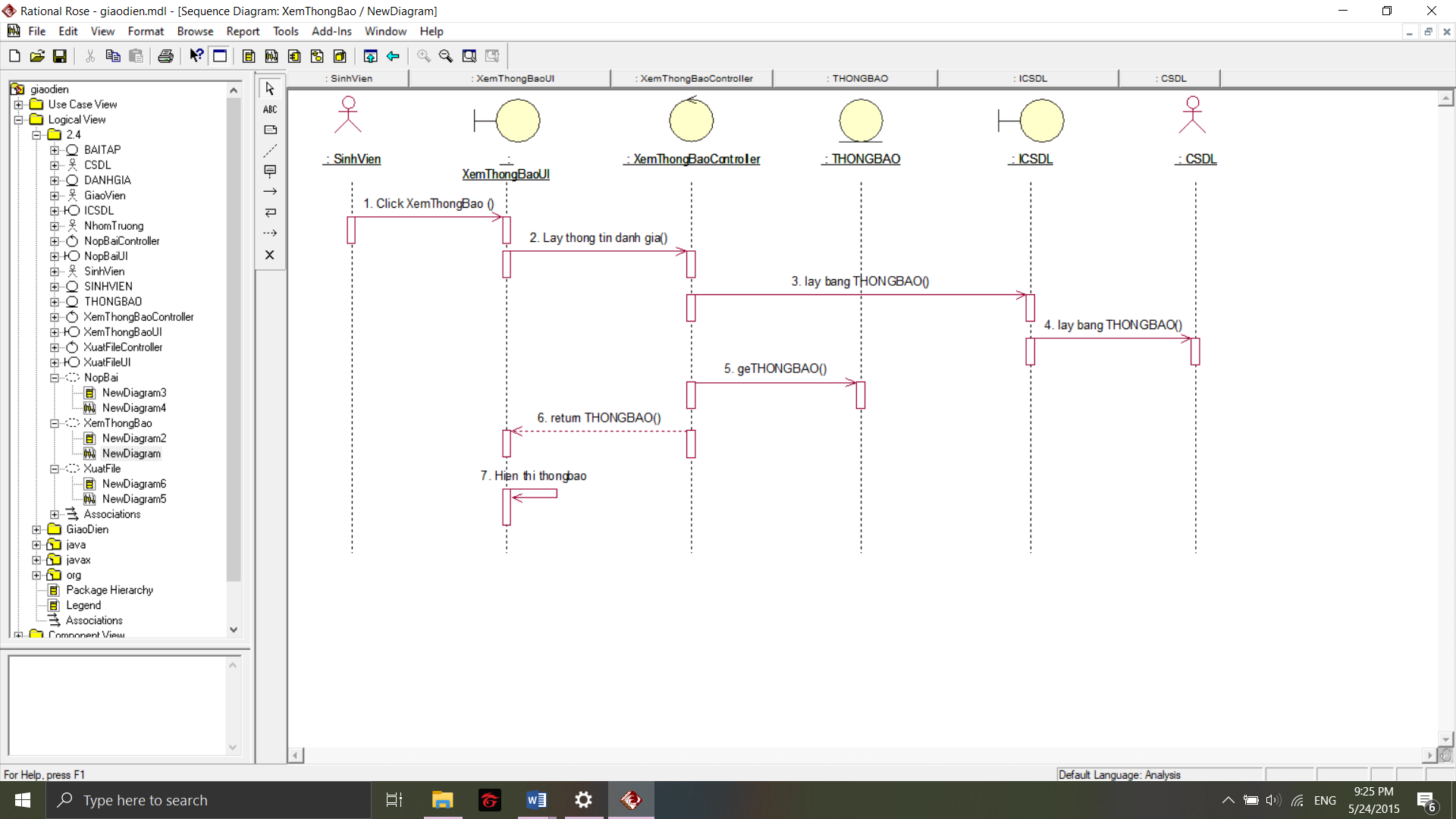
}

### Mẫn Văn Quân – Xem thông báo

* Thiết kế giao diện vào/ra



* Mô tả các đối tượng trên màn hình :
  + 1 table : hiển thị thông tin của các thông báo
  + 1 button quay lại : quay lại màn hình chính của sinh viên
* Mô tả thuật toán kiểm tra tính hợp lệ dữ liệu đầu vào
* Mô tả thuật toán xử lý tương ứng với các tác động của người dùng



* Code mẫu

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import javax.swing.JOptionPane;

import java.util.ArrayList;

public void getDataThongBao(ArrayList dsThongBao) {

String sql = "SELECT \* FROM THONGBAO";

try {

getConnect();

stm = cnn.createStatement();

rs = stm.executeQuery(sql);

while (rs.next()) {

dsThongBao.add(new ThongBao(rs.getString(1), rs.getString(2), rs.getString(3)));

}

closeConnect();

} catch (SQLException ex) {

JOptionPane.showMessageDialog(null, "Loi lay du lieu tu bang THONGBAO \n" + ex.getMessage());

}

}

public void loadTable(ArrayList<ThongBao> list, JTable jTableThongBao) {

selectDSThongBao(list);

jTableThongBao.setModel(new CustomTableThongBao(list));

}

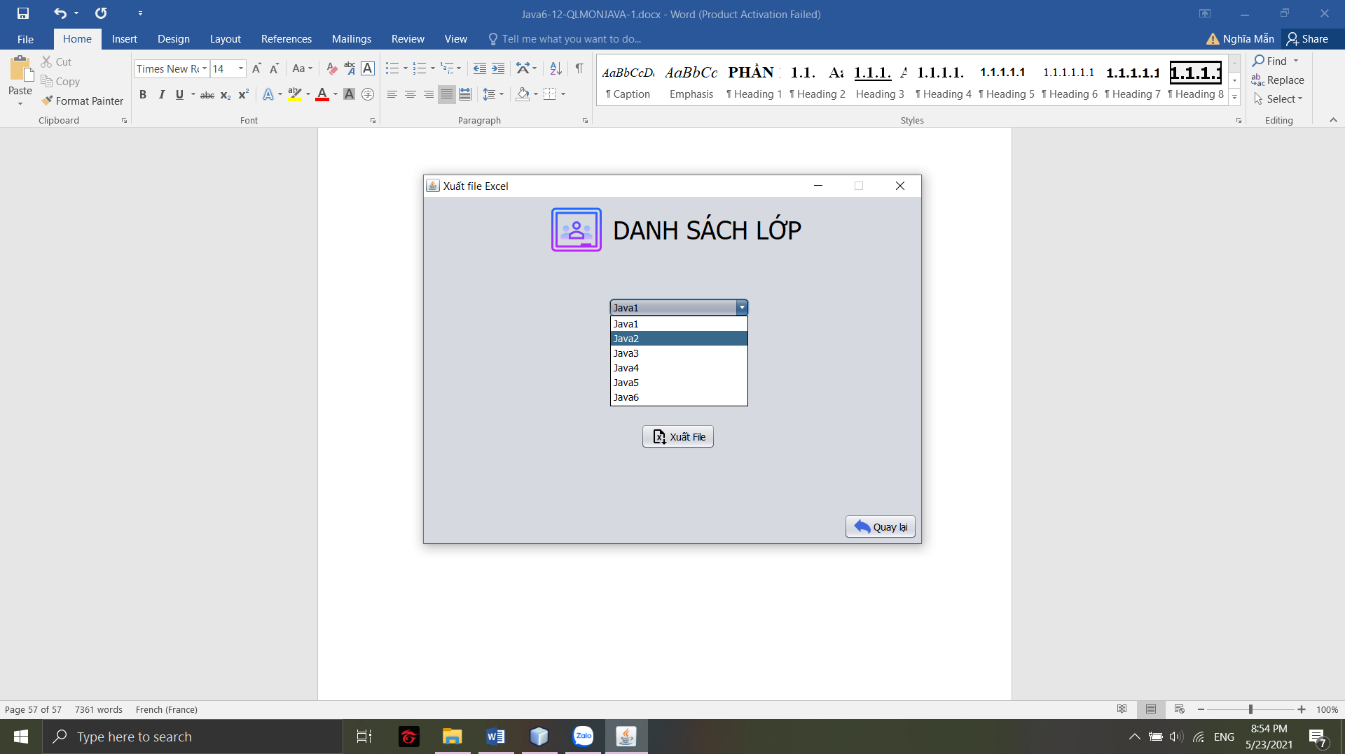
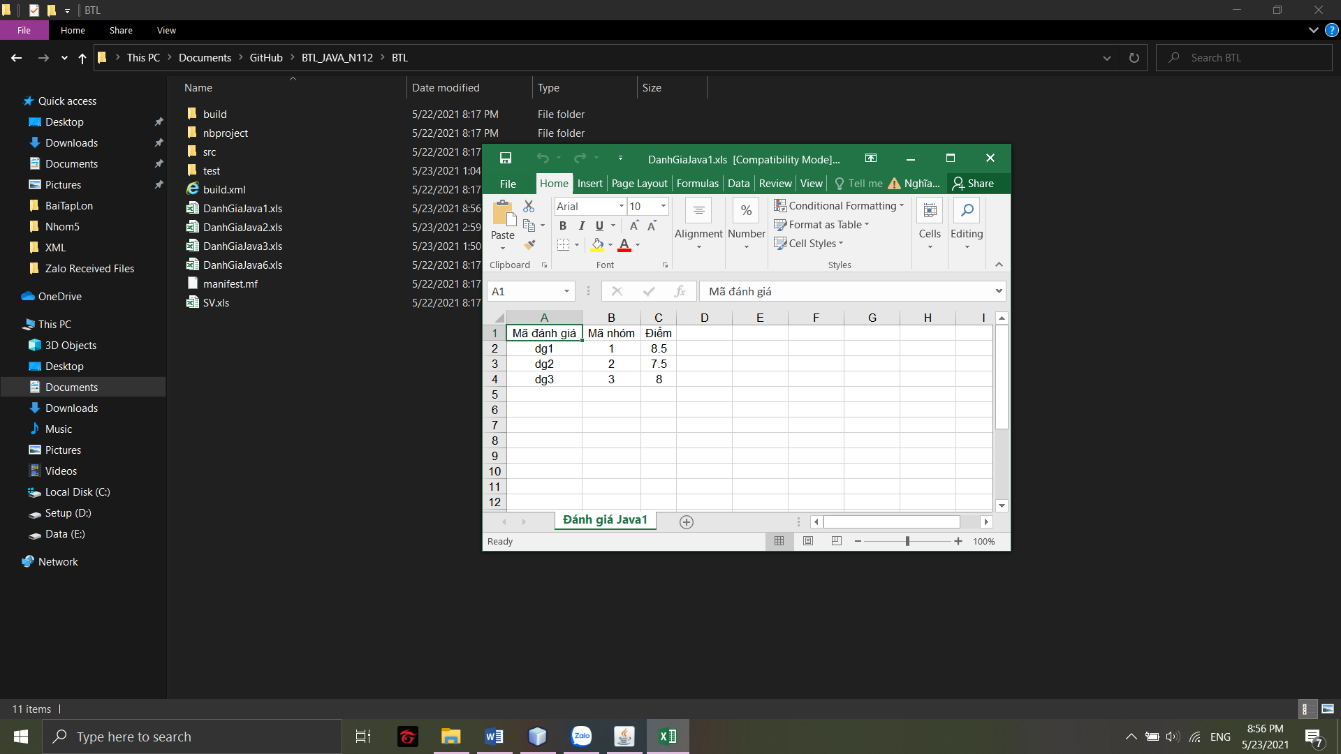
public void selectDSThongBao(ArrayList<ThongBao> dsThongBao) {

db.getDataThongBao(dsThongBao);

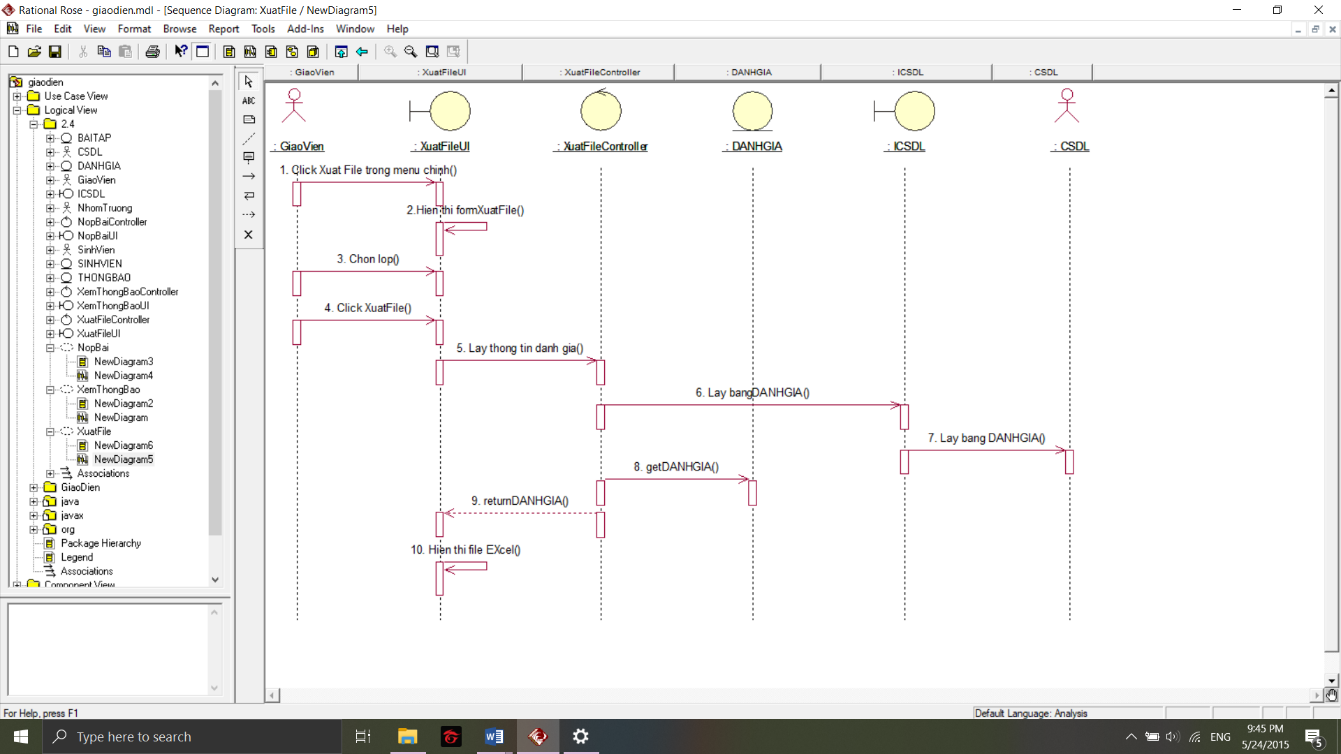
}

### Mẫn Văn Quân – Xuất file

* Thiết kế giao diện vào/ra



* Mô tả các đối tượng trên màn hình :
  + Combobox : hiển thị danh sách các lớp học
  + Button Xuất file : Xuất file Excel
  + File Excel bảng tính : hiển thị danh sách đánh giá của giáo viên
  + Button Quay lại : quay lại màn hình chính của giáo viên
* Mô tả thuật toán kiểm tra tính hợp lệ dữ liệu đầu vào
  + Nếu danh sách lớp trống sẽ không xuất được file
* Mô tả thuật toán xử lý tương ứng với các tác động của người dùng



* Code mẫu

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import javax.swing.JOptionPane;

import java.util.ArrayList;

import org.apache.poi.hssf.usermodel.HSSFRow;

import org.apache.poi.hssf.usermodel.HSSFSheet;

import org.apache.poi.hssf.usermodel.HSSFWorkbook;

import org.apache.poi.ss.usermodel.Cell;

import org.apache.poi.ss.usermodel.CellStyle;

import java.awt.Desktop;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.IOException;

public void initExcelFile(HSSFWorkbook workbook, ArrayList<DanhGia> dsDanhGia, String tenLop) {

CellStyle cellStyle = workbook.createCellStyle();

cellStyle.setAlignment(HorizontalAlignment.CENTER);

cellStyle.setVerticalAlignment(VerticalAlignment.CENTER);

HSSFSheet sheet = workbook.createSheet("Đánh giá " + tenLop);

HSSFRow row;

row = sheet.createRow(0);

Cell cell0 = row.createCell(0);

Cell cell1 = row.createCell(1);

Cell cell2 = row.createCell(2);

cell0.setCellStyle(cellStyle);

cell1.setCellStyle(cellStyle);

cell2.setCellStyle(cellStyle);

cell0.setCellValue("Mã đánh giá");

cell1.setCellValue("Mã nhóm");

cell2.setCellValue("Điểm");

for (int i = 0; i < dsDanhGia.size(); i++) {

row = sheet.createRow(i + 1);

for (int j = 0; j < 3; j++) {

Cell cell = row.createCell(j);

cell.setCellStyle(cellStyle);

if (cell.getColumnIndex() == 0) {

cell.setCellValue(dsDanhGia.get(i).getMadanhgia());

}

if (cell.getColumnIndex() == 1) {

cell.setCellValue(dsDanhGia.get(i).getManhom());

}

if (cell.getColumnIndex() == 2) {

cell.setCellValue(dsDanhGia.get(i).getDiem());

}

}

}

for (int i = 0; i < 3; i++) {

sheet.autoSizeColumn(i);

}

writingExcelFile(workbook, tenLop);

}

public void writingExcelFile(HSSFWorkbook workbook, String tenLop) {

File a = new File("DanhGia" + tenLop + ".xls");

try (FileOutputStream out = new FileOutputStream(a)) {

workbook.write(out);

out.close();

File f = a.getAbsoluteFile();

if (!f.exists()) {

JOptionPane.showMessageDialog(null, "File không tồn tại.", "Message", JOptionPane.OK\_OPTION);

} else if (!Desktop.isDesktopSupported()) {

JOptionPane.showMessageDialog(null, "Desktop không hỗ trợ file này.", "Warning", JOptionPane.OK\_OPTION);

} else {

Desktop.getDesktop().open(f);

}

} catch (FileNotFoundException ex) {

JOptionPane.showMessageDialog(null, ex, "Message", JOptionPane.CLOSED\_OPTION);

} catch (IOException ex) {

JOptionPane.showMessageDialog(null, ex, "Message", JOptionPane.CLOSED\_OPTION);

}

}

# KẾT LUẬN VÀ BÀI HỌC KINH NGHIỆM

## Nội dung đã thực hiện

Thông qua thực hiện bài tập lớn, nhóm 12 học được thêm nhiều kỹ năng cũng như kinh nghiệm có ích như các kiến thức về java cơ bản, như kỹ năng làm việc nhóm, quy trình phát triển phần mềm.

Vận dụng được kiến thức lập trình hướng đối tượng vào lập trình java, hiểu được các khái niệm bẫy lỗi, xử lí gom rác, tuyển tập đối tượng, đơn- đa luồng, hiểu và quản lý được giao diện đồ họa trong java.

Bài học kinh nghiệm rút ra

* Phân công công việc cho từng thành viên rõ ràng.
* Trước khi thảo luận , nhóm trưởng và các thành viên phải chuẩn bị kỹ nội dung thảo luận và phần việc được giao.
* Mỗi thành viên phải có trách nhiệm với kết quả chung của nhóm.
* Các thành viên hoàn thành đúng hạn các công việc đã được giao.

## Hướng phát triển.

Với những kiến thức đã học được trong chương trình học, nhóm 12 đã hoàn thành được những chức năng cơ bản của chương trình và cố gắng phát triển bài tập lớn để có thể ứng dụng được trong thực tế. Đồng thời, khi thực hiện bài tập lớn nhóm 12 học được thêm nhiều kỹ năng cũng như kinh nghiệm có ích như kỹ năng làm việc nhóm, quy trình phát triển phần mềm và nhiều kiến thức khác.

Nhóm 12 sẽ tiếp tục phát triển ứng dụng này trong tương lai để có thể xây dựng một phần mềm quản lý nhóm học môn Java với những nền tảng, kiến thức căn bản đã được nhóm triển khai trong đề tài này và những kiến thức nâng cao để hoàn thiện chương trình.