

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



BÁO CÁO BÀI TẬP LỚN
THIẾT KẾ VÀ LẬP TRÌNH WEB

Đề tài: HỆ THỐNG QUẢN LÝ ĐỒ ÁN

Giảng viên hướng dẫn: ThS. HOÀNG ANH VIỆT

Giảng viên hướng dẫn : TS. Hoàng Anh Việt

Nhóm sinh viên thực hiện : Đàm Minh Tiến,

Vương Ngọc Quỳnh Hương,

Nguyễn Thị Huyền.

Học kỳ : 20172

Github: https://github.com/damminhtien/project_management

MỤC LỤC

LỜI NÓI ĐẦU	5
CHƯƠNG I: KHẢO SÁT, ĐẶC TẢ YÊU CẦU BÀI TOÁN	6
1. Giới thiệu tổng quan về hệ thống.....	6
2. Mô tả yêu cầu hệ thống.....	7
3. Lựa chọn công nghệ giải quyết.....	8
3.1 Thiết kế giao diện người dùng.....	8
Bootstrap	8
Ejs.....	8
3.2 Thiết kế server-side.....	9
NodeJs	9
Express js	9
3.3 Thiết kế cơ sở dữ liệu	10
PostgreSql	10
Pg Module	11
CHƯƠNG II: PHÂN TÍCH YÊU CẦU BÀI TOÁN	12
1. Mô hình chức năng (Functional modeling)	12
1.1. Đặc tả ca sử dụng.....	12
1.1.1. Chức năng quản lý thông tin cá nhân	12
1.1.2. Chức năng Đăng ký giảng viên.....	13
1.1.3. Chức năng Liên hệ	14
1.1.4. Chức năng Nộp bài	15
1.1.5. Chức năng Xem điểm	16
1.1.6. Chức năng Xem đề tài	17
1.1.7. Chức năng Quản lý sinh viên.....	18
1.1.8. Chức năng Chấm điểm.....	19
1.1.9. Chức năng Quản lý người dùng.....	21

1.1.10. Chức năng Quản lý báo cáo đồ án	24
1.2 Biểu đồ ca sử dụng	26
2. Mô hình cấu trúc (Structural modeling)	27
2.1. Biểu đồ lớp chức năng Quản lý thông tin cá nhân	27
2.2. Biểu đồ lớp chức năng Xem đề tài	27
2.3. Biểu đồ lớp chức năng Đăng ký giảng viên	28
2.4. Biểu đồ lớp chức năng Nộp bài	29
2.5. Biểu đồ lớp chức năng Liên hệ.....	30
2.6. Biểu đồ lớp chức năng Quản lý sinh viên.....	31
2.7. Biểu đồ lớp chức năng Chấm điểm	32
2.8. Biểu đồ lớp chức năng Quản lý người dùng.....	33
2.9. Biểu đồ lớp chức năng Quản lý báo cáo đồ án.....	34
2.9. Biểu đồ lớp chức năng Xem điểm	35
CHƯƠNG III: THIẾT KẾ HỆ THỐNG.....	36
1. Mô hình thiết kế.....	36
1.1 Mẫu thiết kế (Design Pattern).....	36
1.2 Thiết kế module và phương thức.....	37
1.3 Thiết kế tầng quản lý cơ sở dữ liệu	37
1.4 Thiết kế giao diện người dùng.....	38
CHƯƠNG IV: THỰC HIỆN	39
GIẢI QUYẾT CÁC BÀI TOÁN	39
1. Đặt vấn đề	39
2. Giải quyết từng vấn đề.....	39
2.3 Bài toán authentication:	42
2.4 Render ảnh:.....	42
TỔNG KẾT.....	43
1. Kết quả đạt được	43
2. Hướng phát triển	45

3. Kiến nghị và lời cảm ơn.....	46
<i>Tài liệu tham khảo</i>	47
<i>Chú thích</i>	47

LỜI NÓI ĐẦU

Trên thế giới cũng như ở Việt Nam, công nghệ thông tin đã trở thành một công nghệ mũi nhọn. Nhịp độ phát triển của ngành công nghệ thông tin đang là một vấn đề rất được các ngành khoa học, giáo dục, kinh tế ... quan tâm, việc ứng dụng Công nghệ thông tin vào các lĩnh vực phải làm sao để đạt được hiệu quả, sự lựa chọn và vận dụng những phần mềm sao cho phù hợp và đạt kết quả tốt.

Xét trong lĩnh vực giáo dục, đối với các trường đại học nói chung và Trường Đại học Bách Khoa Hà Nội nói riêng, việc quản lý đồ án là thiết yếu và rất quan trọng. Để bắt đầu rồi thực hiện cho đến lúc kết thúc đồ án là cả một quá trình nhiều giai đoạn nhỏ, mà một giáo viên lại hướng dẫn cho nhiều sinh viên, vì vậy để quản lý đồ án tốt nhất, hiệu quả và chính xác nhất thì cần một hệ thống thực hiện thay vì làm thủ công. Nhận thấy vấn đề nêu trên là rất cần thiết, với bài tập lớn lần này, chúng em quyết định thực hiện phân tích, thiết kế và xây dựng hệ thống quản lý đồ án.

Trong bài tập lớn này, chúng em sử dụng công nghệ dựa trên lõi nodejs để viết server, postgre làm hệ quản trị cơ sở dữ liệu.

Để hoàn thành được đồ án này, chúng em đã cùng nhau cố gắng làm việc tích cực và không ngừng cải thiện cũng như phát triển hệ thống được tối ưu nhất. Chúng em đã đặt ra các mục tiêu hàng tuần phải đạt được để có thể đi đến kết quả cuối cùng đúng tiến độ và yêu cầu của môn học. Chúng em rất mong nhận được những nhận xét, đánh giá ... để chúng em có thể tích lũy thêm kinh nghiệm cho bản thân.

Chúng em xin chân thành cảm ơn thầy!

Hà Nội, ngày 28 tháng 03 năm 2018

CHƯƠNG I: KHẢO SÁT, ĐẶC TẢ YÊU CẦU BÀI TOÁN

1. Giới thiệu tổng quan về hệ thống

Hệ thống chuyên dụng quản lý đồ án toàn bộ quá trình làm cho đến kết thúc, theo một cách trực quan, thuận tiện và chính xác, phục vụ cho nhiều đối tượng, nhằm tiết kiệm thời gian và phát huy tính hiệu quả.

Hệ thống của chúng em xây dựng sẽ có phạm vi như sau:

- Đối với Giảng viên hướng dẫn đồ án sẽ có những chức năng như sau:

- + Quản lý sinh viên
- + Xem đề tài
- + Trả lời liên hệ từ sinh viên
- + Chấm điểm
- + Quản lý thông tin cá nhân

- Đối với sinh viên:

- + Quản lý thông tin cá nhân
- + Đăng ký giảng viên hướng dẫn
- + Xem đề tài
- + Liên hệ với giảng viên hướng dẫn
- + Nộp bài
- + Xem điểm

- Đối với quản trị viên:

- + Quản lý người dùng
- + Quản lý báo cáo đồ án

2. Mô tả yêu cầu hệ thống

Quản lý đồ án sinh viên là một công việc rất quan trọng và không thể thiếu trong các trường đại học nói chung và Trường Đại học Bách Khoa Hà Nội nói riêng. Sau mỗi kì làm và bảo vệ đồ án của sinh viên, người quản lý đồ án phải lưu lại các tài liệu liên quan và thông tin đồ án. Các tài liệu và thông tin cần phải lưu bao gồm: báo cáo đồ án, các file đính kèm, thông tin sinh viên và thông tin giảng viên hướng dẫn sinh viên đó, điểm của đồ án...

Trong quá trình quản lý đồ án, khi người quản lý muốn tìm kiếm đề tài theo chủ đề hay theo giảng viên hướng dẫn, khóa, hệ đào tạo,...nếu phải tìm kiếm thủ công, thì rất mất nhiều thời gian và công sức cộng với việc lưu trữ lại các đồ án làm tài liệu rất cồng kềnh mất nhiều diện tích mà không đầy đủ nhiều khi còn gây mất mát và hỏng dữ liệu. Sinh viên phải nộp bài theo tiến độ hàng tuần, những nội dung ấy cũng phải được lưu lại theo thứ tự để cho giảng viên hướng dẫn đồ án cũng có thể theo dõi và có những nhận xét. Những bài nộp hàng tuần đó cũng chính là căn cứ để giảng viên chấm điểm và là bằng chứng rõ ràng cho điểm mà giảng viên chấm cho sinh viên. Thêm vào đó, thông tin về sinh viên và giảng viên cũng cần thiết phải lưu lại. Vì vậy mỗi người cần có một tài khoản để có thể dễ dàng quản lý hiệu quả.

Từ những thực tế trên, đặt ra một bài toán Quản lý Đồ án với các yêu cầu hệ thống như sau:

Người quản lý:

- Cập nhật dữ liệu và chỉnh sửa thông tin Đồ án một cách nhanh chóng, chính xác.
- Thống kê, tìm kiếm các thông tin liên quan đến đồ án theo các tiêu chí khác nhau.
- Lưu trữ được bản mềm của đồ án.

- Quản lý người sử dụng hệ thống (giảng viên, sinh viên).

Người sử dụng:

- Xem các thông tin về đồ án một cách trực quan và rõ ràng.
- Liên hệ và giải đáp những thắc mắc về đồ án.
- Tìm kiếm đồ án, giảng viên theo các tiêu chí khác nhau: như tìm kiếm đồ án theo ngành, theo chủ đề, theo tên, theo giảng viên hướng dẫn..., tìm kiếm giảng viên theo hướng nghiên cứu.
- Quản lý thông tin cá nhân: thêm, sửa thông tin cá nhân...

3. Lựa chọn công nghệ giải quyết

3.1 Thiết kế giao diện người dùng

Bootstrap

- Với sự phát triển mạnh mẽ của các thiết bị di động thì xu hướng thiết kế mobile-first ngày càng phổ biến trong các website hiện đại.
- Bootstrap xây dựng sẵn responsive css trên các thiết bị iphones, tablets, và desktops. Tính năng này khiến cho người dùng tiết kiệm được rất nhiều thời gian trong việc tạo ra một website thân thiện với các thiết bị điện tử, thiết bị cầm tay.
- Bootstrap tương thích với tất cả các trình duyệt (Chrome, Firefox, IE, Safari, Opera...). Tuy nhiên với trình duyệt IE, Bootstrap chỉ hỗ trợ từ IE9 trở lên.

Trong bài tập lớn lần này chúng em sẽ sử dụng công nghệ bootstrap 4.0.

Ejs

Ejs là template engine cho phép biên dịch ra mã html và javascript ngay trên server.

Ejs có các đặc điểm nổi bật:

- Tốc độ biên dịch và render nhanh

- Các thẻ mẫu đơn giản: <% %>
- Hỗ trợ cả máy chủ Js và trình duyệt
- Lưu trữ tĩnh các template

3.2 Thiết kế server-side

NodeJs

- NodeJs là một nền tảng được xây dựng trên môi trường Chrome V8 Engine, nó được dùng để xây dựng các ứng dụng web. NodeJs cung cấp các thư viện phong phú ở dạng javascript module khác nhau giúp đơn giản hóa việc lập trình và giảm bớt thời gian ở mức thấp nhất.
- NodeJs sử dụng cơ chế điều hướng sự kiện và vào ra không đồng bộ.
- NodeJs có tốc độ thực thi rất nhanh và có khả năng mở rộng nhanh chóng. NodeJs có khả năng xử lý hàng ngàn kết nối đồng thời, đáp ứng một lượng lớn người dùng sử dụng trang web.
- NodeJs đang ngày càng trở nên lớn mạnh hơn nhờ vào cộng đồng đông đảo.

Express js

Express là một framework nhỏ và tiện ích để xây dựng các ứng dụng web, cung cấp một lượng lớn các tính năng mạnh mẽ để phát triển các ứng dụng web và mobile. Nó rất dễ dàng để phát triển các ứng dụng nhanh dựa trên NodeJs cho các ứng dụng web. Một số tính năng cơ bản của Express:

- Cho phép thiết lập các lớp trung gian để trả về các HTTP request.
- Định nghĩa bảng routing có thể được sử dụng với các hành động khác nhau dựa trên phương thức HTTP và URL.
- Cho phép trả về các trang HTML dựa vào các tham số truyền vào đến template.

ES6

Node.js được xây dựng dựa trên phiên bản mới của V8. Vì vậy, nodejs luôn được cập nhật các tiêu chuẩn mới của engine.

Sơ qua về ES6, là một tập các tiêu chuẩn nâng cao của javascript, được xây dựng bởi hiệp hội các nhà phát triển máy tính Châu Âu nhằm đưa ra một thống nhất chung cho phiên bản của javascript trên các trình duyệt.

Các tính năng của ES6 trong Nodejs được chia thành 3 nhóm:

1. Shipping: Tính năng ổn định và được đặt mặc định trong nodejs
2. Staged: Tính năng gần hoàn thiện và có thể sử dụng bằng việc đặt cờ harmony, nhưng chưa ổn định
3. In progress: Tính năng đang thử nghiệm, có thể sử dụng, V8 team engineer có thể thay đổi mà không cần thông báo.

Full text search

Full text search (gọi tắt là FTS) là cách tự nhiên nhất để tìm kiếm thông tin, hết như Google, ta chỉ cần gõ từ khóa và nhấn enter thế là có kết quả trả về.

Một số ưu điểm của FTS như sau:

1. Kết quả search trả về nhiều.
2. Khi đánh index thì tốc độ search nhanh.
3. Tối ưu hơn việc sử dụng LIKE khi thao tác với các trường text lớn.

3.3 Thiết kế cơ sở dữ liệu

PostgreSql

- PostgreSQL là một hệ quản trị cơ sở dữ liệu được cấp phép bởi BSD, phiên bản mới nhất là 10 beta.
- PostgreSQL không chỉ là cơ sở dữ liệu quan hệ, nó là quan hệ hướng đối tượng. Một đặc điểm cơ bản của cơ sở dữ liệu quan hệ hướng đối tượng là hỗ trợ các đối tượng người dùng tự định nghĩa và các hành vi của chúng bao gồm các kiểu dữ liệu, các hàm, các thao tác, các tên miền và các chỉ mục. Điều này làm cho PostgreSQL cực kỳ mạnh và linh hoạt hơn nhiều so với các hệ quản trị cơ sở dữ liệu khác như MySQL, MariaDb và Firebird.

- MySQL, MariaDb và Firebird không có khả năng này. Để lưu trữ mảng các giá trị như thế này trong một cơ sở dữ liệu truyền thống, một bảng riêng với một dòng cho mỗi giá trị mảng có thể được tạo ra như một cách giải quyết.
- PostgreSQL hỗ trợ 2008 kiểu dữ liệu, ngoài ra có thể lưu trữ hình ảnh, nhạc, video.
- PostgreSQL tuân thủ đầy đủ theo chuẩn ANSI, ACID (4 nguyên tắc xử lý thao tác khi thiết kế hệ quản trị CSDL).
- Còn nhiều những ưu điểm đáng kể khác nữa... PostgreSQL thực sự là hệ quản trị CSDL mã nguồn mở đáng sử dụng.

Vì vậy nhóm chúng em lựa chọn PostgreSQL là hệ quản trị cơ sở dữ liệu cho hệ thống được xây dựng trong bài tập lớn môn học Thiết kế và lập trình web.

Pg Module

Node-postgres là một bộ sưu tập các module NodeJs để tương tác với cơ sở dữ liệu PostgreSQL. Hiện phiên bản mới nhất đã hỗ trợ các hàm async / await theo chuẩn ES7.

CHƯƠNG II: PHÂN TÍCH YÊU CẦU BÀI TOÁN

1. Mô hình chức năng (Functional modeling)

1.1. Đặc tả ca sử dụng

1.1.1. Chức năng quản lý thông tin cá nhân

Tên ca sử dụng: Quản lý thông tin cá nhân	ID: 1	Mức quan trọng: Cao
Tác nhân chính: Giảng viên, Sinh viên.	Loại ca sử dụng: Chi tiết, thiết yếu	
Người liên quan và công việc quan tâm: Giảng viên, sinh viên quản lý thông tin cá nhân của mình trên hệ thống.		
Mô tả tóm tắt: chức năng này cho phép giảng viên, sinh viên quản lý thông tin cá nhân của mình trên hệ thống.		
Kích hoạt: Người dùng chọn chức năng quản lý thông tin cá nhân sau khi đã thực hiện đăng nhập hệ thống.		
Loại: Bên ngoài		
Các mối quan hệ: <div><div>1. Liên quan (Association): Giảng viên, Sinh viên. (người dùng)</div><div>2. Bao hàm (include): Đăng nhập</div><div>3. Mở rộng (extend):</div><div>4. Tổng quát hóa (Generalization):</div></div>		
Các dòng sự kiện chính: <div><div>1. Giảng viên, Sinh viên chọn chức năng Quản lý thông tin cá nhân.</div><div>2. Hệ thống hiển thị các lựa chọn:<div><div>- Xem thông tin cá nhân</div><div>- Sửa thông tin cá nhân</div></div></div></div>		

<p>3. Người dùng chọn chức năng mình muốn thực hiện:</p> <ul style="list-style-type: none"> - Nếu Người dùng chọn chức năng “<i>Xem thông tin cá nhân</i>”, thực hiện S-1. - Nếu Người dùng chọn chức năng “<i>Sửa thông tin cá nhân</i>”, thực hiện S-2.
<p>Các dòng sự kiện rẽ nhánh:</p> <p>S-1: Xem thông tin cá nhân</p> <ol style="list-style-type: none"> 1. Hệ thống hiển thị thông tin cá nhân của Người dùng. <p>S-2: Sửa thông tin cá nhân.</p> <ol style="list-style-type: none"> 1. Hệ thống hiển thị form thông tin cá nhân để Người dùng sửa. 2. Người dùng sửa những thông tin cần sửa. 3. Người dùng chọn Hoàn tất. 4. Hệ thống cập nhật thông tin cá nhân của Người dùng.
<p>Các dòng sự kiện ngoại lệ:</p>

1.1.2. Chức năng Đăng ký giảng viên

Tên ca sử dụng: Đăng ký giảng viên		ID: 2	Mức quan trọng: Cao
Tác nhân chính: Sinh viên.		Loại ca sử dụng: Chi tiết, thiết yếu	
Người liên quan và công việc quan tâm: Sinh viên thực hiện đăng ký giảng viên hướng dẫn đồ án mà mình muốn.			
Mô tả tóm tắt: chức năng này cho phép sinh viên đăng ký giảng viên hướng dẫn đồ án trên hệ thống.			
Kích hoạt: Sinh viên chọn chức năng Đăng ký giảng viên trên hệ thống.			
Loại: Bên ngoài			
Các mối quan hệ:			
5. Liên quan (Association): Sinh viên.			
6. Bao hàm (include):			
7. Mở rộng (extend):			
8. Tổng quát hóa (Generalization):			

Các dòng sự kiện chính:

4. Sinh viên chọn chức năng Đăng ký giảng viên.
5. Hệ thống hiển thị form nhập Đăng ký:
 - Tên giảng viên
 - Mã số giảng viên
 - Bộ môn
 - Đề tài (nếu muốn đăng ký luôn)
 - Thông tin cá nhân của sinh viên đăng ký (Mã số sinh viên, tên sinh viên, lớp, ngành, số điện thoại, email).
6. Sinh viên nhập thông tin vào form và chọn Hoàn tất.
7. Hệ thống kiểm tra thông tin giảng viên và thông tin sinh viên nhập vào và thông báo kết quả.

Các dòng sự kiện rẽ nhánh:

4a: Thông tin không hợp lệ

1. Hệ thống hiển thị thông báo không hợp lệ ở những ô thông tin bị sai và yêu cầu điền lại
2. Sinh viên nhập lại thông tin cho chính xác, quay lại bước 3 trong luồng sự kiện chính.

Các dòng sự kiện ngoại lệ:

1.1.3. Chức năng Liên hệ

Tên chức năng: Liên hệ	ID: 3	Mức quan trọng: Cao
Tác nhân chính: Sinh viên	Loại chức năng: Chi tiết, cần thiết	
Người liên quan và công việc quan tâm: Sinh viên thực hiện liên hệ với giảng viên trên hệ thống về những thắc mắc cũng như các vấn đề liên quan đến đồ án.		

Mô tả tóm tắt: Chức năng này mô tả chức năng Liên hệ trên hệ thống của một Sinh viên.
Kích hoạt: Sinh viên chọn chức năng Liên hệ trên hệ thống sau khi đã đăng nhập hệ thống. Loại: Bên ngoài
Các mối quan hệ: <ol style="list-style-type: none"> 1. Liên quan (Association): Sinh viên. 2. Bao hàm (include): 3. Mở rộng (extend): 4. Tổng quát hóa (Generalization):
Các dòng sự kiện chính: <ol style="list-style-type: none"> 1. Sinh viên chọn Liên hệ. 2. Hệ thống hiển thị form nhập nội dung cần liên hệ: <ul style="list-style-type: none"> - Tiêu đề - Người nhận liên hệ - Nội dung cần trao đổi. - Ô đính kèm file (nếu muốn đính kèm file). 3. Sinh viên nhập vào form và chọn Gửi 4. Hệ thống thực hiện gửi liên hệ và thông báo kết quả, cập nhật vào cơ sở dữ liệu.
Các dòng sự kiện rẽ nhánh:
Các dòng sự kiện ngoại lệ:

1.1.4. Chức năng Nộp bài

Tên chức năng: Nộp bài	ID: 4	Mức quan trọng: Cao
Tác nhân chính: Sinh viên	Loại chức năng: Chi tiết, cần thiết	

Người liên quan và công việc quan tâm: Sinh viên thực hiện nộp bài báo cáo hàng tuần trên hệ thống.
Mô tả tóm tắt: Chức năng này mô tả chức năng Nộp bài trên hệ thống của một Sinh viên.
Kích hoạt: Sinh viên chọn chức năng Nộp bài trên hệ thống sau khi đã đăng nhập hệ thống. Loại: Bên ngoài
Các mối quan hệ: <ul style="list-style-type: none"> 5. Liên quan (Association): Sinh viên. 6. Bao hàm (include): 7. Mở rộng (extend): 8. Tổng quát hóa (Generalization):
Các dòng sự kiện chính: <ul style="list-style-type: none"> 5. Sinh viên chọn tên đồ án. 6. Sinh viên chọn mục Nộp bài. 7. Hệ thống hiển thị ô điền nội dung cần gửi cho giảng viên và ô đính kèm file. 8. Sinh viên đính kèm file và điền trực tiếp vào ô điền nội dung rồi chọn Nộp bài. 9. Hệ thống hiển thị gửi thành công và cập nhật vào hệ thống.
Các dòng sự kiện rẽ nhánh:
Các dòng sự kiện ngoại lệ:

1.1.5. Chức năng Xem điểm

Tên chức năng: Xem điểm	ID: 5	Mức quan trọng: Cao
Tác nhân chính: Sinh viên	Loại chức năng: Chi tiết, cần thiết	
Người liên quan và công việc quan tâm: Sinh viên xem điểm đồ án trên hệ thống.		
Mô tả tóm tắt: Chức năng này mô tả thao tác xem điểm trên hệ thống của sinh viên		
Kích hoạt: Sinh viên chọn chức năng Xem điểm trên hệ thống sau khi đã đăng nhập vào hệ thống.		

Loại: Bên ngoài
Các mối quan hệ: <ol style="list-style-type: none"> 1. Liên quan (Association): Sinh viên. 2. Bao hàm (include): 3. Mở rộng (extend): 4. Tổng quát hóa (Generalization):
Các dòng sự kiện chính: <ol style="list-style-type: none"> 1. Sinh viên chọn mục Xem điểm. 2. Hệ thống hiển thị danh sách tên bộ môn. 3. Người dùng chọn bộ môn. 4. Hệ thống hiển thị danh sách điểm bao gồm các cột: <ul style="list-style-type: none"> - Số thứ tự - Mã số sinh viên - Họ và tên - Bộ môn - Giáo viên hướng dẫn - Điểm - Xếp loại 5. Sinh viên tìm và xem điểm.
Các dòng sự kiện rẽ nhánh:
Các dòng sự kiện ngoại lệ:

1.1.6. Chức năng Xem đề tài

Tên chức năng: Xem đề tài	ID: 6	Mức quan trọng: Cao
Tác nhân chính: Giảng viên, Sinh viên. (người dùng)	Loại chức năng: Chi tiết, cần thiết	

Người liên quan và công việc quan tâm: Người dùng thao tác Xem đề tài đồ án trên hệ thống.
Mô tả tóm tắt: Chức năng này mô tả thao tác Xem đề tài đồ án trên hệ thống của giảng viên, sinh viên
Kích hoạt: Người dùng chọn chức năng Xem đề tài trên hệ thống sau khi đã đăng nhập vào hệ thống. Loại: Bên ngoài
Các mối quan hệ: <ul style="list-style-type: none"> 5. Liên quan (Association): Giảng viên, Sinh viên. 6. Bao hàm (include): 7. Mở rộng (extend): 8. Tổng quát hóa (Generalization):
Các dòng sự kiện chính: <ul style="list-style-type: none"> 6. Người dùng chọn mục Xem đề tài. 7. Hệ thống hiển thị danh sách tên bộ môn. 8. Người dùng chọn bộ môn. 9. Hệ thống hiển thị danh sách tên đề tài 10. Người dùng chọn một tên đề tài 11. Hệ thống hiển thị chi tiết đề tài đó bao gồm: <ul style="list-style-type: none"> - Tên đề tài - Nội dung đề tài - ...
Các dòng sự kiện rẽ nhánh:
Các dòng sự kiện ngoại lệ:

1.1.7. Chức năng Quản lý sinh viên

Tên chức năng: Quản lý sinh viên	ID: 7	Mức quan trọng: Cao
----------------------------------	-------	---------------------

Tác nhân chính: Giảng viên.	Loại chức năng: Chi tiết, cần thiết
Người liên quan và công việc quan tâm: Giảng viên thực hiện chức năng quản lý sinh viên trên hệ thống.	
Mô tả tóm tắt: Chức năng này mô tả thao tác Quản lý sinh viên trên hệ thống của giảng viên.	
Kích hoạt: Giảng viên chọn chức năng Quản lý sinh viên trên hệ thống. Loại: Bên ngoài	
Các mối quan hệ: 9. Liên quan (Association): Giảng viên. 10. Bao hàm (include): 11. Mở rộng (extend): 12. Tổng quát hóa (Generalization):	
Các dòng sự kiện chính: 12. Giảng viên chọn chức năng Quản lý sinh viên. 13. Hệ thống hiển thị danh sách sinh viên mà giảng viên hướng dẫn. 14. Giảng viên chọn vào sinh viên mà mình muốn kiểm tra tiến độ. 15. Hệ thống hiển thị chi tiết về thông tin và tiến độ của sinh viên.	
Các dòng sự kiện rẽ nhánh:	
Các dòng sự kiện ngoại lệ:	

1.1.8. Chức năng Chấm điểm

Tên chức năng: Chấm điểm	ID: 8	Mức quan trọng: Cao
Tác nhân chính: Giảng viên.		Loại chức năng: Chi tiết, cần thiết
Người liên quan và công việc quan tâm: Giảng viên thực hiện thao tác Chấm điểm cho sinh viên trên hệ thống.		
Mô tả tóm tắt: Chức năng này mô tả thao tác Chấm điểm cho sinh viên do Giảng viên thực hiện,		
Kích hoạt: Giảng viên chọn chức năng Chấm điểm trên hệ thống. Loại: Bên ngoài		

Các mối quan hệ:

13. Liên quan (Association): Giảng viên.

14. Bao hàm (include):

15. Mở rộng (extend):

16. Tổng quát hóa (Generalization):

Các dòng sự kiện chính:

16. Giảng viên chọn Chấm điểm.

17. Hệ thống hiển thị các lựa chọn:

- Nhập điểm
- Sửa điểm

18. Giảng viên chọn một trong 3 tùy chọn trên:

- Nếu chọn Nhập điểm, thực hiện S-1.
- Nếu chọn Sửa điểm, thực hiện S-2.

Các dòng sự kiện rẽ nhánh:

S-1: Nhập điểm

1. Hệ thống hiển thị bảng nhập điểm:

- Số thứ tự
- Mã số sinh viên
- Họ và tên
- Bộ môn
- Giáo viên hướng dẫn
- Điểm
- Xếp loại

2. Giảng viên nhập thông tin vào bảng và chọn Hoàn tất.

3. Hệ thống lưu lại cơ sở dữ liệu.

S-2: Sửa điểm

1. Hệ thống hiển thị bảng điểm và ô nhập mã số sinh viên cần sửa
2. Giảng viên nhập mã số sinh viên cần sửa và chọn Tìm
3. Hệ thống truy xuất cơ sở dữ liệu lấy ra thông tin điểm của sinh viên
4. Giảng viên thực hiện sửa và chọn Hoàn tất.
5. Hệ thống cập nhật điểm vào cơ sở dữ liệu và thông báo kết quả.

Các dòng sự kiện ngoại lệ:

S-2, 2a: Mã số sinh viên không hợp lệ

1. Hệ thống thông báo lỗi mã số sinh viên không tồn tại và yêu cầu nhập lại
2. Quay lại bước 2 trong luồng sự kiện S-2.

1.1.9. Chức năng Quản lý người dùng

Tên chức năng: Quản lý người dùng	ID: 9	Mức quan trọng: Cao
Tác nhân chính: Quản trị viên	Loại chức năng: Chi tiết, cần thiết	
Người liên quan và công việc quan tâm: Quản trị viên thực hiện chức năng Quản lý người dùng trên hệ thống.		
Mô tả tóm tắt:Chức năng này mô tả thao tác Quản lý người dùng của quản trị viên.		
Kích hoạt: Quản trị viên chọn chức năng Quản lý người dùng.		
Loại: Bên ngoài		
Các mối quan hệ: <div><div>1. Liên quan (Association): Quản trị viên.</div><div>2. Bao hàm (include):</div><div>3. Mở rộng (extend):</div><div>4. Tổng quát hóa (Generalization):</div></div>		
Các dòng sự kiện chính: <div><div>1. Quản trị viên chọn chức năng Quản lý người dùng.</div><div>2. Hệ thống sẽ hiển thị các lựa chọn:<div><div>- Thêm người dùng</div></div></div></div>		

- Xóa người dùng
- Sửa thông tin người dùng

3. Quản trị viên chọn một chức năng:

- Nếu chọn “Thêm người dùng”, thực hiện S-1.
- Nếu chọn “Xóa người dùng”, thực hiện S-2.
- Nếu chọn “Sửa thông tin người dùng”, thực hiện S-3.

Các dòng sự kiện rẽ nhánh:

S-1: Thêm người dùng

1. Hệ thống hiển thị form điền thông tin người dùng cần thêm:

- Họ và tên:
- Ngày sinh:
- Tên tài khoản (Mã số sinh viên | Mã số giảng viên):
- Mật khẩu (Mã số sinh viên | Mã số giảng viên):
- Email:
- Số điện thoại:
- Chuyên ngành:

2. Quản trị viên nhập thông tin người dùng cần thêm vào form rồi chọn **Thêm**.

3. Hệ thống kiểm tra tính hợp lệ của thông tin nhập vào và thông báo kết quả.

4. Ca sử dụng kết thúc.

S-2: Xóa người dùng

1. Hệ thống hiển thị form nhập mã số người dùng cần xóa.

2. Quản trị viên nhập một mã số người dùng cần xóa rồi chọn **OK**.

3. Hệ thống kiểm tra mã số nhập vào và hiển thị thông tin về người dùng sẽ bị xóa.

4. Quản trị viên xem thông tin đúng người dùng muốn xóa chưa và chọn **XÓA**.

5. Hệ thống hỏi lại “Có chắc chắn muốn xóa không?”

6. Quản trị viên chọn “Có”
7. Hệ thống thực hiện xóa người dùng và cập nhật lại danh sách người dùng.
8. Ca sử dụng kết thúc.

S-3: Sửa thông tin người dùng

1. Hệ thống hiển thị form nhập mã số người dùng cần sửa thông tin.
2. Quản trị viên nhập một mã số người dùng cần sửa thông tin rồi chọn **OK**.
3. Hệ thống hiển thị thông tin người dùng kèm theo nút **Sửa** ở từng mục.
4. Quản trị viên nhấp vào Sửa ở những mục cần sửa thông tin và nhập vào thông tin mới rồi chọn **Hoàn tất**.
5. Hệ thống kiểm tra thông tin nhập vào và cập nhật thông tin mới vào cơ sở dữ liệu .
6. Ca sử dụng kết thúc.

Các dòng sự kiện ngoại lệ:

S-1, 3a: Xử lý nhập thông tin không hợp lệ

1. Hệ thống sẽ hiển thị lại form thông báo đã điền kèm lỗi ở những ô thông tin bị sai.
2. Quản trị viên nhập lại thông tin cho chính xác, quay lại bước 2 trong luồng S-1.

S-2, 3a: Xử lý nhập mã số không hợp lệ

1. Hệ thống sẽ hiển thị thông báo mã số sai và yêu cầu nhập lại.
2. Quay lại bước 2 trong luồng S-2.

S-3, 3a: Xử lý nhập mã số không hợp lệ

1. Hệ thống sẽ hiển thị thông báo mã số sai và yêu cầu nhập lại
2. Quay lại bước 2 trong luồng S-3.

1.1.10. Chức năng Quản lý báo cáo đồ án

Tên chức năng: Quản lý báo cáo đồ án	ID: 10	Mức quan trọng: Cao
Tác nhân chính: Quản trị viên	Loại chức năng: Chi tiết, cần thiết	
Người liên quan và công việc quan tâm: Quản trị viên thực hiện chức năng quản lý báo cáo đồ án trên hệ thống.		
Mô tả tóm tắt:Chức năng này mô tả các thao tác quản lý đồ án trên hệ thống của một quản trị viên.		
Kích hoạt: Quản trị viên chọn chức năng Quản lý báo cáo đồ án trên hệ thống. Loại: Bên ngoài		
Các mối quan hệ: 10.Liên quan (Association): Quản trị viên. 11.Bao hàm (include): 12.Mở rộng (extend): 13.Tổng quát hóa (Generalization):		
Các dòng sự kiện chính: 5. Quản trị viên chọn mục Quản lý báo cáo đồ án. 6. Hệ thống hiển thị màn hình gồm các lựa chọn: - Thêm báo cáo - Xóa báo cáo - Xem báo cáo 7. Quản trị viên chọn một trong 3 lựa chọn trên và thực hiện thao tác. - Nếu chọn Thêm báo cáo, thực hiện S-1. - Nếu chọn Xóa báo cáo, thực hiện S-2. - Nếu chọn Xem báo cáo, thực hiện S-3.		
Các dòng sự kiện rẽ nhánh: S-1: Thêm báo cáo 1. Hệ thống hiển thị giao diện nhập thông tin về báo cáo cần thêm		

2. Quản trị viên nhập thông tin báo cáo cần thêm và chọn Hoàn tất.

3. Hệ thống thực hiện thêm báo cáo vào cơ sở dữ liệu.

S-2: Xóa báo cáo

1. Hệ thống hiển thị danh sách báo cáo và ô nhập tên hoặc nhập mã số báo cáo cần xóa.

2. Quản trị viên nhập tên hoặc nhập mã số báo cáo hoặc chọn trong danh sách báo cáo hiển thị báo cáo cần xóa và chọn vào báo cáo muốn xóa.

3. Hệ thống hiển thị nội dung báo cáo và thông tin liên quan đến báo cáo muốn xóa.

4. Quản trị viên nhấn chọn xác nhận xóa.

5. Hệ thống hiển thị thông báo: “ Có chắc chắn muốn xóa không?”

6. Quản trị viên chọn Xóa.

7. Hệ thống hiển thị thông báo xóa thành công và cập nhật vào cơ sở dữ liệu.

S-3: Xem báo cáo

1. Hệ thống hiển thị danh sách báo cáo và ô nhập tên hoặc nhập mã số báo cáo cần xem.

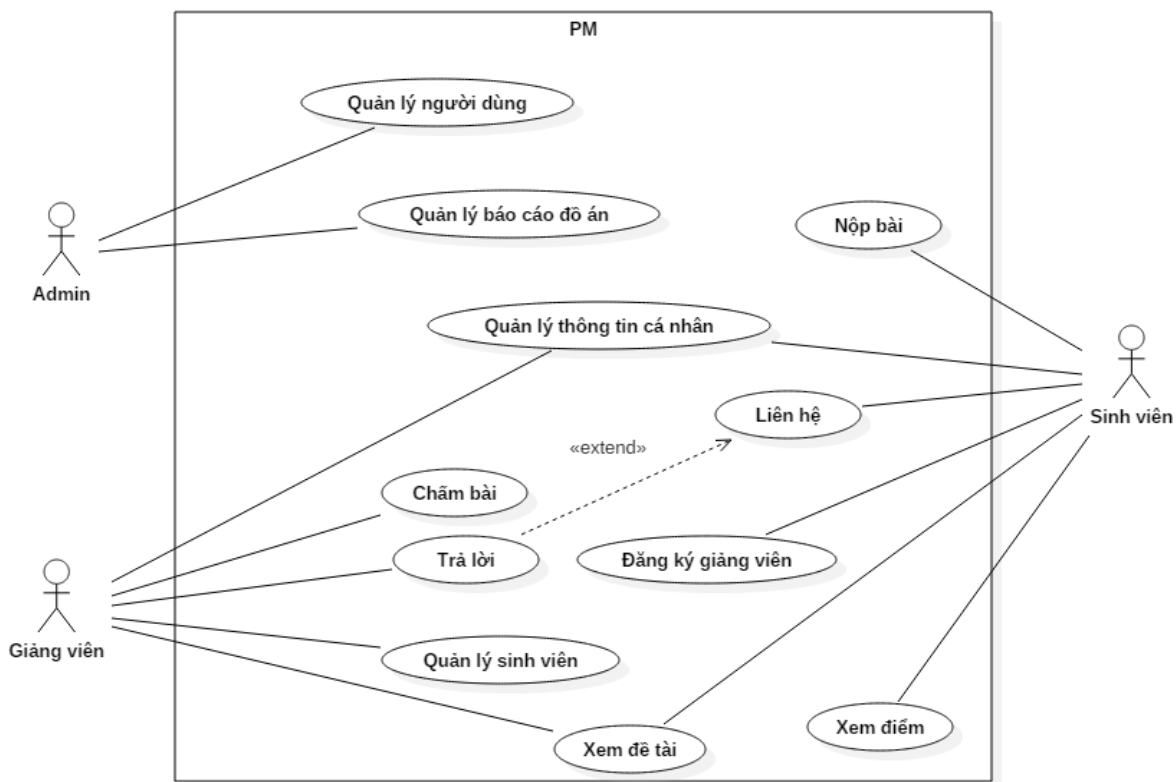
2. Quản trị viên nhập tên hoặc nhập mã số báo cáo hoặc chọn trong danh sách báo cáo hiển thị báo cáo cần xem và chọn vào báo cáo muốn xem.

3. Hệ thống hiển thị nội dung báo cáo đó và các thông tin liên quan đến báo cáo.

Các dòng sự kiện ngoại lệ:

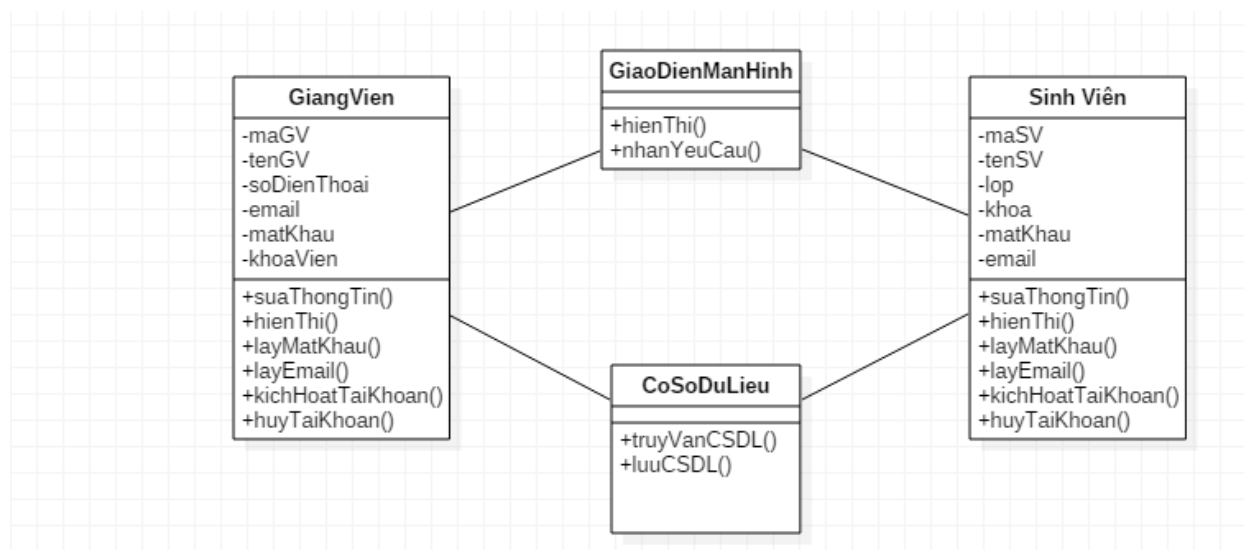
1.2 Biểu đồ ca sử dụng

Biểu đồ ca sử dụng cho cả hệ thống như sau:

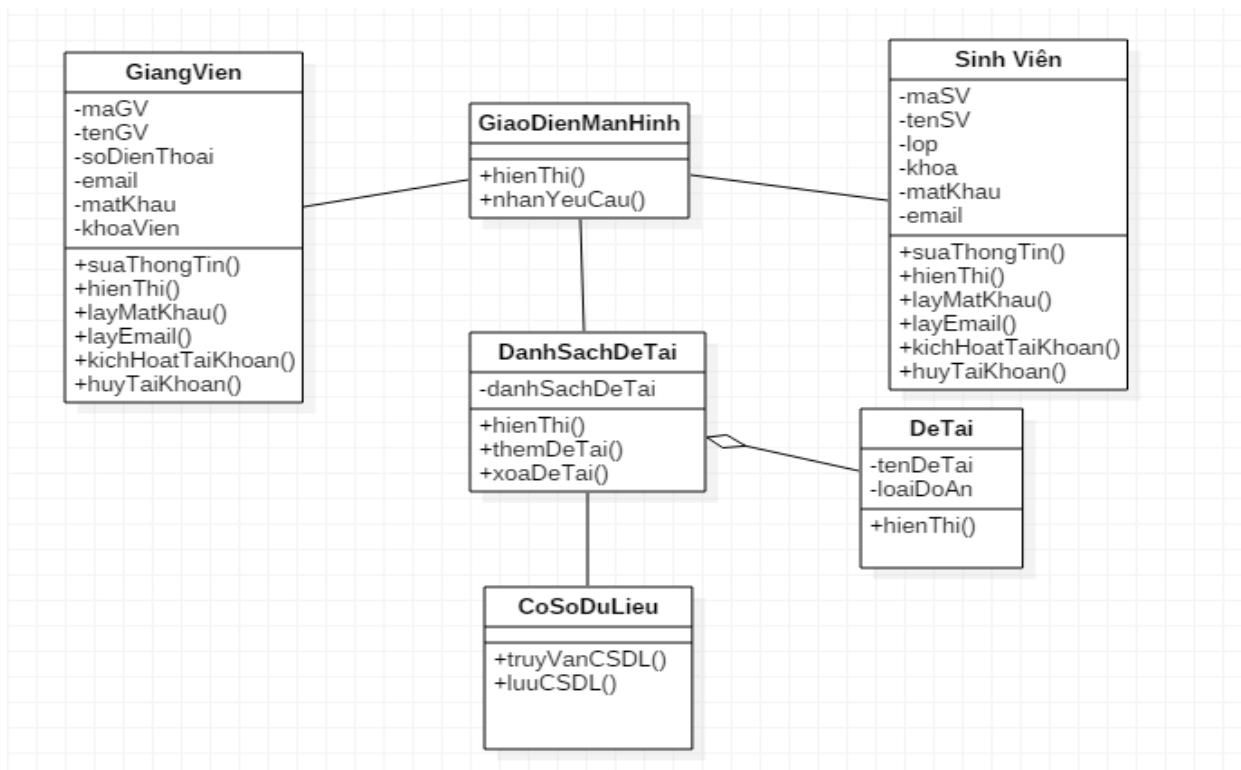


2. Mô hình cấu trúc (Structural modeling)

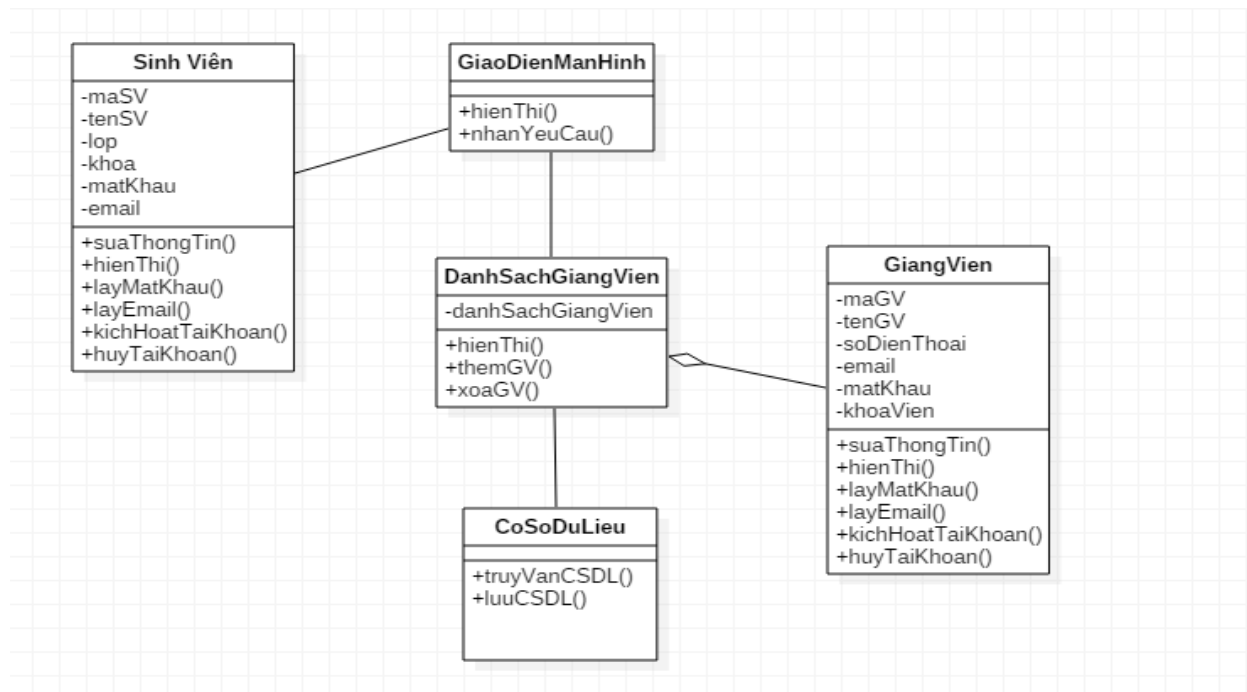
2.1. Biểu đồ lớp chức năng Quản lý thông tin cá nhân



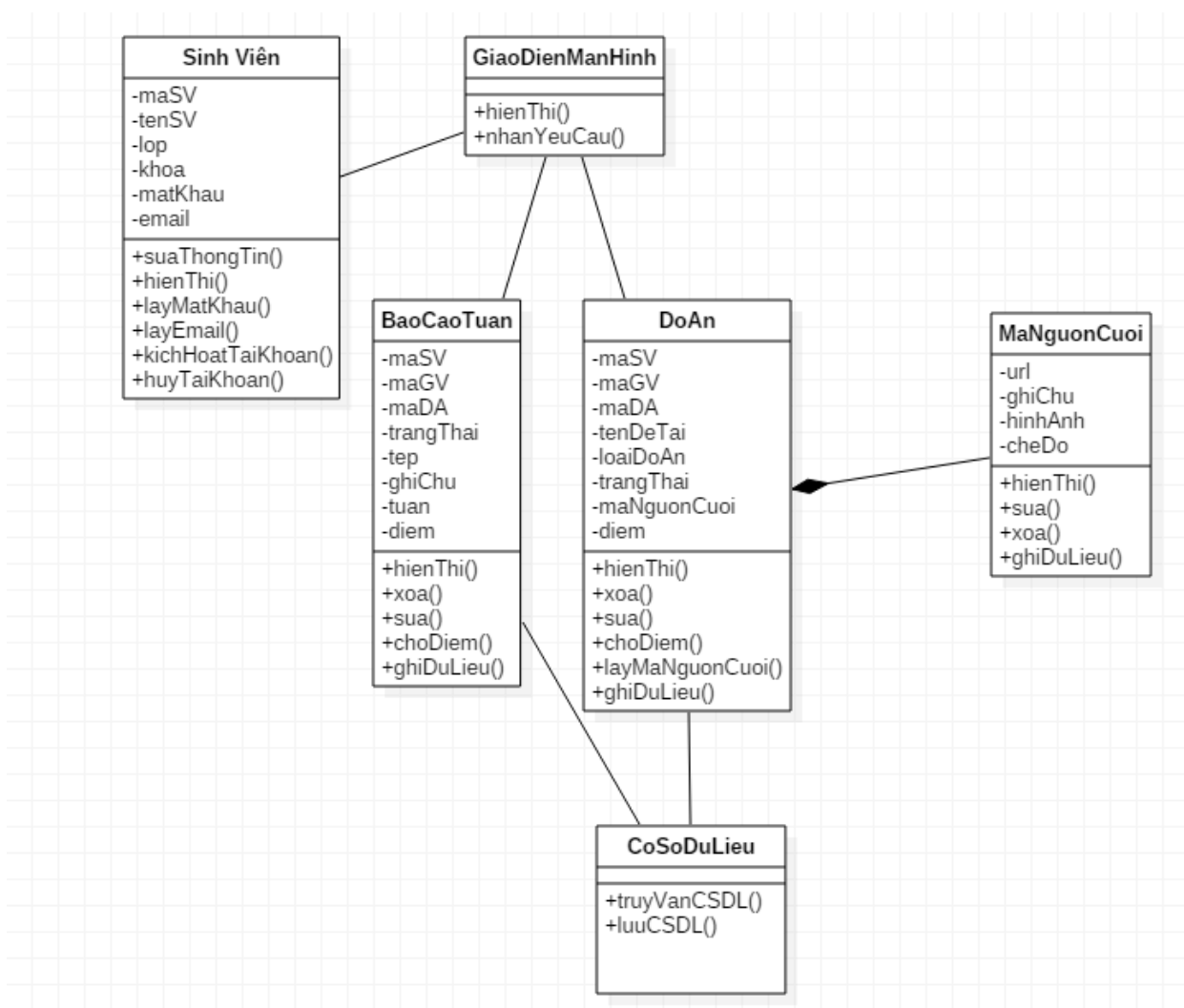
2.2. Biểu đồ lớp chức năng Xem đề tài



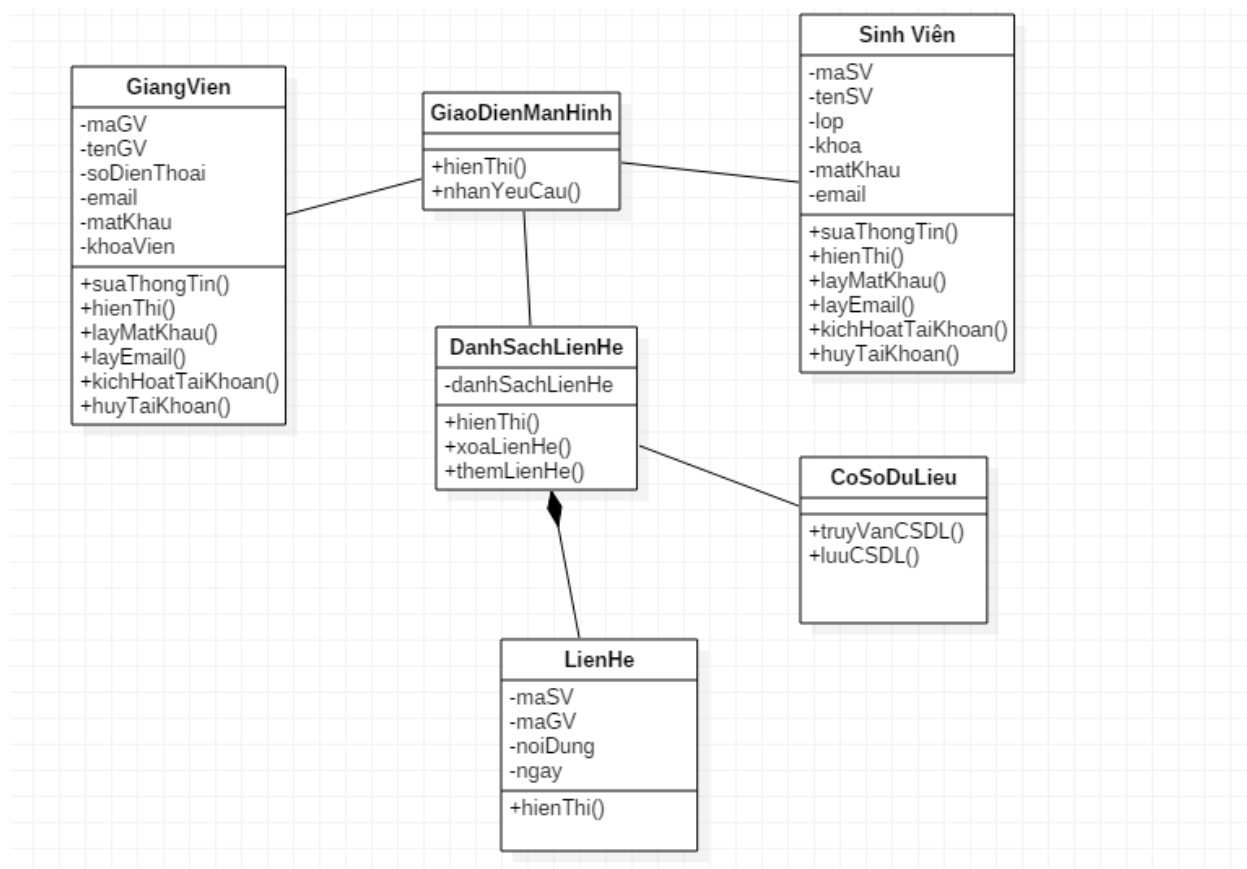
2.3. Biểu đồ lớp chức năng Đăng ký giảng viên



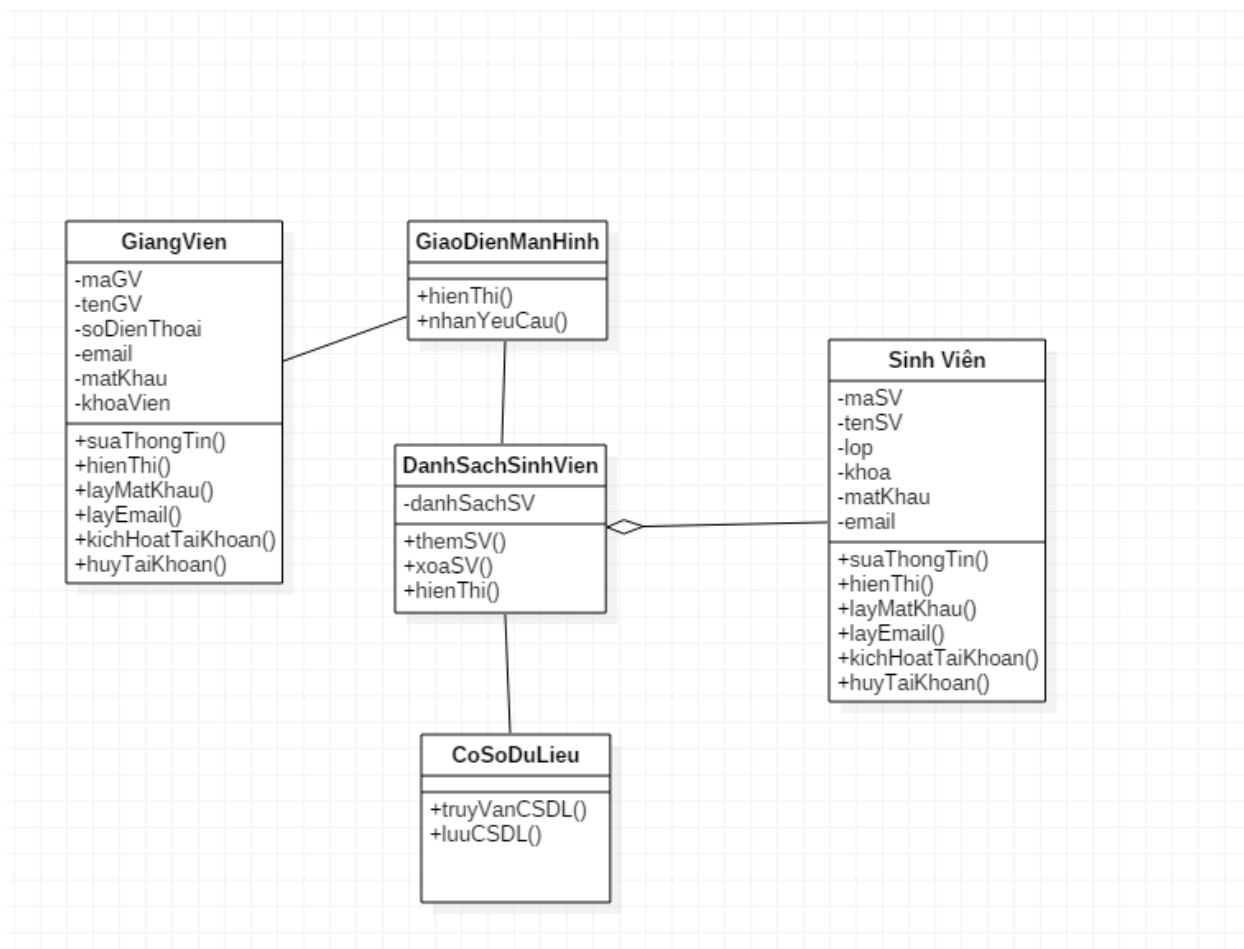
2.4. Biểu đồ lớp chức năng Nộp bài



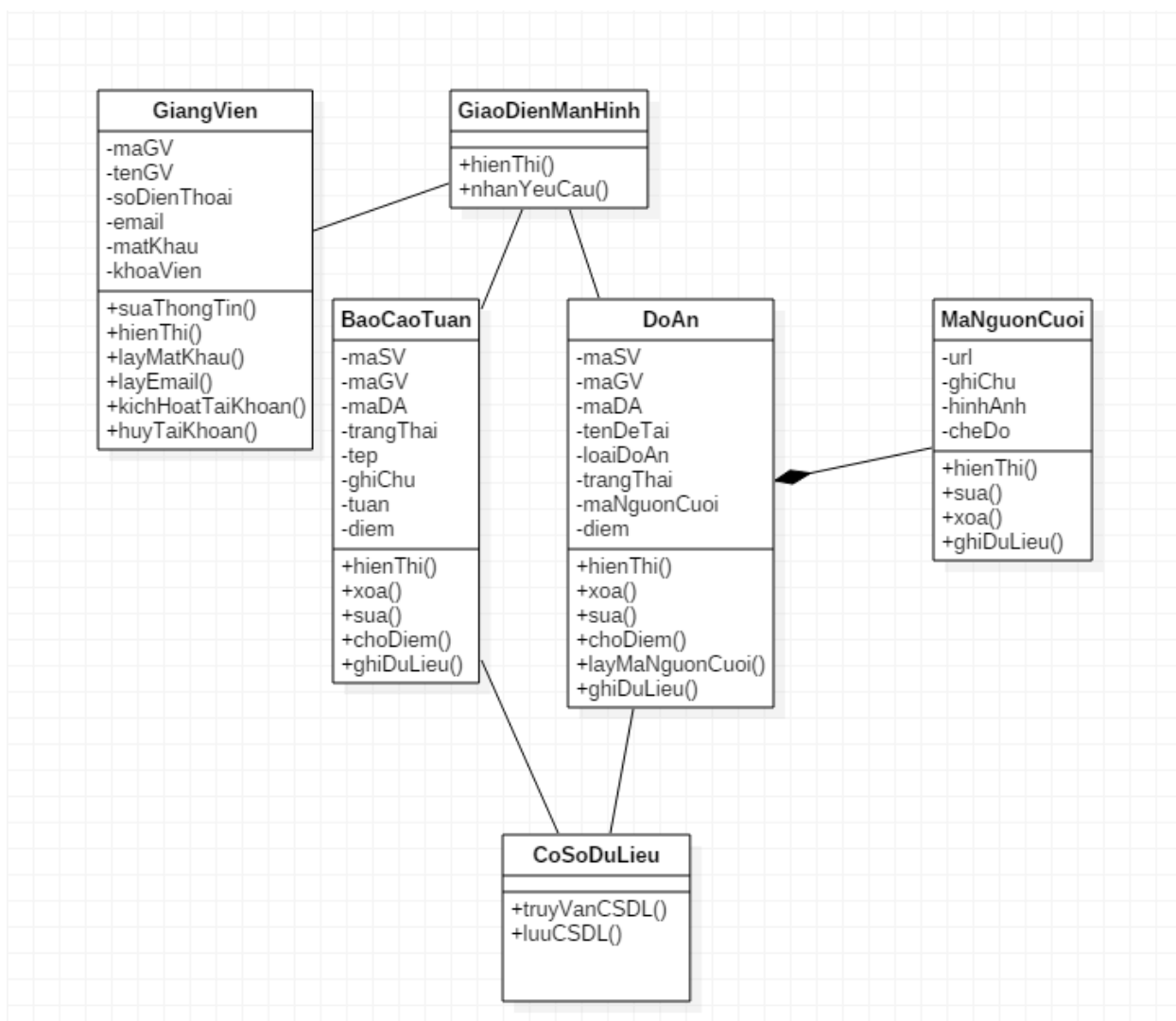
2.5. Biểu đồ lớp chức năng Liên hệ



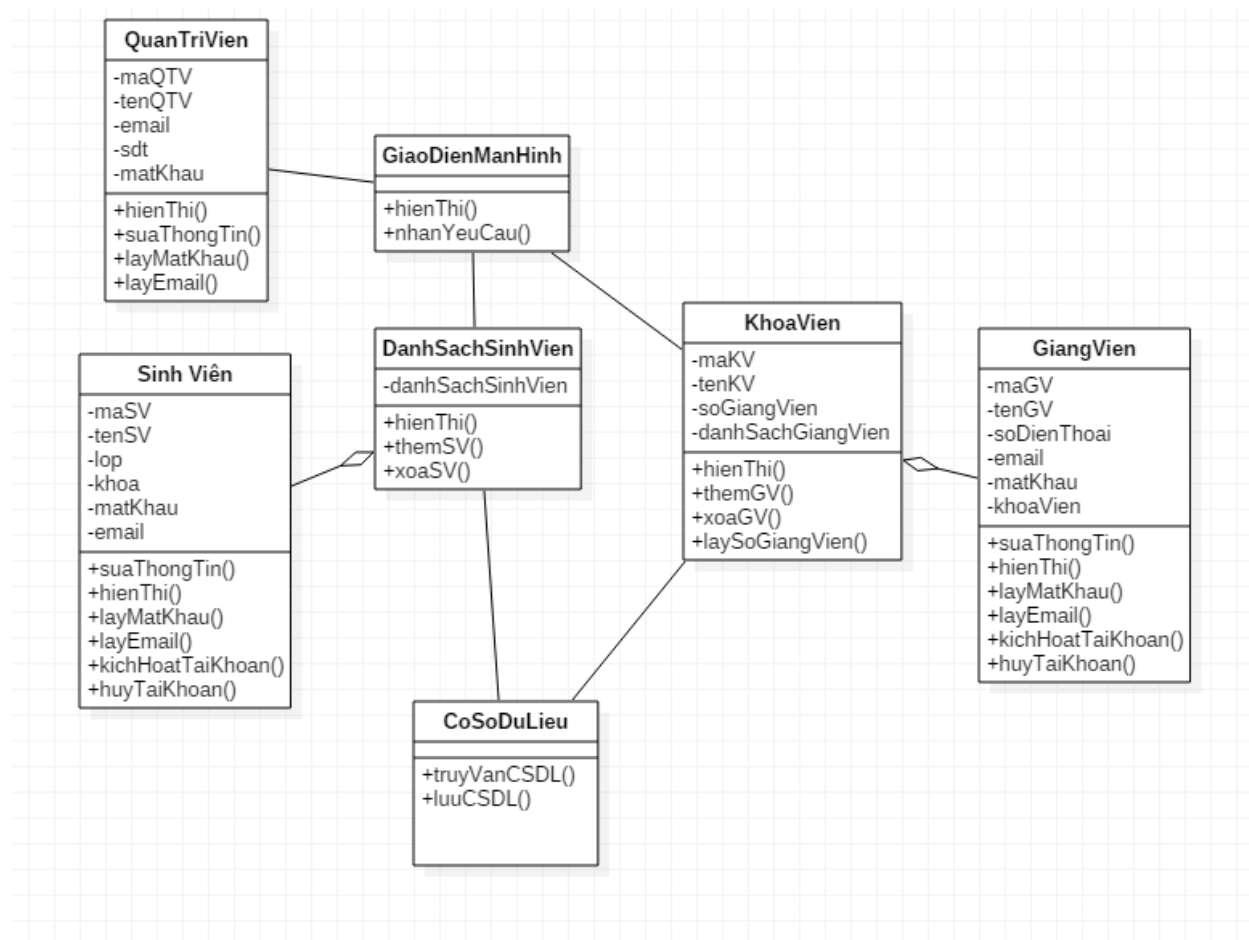
2.6. Biểu đồ lớp chức năng Quản lý sinh viên



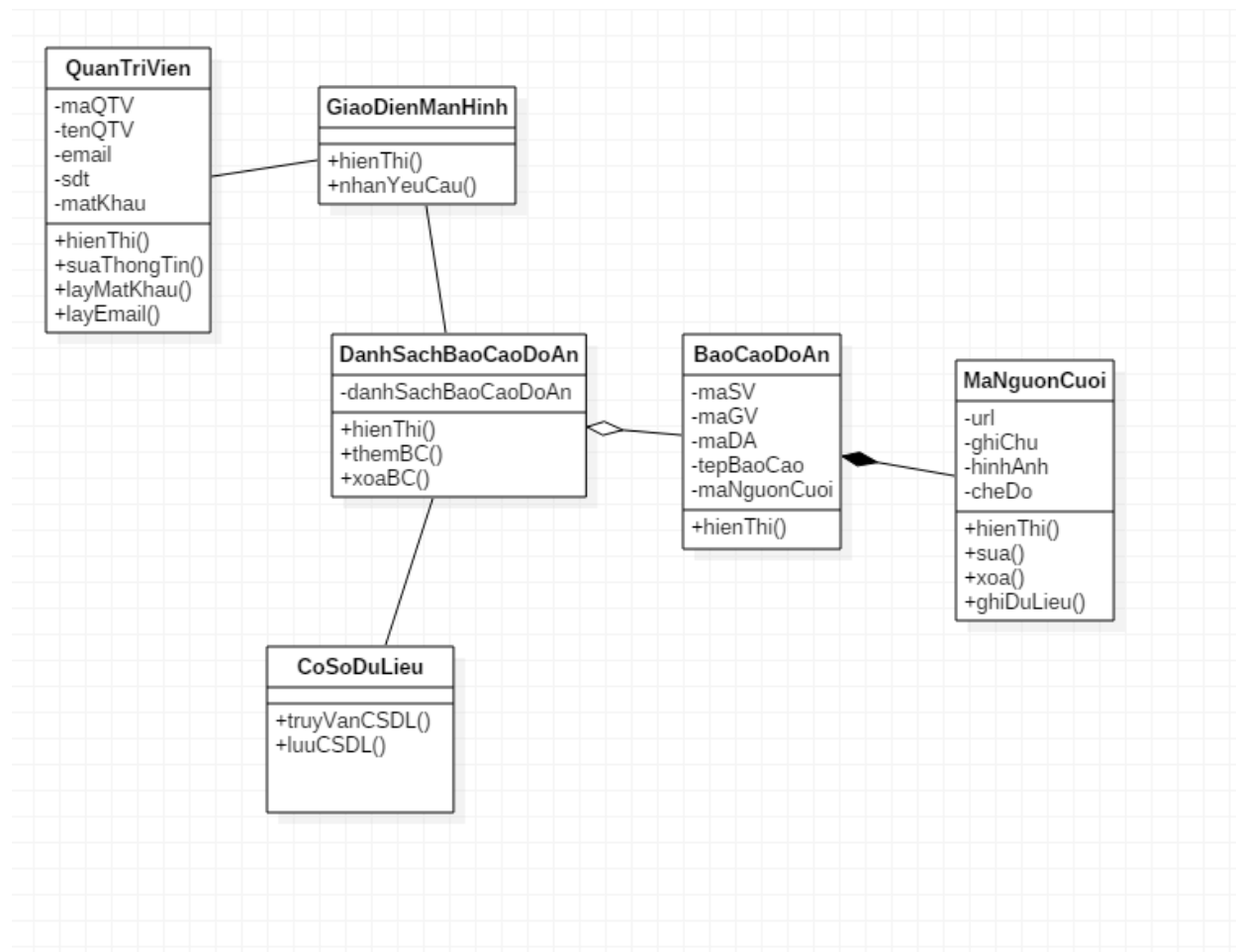
2.7. Biểu đồ lớp chức năng Chấm điểm



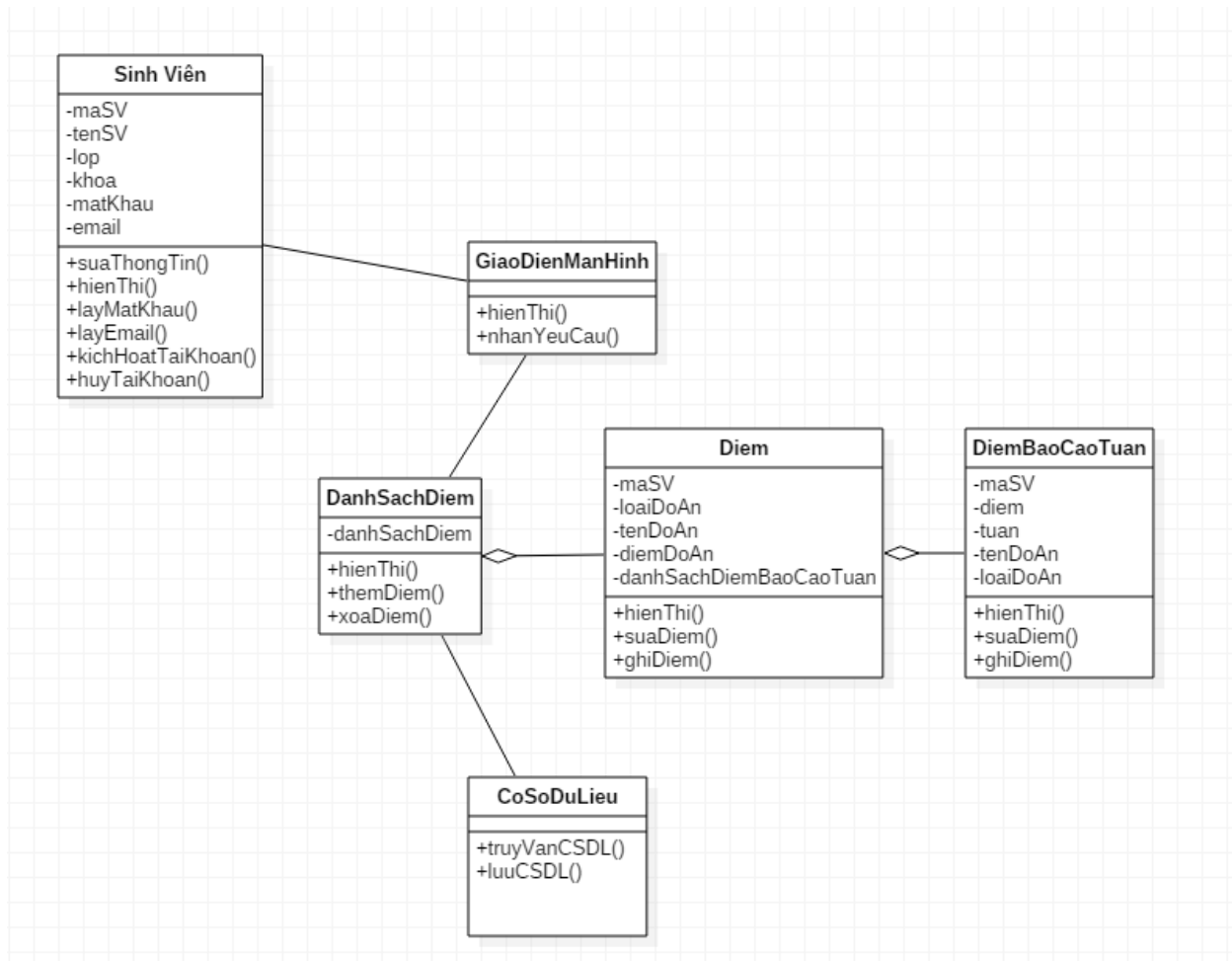
2.8. Biểu đồ lớp chức năng Quản lý người dùng



2.9. Biểu đồ lớp chức năng Quản lý báo cáo đồ án



2.9. Biểu đồ lớp chức năng Xem điểm



CHƯƠNG III: THIẾT KẾ HỆ THỐNG

1. Mô hình thiết kế

1.1 Mẫu thiết kế (Design Pattern)

Express generator giúp chúng ta tạo ra nhanh một khung ứng dụng.

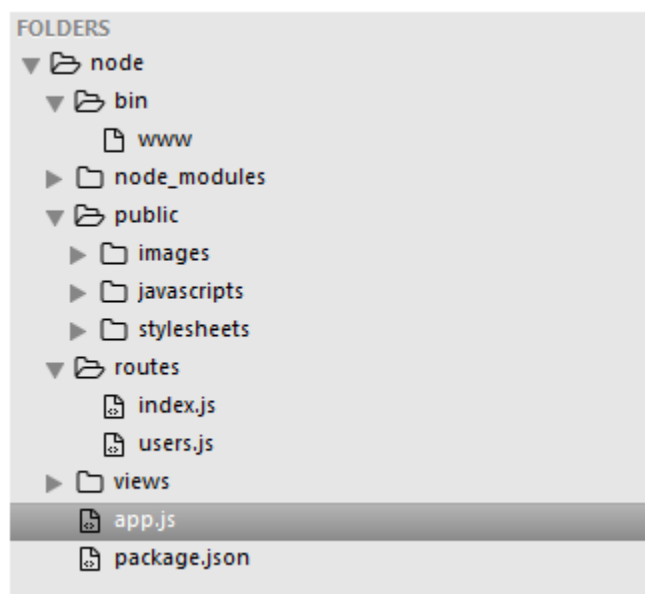
Sử dụng câu lệnh sau để cài đặt gói express-generator:

\$ npm install express-generator -g

Sau khi cài xong bạn cần cd đến folder muốn tạo project và khởi tạo project bằng lệnh: ***express***

Cài đặt những NodeJS module cần thiết cho project: ***npm install***

Cấu trúc thư mục sau khi tạo ra ứng dụng bao gồm một số tệp và thư mục quan trọng như:



- File app.js sẽ require các module để sử dụng cho cả ứng dụng và quan trọng nhất là require module ***express***. File app.js cũng tương tự như một file server.js, nó định nghĩa các đường dẫn tuyệt đối đến thư mục view và public cũng như xác định

view engine và các route đặc biệt.

- File package.json là file đầu tiên được tạo khi cấu trúc dự án. File này định nghĩa một json chứa thông tin của dự án như tên version và các dependencies

1.2 Thiết kế module và phương thức

Trong router, hệ thống có những module là: sinhvien.js, giangvien.js, doan.js, baocaotuan.js, manguoncuoi.js, thongbao.js, loaidoan.js, khoavien.js, trangthai.js, quantrivien.js.

Trong đó, hầu hết các module đều thực hiện các chức năng thêm, sửa, xóa, tìm kiếm.

Một số module đặc biệt như:

- index.js: định nghĩa các đường dẫn.

- home.js: là controller của trang chủ

Các module được thiết kế với các chức năng khác nhau cho từng đối tượng: sinh viên, giảng viên, quản trị viên.

1.3 Thiết kế tầng quản lý cơ sở dữ liệu

Tầng model là tầng kết nối và thao tác cơ sở dữ liệu, ở đây cụ thể là thao tác với api của postgresSQL.

Node-postgres (N-P) là tập hợp các mô-đun node.js để giao tiếp với cơ sở dữ liệu PostgreSQ. N-P có hỗ trợ callback, promise, async / await, connection pooling, truy vấn, con trỏ, stream result, liên kết C / C ++, richtext parsing và v.v

Trong thư mục /model/ định nghĩa file index.js để tạo pool connect đến CSDL:

```
const { Pool, Client } = require('pg');  
  
const pool = new Pool({  
  user: 'postgres',
```

```
host: 'localhost',  
database: 'pm',  
password: 'dmt',  
port: 2197,  
});
```

```
module.exports = pool
```

Các module khác required module này và thực hiện truy vấn:

```
const { Client } = require('pg')  
const client = new Client()  
  
await client.connect()  
  
const res = await client.query('SELECT $1::text as message', ['Hello world!'])  
console.log(res.rows[0].message) // Hello world!  
await client.end()
```

1.4 Thiết kế giao diện người dùng

- Responsive với bootstrap 3, chạy được trên tất cả kích cỡ mang hình
- Sử dụng view engine EJS – Effective Javascript template
 - Tính năng:
 - Biên dịch html nhanh nhờ sử dụng caching
 - Template tags đơn giản: <% %>
 - Custom delimiters (e.g., use <? ?> instead of <% %>)
 - Hỗ trợ includes
 - Hỗ trợ cả phía server side và client side
 - Static caching mã JS tĩnh
 - Phổ biến với [Express](#) view

CHƯƠNG IV: THỰC HIỆN GIẢI QUYẾT CÁC BÀI TOÁN

1. Đặt vấn đề

- Bài toán tìm kiếm
- Bài toán sử dụng đồng bộ vs bất đồng bộ
- Bài toán authentication
- Render ảnh

2. Giải quyết từng vấn đề

2.1. Bài toán tìm kiếm

2.1.1. Basic text matching

- `To_tsvector()`: FTS dùng hàm này để phân tích Document(đơn vị tài liệu tìm kiếm ví dụ cột `ten_de_tai`) thành các tokens, rồi từ tokens thành lexemes. Tokens là các từ trong Document, còn lexemes là tokens được chuẩn hóa, trong Tiếng anh là bỏ hậu tố. Ví dụ: token: cats -> lexemes: cat
- `To_tsquery()`: hàm nhận vào các từ cần tìm kiếm. Dùng `plainto_tsquery(cụm từ)` để search theo cụm từ người dùng nhập vào. Ví dụ `plainto_tsquery(Tiến Huyền Hương)` sẽ tìm Document chứa Tiến & Huyền & Hương.

- Cú pháp: `to_tsvector(...) @@ plainto_tsquery(...)`

2.1.2. Xếp hạng

- Dùng `ts_rank_cd()` để xếp hạng dựa theo số lần xuất hiện cụm từ tìm kiếm trong Document. Càng nhiều lần xuất hiện thì hạng càng cao(không kể đến khoảng cách giữa các cụm từ xa hay gần)
- Cú pháp `ts_rank_cd(vector tsvector, query tsquery);`
Cụ thể chúng em xếp hạng theo thứ tự xếp hạng của tần xuất xuất hiện cụm từ tìm kiếm giảm dần:
... ORDER BY `ts_rank_cd(to_tsvector(ten_de_tai) , query)`
DESC

2.1.3. Highlighting Result (Đánh dấu cụm từ tìm kiếm ở kết quả trả về)

- Sử dụng `ts_headline([config], vector tsvector, query tsquery, [option])`.
- Cụ thể, để highlight cụm từ tìm kiếm searchterm trong cột `ten_de_tai`, option `maxword=1023`, `minwords=50` là hiển thị `ten_de_tai` với ít nhiều nhất 1023 kí tự, ít nhất 50 kí tự, chúng em đã dùng:
`ts_headlind(ts_headline(ten_de_tai, plainto_tsquery("'" + searchterm + "'"), 'maxwords=1023, minwords=50')`

2.1.4. Câu truy vấn sql

- Kết nối 3 bảng sinh viên, giảng viên, đồ án. “Select” từ đó “`ts_headlind(...) AS ten_de_tai`”
- Cuối cùng, “order” theo “Rank”

2.1.5. Để tăng tốc độ tìm kiếm

- Lập chỉ mục cho cột “`ten_de_tai`”
- Cú pháp:

```
Create INDEX doan_idx ON doan USING GIN  
(to_tsvector(' ten_de_tai '));
```

Trong đó doan_idx chỉ là tên gọi của index.

2.1.6. Form để người dùng nhập cụm từ tìm kiếm (trong file top_content.ejs)

- Action: /search
- Gán cho input người dùng nhập vào name là searchterm

2.1.7. Thêm route /search vào home.js

- Viết câu lệnh SQL với biến searchterm được truyền vào

2.1.8. Thiết kế giao diện kết quả tìm kiếm (file searchresult.ejs)

2.2. Bài toán sử dụng đồng bộ và bất đồng bộ

- Trong nodejs chủ yếu các câu lệnh được xử lý theo quy tắc bất đồng bộ (async - câu lệnh sau được thực hiện không cần câu lệnh trước được thực hiện xong).
- Để xử lý các đoạn code bất đồng bộ thì trong phiên bản ES6, JavaScript đã được bổ xung thêm (.then()) Promise, tuy nhiên vẫn có nhiều nhược điểm.
- Đến phiên bản ES7, Async / Await đã được bổ xung để việc viết code bất đồng bộ trong JavaScript tốt hơn, code dễ nhìn hơn và dễ sử dụng hơn.
- **Async** - khai báo một hàm bất đồng bộ, còn **Await** - tạm dừng việc thực hiện các hàm async
- Chúng em sử dụng mô-đun express-promise-router, sử dụng await trước những câu lệnh (bất đồng bộ) để cho câu lệnh đó thực hiện xong mới chuyển sang câu lệnh khác.

2.3 Bài toán authentication:

Sử dụng module passport.js.

- Tính năng:
 - Middleware xác thực cho các request.
 - Đóng gói và cung cấp các hàm bảo mật.
 - Đăng nhập thông qua facebook, google,...
 - Đang được phát triển mạnh mẽ
- Khó khăn: Documentation ít và cần tìm hiểu.
- Cài đặt:
 - Require module passport và passport-local cũng như các module liên quan đến middleware & cookie-session.
 - Trong app.js khởi tạo các hàm serializeUser và deserializeUser các hàm này sẽ kiểm tra middleware tại mỗi router.
 - Trong app.js định nghĩa biến LocalStrategy dùng để kiểm tra đăng nhập của user có đúng và tạo cookie.
 - Các route được cần được kiểm tra middleware sẽ check session nhờ sử dụng hàm isAuthenticated()

2.4 Render ảnh:

NodeJS là single thread, nếu có một tác vụ (ở đây là đọc file ảnh) xử lý quá lâu, thì sẽ lock các tác vụ khác. Ngoài ra, việc đọc file theo cách bình thường gây tiêu tốn tài nguyên khá là lớn.

Khi render bất kỳ file nào bằng NodeJS cũng nên sử dụng stream vì nó sẽ giúp giảm thiểu bộ nhớ sử dụng của ứng dụng.

Tham khảo: [Node.js Streams: Everything you need to know](#).

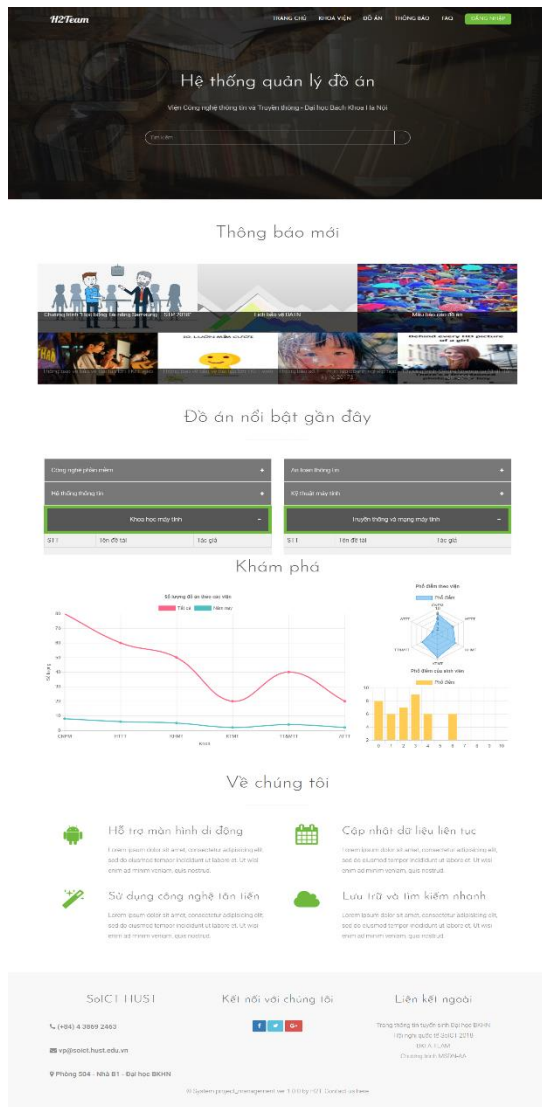
TỔNG KẾT


1. Kết quả đạt được

Sau thời gian làm việc 1 kỳ cố gắng, chúng em đã hoàn thành được 95% các mục tiêu đặt ra khi phân tích thiết kế hệ thống.

Github: https://github.com/damminhtien/project_management

Tổng cộng hơn 120 commits, với 2 version phát triển trên 2 nhánh.





Dam Minh Tien

"Thích du lịch, đọc sách, nghe nhạc và khám phá thế giới"


THÔNG TIN CÁ NHÂN

H2Team
THÔNG TIN ĐỒ ÁN
ĐỒ ÁN CỦA TÔI
ĐĂNG XUẤT

Lớp	CNTT03	Kỳ	Tên học phần	Tên đề tài	Điểm	Giảng viên HD	Xem chi tiết
♥ Khóa	60	20172	Đồ án 1	Lập trình kịch bản với JavaScript	10	Đỗ Bá Lâm	👁
✉ Email	damminhtienchi@gmail.com	20171	Đồ án 2	Lập trình web với Nodejs	10	Hoàng Anh Việt	👁
THAO TÁC HỆ THỐNG		20172	Đồ án 3	Student rollout App with Deep Learning	10	Trịnh Thành Trung	👁
✓ Nguyên vọng đồ án		20172	Đồ án tốt nghiệp			Hoàng Anh Việt	👁
📄 Đồ án tham khảo							
⚙ Cập nhật thông tin cá nhân							
⚙ Đổi mật khẩu							


Login Form

©2018 All Rights Reserved. H2Team - we are awesome.
Privacy and Terms



Hoàng Anh Việt
ThS - Bộ môn Công nghệ phần mềm

THÔNG TIN CÁ NHÂN		H2Team DANH SÁCH SINH VIÊN CỦA TÔI DANH SÁCH ĐỀ TÀI CỦA TÔI ĐĂNG XUẤT					
Email	hav@gmail.com	Kỳ	Tên học phần	Tên đề tài	Điểm	Sinh viên thực hiện	Xem chi tiết
Số điện thoại	0934255255	20171	Đồ án 2	Lập trình web với Nodejs	10	Dam Minh Tien	
THAO TÁC HỆ THỐNG		20172	Đồ án 3	Scrum - Agile	10	Nguyễn Thị Huyền	
		20172	Đồ án 3	Tensorflow	10	Vương Ngọc Quỳnh Hương	
		20172	Đồ án tốt nghiệp			Dam Minh Tien	
Cập nhật thông tin cá nhân	2012	Đồ án tốt nghiệp	Blockchain trong mô hình liên ngân hàng	10	Vương Ngọc Quỳnh Hương		
Đổi mật khẩu							



H2T team

Chào mừng,
H2T

TỔNG QUAN

- Trang chủ
- Quản lý người dùng
 - Sinh viên
 - Giảng viên
- Quản lý dữ liệu
- Thống kê
- Mạng xã hội
- KHÁC
 - Feed back
 - Hướng dẫn sử dụng
- Extras
 - Coming Soon

WE ARE FROM HUST

Sinh viên

Show 10 entries

Copy CSV Print

Search:

Mã sinh viên	Tên Sinh viên	Khoa	Lớp	Email
20156593	Vương Ngọc Quỳnh Hương	60	CNCNTT01	huongmeo@gmail.com
20156594	Nguyễn Thị Huyền	60	CNCNTT03	huyentra97@gmail.com
20156599	Dam Minh Tien	60	CNTT03	damminhtienchl@gmail.com

Showing 1 to 3 of 3 entries

Previous 1 Next

2. Hướng phát triển

Trong thời gian tiếp theo, chúng em sẽ tiếp tục hoàn thiện và phát triển các version mới và tối ưu code phía server.

3. Kiến nghị và lời cảm ơn

Để hoàn thành được đồ án này, chúng em đã cùng nhau cố gắng làm việc tích cực và không ngừng cải thiện cũng như phát triển hệ thống được tối ưu nhất. Chúng em đã đặt ra các mục tiêu hàng tuần phải đạt được để có thể đi đến kết quả cuối cùng đúng tiến độ và yêu cầu của môn học. Chúng em rất mong nhận được những nhận xét, đánh giá và góp ý để chúng em có thể tích lũy thêm kinh nghiệm cho bản thân.

Chúng em xin chân thành cảm ơn thầy **Hoàng Anh Việt** – Thạc sĩ bộ môn Công nghệ phần mềm, đã hướng dẫn chúng em hoàn thành bài tập lớn này.

Tài liệu tham khảo

- NodeJS v9.8: <https://nodejs.org/>
- Express: <https://expressjs.com/>
- Express promise router: <https://github.com/express-promise-router/express-promise-router>
- FE checklist: <https://github.com/damminhtien/Front-End-Checklist>
- PG: <https://node-postgres.com/>
- PostgreSQL Database: <https://www.postgresql.org/docs/manuals/>
- jQuery: <https://jquery.com/>
- Bootstrap 4: <http://getbootstrap.com/docs/4.0/>
- Multer: <https://github.com/expressjs/multer>
- Passport JS: <http://www.passportjs.org/>
- EJS: <http://ejs.co/> (Effective JavaScript templating)
- Body-parser: <https://www.npmjs.com/package/body-parser>

Chú thích