TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN ĐÀO TẠO QUỐC TẾ

──────── \* ───────

ĐỒ ÁN

**TỐT NGHIỆP ĐẠI HỌC**

NGÀNH THIẾT KẾ VÀ QUẢN TRỊ HỆ THỐNG THÔNG TIN (INPG)

**HỆ THỐNG ĐĂNG KÝ VÀ QUẢN LÝ THỰC TẬP TẠI DOANH NGHIỆP**

Sinh viên thực hiện: **Nguyễn Công Sơn**

Lớp INPG12 – K58

Giáo viên hướng dẫn: **ThS.Nguyễn Hồng Phương**

HÀ NỘI 06-2018

**MỤC LỤC**

[DANH MỤC HÌNH ẢNH 4](#_Toc514862508)

[DANH MỤC BẢNG BIỂU 5](#_Toc514862509)

[DANH MỤC THUẬT NGỮ, TỪ VIẾT TẮT 6](#_Toc514862510)

[LỜI CẢM ƠN 7](#_Toc514862511)

[CHƯƠNG 1: KHẢO SÁT, PHÂN TÍCH VÀ ĐẶC TẢ YÊU CẦU 8](#_Toc514862512)

[1.1. Khảo sát, đặt vấn đề 9](#_Toc514862513)

[1.2. Mô tả nghiệp vụ 10](#_Toc514862514)

[1.3. Mô tả hệ thống 11](#_Toc514862515)

[CHƯƠNG 2: PHÂN TÍCH HỆ THỐNG 14](#_Toc514862516)

[2.1. Biểu đồ Use-Case 14](#_Toc514862517)

[2.1.1. Biểu đồ use-case tổng quan 14](#_Toc514862518)

[2.1.2. Biểu đồ use-case sinh viên 15](#_Toc514862519)

[2.1.3. Biểu đồ use-case giảng viên hướng dẫn 21](#_Toc514862520)

[2.1.4. Biểu đồ use-case người hướng dẫn 27](#_Toc514862521)

[2.1.5. Biểu đồ use-case đại diện công ty 30](#_Toc514862522)

[2.1.6. Biều đồ use-case admin 35](#_Toc514862523)

[2.1.7. Biểu đồ use-case hệ thống 42](#_Toc514862524)

[2.2. Biểu đồ Hoạt động 42](#_Toc514862525)

[2.2.1. Biểu đồ hoạt động chung cho các tác nhân 42](#_Toc514862526)

[2.2.2. Biểu đồ hoạt động sinh viên 45](#_Toc514862527)

[2.2.3. Biểu đồ hoạt động giảng viên hướng dẫn 50](#_Toc514862528)

[2.2.4. Biểu đồ hoạt động người hướng dẫn 58](#_Toc514862529)

[2.2.5. Biểu đồ hoạt động đại diện công ty 58](#_Toc514862530)

[2.2.6. Biểu đồ hoạt động admin 64](#_Toc514862531)

[2.2.7. Biểu đồ hoạt động hệ thống 68](#_Toc514862532)

[2.3. Các biểu đồ tuần tự 69](#_Toc514862533)

[2.3.1. Biểu đồ tuần tự cho các chức năng của sinh viên 69](#_Toc514862534)

[2.3.2. Biểu đồ tuần tự giảng viên hướng dẫn 74](#_Toc514862535)

[2.3.3. Nhóm biểu đồ tuần tự đại diện công ty 81](#_Toc514862536)

[2.3.4. Nhóm biểu đồ tuần tự người hướng dẫn 87](#_Toc514862537)

[2.3.5. Nhóm biểu đồ tuần tự admin 87](#_Toc514862538)

[2.4. Biểu đồ Class 88](#_Toc514862539)

[2.4.1. Biểu đồ lớp tổng thể 88](#_Toc514862540)

[2.4.2. Biểu đồ lớp chi tiết 88](#_Toc514862541)

[CHƯƠNG 3: THIẾT KẾ HỆ THỐNG 88](#_Toc514862542)

[3.1. Thiết kế tổng thể 88](#_Toc514862543)

[3.2. Thiết kế chi tiết 89](#_Toc514862544)

[3.2.1. Thiết kế giao diện 89](#_Toc514862545)

[3.2.2. Thiết kế logic (thực thể) 89](#_Toc514862546)

[3.2.3. Thiết kế cơ sở dữ liệu 90](#_Toc514862547)

[3.2.4. Thuật toán so khớp 104](#_Toc514862548)

[CHƯƠNG 4: CÀI ĐẶT THỰC NGHIỆM 105](#_Toc514862549)

[4.1. Công cụ lập trình và và các Framework 105](#_Toc514862550)

[4.1.1. Công cụ lập trình 105](#_Toc514862551)

[4.1.2. Các ngôn ngữ lập trình sử dụng 105](#_Toc514862552)

[4.1.3. Strust 2 Framework 105](#_Toc514862553)

[4.1.4. Hibernate Framework 109](#_Toc514862554)

[4.2. Cấu trúc hệ thống 116](#_Toc514862555)

[4.3. Triển khai trên Tomcat server 8 116](#_Toc514862556)

[4.4. Kết quả 116](#_Toc514862557)

[4.4.1. Kết quả chương trình dành chung cho các đối tượng 116](#_Toc514862558)

[4.4.2. Kết quả chương trình cho người dùng là sinh viên 118](#_Toc514862559)

[4.4.3. Kết quả chương trình cho người dùng là giảng viên hướng dẫn 118](#_Toc514862560)

[4.4.4. Kết quả chương trình cho người dùng là đại diện công ty 118](#_Toc514862561)

[4.4.5. Kết quả chương trình cho người dùng là người hướng dẫn 118](#_Toc514862562)

[4.4.6. Kết quả chương trình cho người dùng là admin 118](#_Toc514862563)

[KẾT LUẬN 119](#_Toc514862564)

[5.1. Các kết quả đạt được 119](#_Toc514862565)

[5.2. Những tồn tại trong quá trình phát triển 119](#_Toc514862566)

[5.3. Hướng phát triển 119](#_Toc514862567)

[TÀI LIỆU THAM KHẢO 120](#_Toc514862568)

# DANH MỤC HÌNH ẢNH

[Hình 1: Biểu đồ use-case tổng quan **Error! Bookmark not defined.**](file:///C:\Users\sonnc\Documents\NetBeansProjects\QT5315\BÁO%20CÁO%20ĐỒ%20ÁN%20TỐT%20NGHIỆP.docx#_Toc513016758)

[Hình 2: Biều đồ use-case dành cho sinh viên **Error! Bookmark not defined.**](#_Toc513016759)

[Hình 3: Biểu đồ use-case Giảng viên hướng dẫn **Error! Bookmark not defined.**](file:///C:\Users\sonnc\Documents\NetBeansProjects\QT5315\BÁO%20CÁO%20ĐỒ%20ÁN%20TỐT%20NGHIỆP.docx#_Toc513016760)

# DANH MỤC BẢNG BIỂU

# DANH MỤC THUẬT NGỮ, TỪ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **STT** | **Từ viết tắt** | **Giải thích** |
| 1 | HTDKQLTT | Hệ thống đăng ký và quản lý thực tập DN |
| 2 | DN | Doanh nghiệp |
| 3 | CSTT | Cơ sở thực tập |
| 4 |  |  |
| 5 |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  | CSDL | Cơ sở dữ liệu |
|  | SF2 | Strust 2 Framework |
|  | HF | Hibernate Framework |
|  |  |  |
|  |  |  |
|  |  |  |

# LỜI CẢM ƠN

Lời đầu tiên, em xin được gửi tới các thầy cô trong Ban giám hiệu, Viện công nghệ thông tin và truyền thông, Viện đào tạo Quốc tế, trường Đại học Bách Khoa Hà Nội đã tạo điều kiện tốt nhất cho em trong quá trình học tập và xây dựng đồ án tốt nghiệp.

Em xin chân thành cảm ơn thầy Nguyễn Hồng Phương, bộ môn Hệ thống thông tin, Viện công nghệ thông tin và truyền thông, người đã tận tình chỉ bảo, hướng dẫn em trong quá trình thực hiện đồ án này.

Em xin gửi lời cảm ơn sâu sắc tới bố mẹ, người thân, bạn bè đã động ủng hộ, giúp đỡ và động viên em trong những lúc khó khăn, trong quá trình học tập cũng như trong quá trình thực hiện đồ án tốt nghiệp.

Em đã cố gắng để hoàn thành đồ án tốt nghiệp “Hệ thống đăng ký và quản lý thực tập trực tuyến”, nhưng có thể không tránh khỏi những sai sót hoặc hệ thống chưa được hoàn thiện, tối ưu. Em hy vọng rằng, những đóng góp của thầy cô, các bạn sẽ là động lực để em có thể tiến bộ hơn.

Em xin chân thành cảm ơn.

# CHƯƠNG 1: KHẢO SÁT YÊU CẦU

* 1. Khảo sát, đặt vấn đề

Sau quá trình tiếp thu và học tập tại trường, sinh viên cần phải có trải nghiệm làm việc thực tế trước khi ra trường để có thêm kinh nghiệm cho bản thân và thấy được mình đã đạt được gì, cần bổ xung những gì và tiếp thu những gì khi thực tập tại DN.

Việc thực tập tại DN và học phần thực tập kỹ thuật là một phần trọng trong quá trình đào tạo sinh viên. Sau quá trình thực tập, sinh viên không chỉ tiếp thu được những kiến thức cơ bản, các kiến thức thực thế mà còn được bổ xung thêm kỹ năng mềm. Kỹ năng mềm là một phần rất quan trọng trong quá trình phát triển bản thân. Từ đó, sinh viên sẽ chủ động áp dụng các kiến thức đã được học và kỹ năng vào môi trường làm việc thực tế tại DN, biết cách làm việc nhóm, giao tiếp và trách nhiệm với công việc.

Mục đích của thực tập tại DN: Thực tập tại DN được coi như là một học phần, cho phép sinh viên trải nghiệm môi trường làm việc thực tế, áp dụng các kiến thức đã học được tại trường vào các dự án của DN và đặc biệt là phát triển kỹ năng mềm, phong cách làm việc cá nhân và tập thể trong các DN hoạt động trong lĩnh vực Công nghệ Thông tin và Truyền thông hoặc các DN có ứng dụng Công nghệ Thông tin và Truyền thông trong hoạt động chuyên môn nghiệp vụ.

Đối với sinh viên: áp dụng đối với sinh viên học các chương trình đào tạo Kỹ sư các chuyên ngành Công nghệ Thông tin, Kỹ thuật phần mềm, Thiết kế và quản trị hệ thống thông tin và yêu cầu phải tìm được một đơn vị thực tập có uy tín, đảm bảo được chất lượng và mức độ đầu ra, đạt được những gì sau khi thực tập. Một mặt, sinh viên phải tìm được đề tài sao cho đúng với chuyên ngành của mình ở mức độ cao nhất, tránh tình trạng thực tập không đúng chuyên môn được đào tạo. Mặt khác, sinh viên sẽ phải báo cáo lại cho trường học về việc thực tập tại DN, có đúng chuyên ngành, việc đào tạo tại trường có đáp ứng được yêu cầu của DN hay không?

Về phía nhà trường: yêu cầu phải tổ chức sao cho việc quản lý sinh viên thực tập tại doanh ngiệp phải được thống nhất. Các giảng viên có thể biết được tình hình thực tập của sinh viên. Giảng viên cũng cần phải trao đổi với đại diện DN thông qua một kênh truyền thông nào đó để xác nhận đơn vị thực tập này có đáp ứng đủ tiêu chuẩn để sinh viên thực tập hay không? Một mặt khác là giảng viên cần quản lý các đề tài, báo cáo của sinh viên, đơn vị thực tập, từ đó thông kê báo cáo lên nhà trường về tình trạng đào tạo có đáp ứng được DN hay không?

Về phía DN, yêu cầu của DN là tìm kiếm được các sinh viên có nhu cầu thực để học hỏi, tiếp thu kinh nghiệm và tham gia vào các dự án thực tế? Qua quá trình thực tập sẽ là một lực lượng sinh viên có khả năng đáp ứng được công việc thực tế và từ đó có thể tìm kiếm nguồn nhân lực cho DN của mình hay không?

* 1. Mô tả nghiệp vụ

Nghiệp vụ 1: Hệ thống khởi tạo. Sau khi phát triển hoàn thành và đáp ứng được nghiệp vụ ứng dụng thì tiến thành triển khai lên mạng.

Nghiệp vụ 2: Đăng ký công ty. Mỗi công ty muốn thực hiện nghiệp vụ đăng đề tài thì cần phải đăng ký tài khoản, thông tin và tham gia chương trình thực tập tại DN. Đại diện công ty có trách nhiệm đăng đề tài và đăng ký người hướng dẫn cho mỗi đề tài.

Nghiệp vụ 3: Sinh viên đăng ký thực tập. Sinh viên phải đăng ký thực tập trên hệ thống. Mọi trường hợp sinh viên đăng ký thực tập ngoài sẽ không được tính vào học phần và bắt buộc phải học lại học phần thực tập kỹ thuật.

* Sinh viên phải kê khai đầy đủ thông tin cá nhân trong hệ thống và chịu trách nhiệm trước những bản khai đăng ký của mình.
* Nếu công ty tiếp nhận sinh viên, sinh viên phải có trách nhiệm yêu cầu đơn vị thực tập xác nhân (ký và đánh dấu).
* Nộp các biểu mẫu cho quản ngành.

Nghiệp vụ 4: Quản ngành tổng hợp. Quản ngành có trách nhiệm tổng hợp các biều mẫu và gửi lại cho giảng viên hướng dẫn thực tập.

Nghiệp vụ 5: Giảng viên hướng dẫn duyệt thông tin. Trên cơ sở thông tin do CSTT cung cấp, giảng viên hướng dẫn tiến hành duyệt thông tin công ty, đề tài và xác nhận cho phép hoạt động trên hệ thống. Mặt khác, giảng viên hướng dẫn cũng cần phải duyệt đề tài sinh viên và xác nhận cho phép sinh viên thực tập tại CSTT đã đăng ký.

Nghiệp vụ 6: Giảng viên hướng dẫn tiến hành liên lạc, thống nhất thời gian gặp mặt sinh viên, trao đổi và các yêu cầu trong quá trình thực tập.

Nghiệp vụ 7: Giảng viên hướng dẫn đưa nhóm sinh viên được phân công phụ trách tới CSTT. Trong buổi gặp mặt lần đầu này, giảng viên, sinh viên và công ty sẽ trao đổi và thống nhất kế hoạch thực tập cụ thể: lịch thực tập, nội dung công việc và kết quả cần đạt, phương thức đánh giá, quy định làm việc

Nghiệp vụ 8: Sinh viên yêu cầu CSTT xác nhận (ký, đóng dấu) vào Phiếu giao nhiệm vụ cho sinh viên. Nộp lại biểu mẫu trên (bản cứng) cho cán bộ quản ngành theo lớp. Tiến hành thực tập tại CSTT dưới sự giám sát của người đã được CSTT chỉ định phụ trách việc thực tập. Tham gia đầy đủ các buổi thực tập theo kế hoạch của CSTT. Thực hiện các công việc theo đúng kế hoạch và theo quy định làm việc của CSTT. Thông báo cho điều phối viên nếu có các khúc mắc với CSTT

Nghiệp vụ 9: Giảng viên hướng dẫn liên lạc thường xuyên với sinh viên và CSTT để nắm được tình hình, xử lý các vấn đề phát sinh.

Nghiệp vụ 10: Người hướng dẫn liên lạc thường xuyên với giảng viên hướng dẫn để nắm được tình hình, xử lý các vấn đề phát sinh.

Nghiệp vụ 11: DN tổ chức cho sinh viên thực tập theo đúng kế hoạch đã thống nhất với giảng viên và sinh viên. Thực hiện chấm công cho sinh viên thực tập. Định kỳ 2 tuần 1 lần, CSTT thông báo cho giảng viên danh sách sinh viên không đủ điều kiện tiếp tục thực tập. Sinh viên bị xếp loại không đủ điều kiện tiếp tục thực tập tại cơ sở khi:

* Vắng 1/3 thời gian thực tập tính đến thời điểm báo cáo định kỳ của CSTT
* Không có ý thức khi đi thực tập
* Không hoàn thành kế hoạch đã cam kết giữa sinh viên, giảng viên phụ trách và CSTT.

Nghiệp vụ 12: Sinh viên viết báo cáo quá trình thực tập cũng như kết quả thu được. Báo cáo phải có ý kiến xác nhận của CSTT.

* Lưu ý: nếu sinh viên bị xếp loại không đủ điều kiện tiếp tục thực tập tại CSTT thì sẽ phải học lại học phần thực tập kỹ thuật vào đợt sau
* Sinh viên phải nộp báo cáo bản cứng, đối với bản mềm thì nộp trên hệ thống.

Nghiệp vụ 13: CSTT đánh giá sinh viên tham gia thực tập và nộp trên hệ thống đồng thời phản hồi cho giảng viên hướng dẫn một cách trung thực nhất về quá trình thực tập của sinh viên và những tồn tại trong quá trình thực tập

Nghiệp vụ 14: Giảng viên hướng dẫn căn cứ vào bảng đánh giá của công ty để cho điểm quá trình học phần. Căn cứ vào báo cáo của sinh viên để cho điểm thi kết thúc học phần. Viết báo cáo kết quả thực tập chung của đoàn và tổng hợp đăng điểm trên hệ thống.

* 1. Mô tả hệ thống

HTDKQLTT là một hệ thống cho phép hai bên là nhà trường và DN (đơn vị thực tập) liên kết với nhau để quản lý sinh viên. Đối với nhà trường, hệ thống sẽ quản lý sinh viên thực tập tại DN như một môn học, từ đó đưa ra các yêu cầu và đánh giá sinh viên về mức độ thực tập tại DN của sinh viên. Đối với DN, đây có thể là một kênh trực truyến để tìm kiếm nguồn nhân lực tương lai hoặc sinh viên muốn đăng ký thực tập tại công ty.

HTDKQLTT cho phép sinh viên lựa chọn các đề tài phù hợp với trình độ và khả năng thực tế của bản thân bao gồm: kiến thức, chuyên môn, ngại ngữ, các kỹ năng mềm,…Đáp ứng nhu cầu thực tập của sinh viên với đề tài đa dạng, nhiều lĩnh vực trong cuộc sống

Cho phép của hệ thống đối với giảng viên hd

Cho phép đối với cong ty

Xác thực đánh giá

Quy trình của HTDKQLTT bắt đầu từ việc đăng ký thông tin cá nhân của sinh viên, giảng viên hướng dẫn, người hướng dẫn và đại diện công ty. Các đối tượng khi tham gia, đăng ký vào hệ thống phải có trách nhiệm trong quá trình sử dụng và xác thông tin phải chính xác.

* Đối với DN
  + Trước tiên, các DN có nhu cầu đăng ký tham gia vào chương trình thực tập tại doanh nghiệp trên hệ thống sẽ truy cập vào hệ thống và đăng ký tài khoản, thông tin DN. Sau khi đăng ký thành công sẽ chờ cho giảng viên hướng dẫn xác thực thông tin và duyệt công ty trên hệ thống.
  + DN sẽ đăng ký nhân viên, giảng viên hướng dẫn cho các đề tài, DN sẽ chịu trách nhiệm quản lý người hướng dẫn đề tài của DN mình.
  + DN đăng các đề tài lên HTDKQLTT, hệ thống có nhiệm vụ lưu trữ thông tin và kiểm tra tính hợp lệ các đối tượng cần lưu
  + Hệ thống phải phản hồi cho DN khi DN thực hiện các chức năng có thành công hay không.
* Đối với sinh viên:
  + Hệ thống có trách nhiệm xác thực tài khoản khi đăng ký đối với sinh viên. Kiểm tra tính hợp lệ của tài khoản sinh viên ở định dạng: mssv@student.hust.edu.vn.
  + Khi sinh viên đăng ký đề tài, hệ thống cần phải kiểm tra xem sinh viên đã cập nhật đầy đủ thông cá nhân chưa. Nếu sinh viên muốn đăng ký đề tài thì HTDKQLTT kiểm ta xem đã đăng ký kỳ thực tập chưa.
  + Đối với mỗi sinh viên sẽ được đăng ký tối đa 3 đề tài, 3 đề tài này phải khác nhau, không được đăng ký trùng đề tài
  + Sau khi kiểm tra đăng ký, hệ thống tiến hành so khớp thông tin và các về vấn đề trình độ, kỹ năng và chuyên môn về lập trình của sinh viên. Đây là cơ sở để quyết định sinh viên có được duyệt đề tài thành công hay không.
  + Hệ thông sẽ thông báo các thông tin mới, các yêu cầu trong quá trình thực tập, sinh viên sẽ theo dõi qua quy trình thực tập để biết thời gian, địa điểm và các hạn quy trình.
  + Sau khoảng thời gian thực tập, sinh viên sẽ phải nộp báo cáo giữa kỳ và cuối kỳ lên hệ thống. Hệ thống tiếp nhận và xử lý gửi cho giảng viên để chấm điểm thi.
  + Hệ thống sẽ cung cấp điểm thi cho sinh viên sau quá trình thực tập. Điểm thi sinh viên sẽ bao gồm các thành phần sau:
  + Trong đó:
    - DKT: Điểm kết thúc quá trình thực tập
    - DPH: Điểm phản hồi là số lượng phản hổi của DN với giảng viên hướng dẫn
    - DBCQT: Điểm báo cáo quá trình
    - DBCCK: Điểm báo cáo cuối kỳ
    - HS: hệ sô giữa các điểm
  + Nếu sinh viên có thắc măc gì, phải gửi mail thông báo trên hệ thống tới các đối tượng cần giải quyết.
* Đối với người hướng dẫn
  + Hệ thống sẽ cho phép người hướng dẫn quản lý được sinh viên đang thực tập tại công ty.
  + Người hướng dẫn có trách nhiệm upload các file báo cáo/ đánh giá sinh viên sau khi kết thúc quá trình thực tập
  + Người hướng dẫn cũng có trách nhiệm sử dụng mail của hệ thống để thông báo và phàn hồi tình trạng thực tập của sinh viên trong quá trình thwuc tập tại công ty
  + Theo định kỳ 2 tuần 1 lần, người hướng dẫn sẽ thay mặt đại diện công ty phản hồi về cho giảng viên hướng dẫn
* Đối với Giảng viên hướng dẫn
  + Hệ thống cho phép giảng viên hướng dẫn quản lý được sinh viên đã/ đang và sẽ thực tập sau khi sinh viên đã đăng ký.
  + Giảng viên hướng dẫn sẽ xem danh sách đề tài sinh viên đăng ký và tiến hành duyệt theo mức đồ phù hợp, tình trạng so khớp của thông tin của sinh viên và hiển thị theo nhóm đề tài của sinh viên
  + Nếu giảng viên không chấp nhận một đề tài của sinh viên thì hệ thống sẽ từ chối đề tài cho sinh viên đó. Nếu giảng viên chấp nhận đề tài đó cho sinh viên và sinh viên có đăng ký số lượng đề tài lớn hơn 3 thì hệ thống sẽ chấp nhận đề tài đang duyệt và từ chối các đề tài đăng ký còn lại cho sinh viên.
  + Hệ thống hiển thị ra danh sách công ty, giảng viên cần xác thực thông tin công ty và duyệt có đồng ý cho công ty có được đăng đề tài vào hệ thống hay không.
  + Hệ thống cũng hiển thị danh sách các đề tài của các công ty, giảng viên sẽ tiến hành duyệt theo một định kỳ nào đó.
  + Trong quá trình thực tập, giảng viên phải đăng các thông báo về quy trình thực tập. Sau mỗi đợt thực tập, giảng viên sẽ phải chấm điểm thi cho sinh viên.
* Hệ thống:
  + Hệ thống sẽ thực hiện kiểm tra thời hạn phản hồi của mỗi sinh viên và tiến hành gửi mail thông báo đến hạn cần phản hồi thông tin từ người hướng dẫn tời giảng viên hướng dẫn.
  + Hệ thống sẽ gửi một số Gmail, eSMS nếu có yêu cầu từ từ người dùng và mặc định sẽ gửi ở một số chức năng
* Admin:
  + Là người quản lý cấp cao nhất trong hệ thống
  + Admin sẽ quản lý hệ thống, người dùng, email, thực tập, công ty, đề tài,…

# CHƯƠNG 2: PHÂN TÍCH HỆ THỐNG

1. Biểu đồ Use-Case
   * 1. Biểu đồ use-case tổng quan

Biểu đồ use-case tổng quan là biểu đồ thể hiện một cách tổng thể các đối tượng, tác nhân và các chức năng có trong hệ thống. Biểu đồ này sẽ được phân rã thành các biểu đồ có mức chi tiết cao hơn theo các tác nhân tác động vào hệ thống. Từ biểu đồ Use-case tổng quan, chúng ta bao quát một cách tổng thể trong hệ thống có những gì, chức năng ra sao.

Các tác nhân và chức năng theo từng tác nhân của hệ thống bao gồm:

1. Sinh viên

Sinh là một trong những tác nhân quan trọng trong hệ thống.

1. Giảng viên hướng dẫn
2. Đại diện công ty
3. Người hướng dẫn (tại công ty)
4. Người quản trị (Admin)
   * 1. Biểu đồ use-case sinh viên
5. Use-case đăng nhập

Đặc tả use-case đăng nhập

* Mục đích: Đảm bảo xác thực thông tin người dùng và định danh người dùng trong hệ thống
* Tác nhân/ đối tượng sử dụng: tất cả các actor là con người
* Mổ tả chung: Người dùng muốn sử dụng được hệ thống thì bắt buộc phải đăng nhập vào hệ thống.
* Luồng sự kiện chính:
  + - Người dùng chọn đăng nhập trên hệ thống
    - Hệ thống yêu cầu người dùng nhập tài khoản và mật khẩu
    - Người dùng nhập xong thông tin đăng nhập và click nút đăng nhập.
    - Hệ thống kiểm tra thông tin đăng nhập. Nếu đăng nhập thành công thì chuyển hướng tới trang chủ. Ngược lại thì thông báo lỗi và yêu cầu nhập lại.
* Luồng thay thế: Sai quá số lần quy định thì yêu cầu cài đặt lại mật khẩu thông qua xác thực.
* Yêu cầu cụ thể: Tên đăng nhập và mật khẩu phải khớp với thông tin đã đăng ký trong hệ thống.
* Điều kiện trước: Người dùng đã đăng ký thành công thành viên trong hệ thống
* Điều kiện sau: trả về trang chủ

1. Use-case đăng xuất

Đặc tả use-case đăng xuất

* Mục đích: đảm bảo thoát toàn bộ kết nối của người dùng tới hệ thống.
* Tác nhân/ đối tượng sử dụng: Tât cả đối tượng là con người
* Mổ tả chung: người dùng sau khi sử dụng xong các chức năng trong hệ thống và tạm thời không có nhu cầu tiếp tục sử dụng hệ thống.
* Luồng sự kiện chính:
  + - Người dùng nhấn chức năng đăng xuất.
    - Hệ thống tiến hành đăng xuất người dùng, xóa các session ra khỏi hệ thống.
    - Hệ thống thông báo và trở về trang đăng nhập.
* Luồng thay thế: Không có
* Yêu cầu cụ thể: Người dùng đã đăng nhập vào hệ thống trước đó.
* Điều kiện trước: Người dùng đã đăng nhập vào hệ thống trước đó.
* Điều kiện sau: Trả về thông báo và trở về trang đăng nhập vào hệ thống

1. Use-case đăng ký thông tin cá nhân

Đặc tả use-case đăng ký thông tin cá nhân

* Mục đích: Đảm bảo thông danh tính, thông tin người dùng trong hệ thống.
* Tác nhân/ đối tượng sử dụng: Tất cà đối tượng là con người
* Mổ tả chung: Sau khi đăng ký tài khoản thành công, người dùng bắt buộc phải đăng ký thông tin cá nhân.
* Luồng sự kiện chính:
  + - Người dùng đăng nhập
    - Nếu thành công, hệ thống chuyển đến trang đăng ký thông tin cá nhân
    - Người dùng nhập thông tin cá nhân
    - Hệ thống bắt lỗi nhập liệu
    - Nếu không có lỗi nhập liệu, người dùng nhấn nút đăng ký thông tin
    - Hệ thống kiểm tra tính hợp lệ trong hệ cơ sở dữ liệu và trả về kết quả cho người dùng
* Luồng thay thế: Hệ thống yêu cầu nhập lại thông tin
* Yêu cầu cụ thể: Các thông tin đăng ký phải chính xác với từng đối tượng. Chưa từng đăng ký thông tin trên hệ thống trước đó.
* Điều kiện trước: Người dùng đã đăng ký thành công tài khoản trên hệ thống.
* Điều kiện sau: Thông báo cho người dùng nếu có lỗi hoặc đăng ký thành công trong hệ thống.

1. Use-case cập nhật thông tin cá nhân

Đặc tả use-case cập nhật thông tin cá nhân

* Mục đích: Thay đổi các thông tin cá nhân bị sai hoặc muốn thay đổi theo phù hợp với yêu cầu của người dùng.
* Tác nhân/ đối tượng sử dụng: tất cả các đối tượng là con người.
* Mổ tả chung: Người dùng muốn thay đổi thông tin cá nhân của mình. Một số thông tin sẽ không được thay đổi hay cập nhật (ví dụ như email).
* Luồng sự kiện chính:
  + - Người dùng đăng nhập
    - Hệ thống trả về giao diện trang chủ
    - Người dùng nhấn vào nút cập nhật thông tin cá nhân
    - Hệ thống trả về giao diện cập nhật thông tin
    - Người dùng sẽ cập nhật các thông tin được phép thay đổi thông tin
    - Hệ thống bắt lỗi nhập liệu trên giao diện
    - Người dùng nhấn nút cập nhật
    - Hệ thống kiểm tra tính hợp lệ dữ liệu và lưu vào CSDL
    - Hệ thông báo cho người dùng về việc cập nhật thông tin có thành công hay không.
* Luồng thay thế: Người dùng nhập sai thì hệ thống yêu cầu nhập lại
* Yêu cầu cụ thể: Thông tin cá nhân của người dùng đã được lưu trước đó và chỉ được phép thay đổi một số thông tin cho chép của hệ thống.
* Điều kiện trước: Người dùng đã đăng ký thông tin cá nhân trên hệ thống
* Điều kiện sau: Thông báo cho người dùng đã cập nhật thông tin cá nhân thành công hay không.

1. Use-case xem đề tài

Đặc tả use-case xem đề tài

* Mục đích: Sinh viên xem các thông tin đề tài, thông tin công ty, chuyên môn, lĩnh vực hoạt động hoặc các đề tài mà sinh viên quan tâm.
* Tác nhân/ đối tượng sử dụng: sinh viên
* Mổ tả chung: Sinh viên muốn xem thông tin đề tài, xem chi tiết đề tài trước khi đăng ký đề tài thực tập.
* Luồng sự kiện chính:
  + - Sinh viên nhấn nút đề tài
    - Hệ thống truy vấn dữ liệu
    - Hệ thống trả về dữ liệu cho người dùng
* Luồng thay thế: Không có
* Yêu cầu cụ thể:
* Điều kiện trước: Dữ liệu, thông tin về đề tài phải được duyệt trước đó từ giảng viên hướng dẫn
* Điều kiện sau: hiển thị toàn bộ danh sách đề tài.

1. Use-case đăng ký đề tài

Đặc tả use-case đăng ký đề tài

* Mục đích: Sinh viên muốn đăng ký các đề tài thực tập cho mình
* Tác nhân/ đối tượng sử dụng: sinh viên
* Mô tả chung: Sau khi đăng ký thành công đợt thực tập, sinh viên cần đăng ký tối đa 3 đề tài để thực tập.
* Luồng sự kiện chính:
  + - Sinh viên nhấn nút đề tài
    - Hệ thống truy vấn dữ liệu
    - Hệ thống trả về dữ liệu cho người dùng
    - Sinh viên nhấn nút đăng ký đề tài
    - Hệ thống thông báo cảnh báo thực hiện hành động
    - Hệ thống kiểm tra xem sinh viên đã đăng ký đợt thực tập chưa
    - Nếu sinh viên chưa đăng ký đợt thực tập, hệ thống thông báo và chuyển sang trang đăng ký đợt thực tập. Nếu sinh viên đã đăng ký đợt thực tập, hệ thống lưu đề tài đăng ký cho sinh viên và trả về thông báo đăng ký cho sinh viên
    - Sinh viên tiếp tục đăng ký đề tài
    - Hệ thống thông báo cảnh báo thực hiện hành động
    - Hệ thống kiểm tra sinh viên đăng ký có bị trùng đề tài hay quá giới hạn 3 đề tài hay không.
    - Hệ thống thông báo cho sinh viên về tình trạng đăng ký
* Luồng thay thế:
  + - Nếu sinh viên chưa đăng ký đợt thực tập, hệ thống chuyển đến trang đăng ký đợt thực tập
    - Hủy bỏ việc đăng ký đề tài đã chọn nếu sinh viên đăng ký trùng đề tài hoặc quá giới hạn đề tài
* Yêu cầu cụ thể: Sinh viên phải đăng ký tối đa 3 đề tài và không được trùng nhau.
* Điều kiện trước: sinh viên đang trong trạng thái hoạt động, đề tài đăng ký không quá 3 đề tài, đã đăng ký thành công đợt thực tập
* Điều kiện sau: Thông báo cho sinh viên về tình trạng đăng ký đề tài, các trường hợp quá 3 đề tài, chưa đăng ký đợt thực tập hoặc trùng đề tài.

1. Use-case xem điểm thi

Đặc tả use-case đăng xem điểm thi

* Mục đích: hiển thị thông tin điểm thi của sinh viên sau khi kết thúc đợt thực tập tại DN
* Tác nhân/ đối tượng sử dụng: Sinh viên
* Mô tả chung: Sau khi kết thúc đợt thực tập, sinh viên muốn xem điểm thi của mình.
* Luồng sự kiện chính:
  + - Sinh viên nhấn vào nút điểm thi
    - Hệ thống lấy toàn bộ thông tin điểm thi
    - Hệ thống lấy toàn bộ thông tin hệ số điểm
    - Hệ thống hiển thị điểm thi
* Luồng thay thế: Nếu không có điểm trong hệ thống, hệ thống thông báo không có điểm thi
* Yêu cầu cụ thể: Không có
* Điều kiện trước:
  + - Sinh viên đã hoàn thành đợt thực tập tại công ty
    - Sinh viên đã nộp đầy đủ báo cáo
    - Giảng viên hướng dẫn đã phê duyệt và chấm điểm
* Điều kiện sau: Hiển thị điểm thi cho sinh viên và hệ số điểm

1. Use-case xem lịch trình

Đặc tả use-case xem lịch trình

* Mục đích: Sinh viên muốn xem lịch trình thực tập trong đợt thực của mình có những yêu cầu gì.
* Tác nhân/ đối tượng sử dụng: Sinh viên
* Mô tả chung: Trong quá trình thực tập, sinh viên phải xem lịch trình thực tập, xem các thông báo của giảng viên.
* Luồng sự kiện chính:
  + - Sinh viên nhấn vào nút lịch trình
    - Hệ thống lấy toàn bộ lịch trình thực tập
    - Hệ thống chuyển đổi dữ liệu sang JSON
    - Hệ thống hiển thị lịch trình
    - Sinh viên nhấn vào mỗi lịch trình trên giao diện
    - Hệ thống hiển thị chi tiết lịch trình cho sinh viên
* Luồng thay thế: không có
* Yêu cầu cụ thể: không có
* Điều kiện trước: lịch trình thực tập đã được đăng bởi giảng viên hướng dẫn.
* Điều kiện sau: Hiển thị thông tin lịch trình

1. Use-case mail

Đặc tả use-case email

* Mục đích: trao đổi thông tin, các vướng mắc trong quá trình thực tập hoặc muốn phản hồi cho người dùng, hệ thống.
* Tác nhân/ đối tượng sử dụng: tất cả người dùng
* Mô tả chung: Khi người dùng muốn trao đổi thông tin cho nhau thì cần sử dụng chức năng email trên hệ thống.
* Luồng sự kiện chính:
  + - Người dùng nhấn vào nút email
    - Hệ thống lấy toàn bộ thông tin về email, bao gồm các email đã gửi, đã đọc, chưa đọc, đã nhận,…
    - Hệ thống hiển thị thông tin cho người dùng.
* Luồng thay thế:
  + - Gửi mail: trường hợp người dùng muốn gửi mail tới các người dùng khác trong hệ thống
    - Xem mail: người dùng muốn xem mail
    - Đọc mail: người dùng muốn đọc mail
    - Xóa mail: người dùng muốn xóa mail
* Yêu cầu cụ thể: không có
* Điều kiện trước: tài khoản đã được xác thực
* Điều kiện sau: hiển thị toàn bộ thông tin mail cho người dùng

1. Use-case xem danh sách công ty

Đặc tả use-case xem danh sách công ty

* Mục đích: người dùng muốn xem danh sách công ty
* Tác nhân/ đối tượng sử dụng: sinh viên/ người giảng viên hướng dẫn
* Mô tả chung: người dùng muốn xem thông tin về công ty, đại diện công ty, các đề tài của công ty
* Luồng sự kiện chính:
  + - Người dùng nhấn nút công ty
    - Hệ thống lấy danh dách và hiển thị thông tin công ty, người đại diện cho người dùng
* Luồng thay thế: người dùng muốn xem chi tiết về một công ty
* Yêu cầu cụ thể: không có
* Điều kiện trước: thông tin công ty, đại điện công ty đã được giảng viên hướng dẫn duyệt và chấp nhận
* Điều kiện sau: hiển thị danh sách công ty, hiển thị chi tiết công ty, đại diện công ty

1. Use-case upload/download/ xóa file

Đặc tả use-case upload/download file

* Mục đích: sinh viên muốn upload/download các file cần thiết lên hệ thống.
* Tác nhân/ đối tượng sử dụng: tất cả sinh viên
* Mô tả chung: khi sinh viên muốn đăng hoặc tải xuống các file cần thiết từ hệ thống.
* Luồng sự kiện chính:
  + - Sinh viên nhấn nút file hoặc nút tài liệu
    - Hệ thống lấy danh sách file và hiển thị cho sinh viên
    - Sinh viên nhấn nút tải file
    - Hệ thống truy cập file trong server và gửi về cho sinh viên
    - Sinh viên nhất nút lưu file
    - Nếu sinh viên muốn upload file: hệ thống hiển thị giao diện upload file
    - Sinh viên chọn file muốn upload và điền đầy đủ thông tin
    - Sinh viên lựa chọn đặc tính của file là báo cáo, tài liệu, …
    - Hệ thống kiểm tra. Nếu đáp ứng yêu cầu thì lưu file vào server
    - Hệ thống thông báo cho sinh viên
    - Nếu sinh viên muốn xóa file, chọn nút xóa file
    - Hệ thống kiểm tra, lấy mã file và tiến hành cập nhật dữ liệu
    - Hệ thống trả về cho sinh viên sau khi thực hiện hành động xóa.
* Luồng thay thế: hệ thống yêu cầu nhập lại, chọn lại file nếu quá dung lượng quy định của hệ thống.
* Yêu cầu cụ thể: Các file upload lên hệ thống phải có tính xác thực, sinh viên phải tự chịu trách nhiệm về hành động upload file của mình lên hệ thống.
* Điều kiện trước: sinh viên đã đăng nhập và chọn file muốn upload hoặc chọn file muốn tải/ xóa. Hiển thị thông báo cảnh báo hành động.
* Điều kiện sau: Hiện thị thông báo sau hành động
  + 1. Biểu đồ use-case giảng viên hướng dẫn
       1. Use-case đăng nhập

Mô tả được thể hiện giống như use-case đăng nhập trong phần sinh viên

* + - 1. Use-case đăng ký thông tin cá nhân
      2. Use-case cập nhật thông tin cá nhân
      3. Use-case xem danh sách công ty
      4. Use-case duyệt công ty

Đặc tả use-case duyệt công ty

* Mục đích: duyệt các công ty có các tiêu chuẩn đáp ứng đủ tiêu chí đặt ra trong quá trình thực tập của sinh viên
* Tác nhân/ đối tượng sử dụng: giảng viên hướng dẫn
* Mô tả chung: khi công ty muốn tham gia vào chương trình thực tập thì cần phải trải qua quá trình duyệt các tiêu chí. Mặt khác giảng viên sẽ loại bỏ các công ty không đủ yêu cầu.
* Luồng sự kiện chính:
  + - Giảng viên hướng dẫn nhấn nút danh sách công ty
    - Hệ thống lấy toàn bộ thông tin về công ty
    - Hệ thống hiển thị danh sách công ty
    - Giảng viên nhấn chấp nhận/ từ chối
    - Hệ thống đưa ra cảnh báo hành động
    - Nếu giảng viên đồng ý, hệ thống tiến hành cập nhật dữ liệu. Nếu giảng viên từ chối, hệ thống hủy bỏ hành động hiện tại
    - Hệ thống thông báo hành động thực hiện của giảng viên
* Luồng thay thế:
  + - Giảng viên muốn xem toàn bộ danh sách công ty
    - Giảng viên muốn xem danh sách công ty đã được duyệt
    - Giảng viên muốn xem danh sách công ty chưa duyệt
    - Giảng viên muốn xem chi tiết công ty
* Yêu cầu cụ thể: thông tin công ty đã được cập nhật đầy đủ
* Điều kiện trước: giảng viên phải đăng nhập vào hệ thống, công ty phải được cập nhật đầy đủ và đang chờ duyệt
* Điều kiện sau: hiển thị thông báo cho người dùng
  + - 1. Use-case xem danh sách đề tài công ty

Đặc tả use-case xem danh sách đề tài công ty

* Mục đích: xem danh sách đề tài đề duyệt đề tài phù hợp
* Tác nhân/ đối tượng sử dụng: giảng viên hướng dẫn
* Mô tả chung: Giảng viên hướng dẫn muốn hiển thị danh sách đề tào công ty, hiển thị toàn bộ danh sách đề tài, hiển thị đề tài đã duyệt hoặc hiển thị đề tài chưa duyệt.
* Luồng sự kiện chính:
  + - Giảng viên hướng dẫn nhấn vào nút đề tài công ty
    - Hệ thống lấy toàn bộ dữ liệu và trả về cho người dùng
    - Hệ thống hiển thị danh sách đề tài
* Luồng thay thế:
  + - Xem toàn bộ danh sách đề tài
    - Xem đề tài đã duyệt
    - Xem đề tài chưa duyệt
* Yêu cầu cụ thể: đề tài đã được đăng bởi công ty đã được chấp nhận.
* Điều kiện trước: giảng viên hướng dẫn phải đăng nhập, dữ liệu các đề tài có trong hệ thống
* Điều kiện sau: hiển thị danh sách đề tài
  + - 1. Use-case duyệt đề tài công ty

Đặc tả use-case xem duyệt đề tài công ty

* Mục đích: duyệt các đề tài có phù hợp với yêu cầu chuyên môn, phù hợp với yêu cầu thực tập của sinh viên.
* Tác nhân/ đối tượng sử dụng: giảng viên hướng dẫn
* Mô tả chung: Sinh viên muốn đăng ký được đề tài thì giảng viên hướng dẫn phải duyệt các đề tài phù hợp.
* Luồng sự kiện chính:
  + - Giảng viên hướng dẫn nhấn nút đề tài
    - Hệ thống lấy toàn bộ danh sách đề tài
    - Hệ thống hiển thị danh sách đề tài
    - Giảng viên duyệt đề tài (chấp nhận/ từ chối)
    - Hệ thống thông báo cảnh báo hành động
    - Giảng viên chấp nhận đề tài
    - Hệ thống thông báo
* Luồng thay thế: giảng viên không chấp nhận đề tài
* Yêu cầu cụ thể: đề tài của công ty nào thì công ty đó phải được duyệt thành công trước đó.
* Điều kiện trước: giảng viên đã đăng nhập, đã có dữ liệu về các đề tài trước đó.
* Điều kiện sau: mỗi đề tài sau khi duyệt thì hệ thống phải thông báo cho người dùng về tình trạng duyệt thành công hay không và hiển thị lại toàn bộ danh sách đề tài.
  + - 1. Use-case xem danh sách sinh viên

Đặc tả use-case xem danh sách sinh viên

* Mục đích: xem/ biết danh sách sinh viên đang thực tập/ đã kết thúc thực tập hoặc đã bị hủy.
* Tác nhân/ đối tượng sử dụng: giảng viên hướng dẫn
* Mô tả chung: giảng viên muốn biết trong kỳ thực tập thì có những sinh viên nào thực tập hoặc toàn bộ danh sách sinh viên đã/ đang/ sẽ thực tập.
* Luồng sự kiện chính:
  + - Giảng viên nhấn nút danh sách sinh viên
    - Hệ thống kiểm tra và lấy toàn bộ danh sách
    - Hệ thống hiển thị danh sách sinh viên
* Luồng thay thế: không có
* Yêu cầu cụ thể: sinh viên đã đăng ký thực tập
* Điều kiện trước: giảng viên đã đăng nhập, danh sách sinh viên thực tập đã được đăng ký trong hệ thống.
* Điều kiện sau: hiển thị toàn bộ danh sách sinh viên
  + - 1. Use-case xem đề tài sinh viên đăng ký

Đặc tả use-case xem đề tài sinh viên đăng ký

* Mục đích: xem danh sách các đề tài sinh viên và là tiền đề để duyệt mỗi đề tài cho sinh viên.
* Tác nhân/ đối tượng sử dụng: giảng viên hướng dẫn
* Mô tả chung: giảng viên muốn duyệt được đề tài cho sinh viên thì bắt buộc phải xem danh sách các đề tài.
* Luồng sự kiện chính:
  + - Giảng viên nhấn nút đề tài sinh viên
    - Hệ thống lấy danh sách đề tài sinh viên
    - Hệ thống hiển thị danh sách đề tài sinh viên
* Luồng thay thế: không có
* Yêu cầu cụ thể: lấy được toàn bộ danh sách sinh viên
* Điều kiện trước: giảng viên đã đăng nhập, sinh viên đã đăng ký đề tài thực tập trên hệ thống
* Điều kiện sau: hệ thống hiển thị danh sách sinh viên
  + - 1. Use-case duyệt đề tài sinh viên

Đặc tả use-case duyệt đề tài sinh viên

* Mục đích: duyệt các đề tài mà sinh viên đã đăng ký trong kỳ thực tập hiện tại.
* Tác nhân/ đối tượng sử dụng: giảng viên hướng dẫn
* Mô tả chung: Sau khi sinh viên đăng ký kỳ thực tập và đăng ký thành công các đề tài thì giảng viên hướng dẫn phải duyệt đề tài cho sinh viên.
* Luồng sự kiện chính:
  + - Giảng viên nhấn nút đề tài sinh viên
    - Hệ thống lấy danh sách đề tài sinh viên
    - Hệ thống hiển thị danh sách đề tài sinh viên
    - Giảng viên nhấn nút chấp nhận/ từ chối
    - Hệ thống thông báo cánh báo hành động
    - Giảng viên chấp nhận đề tài
    - Hệ thống kiểm trả dữ liệụ
    - Nếu sinh viên đó có nhiều hơn 1 đề tài, hệ thống tự động từ chối các đề tài còn lại trong hệ thống của sinh viên đó
    - Hệ thống hiển thị thông báo
* Luồng thay thế: giảng viên từ chối đề tài cho sinh viên
* Yêu cầu cụ thể: giảng viên sẽ duyệt đề từng đề tài sinh viên. Các đề tài được sắp xếp theo mã số sinh viên. Giảng viên sẽ xem nhóm đề tài của sinh viên, nếu đề tài nào có chỉ số so khớp cao thì nên duyệt chấp nhận.
* Điều kiện trước: sinh viên đã đăng ký thực tập, sinh viên đã đăng ký đề tài, giảng viên đã đăng nhập vào hệ thống
* Điều kiện sau: hệ thống hiển thị thông báo hành động duyệt đề tài sinh viên thành công hay không thành công.
  + - 1. Use-case đăng/ xóa lịch trình

Đặc tả use-case đăng/ xóa lịch trình

* Mục đích: đăng các thông tin các sự kiện trong thời gian thực tập của sinh viên nhằm thông báo chó sinh viên biết cần làm những việc gì.
* Tác nhân/ đối tượng sử dụng: giảng viên hướng dẫn
* Mô tả chung: trong mỗi đợt thực tập, giảng viên hướng dẫn phải đăng các lịch trình thực tập.
* Luồng sự kiện chính:
  + - Giảng viên ướng dẫn nhấn nút lịch trình
    - Hệ thống lấy dữ liệu và chuyển dữ liệu sang định dạng JSON
    - Hệ thống hiển thị thông tin các lịch trình
    - Trường hợp giảng viên muốn đăng lịch trình: giảng viên hướng dẫn nhấn nút đăng lịch trình
    - Giảng viên hướng dẫn nhập liệu các thông tin
    - Hệ thống kiểm tra lỗi nhập liệu
    - Giảng viên hướng dẫn nhấn nút đăng lịch trình
    - Hệ thống đưa ra cảnh báo hành động
    - Giảng viên chấp nhận, hệ thống thực hiện hành động ghi dữ liệu
    - Hệ thống hiển thị thông báo kết thúc hành động
    - Trường hợp giảng viên hướng dẫn muốn xóa lịch trình: giảng viên nhấn nút xóa
    - Hệ thống đưa ra cảnh báo hành động
    - Giảng viên thực hiện hành động
    - Hệ thống thông báo kết thúc hành động
* Luồng thay thế:
  + - Trường hợp giảng viên đăng lịch trình, hệ thống sẽ yêu cầu đăng lại nếu lịch trình đó có lỗi
    - Trường hợp giảng viên muốn xóa lịch trình, hệ thống thông báo không thành công nếu có lỗi xảy ra
* Yêu cầu cụ thể: giảng viên cần xác đinh khoảng thời gian lịch trình
* Điều kiện trước: giảng viên đã đăng nhập, lịch trình sau không được trùng với lịch trình trước, xóa lịch trình thì cần xem có bị xung đột thời gian hay không.
* Điều kiện sau: hệ thống thông báo đăng lịch trình/ xóa lịch trình có thành công hay không.
  + - 1. Use-case upload/download file

Đặc tả use-case upload/download file cho giảng viên hướng dẫn

* Mục đích: giảng viên hướng dẫn muốn upload/download các file cần thiết lên hệ thống để cho sinh viên nắm bắt được thông tin về tài liệu.
* Tác nhân/ đối tượng sử dụng: giảng viên hướng dẫn
* Mô tả chung: khi giảng viên hướng dẫn muốn đăng hoặc tải xuống các file cần thiết từ hệ thống.
* Luồng sự kiện chính:
  + - Giảng viên hướng dẫn nhấn nút file hoặc nút tài liệu
    - Hệ thống lấy danh sách file và hiển thị cho giảng viên hướng dẫn
    - Giảng viên hướng dẫn nhấn nút tải file
    - Hệ thống truy cập file trong server và gửi về cho giảng viên hướng dẫn
    - Giảng viên hướng dẫn nhất nút lưu file
    - Nếu giảng viên hướng dẫn muốn upload file: hệ thống hiển thị giao diện upload file
    - Giảng viên hướng dẫn chọn file muốn upload và điền đầy đủ thông tin
    - Hệ thống kiểm tra. Nếu đáp ứng yêu cầu thì lưu file vào server
    - Hệ thống thông báo cho giảng viên hướng dẫn
    - Nếu giảng viên hướng dẫn muốn xóa file, chọn nút xóa file
    - Hệ thống kiểm tra, lấy mã file và tiến hành cập nhật dữ liệu
    - Hệ thống trả về cho giảng viên hướng dẫn sau khi thực hiện hành động xóa.
* Luồng thay thế: hệ thống yêu cầu nhập lại, chọn lại file nếu quá dung lượng quy định của hệ thống.
* Yêu cầu cụ thể: Các file upload lên hệ thống phải có tính xác thực, giảng viên hướng dẫn phải tự chịu trách nhiệm về hành động upload file của mình lên hệ thống.
* Điều kiện trước: giảng viên hướng dẫn đã đăng nhập và chọn file muốn upload hoặc chọn file muốn tải/ xóa. Hiển thị thông báo cảnh báo hành động.
* Điều kiện sau: Hiện thị thông báo sau hành động, hiển thị file đã upload hoặc không hiển thị file đã xóa
  + - 1. Use-case email

Đặc tả use-case email cho giảng viên hướng dẫn

* Mục đích: trao đổi thông tin với sinh viên/ người hướng dẫn/ đại diện công ty.
* Tác nhân/ đối tượng sử dụng: giảng viên hướng dẫn
* Mô tả chung: Khi giảng viên hướng dẫn muốn trao đổi thông tin với sinh viên/ người hướng dẫn/ đại diện công ty.
* Luồng sự kiện chính:
  + - Giảng viên hướng dẫn nhấn vào nút email
    - Hệ thống lấy toàn bộ thông tin về email, bao gồm các email đã gửi, đã đọc, chưa đọc, đã nhận,…
    - Hệ thống hiển thị thông tin cho giảng viên hướng dẫn.
* Luồng thay thế:
  + - Gửi mail: trường hợp giảng viên hướng dẫn muốn gửi mail tới các người dùng khác trong hệ thống
    - Xem mail: giảng viên hướng dẫn muốn xem mail
    - Đọc mail: giảng viên hướng dẫn muốn đọc mail
    - Xóa mail: giảng viên hướng dẫn muốn xóa mail
* Yêu cầu cụ thể: không có
* Điều kiện trước: tài khoản đã được xác thực
* Điều kiện sau: hiển thị toàn bộ thông tin mail cho giảng viên hướng dẫn
  + - 1. Use-case chấm điểm

Đặc tả use-case chấm điểm cho sinh viên

* Mục đích: chấm điểm cho sinh viên sau khi sinh viên kết thúc thực tập
* Tác nhân/ đối tượng sử dụng: giảng viên hướng dẫn
* Mô tả chung: sau khi sinh viên kết thúc kỳ thực tập, giảng viên hướng dẫn tiến hành tải các báo cáo giữa kỳ, báo cáo cuối kỳ. Sau khi giảng viên đọc các báo cáo thì tiến hành chấm điểm cho sinh viên trên hệ thống cho từng sinh viên.
* Luồng sự kiện chính:
  + - Giảng viên nhấn vào nút chấm điểm
    - Hệ thống lấy toàn bộ danh sách sinh viên và hiển thị cho giảng viên hướng dẫn
    - Giảng viên hướng dẫn nhập điểm cho sinh viên
    - Hệ thống bắt lỗi nhập liệu
    - Giảng viên nhấn nút chấm điểm
    - Hệ thống báo cảnh báo hành động
    - Giảng viên hướng dẫn xác thực hành động
    - Hệ thống cập nhật dữ liệu
    - Hệ thống thông báo kết thúc hành động
* Luồng thay thế: hệ thống yêu cầu nhập lại điểm nếu điểm quá giá trị trong khoảng 0 đến 10
* Yêu cầu cụ thể: mỗi lần chấm điểm chỉ có thể chấm điểm cho một sinh viên, không thể chấm điểm cho nhiều sinh viên cùng lúc.
* Điều kiện trước: giảng viên đã đăng nhập, sinh viên đã kết thúc thực tập, sinh viên đã nộp báo cáo, giảng viên đã đọc báo cáo của sinh viên
* Điều kiện sau: hệ thống thông báo hành động, lấy danh sách sinh viên chưa chấm điểm, sinh viên nào đã được chấm thì không được hiển thị.
  + 1. Biểu đồ use-case người hướng dẫn
       1. Use-case đăng nhập

Đặc tả use-case đăng nhập

* Mục đích: Đảm bảo xác thực thông tin người dùng và định danh người dùng trong hệ thống
* Tác nhân/ đối tượng sử dụng: tất cả các actor là con người
* Mổ tả chung: Người dùng muốn sử dụng được hệ thống thì bắt buộc phải đăng nhập vào hệ thống.
* Luồng sự kiện chính:
  + - Người dùng chọn đăng nhập trên hệ thống
    - Hệ thống yêu cầu người dùng nhập tài khoản và mật khẩu
    - Người dùng nhập xong thông tin đăng nhập và click nút đăng nhập.
    - Hệ thống kiểm tra thông tin đăng nhập. Nếu đăng nhập thành công thì chuyển hướng tới trang chủ. Ngược lại thì thông báo lỗi và yêu cầu nhập lại.
* Luồng thay thế: Sai quá số lần quy định thì yêu cầu cài đặt lại mật khẩu thông qua xác thực.
* Yêu cầu cụ thể: Tên đăng nhập và mật khẩu phải khớp với thông tin đã đăng ký trong hệ thống.
* Điều kiện trước: Người dùng đã đăng ký thành công thành viên trong hệ thống
* Điều kiện sau: trả về trang chủ
  + - 1. Use-case xem danh sách sinh viên

Đặc tả use-case xem danh sách sinh viên cho người hướng dẫn

* Mục đích: xem/ biết danh sách sinh viên đang thực tập/ đã kết thúc thực tập hoặc đã bị hủy do người hướng dẫn quản lý.
* Tác nhân/ đối tượng sử dụng: người hướng dẫn
* Mô tả chung: người hướng dẫn muốn biết trong kỳ thực tập thì có những sinh viên nào thực tập do mình quản lý
* Luồng sự kiện chính:
  + - Người hướng dẫn nhấn nút danh sách sinh viên
    - Hệ thống kiểm tra và lấy toàn bộ danh sách
    - Hệ thống hiển thị danh sách sinh viên
* Luồng thay thế: không có
* Yêu cầu cụ thể: lấy toàn bộ danh sách sinh viên đang do người hướng dẫn thực tập quản lý.
* Điều kiện trước: người đã đăng nhập, danh sách sinh viên thực tập đã được đăng ký trong hệ thống, sinh viên đã đăng ký đề tài và được giảng viên hướng dẫn đồng ý đề tài
* Điều kiện sau: hiển thị toàn bộ danh sách sinh viên
  + - 1. Use-case upload/download file

Đặc tả use-case upload/download file cho người hướng dẫn

* Mục đích: người hướng dẫn đăng các file đánh giá/ báo cáo của sinh viên cho giảng viên hướng dẫn.
* Tác nhân/ đối tượng sử dụng: người hướng dẫn
* Mô tả chung: người hướng dẫn đăng file đánh giá sau khi sinh viên kết thúc thực tập, thời gian tối đa là 15 ngày kể từ ngày kết thúc thực tập.
* Luồng sự kiện chính:
  + - Sinh viên nhấn nút file hoặc nút tài liệu
    - Hệ thống lấy danh sách file và hiển thị cho sinh viên
    - Sinh viên nhấn nút tải file
    - Hệ thống truy cập file trong server và gửi về cho sinh viên
    - Sinh viên nhất nút lưu file
    - Nếu sinh viên muốn upload file: hệ thống hiển thị giao diện upload file
    - Sinh viên chọn file muốn upload và điền đầy đủ thông tin
    - Sinh viên lựa chọn đặc tính của file là báo cáo, tài liệu, …
    - Hệ thống kiểm tra. Nếu đáp ứng yêu cầu thì lưu file vào server
    - Hệ thống thông báo cho sinh viên
    - Nếu sinh viên muốn xóa file, chọn nút xóa file
    - Hệ thống kiểm tra, lấy mã file và tiến hành cập nhật dữ liệu
    - Hệ thống trả về cho sinh viên sau khi thực hiện hành động xóa.
* Luồng thay thế: hệ thống yêu cầu nhập lại, chọn lại file nếu quá dung lượng quy định của hệ thống.
* Yêu cầu cụ thể: Các file upload lên hệ thống phải có tính xác thực, sinh viên phải tự chịu trách nhiệm về hành động upload file của mình lên hệ thống.
* Điều kiện trước: sinh viên đã đăng nhập và chọn file muốn upload hoặc chọn file muốn tải/ xóa. Hiển thị thông báo cảnh báo hành động.
* Điều kiện sau: Hiện thị thông báo sau hành động
  + - 1. Use-case cập nhật thông tin

Đặc tả use-case cập nhật thông tin cá nhân cho người hướng dẫn

* Mục đích: Thay đổi các thông tin cá nhân bị sai hoặc muốn thay đổi theo phù hợp với yêu cầu của người hướng dẫn.
* Tác nhân/ đối tượng sử dụng: tất cả các đối tượng là con người.
* Mổ tả chung: Người hướng dẫn muốn thay đổi thông tin cá nhân của mình. Một số thông tin sẽ không được thay đổi hay cập nhật (ví dụ như email).
* Luồng sự kiện chính:
  + - Người hướng dẫn đăng nhập
    - Hệ thống trả về giao diện trang chủ
    - Người hướng dẫn nhấn vào nút cập nhật thông tin cá nhân
    - Hệ thống trả về giao diện cập nhật thông tin
    - Người hướng dẫn sẽ cập nhật các thông tin được phép thay đổi thông tin
    - Hệ thống bắt lỗi nhập liệu trên giao diện
    - Người hướng dẫn nhấn nút cập nhật
    - Hệ thống kiểm tra tính hợp lệ dữ liệu và lưu vào CSDL
    - Hệ thông báo cho người hướng dẫn về việc cập nhật thông tin có thành công hay không.
* Luồng thay thế: Người hướng dẫn nhập sai thì hệ thống yêu cầu nhập lại
* Yêu cầu cụ thể: Thông tin cá nhân của người hướng dẫn đã được lưu trước đó và chỉ được phép thay đổi một số thông tin cho chép của hệ thống.
* Điều kiện trước: Người hướng dẫn đã đăng ký thông tin cá nhân trên hệ thống
* Điều kiện sau: Thông báo cho người hướng dẫn đã cập nhật thông tin cá nhân thành công hay không.
  + - 1. Use-case email

Đặc tả use-case email cho người hướng dẫn

* Mục đích: trao đổi thông tin, các vướng mắc trong quá hướng dẫn thưc tập sinh viên và phản hồi định kỳ 2 tuần 1 lần
* Tác nhân/ đối tượng sử dụng: tất cả người hướng dẫn
* Mô tả chung: Khi hướng dẫn dùng muốn trao đổi thông tin hoặc muốn phản hồi thông tin cho giảng viên hướng dẫn thì cần sử dụng chức năng email trên hệ thống.
* Luồng sự kiện chính:
  + - Người hướng dẫn nhấn vào nút email
    - Hệ thống lấy toàn bộ thông tin về email, bao gồm các email đã gửi, đã đọc, chưa đọc, đã nhận,…
    - Hệ thống hiển thị thông tin cho người dùng.
    - Nếu người hướng dẫn muốn gửi phản hổi thì cần tích thêm vào ô gửi phản hồi
* Luồng thay thế:
  + - Gửi mail: trường hợp người hướng dẫn muốn gửi mail tới các người dùng khác trong hệ thống
    - Xem mail: người hướng dẫn muốn xem mail
    - Đọc mail: người hướng dẫn muốn đọc mail
    - Xóa mail: người hướng dẫn muốn xóa mail
* Yêu cầu cụ thể: không có
* Điều kiện trước: tài khoản đã được xác thực, nếu muốn phản hồi thì phải do hệ thống kiểm tra theo hạn phản hổi. Nếu quá hạn sẽ không phản hồi cộng dồn được.
* Điều kiện sau: hiển thị toàn bộ thông tin mail cho người hướng dẫn
  + 1. Biểu đồ use-case đại diện công ty
       1. Use-case đăng nhập

Đặc tả use-case đăng nhập

* Mục đích: Đảm bảo xác thực thông tin người dùng và định danh người dùng trong hệ thống
* Tác nhân/ đối tượng sử dụng: tất cả các actor là con người
* Mổ tả chung: Người dùng muốn sử dụng được hệ thống thì bắt buộc phải đăng nhập vào hệ thống.
* Luồng sự kiện chính:
  + - Người dùng chọn đăng nhập trên hệ thống
    - Hệ thống yêu cầu người dùng nhập tài khoản và mật khẩu
    - Người dùng nhập xong thông tin đăng nhập và click nút đăng nhập.
    - Hệ thống kiểm tra thông tin đăng nhập. Nếu đăng nhập thành công thì chuyển hướng tới trang chủ. Ngược lại thì thông báo lỗi và yêu cầu nhập lại.
* Luồng thay thế: Sai quá số lần quy định thì yêu cầu cài đặt lại mật khẩu thông qua xác thực.
* Yêu cầu cụ thể: Tên đăng nhập và mật khẩu phải khớp với thông tin đã đăng ký trong hệ thống.
* Điều kiện trước: Người dùng đã đăng ký thành công thành viên trong hệ thống
* Điều kiện sau: trả về trang chủ
  + - 1. Use-case đăng ký tài khoản người hướng dẫn

Đặc tả use-case đăng ký tài khoản người hướng dẫn

* Mục đích: tạo tài khoản người hướng dẫn của công ty
* Tác nhân/ đối tượng sử dụng: đại diện công ty
* Mô tả chung: Mỗi công ty sau khi đăng ký và được duyệt thành công cần phải đăng ký người hướng dẫn cho sinh viên trước khi có thể đăng đề tài thực tập
* Luồng sự kiện chính:
  + - Đại diện công ty nhấn nút nhân viên
    - Hệ thống lấy danh sách nhân viên
    - Hệ thống hiển thị danh sách nhân viên và hiển thị giao diện biểu mẫu thêm mới nhân viên
    - Nếu đại diện công ty muốn thêm mới nhân viên: đại diện công ty cần nhập vào biểu mẫu
    - Hệ thống kiểm tra nhập liệu
    - Đại diện công ty nhấn nút thêm mới nhân viên
    - Hệ thống thông báo cảnh báo
    - Đại diện công ty xác nhận
    - Hệ thống hiển thị thông báo kết quả hành động.
* Luồng thay thế: hệ thống yêu cầu nhập lại nếu sai thông tin hoặc tài khoản email đã tồn tại trong hệ thống
* Yêu cầu cụ thể: đại diện cần phải thêm mới tối thiểu một người hướng dẫn cho công ty của mình.
* Điều kiện trước: đại diện công ty đã đăng nhập
* Điều kiện sau: hiển thị thông báo kết quả hành động thêm mới nhân viên, hiển thị danh sách nhân viên.
  + - 1. Use-case đăng ký công ty

Đặc tả use-case xem đăng ký công ty

* Mục đích: đăng ký thông tin công ty
* Tác nhân/ đối tượng sử dụng: đại diện công ty
* Mô tả chung: công ty muốn sử dụng hệ thống phải đăng ký thành công công ty và chờ giảng viên hướng dẫn xác thực và duyệt công ty.
* Luồng sự kiện chính:
  + - Đại diện công ty đăng ký mới và đăng nhập thành công
    - Hệ thống chuyển về trang đăng ký thông tin công ty
    - Đại diện công ty nhập các thông tin trên biểu mẫu
    - Hệ thống xác thực thông tin
    - Đại diện công nhấn nút đăng ký thông tin công ty
    - Hệ thống thông báo cảnh báo
    - Đại diện công ty xác nhận thông tin
    - Hệ thống kiểm tra và cập nhật dữ liệu
    - Hệ thống thông báo và trả về kết quả
* Luồng thay thế: yêu cầu nhập lại nếu sai thông tin đăng ký
* Yêu cầu cụ thể: thông tin công ty phải xác thực và chính xác, đại diện công ty sẽ phải chịu trách nhiệm trước những thông tin đã khai trên hệ thống.
* Điều kiện trước: đại diện công ty đã đăng nhập thành công và chưa đăng ký công ty lần nào.
* Điều kiện sau: hiển thị thông báo sau khi kết thúc hành động, thông báo đã đăng ký thành công hay không.
  + - 1. Use-case cập nhật công ty

Đặc tả use-case cập nhập công ty

* Mục đích: cập nhật thông tin công ty về lĩnh vực, mô hình hoạt động, các thông tin chi tiết về công ty nếu có sự thay đổi.
* Tác nhân/ đối tượng sử dụng: đại diện công ty
* Mô tả chung: quá trình hoạt động, công ty có sự thay đổi thông tin, cần cập nhật thông tin.
* Luồng sự kiện chính:
  + - Đại diện công ty nhấn nút cập nhật công ty
    - Hệ thống lấy dữ liệu của công ty và hiển thị thông tin
    - Đại diện công ty xem xét các trường thông tin cần thay đổi và nhập liệu trên biểu mẫu. Các trường không được phép thay đổi, đại diện công ty sẽ không thể cập nhật
    - Hệ thống bắt lỗi nhập liệu
    - Đại diện công ty nhấn nút cập nhật thông tin
    - Hệ thống hiện thị thông báo cảnh báo
    - Đại diện công ty xác nhận
    - Hệ thống tiến hành kiểm tra và cập nhật dữ liệu
    - Hệ thống thông báo kết quả của hành động.
* Luồng thay thế: yêu cầu đại diện công ty nhập lại thông tin nếu nhập sai.
* Yêu cầu cụ thể: thông tin công ty phải chính xác, đại diện công ty phải tự chịu trách nhiệm trước những thông tin mà mình đã khai trên hệ thống.
* Điều kiện trước: đại diện công ty đã đăng nhập, thông tin công ty đã được đăng ký trên hệ thống
* Điều kiện sau: hiển thị thông báo kết quả của hành động cập nhật thông tin công ty.
  + - 1. Use-case đăng đề tài

Đặc tả use-case đăng đề tài

* Mục đích: đăng các đề tài mà công ty đang có nhu cầu tuyển thực tập.
* Tác nhân/ đối tượng sử dụng: đại diện công ty
* Mô tả chung: công ty có nhu cầu tuyển sinh viên thực tập và đăng ký tham gia chương trình thực tập tại doanh nghiệp
* Luồng sự kiện chính:
  + - Đại diện công ty nhấn nút đăng đề tài
    - Hệ thống lấy danh sách đề tài trước đó, danh sách người hướng dẫn tại công ty và hiển thị biểu mẫu thêm mới đề tài.
    - Đại diện công ty sẽ nhập liệu cho đề tài mới
    - Đại diên công ty lựa chọn người hướng dẫn cho đề tài
    - Hệ thống kiểm tra lỗi nhập liệu
    - Đại diện công ty nhấn nút thêm đề tài
    - Hệ thống thông báo cảnh báo
    - Đại diện công ty xác thực hành động
    - Hệ thống kiểm tra và cập nhật dữ liệu
    - Hệ thống thông báo kết quả sau kết thúc hành động
* Luồng thay thế: yêu cầu đại diện công ty nhập lại thông tin nếu như có lỗi xảy ra.
* Yêu cầu cụ thể: các đề tài phải nằm trong lĩnh vực công nghệ thông tin, đối phần yêu cầu lập trình phải nhập đúng theo quy định
* Điều kiện trước: đại diện công ty đã đăng nhập
* Điều kiện sau: hiện thị thông báo sau khi kết thúc hành động thêm mới đề tài.
  + - 1. Use-case xóa đề tài

Đặc tả use-case xóa đề tài

* Mục đích: xóa các đề tài không còn có nhu cầu tuyển sinh viên thực tập.
* Tác nhân/ đối tượng sử dụng: đại diện công ty
* Mô tả chung: đại diện công ty muốn xóa các đề tài không còn thực tế hoặc không còn có nhu cầu tuyển thực tập
* Luồng sự kiện chính:
  + - Đại diện công ty nhấn nút đề tài
    - Hệ thống lấy danh sách đề tài trước đó, danh sách người hướng dẫn tại công ty và hiển thị danh sách đề tài
    - Đại diện công ty lựa chọn đề tài muốn xóa
    - Hệ thống thông báo cảnh báo
    - Đại diện công ty xác nhận
    - Hệ thống kiểm tra và cập nhật dữ liệu
    - Hệ thống trả về kết quả hành động
* Luồng thay thế: thông báo không thể xóa đề tài nếu có sinh viên đang thực tập, thay vào đó sẽ chuyển sang trạng thái không sử dụng
* Yêu cầu cụ thể: đề tài cần xóa không có sinh viên nào đăng ký hoặc đã hết hạn đăng ký.
* Điều kiện trước: đại diện công ty đã đăng nhập, đề tài có trong hệ thống.
* Điều kiện sau: hiện thị thông báo kết quả xóa đề tài, cập nhật lại và hiện thị các đề tài khác chưa xóa.
  + - 1. Use-case email

Đặc tả use-case email cho đại diện công ty

* Mục đích: trao đổi thông tin, phản hồi cho người dùng, hệ thống đặc biệt là giảng viên hướng dẫn.
* Tác nhân/ đối tượng sử dụng: tất cả người dùng
* Mô tả chung: Khi đại diện công ty muốn trao đổi thông tin thì cần sử dụng chức năng email trên hệ thống.
* Luồng sự kiện chính:
  + - Đại diện công ty nhấn vào nút email
    - Hệ thống lấy toàn bộ thông tin về email, bao gồm các email đã gửi, đã đọc, chưa đọc, đã nhận,…
    - Hệ thống hiển thị thông tin cho đại diện công ty.
* Luồng thay thế:
  + - Gửi mail: trường hợp đại diện công ty muốn gửi mail tới các người dùng khác trong hệ thống
    - Xem mail: đại diện công ty muốn xem mail
    - Đọc mail: đại diện công ty muốn đọc mail
    - Xóa mail: đại diện công ty muốn xóa mail
* Yêu cầu cụ thể: không có
* Điều kiện trước: tài khoản đã được xác thực
* Điều kiện sau: hiển thị toàn bộ thông tin mail cho đại diện công ty
  + - 1. Use-case
    1. Biều đồ use-case admin
       1. Use-case đăng nhập admin

Đặc tả use-case đăng nhập

* Mục đích: Đảm bảo xác thực thông tin người dùng và định danh người dùng trong hệ thống
* Tác nhân/ đối tượng sử dụng: tất cả các actor là con người
* Mổ tả chung: Người dùng muốn sử dụng được hệ thống thì bắt buộc phải đăng nhập vào hệ thống.
* Luồng sự kiện chính:
  + - Admin chọn đăng nhập trên hệ thống
    - Hệ thống yêu cầu admin nhập tài khoản và mật khẩu
    - Admin nhập xong thông tin đăng nhập và nhấn nút đăng nhập.
    - Hệ thống kiểm tra thông tin đăng nhập. Nếu đăng nhập thành công thì chuyển hướng tới trang quản trị. Ngược lại thì thông báo lỗi và yêu cầu nhập lại.
* Luồng thay thế: Sai quá số lần quy định thì yêu cầu cài đặt lại mật khẩu thông qua xác thực.
* Yêu cầu cụ thể: Tên đăng nhập và mật khẩu phải khớp với thông tin đã đăng ký trong hệ thống.
* Điều kiện trước: Admin đã đăng ký thành công thành viên trong hệ thống
* Điều kiện sau: trả về trang quản trị
  + - 1. Use quản lý tài khoản

Đặc tả use-case quản lý tải khoản

* Mục đích: quản lý các tài khoản trong hệ thống
* Tác nhân/ đối tượng sử dụng: quản trị viên (admin)
* Mô tả chung: admin có quyền quản lý tất cả các tài khoản trong hệ thống, có quyền thêm, sửa hoặc xóa tài khoản
* Luồng sự kiện chính:
  + - Admin nhấn nút quản lý tải khoản
    - Hệ thống tiến hành lấy dữ liệu và trả kết quả về cho admin
    - Hệ thống hiển thị dữ liệu trên màn hình
    - Trường hợp Admin muốn thêm mới tài khoản:
      * Admin nhấn nút thêm mới tài khoản
      * Hệ thống hiển thị giao diện thêm mới
      * Admin nhập email và phân quyền
      * Admin nhấn thêm tài khoản
      * Hệ thống thông báo cảnh báo
      * Admin xác nhận
      * Hệ thống kiểm tra, thêm mới và ghi lịch sử
      * Hệ thống trả về thông báo cho admin
    - Trường hợp admin muốn sửa tài khoản
      * Admin nhấn vào tài khoản muốn sửa
      * Admin sửa phân quyền
      * Admin nhấn nút sửa
      * Hệ thống hiển thị cảnh báo
      * Admin xác nhận
      * Hệ thống cập nhật dữ liệu và ghi lịch sử
      * Hệ thống hiển thị thông báo cho admin
    - Trường hợp admin muốn xóa tài khoản
      * Admin nhấn vào tài khoản muốn xóa
      * Hệ thống hiển thị cảnh báo
      * Admin xác nhận
      * Hệ thống cập nhật dữ liệu và ghi lịch sử
      * Hệ thống hiển thị thông báo cho admin
* Luồng thay thế: ghi lịch sử cho các trường hợp bị lỗi
* Yêu cầu cụ thể: bắt buộc phải phân quyền người dùng, không được trùng tài khoản email trong hệ thống
* Điều kiện trước: admin đã đăng nhập
* Điều kiện sau: hiển thị trạng thái/ dữ liệu cập nhật trên màn hình
  + - 1. Use-case quản lý công ty

Đặc tả use-case quản lý công ty

* Mục đích: quản lý các công ty đăng ký tham gia chương trình thực tập trên hệ thống.
* Tác nhân/ đối tượng sử dụng: admin
* Mô tả chung: admin có quyền xóa công ty và không thể thêm hoặc sửa công ty, ngoài ra admin cũng có quyền duyệt các công ty như giảng viên hướng dẫn
* Luồng sự kiện chính:
  + - Admin nhấn nút quản lý công ty
    - Hệ thống tiến hành lấy dữ liệu và trả kết quả về cho admin
    - Hệ thống hiển thị dữ liệu trên màn hình
    - Admin muốn xóa công ty
    - Admin nhấn vào tài khoản muốn xóa
    - Hệ thống hiển thị cảnh báo
    - Admin xác nhận
    - Hệ thống cập nhật dữ liệu và ghi lịch sử
    - Hệ thống hiển thị thông báo cho admin
* Luồng thay thế: chuyển trạng thái của công ty từ đang hoạt động sang trạng thái tạm dừng nếu công ty đó đã được xác thực.
* Yêu cầu cụ thể: admin phải kiểm tra thông tin về công ty trước khi xóa, mọi dữ liệu không được xóa khỏi cơ sở dữ liệu mà chỉ được phép chuyển trang thái.
* Điều kiện trước: admin đã đăng nhập, công ty đã đăng ký trong hệ thống
* Điều kiện sau: hiển thị thông báo kết quả hành động và tải lại dữ liệu của danh sách công ty.
  + - 1. Use-case quản lý đề tài

Đặc tả use-case quản lý đề tài

* Mục đích: quản lý các đề tài thực tập trong hệ thống
* Tác nhân/ đối tượng sử dụng: admin
* Mô tả chung: admin có quyền quản lý đề tài, do đó admin có thể xóa các đề tài nhưng không thể thêm đề tài. Việc thêm đề tài là do phía công ty đăng đề tài theo nhu cầu thực tế của công ty.
* Luồng sự kiện chính:
  + - Admin nhấn nút quản lý đề tài
    - Hệ thống tiến hành lấy dữ liệu và trả kết quả về cho admin
    - Hệ thống hiển thị dữ liệu trên màn hình
    - Admin nhấn vào đề tài muốn xóa
    - Hệ thống hiển thị cảnh báo
    - Admin xác nhận
    - Hệ thống cập nhật dữ liệu và ghi lịch sử
    - Hệ thống hiển thị thông báo cho admin
* Luồng thay thế: chuyển trạng thái đề tài từ hoạt động sang chế độ tạm dừng nếu đề tài đó đã có sinh viên đăng ký thực tập
* Yêu cầu cụ thể: admin phải xác định các đề tài nào đang có sinh viên thực tập và không được xóa khỏi cơ sở dữ liệu nếu có sinh viên đã đăng ký đề tài.
* Điều kiện trước: admin đã đăng nhập, đề tài đã tồn tại trong hệ thống
* Điều kiện sau: hiển thị thông báo kết thúc hành động và hiện thị lại danh sách đề tài.
  + - 1. Use-case quản lý điểm

Đặc tả use-case quản lý điểm thi sinh viên

* Mục đích: quản lý toàn bộ điểm thi của sinh viên đã tham gia thực tập
* Tác nhân/ đối tượng sử dụng: admin
* Mô tả chung: sau khi kết thúc đợt thực tập, giảng viên hướng dẫn phải chấm điểm thi cho sinh viên. Admin có quyền sửa điểm và quyền chấm điểm thi cho sinh viên nếu như giảng viên hướng dẫn có yêu cầu.
* Luồng sự kiện chính:
  + - Admin nhấn nút quản lý điểm thi
    - Hệ thống lấy toàn bộ dữ liệu điểm thi
    - Hệ thống hiển thị danh sách điểm thi
    - Trường hợp admin muốn sửa điểm thi
      * Admin nhấn nút sửa điểm thi
      * Hệ thống hiển thị giao diện sửa điểm thi
      * Admin nhập điểm thi
      * Hệ thống bắt lỗi nhập liệu điểm từ 0 đến 10
      * Admin nhấn nút sửa điểm
      * Hệ thống hiển thị thông báo cảnh báo
      * Admin xác nhận
      * Hệ thống xử lý cập nhật dữ liệu và ghi lịch sử
      * Hệ thống hiển thị thông báo kết thúc hành động.
    - Trường hợp admin muốn xóa điểm thi
      * Admin nhấn nút xóa điểm thi
      * Hệ thống hiển thị thông báo cảnh báo
      * Admin xác nhận xóa điểm thi
      * Hệ thống xử lý, cập nhật dữ liệu và ghi lịch sử
      * Hệ thống thông báo kết quả kết thúc hành động
* Luồng thay thế: hệ thống yêu cầu admin nhập lại điểm nếu sai.
* Yêu cầu cụ thể: admin chỉ được phép sửa từng điểm thi của từng sinh viên. Việc sửa điểm thi phải có yêu cầu từ giảng viên hướng dẫn. Việc xóa điểm thi không được xóa khỏi cơ sở dữ liệu mà chỉ được chuyển trạng thái sang trạng thái dừng, hoặc đã hết hạn.
* Điều kiện trước: admin đã đăng nhập, điểm thi của sinh viên đã có trong hệ thống.
* Điều kiện sau: hiển thị thông báo kết thúc hành động
  + - 1. Use-case quản lý file

Đặc tả use-case quản lý file

* Mục đích: quản lý toàn bộ file được lên hệ thống
* Tác nhân/ đối tượng sử dụng: admin
* Mô tả chung: Admin sẽ quản lý file được tải lên hệ thống, admin sẽ có quyền thêm mới file, chỉnh sửa thông tin file hoặc xóa file
* Luồng sự kiện chính:
  + - Amin nhấn nút quản lý file
    - Hệ thống tiến hành lấy dữ liệu và trả về kết quả
    - Hệ thống hiển danh sách file
    - Trường hợp admin muốn tải file lên hệ thống
      * Admin nhấn nút upload file
      * Hệ thống hiển thị giao diện upload file
      * Admin nhập dữ liệu và lựa chọn file
      * Hệ thống kiểm tra nhập liệu
      * Admin nhấn nút tải file lên
      * Hệ thống thông báo cảnh báo
      * Admin xác nhận hành động
      * Hệ thống xử lý, cập nhật dữ liệu và ghi lịch sử
      * Hệ thống thông báo kết quả của hành động
    - Trường hợp admin muốn sửa file
      * Admin nhấn nút sửa file
      * Hệ thống hiển giao diện sửa thông tin file
      * Admin nhập liệu
      * Hệ thống kiểm tra nhập liệu
      * Admin nhấn nút cập nhật
      * Hệ thống thông báo cảnh báo
      * Admin xác nhận hành động
      * Hệ thống xử lý, cập nhật dữ liệu và ghi lịch sử
      * Hệ thống thông báo kết quả của hành động
    - Trường hợp admin muốn xóa file
      * Admin nhấn nút xóa file
      * Hệ thống thông báo cảnh báo
      * Admin xác nhận hành động
      * Hệ thống xử lý, cập nhật dữ liệu và ghi lịch sử
      * Hệ thống thông báo kết quả của hành động
* Luồng thay thế: hủy các hành động nếu có lỗi.
* Yêu cầu cụ thể: không được phép chỉnh sửa/ xóa các file của sinh viên
* Điều kiện trước: admin đã đăng nhập, các file muốn xóa hay cập nhật phải có trong cơ sở dữ liệu.
* Điều kiện sau: hiện thị thông báo kết quả sau mỗi hành động.
  + - 1. Use-case quản lý thực tập

Đặc tả use-case xem quản lý thực tập của sinh viên

* Mục đích: quản lý các đề tài đã đăng ký của sinh viên, quản lý đợt thực tập của sinh viên, quản lý thực tập của sinh viên
* Tác nhân/ đối tượng sử dụng: admin
* Mô tả chung: admin sẽ có quyền trong quản lý các đề tài đã đăng ký của sinh viên, quản lý đợt thực tập của sinh viên, quản lý thực tập của sinh viên trong hệ thống.
* Luồng sự kiện chính trường hợp admin muốn xem toàn bộ danh sách đề tài đăng ký của sinh viên:
  + - Admin nhấn nút quản lý đề tài sinh viên
    - Hệ thống xử lý lấy ra danh sách sinh viên và đề tài của sinh viên đã đăng ký và đã được duyệt thành công từ giảng viên hướng dẫn
    - Hệ thống hiển thị danh sách đề tài sinh viên đăng ký
* Luồng sự kiện chính trường hợp admin muốn thay đổi trạng thái đăng ký của sinh viên:
  + - Admin nhấn nút thay đổi trạng thái
    - Hệ thống hiện thông báo cảnh báo
    - Admin xác nhận hành động
    - Hệ thống xử lý, cập nhật và ghi lịch sử
    - Hệ thống hiển thị danh sách đề tài sinh viên đăng ký
* Luồng sự kiện chính trường hợp admin quản lý kỳ thực tập:
  + - Admin nhấn nút quản lý kỳ thực tập
    - Hệ thống hiện thông báo cảnh báo
    - Admin xác nhận hành động
    - Hệ thống xử lý, cập nhật và ghi lịch sử
    - Hệ thống hiển thị danh sách đề tài sinh viên đăng ký
* Luồng sự kiện chính trường hợp admin quản lý sinh sinh viên thực tập:
  + - Admin nhấn nút quản lý sinh viên thực tập
    - Hệ thống xử lý và lấy ra danh sách sinh viên thực tập
    - Hệ thống hiển thị cho admin
    - Trường hợp admin muốn chuyển trạng thái
      * Admin nhấn nút chuyển trạng thái
      * Hệ thống hiển thị thông báo cảnh báo
      * Admin xác nhận hành động
      * Hệ thống xử lý, cập nhật và ghi lịch sử
      * Hệ thống hiển thị và thông báo cho admin
* Luồng thay thế: hủy các hành động nếu không thành công
* Yêu cầu cụ thể:
* Điều kiện trước: admin đã đăng nhập vào hệ thống
* Điều kiện sau: hiện thị thông báo kết thúc hành động.
  + - 1. Use-case email

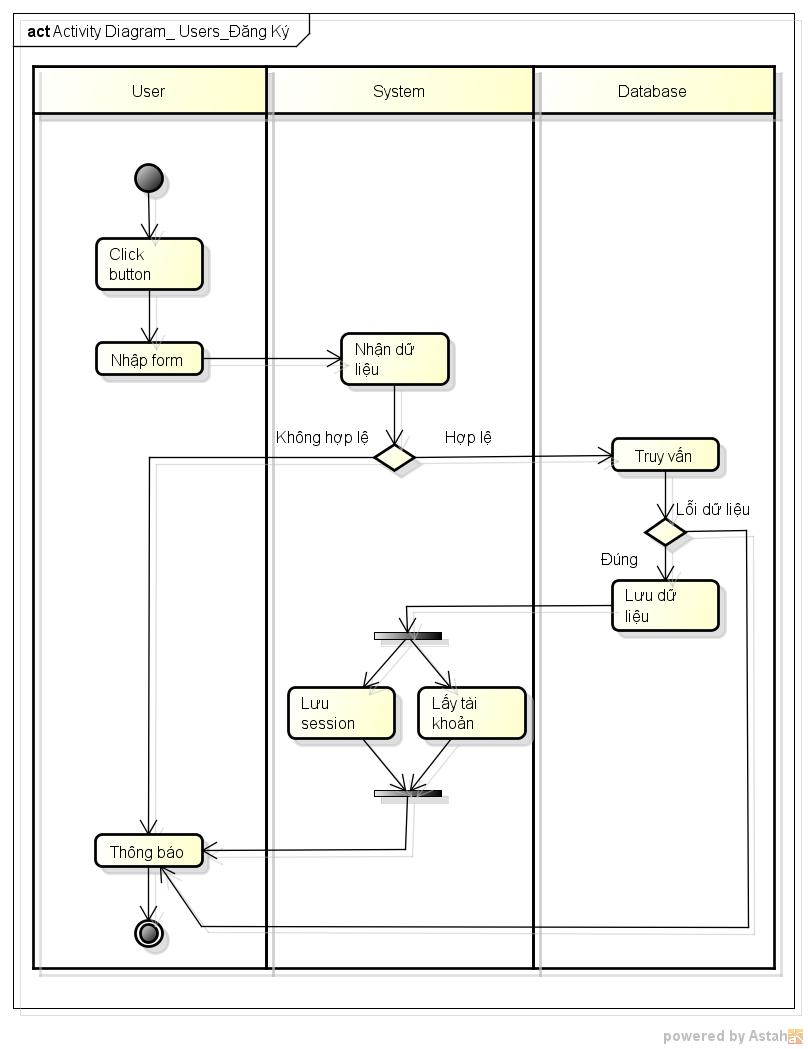
Đặc tả use-case email

* Mục đích: trao đổi thông tin, các vướng mắc trong quá trình thực tập hoặc muốn phản hồi cho người dùng, hệ thống.
* Tác nhân/ đối tượng sử dụng: tất cả người dùng
* Mô tả chung: Khi người dùng muốn trao đổi thông tin cho nhau thì cần sử dụng chức năng email trên hệ thống.
* Luồng sự kiện chính:
  + - Người dùng nhấn vào nút email
    - Hệ thống lấy toàn bộ thông tin về email, bao gồm các email đã gửi, đã đọc, chưa đọc, đã nhận,…
    - Hệ thống hiển thị thông tin cho người dùng.
* Luồng thay thế:
  + - Gửi mail: trường hợp người dùng muốn gửi mail tới các người dùng khác trong hệ thống
    - Xem mail: người dùng muốn xem mail
    - Đọc mail: người dùng muốn đọc mail
    - Xóa mail: người dùng muốn xóa mail
* Yêu cầu cụ thể: không có
* Điều kiện trước: tài khoản đã được xác thực
* Điều kiện sau: hiển thị toàn bộ thông tin mail cho người dùng
  + - 1. Use-case quản lý thông báo

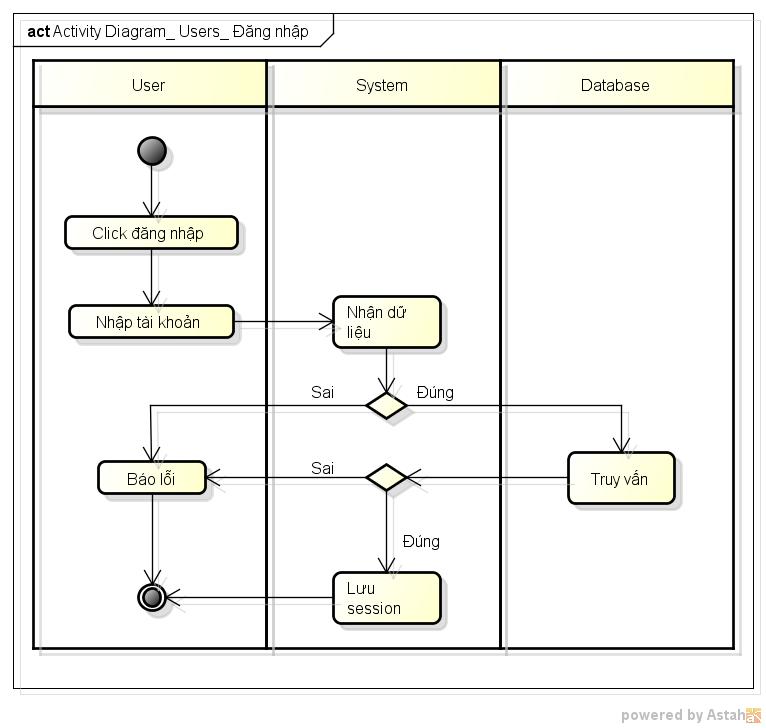
Đặc tả use-case xem danh sách công ty

* Mục đích: admin quản lý các thông báo trên trang chủ
* Tác nhân/ đối tượng sử dụng: admin
* Mô tả chung: admin sẽ có quyền đăng, sửa hoặc xóa các thông báo trên hệ thống. Các thông báo này sẽ được hiển thị trên trang chủ của hệ thống.
* Luồng sự kiện chính thêm mới thông báo:
  + - Admin nhấn nút thêm mới thông báo
    - Hệ thống hiện thị giao diện biểu mẫu
    - Admin nhập dữ liệu
    - Hệ thống bắt lỗi nhập liệu
    - Admin nhấn nút thêm mới
    - Hệ thống hiện thông báo cảnh báo
    - Admin xác thực hành động
    - Hệ thống xử lý, cập nhật dữ liệu và ghi lịch sử
    - Hệ thống thông báo kết quả hành động
* Luồng thay thế: hệ thống yêu cầu nhập lại nếu có lỗi.
* Luồng sự kiện chính chỉnh sửa thông báo:
  + - Admin nhấn nút chỉnh sửa thông báo
    - Hệ thống hiện thị giao diện biểu mẫu
    - Admin nhập dữ liệu
    - Hệ thống bắt lỗi nhập liệu
    - Admin nhấn chỉnh sửa
    - Hệ thống hiện thông báo cảnh báo
    - Admin xác thực hành động
    - Hệ thống xử lý, cập nhật dữ liệu và ghi lịch sử
    - Hệ thống thông báo kết quả hành động
* Luồng thay thế: hệ thống yêu cầu nhập lại nếu có lỗi.
* Luồng sự kiện chính xóa thông báo:
  + - Admin nhấn nút xóa thông báo
    - Hệ thống hiện thông báo cảnh báo
    - Admin xác thực hành động
    - Hệ thống xử lý, cập nhật dữ liệu và ghi lịch sử
    - Hệ thống thông báo kết quả hành động
* Luồng thay thế: không có
* Yêu cầu cụ thể:
* Điều kiện trước: admin đăng nhập vào hệ thống
* Điều kiện sau: hiện thị thông báo kết quả hành động và lấy danh sách thông báo trong hệ thống.
  + 1. Biểu đồ use-case hệ thống

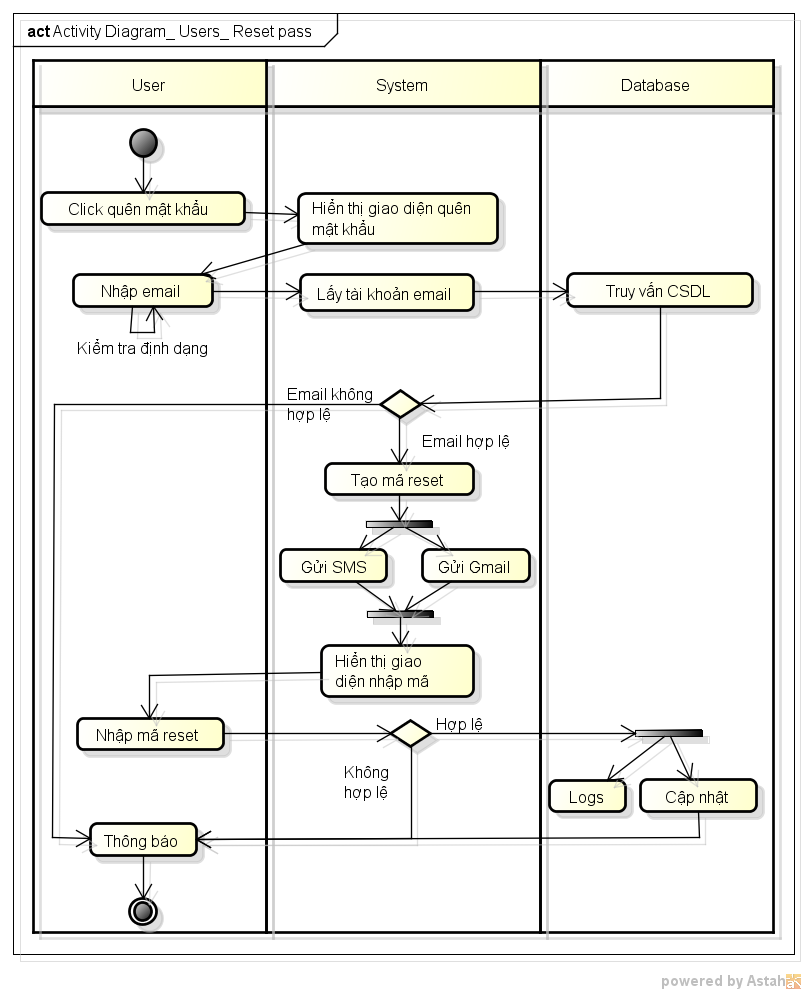
1. Use-case xác thực email
2. Use-case logs
3. Use-case xác thực tin nhắn
4. Use-case so khớp tự động
5. Use-case nhắc nhở thực tập
6. Biểu đồ Hoạt động
7. Biểu đồ hoạt động chung cho các tác nhân
   * + 1. Biểu đồ hoạt động chức năng đăng ký tài khoản



* + - 1. Biểu đồ hoạt động chức năng đăng nhập

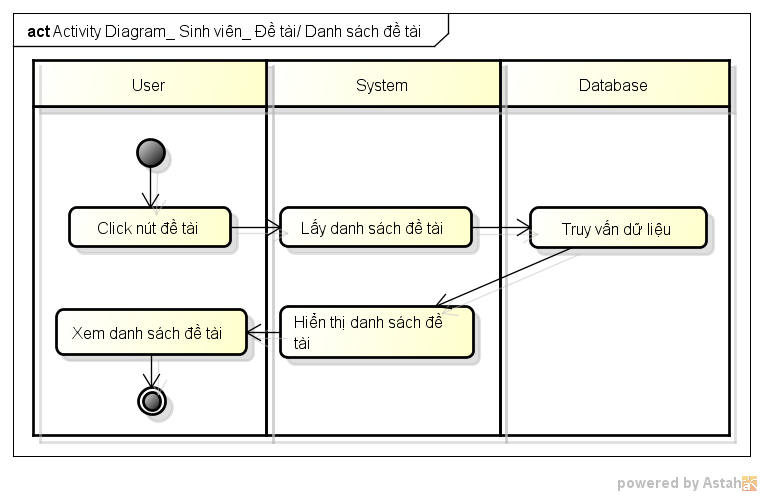


* + - 1. Biểu đồ hoạt động chức năng quên mật khẩu

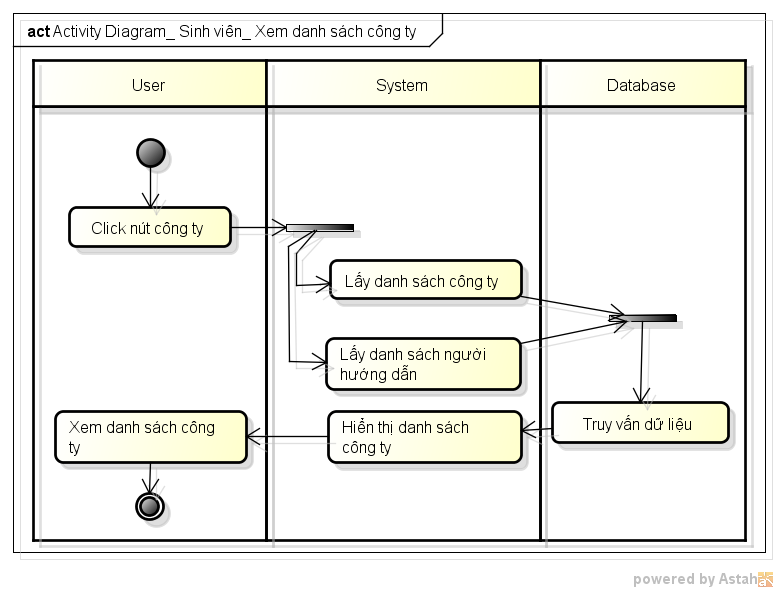


* + - 1. Biểu đồ hoạt động chức năng mail/ phản hồi
      2. Biểu đồ hoạt động chức năng xem danh sách sinh viên
      3. Biểu đồ hoạt động chức năng xem danh sách đề tài
      4. Biểu đồ hoạt động chức năng tải/ xóa file

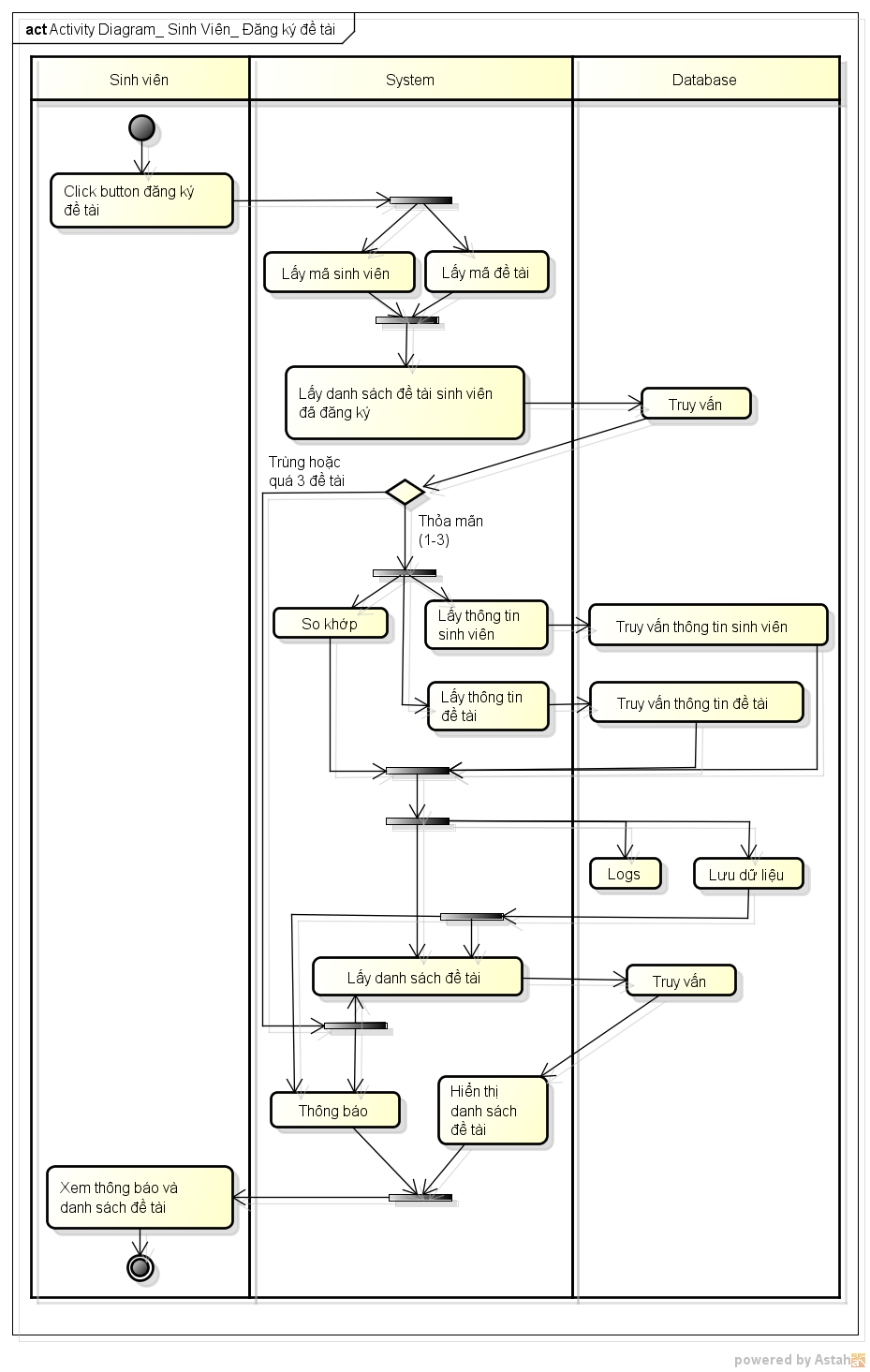
1. Biểu đồ hoạt động sinh viên
   * + 1. Biểu đồ hoạt động chức năng xem đề tài/ danh sách đề tài



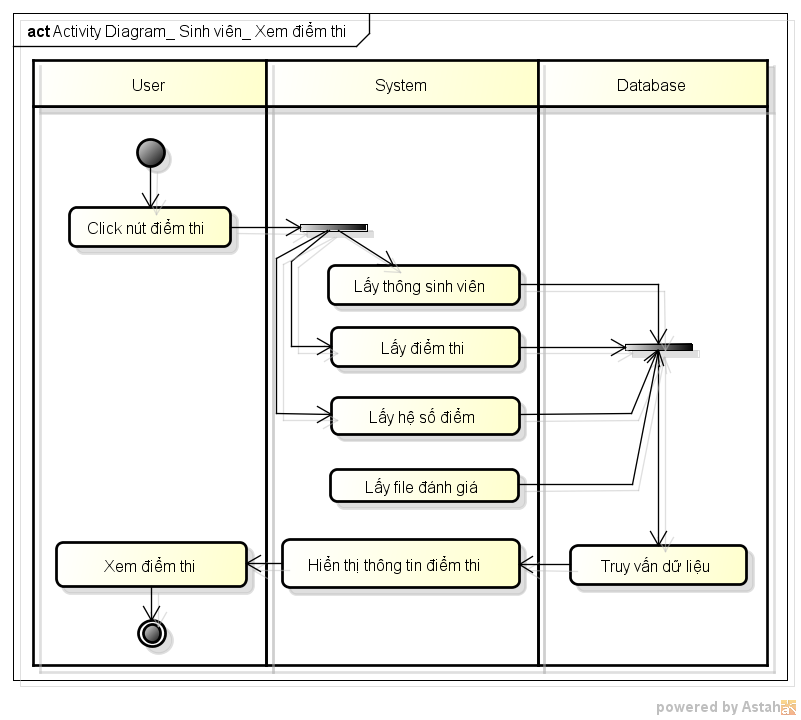
* + - 1. Biểu đồ hoạt động chức năng xem công ty/ danh sách công ty



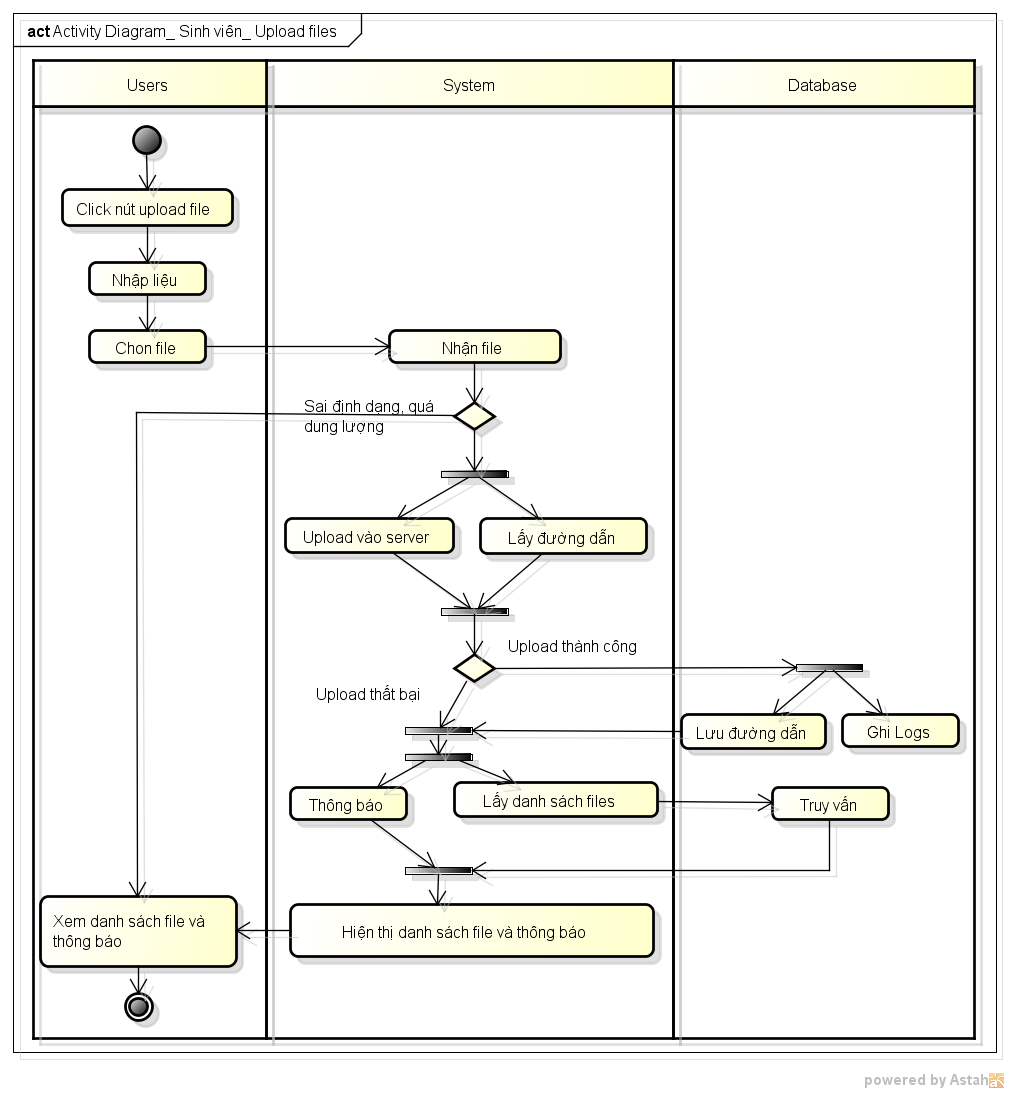
* + - 1. Biểu đồ hoạt động chức năng đăng ký đề tài



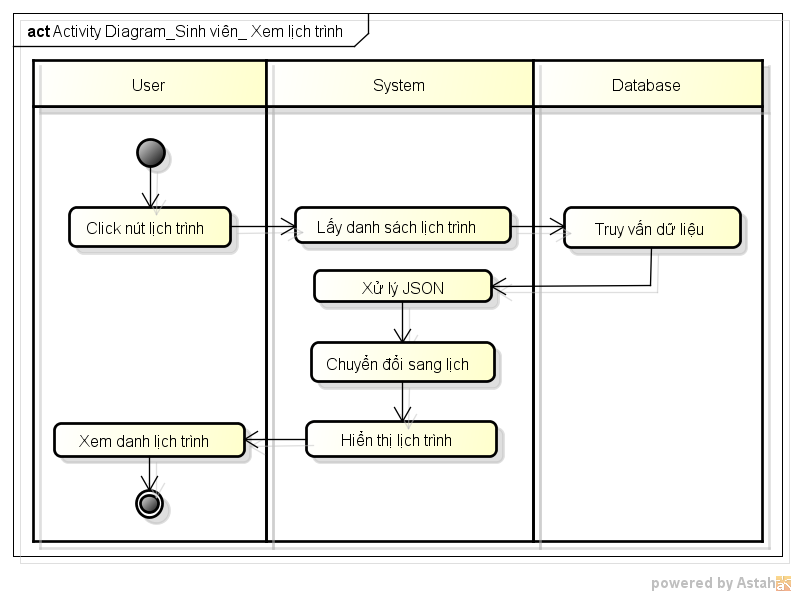
* + - 1. Biểu đồ hoạt động chức năng xem điểm thi



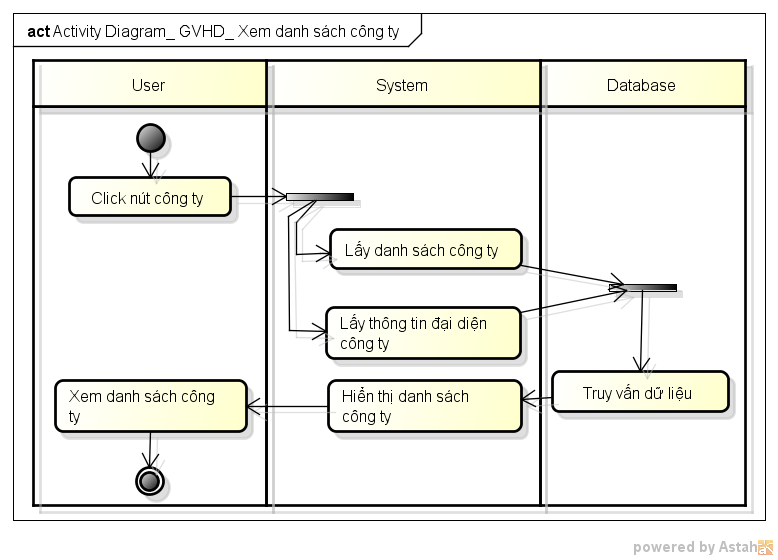
* + - 1. Biều đồ hoạt động chức năng upload file



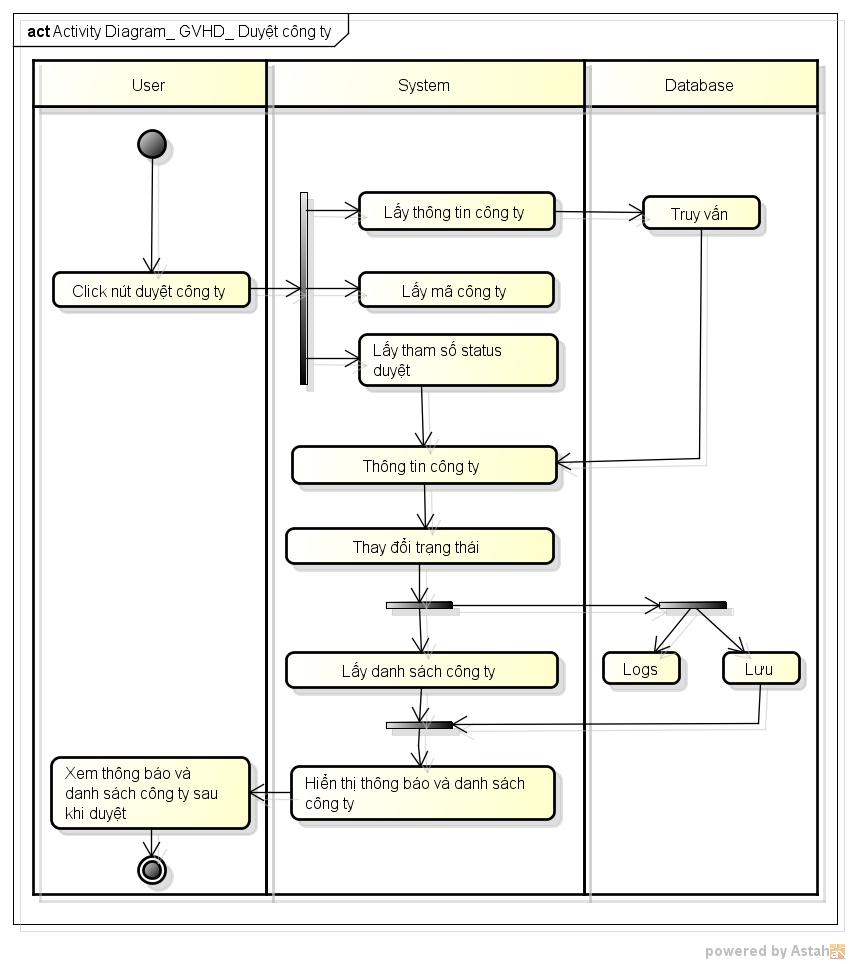
* + - 1. Biểu đồ hoạt động chức năng xem lịch trình



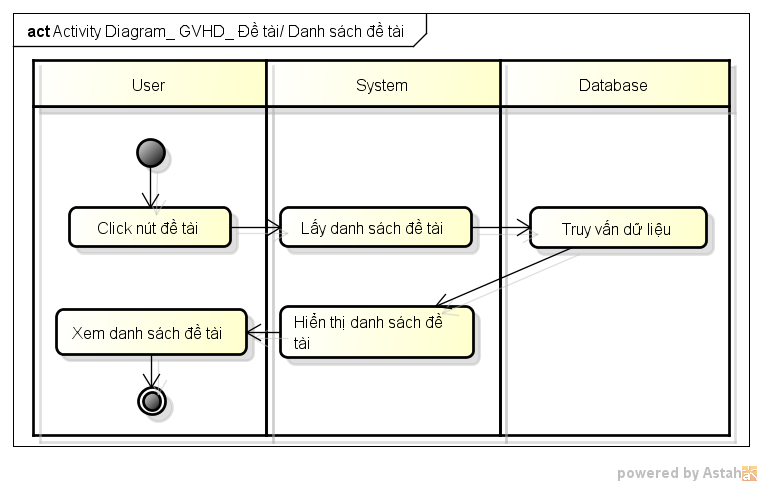
1. Biểu đồ hoạt động giảng viên hướng dẫn
2. Biểu đồ hoạt động chức năng xem danh sách công ty



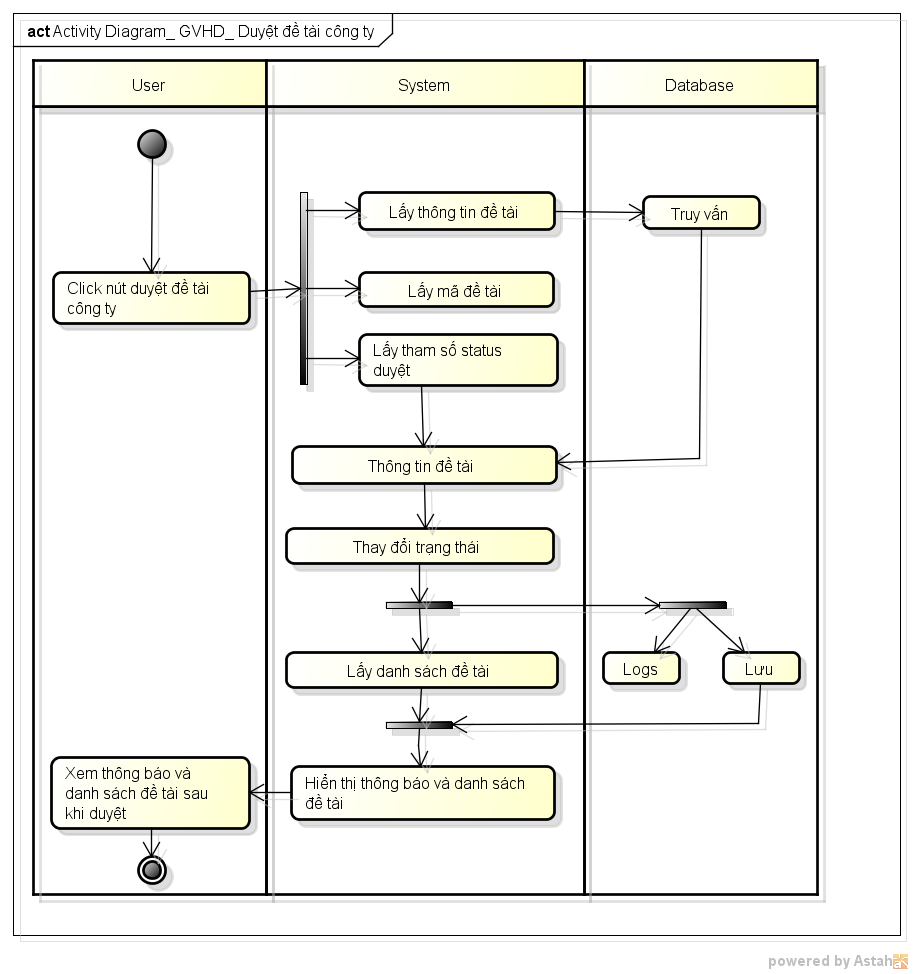
1. Biểu đồ hoạt động chức năng duyệt công ty



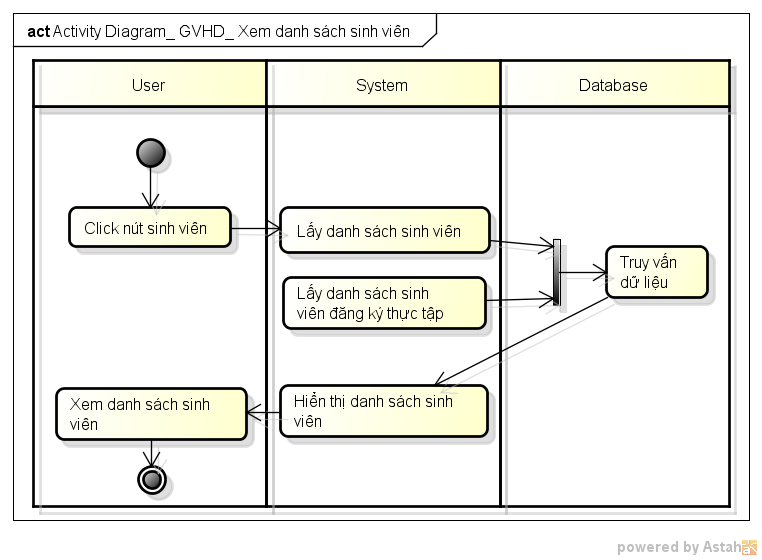
1. Biểu đồ hoạt động chức năng xem danh sách đề tài



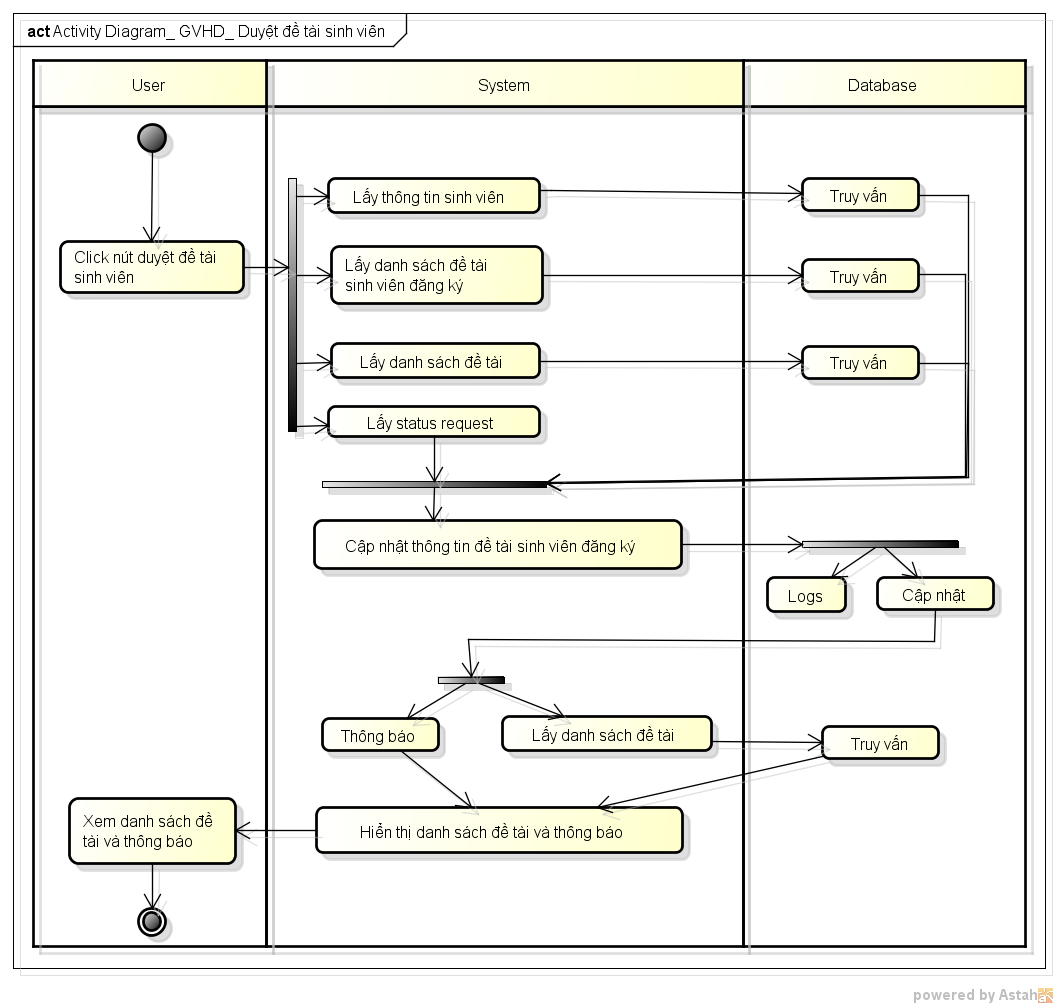
1. Biểu đồ hoạt động chức năng duyệt đề tài công ty



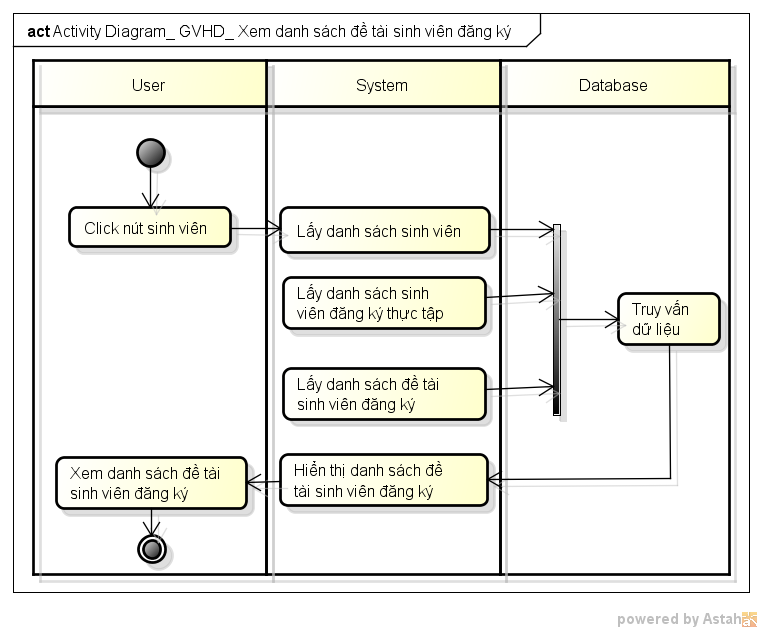
1. Biểu đồ hoạt động chức năng xem danh sách sinh viên



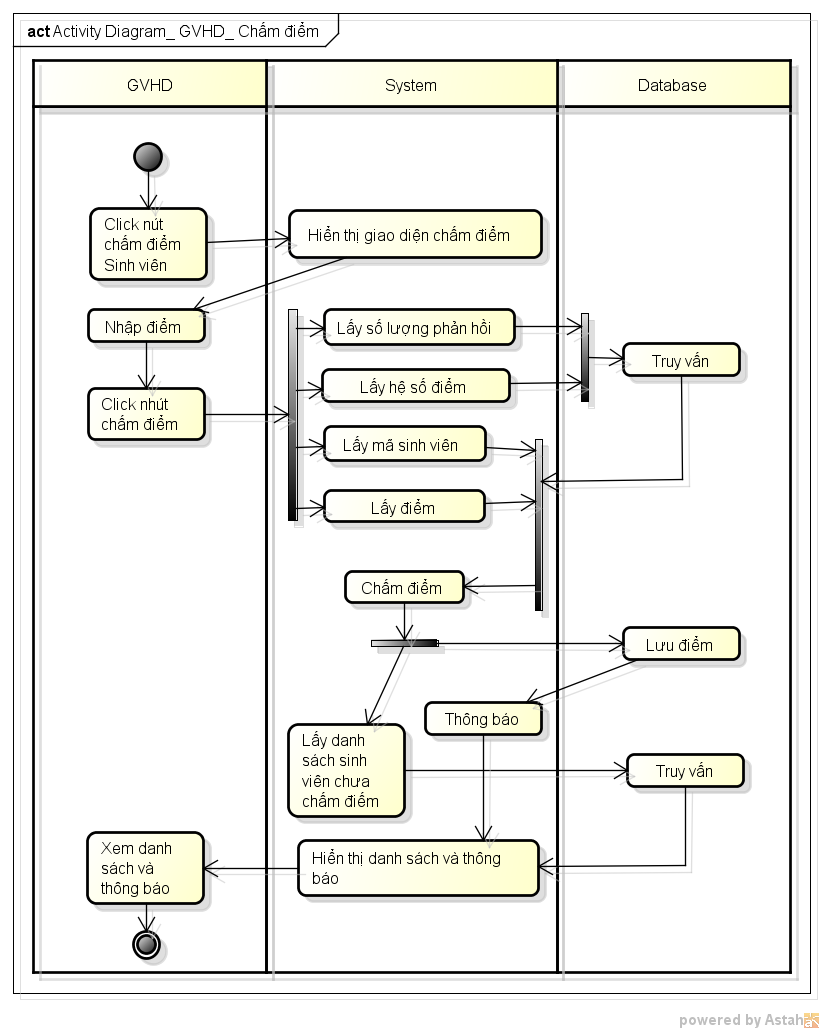
1. Biểu đồ hoạt động chức năng duyệt đề tài sinh viên



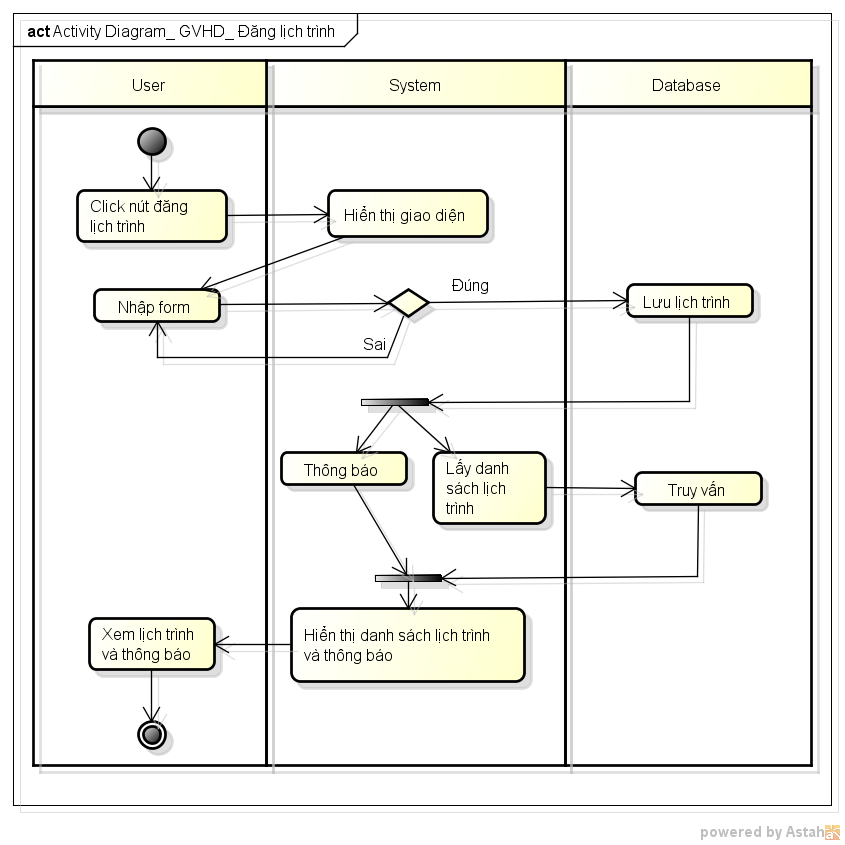
1. Biểu đồ hoạt động chức năng xem đề tài sinh viên đăng ký



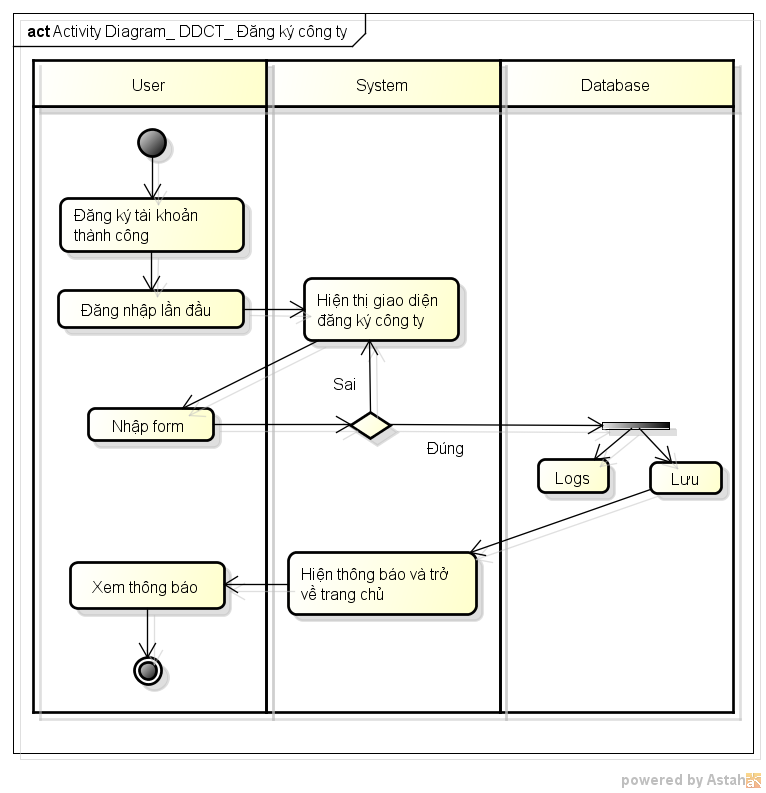
1. Biểu đồ hoạt động chức năng chấm điểm



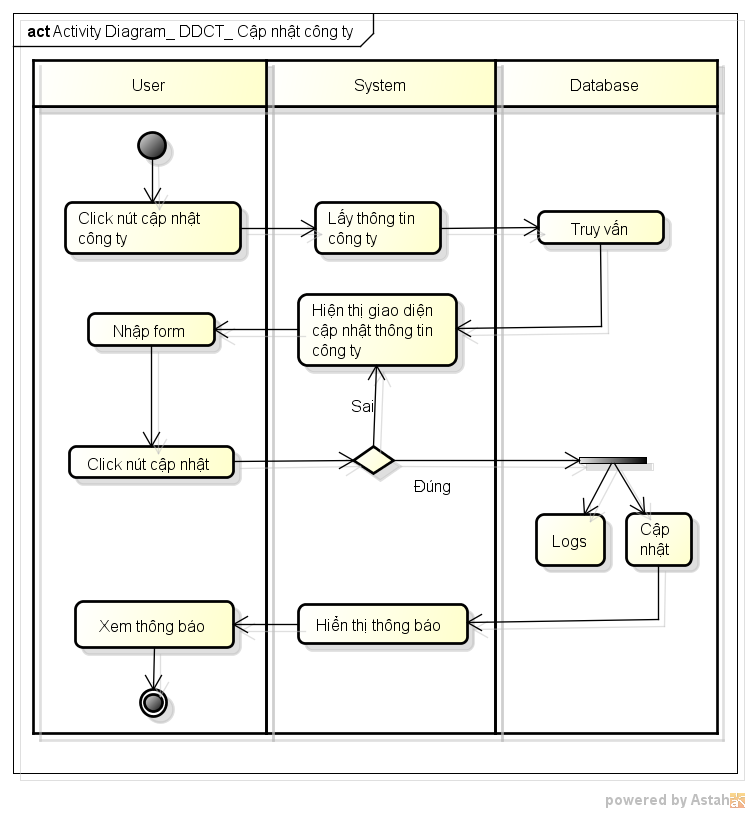
1. Biểu đồ hoạt động chức năng đăng lịch trình/ thông báo



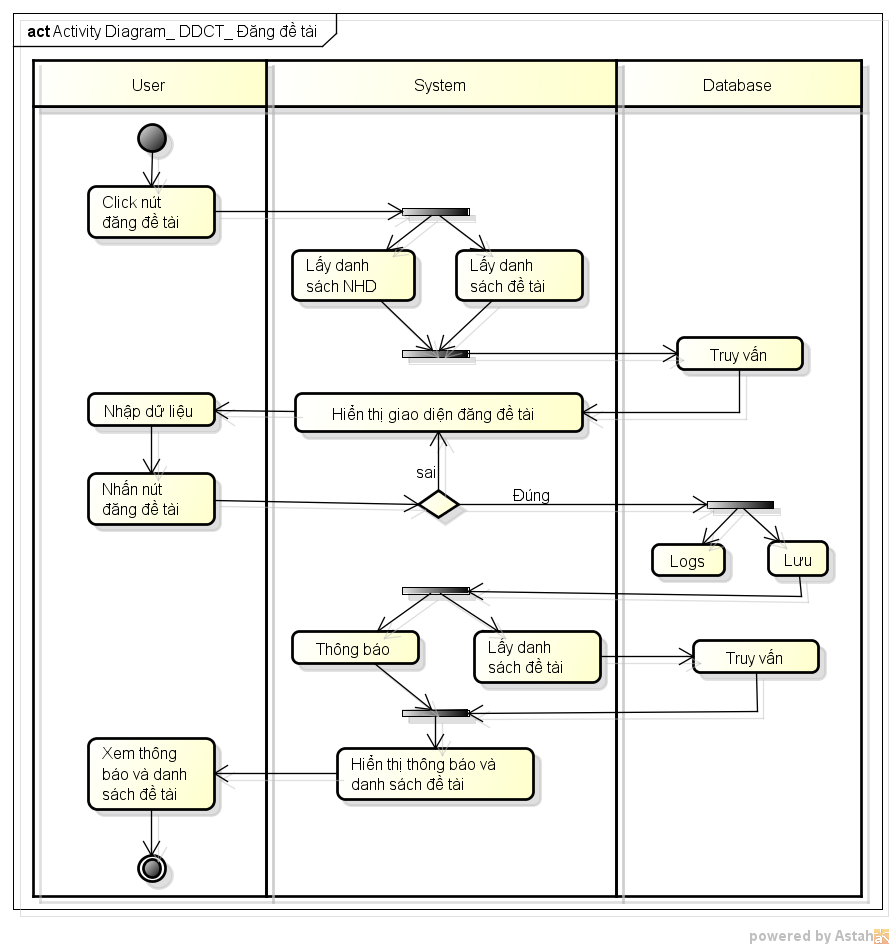
1. Biểu đồ hoạt động chức năng duyệt báo cáo sinh viên
2. Biểu đồ hoạt động người hướng dẫn
3. Biều đồ hoạt động chức năng đánh giá sinh viên
4. Biểu đồ hoạt động chức năng
5. Biểu đồ hoạt động đại diện công ty
6. Biểu đồ hoạt động chức năng đăng ký công ty



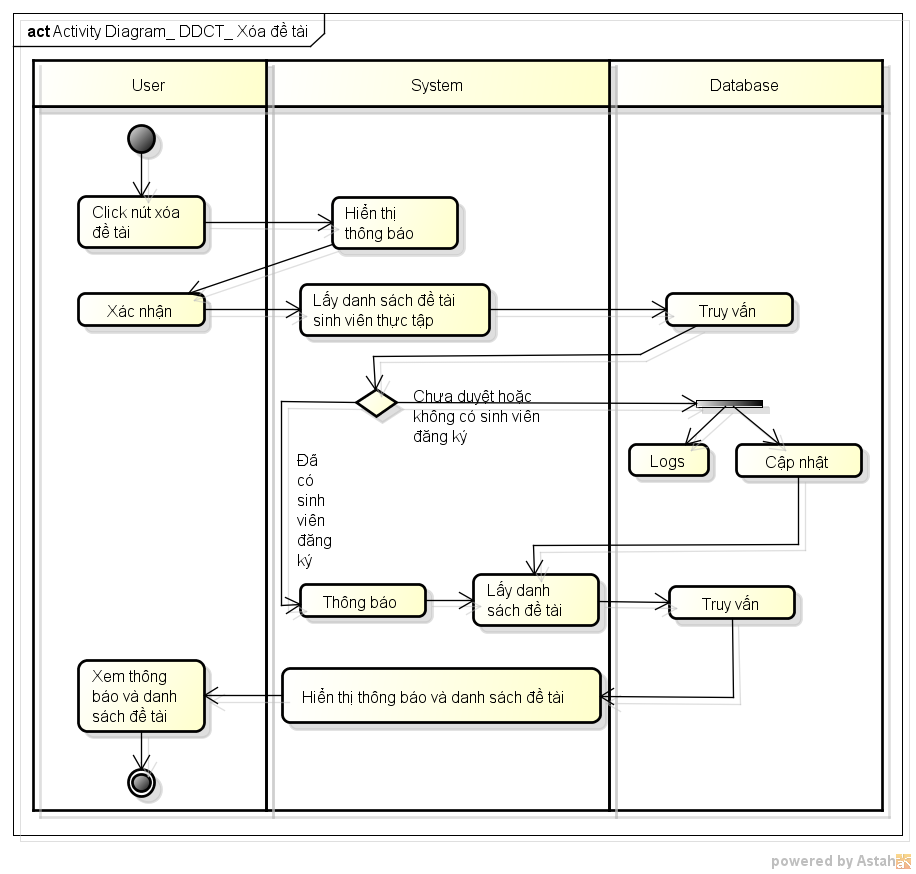
1. Biểu đồ hoạt động chức năng cập nhật công ty



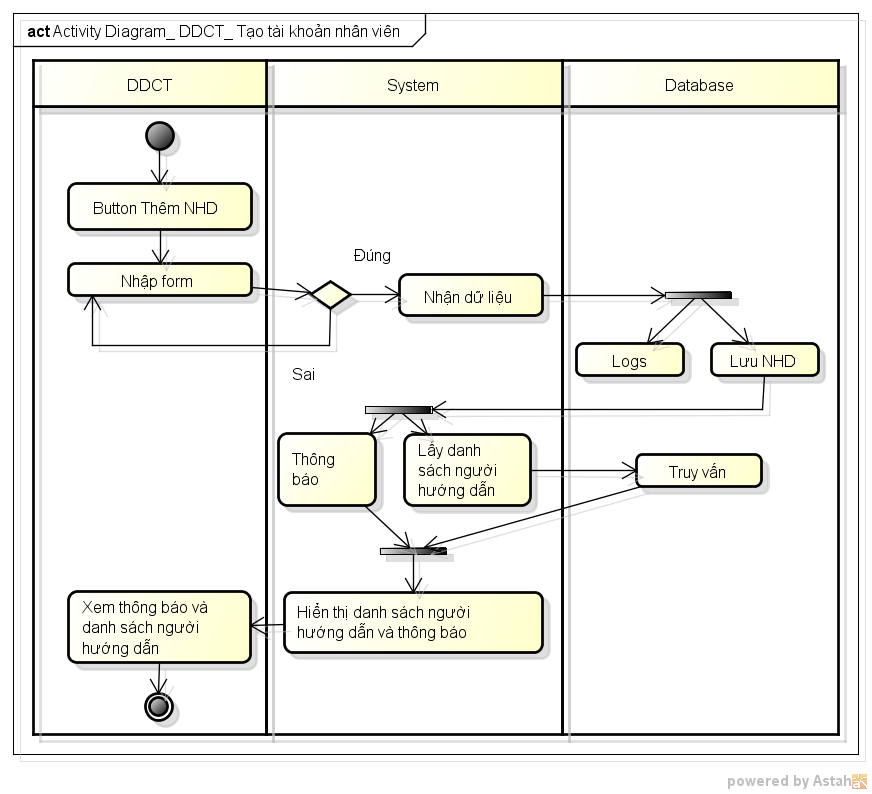
1. Biểu đồ hoạt động chức năng đăng đề tài



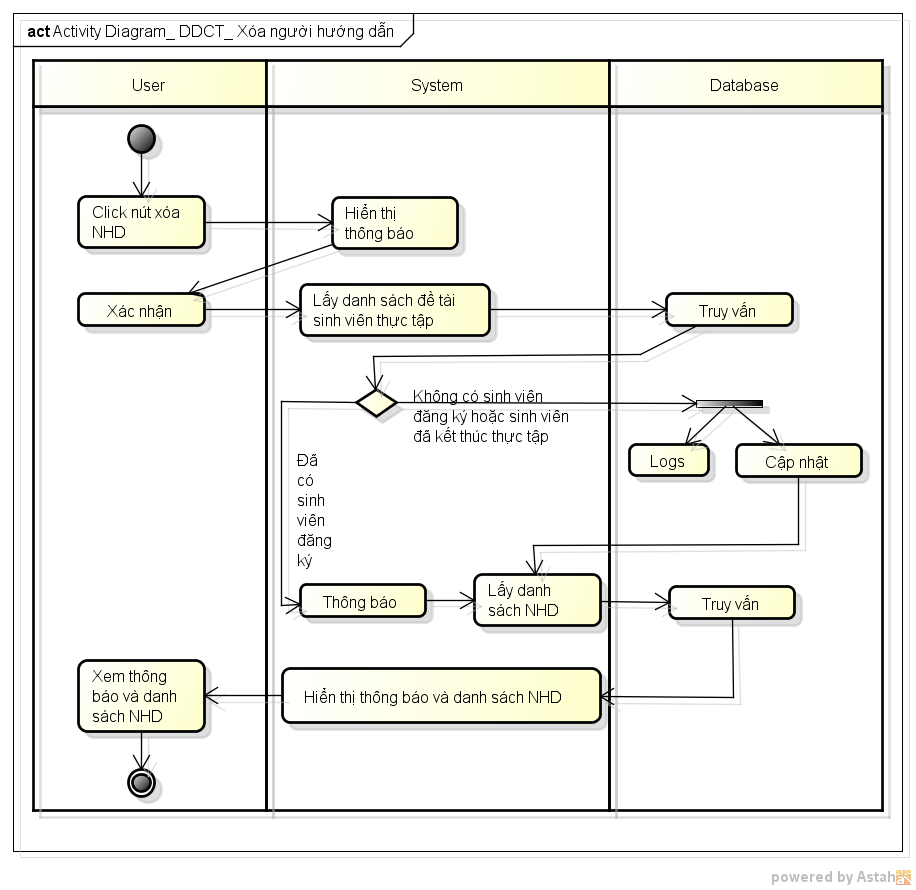
1. Biểu đồ hoạt động chức năng xóa đề tài



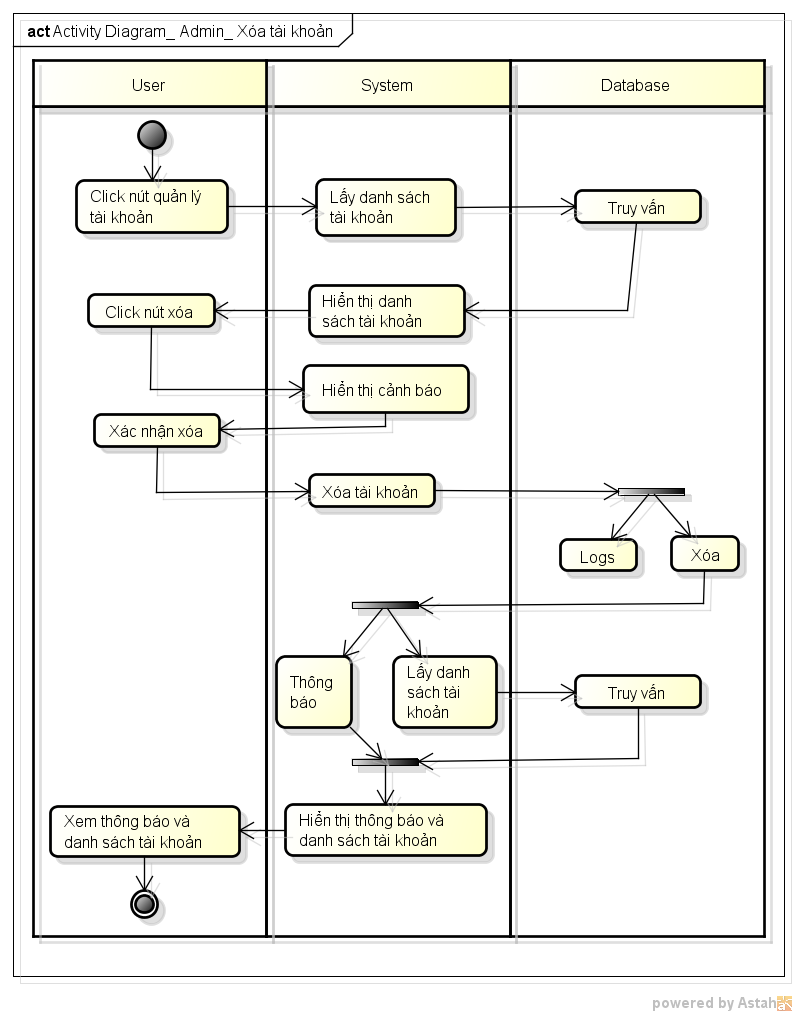
1. Biều đồ hoạt động chức năng thêm mới người hướng dẫn



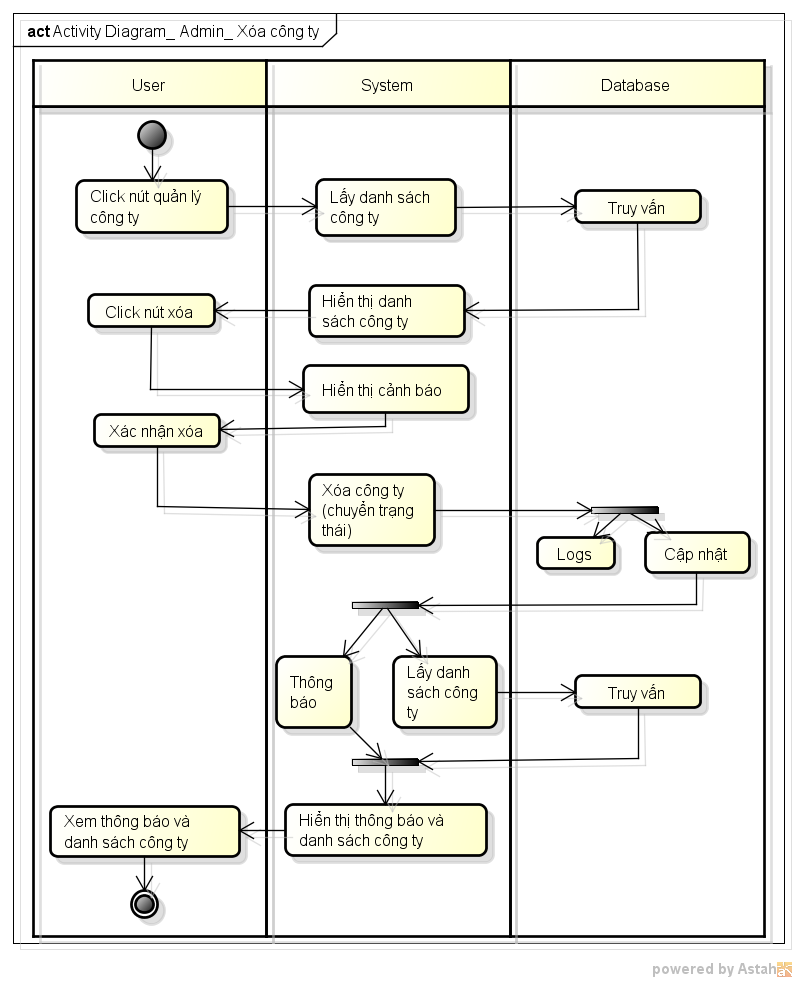
1. Biểu đồ hoạt động chức năng xóa người hướng dẫn



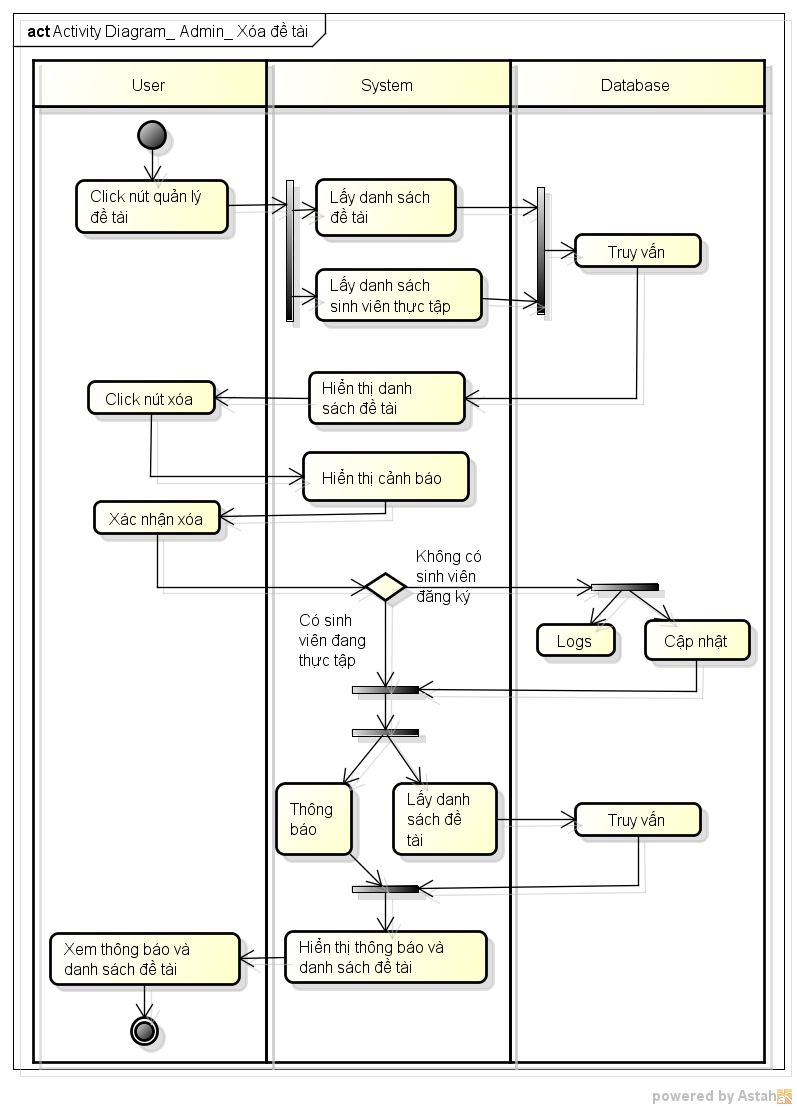
1. Biểu đồ hoạt động admin
2. Biểu đồ hoạt động quản lý tài khoản (chức năng xóa tài khoản)



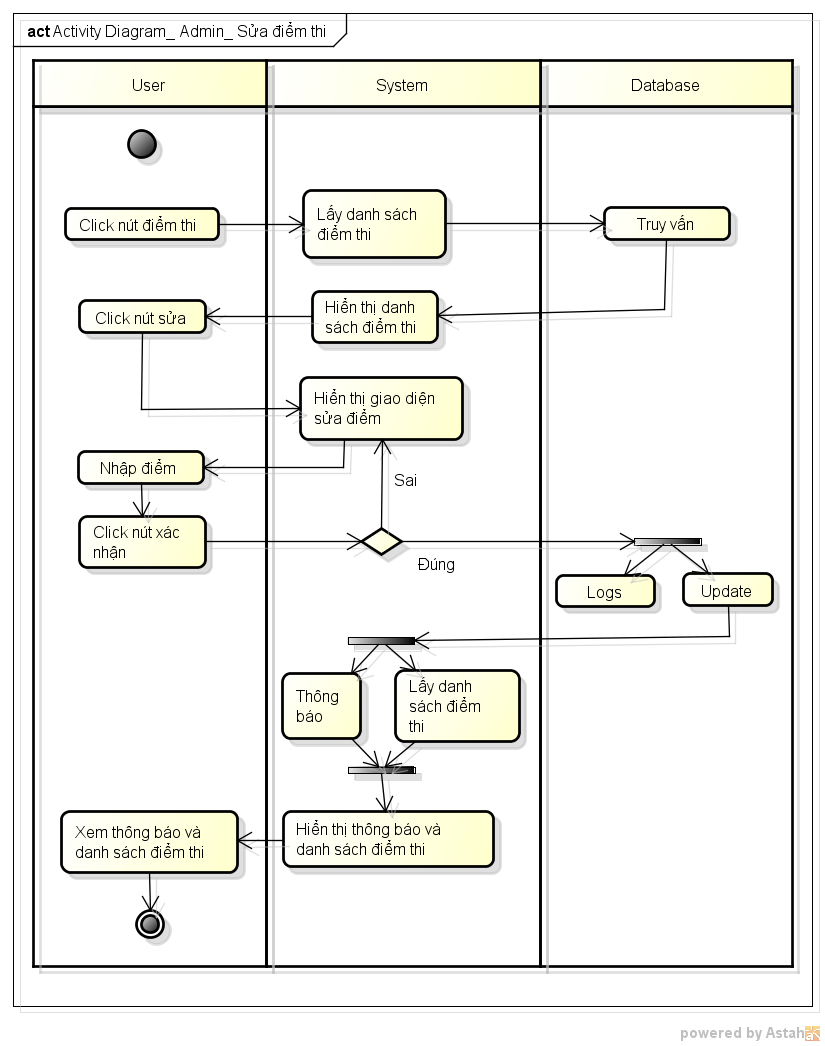
1. Biểu đồ hoạt động quản lý công ty (chức năng xóa công ty)



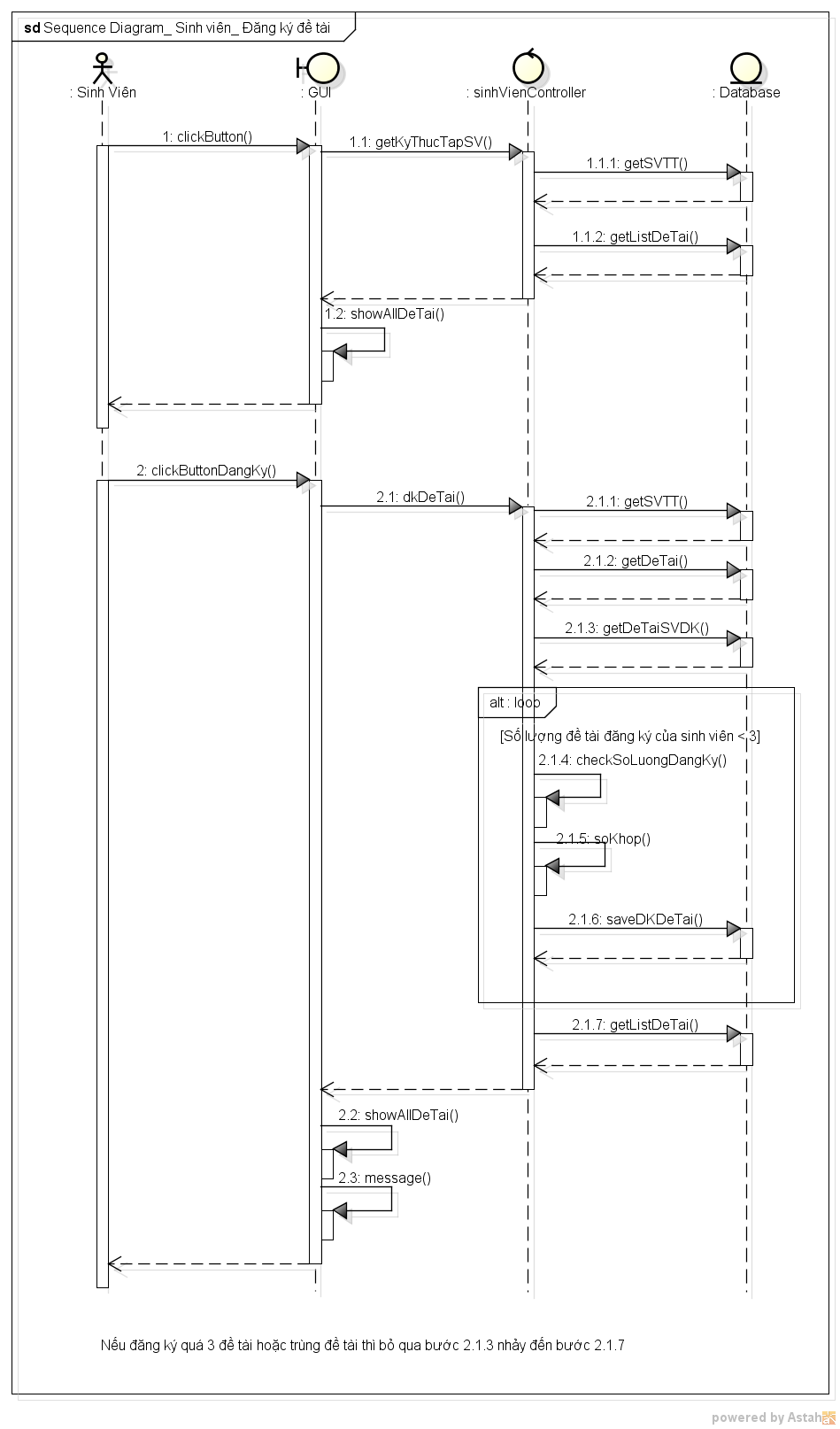
1. Biểu đồ hoạt động quản lý đề tài (chức năng xóa đề tài)



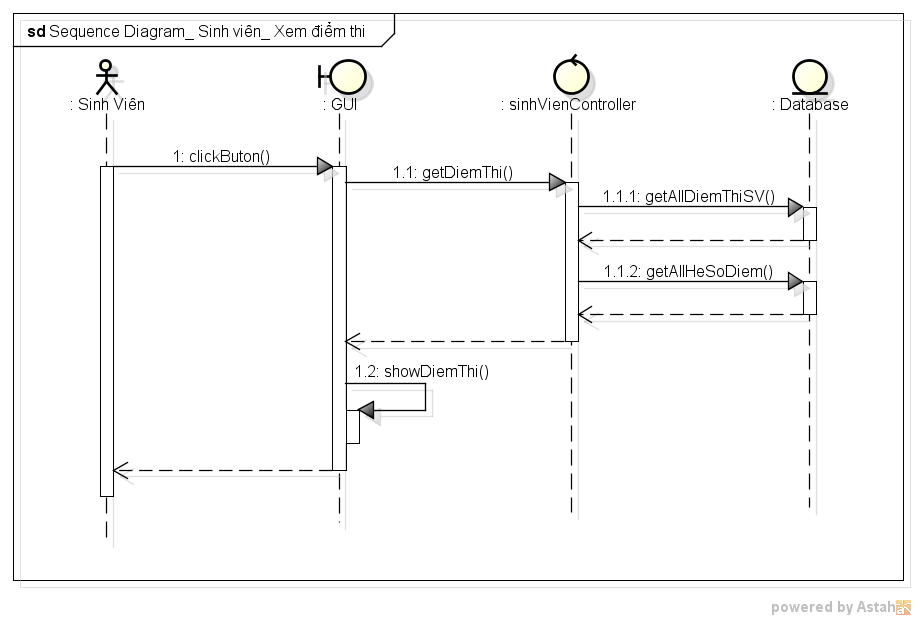
1. Biểu đồ hoạt động quản lý điểm (sửa điểm thi)



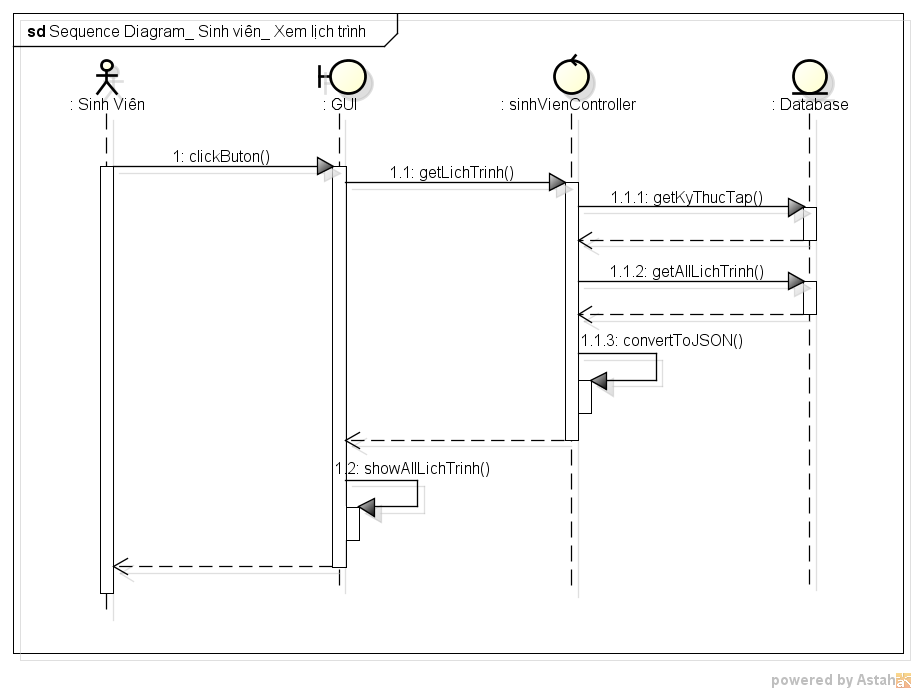
1. Biều đồ hoạt động quản lý thực tập
2. Biểu đồ hoạt động quản lý thông báo
3. Biểu đồ hoạt động hệ thống
4. Biểu đồ hoạt động chức năng gửi Gmail
5. Biểu đồ hoạt động chức năng gửi eSMS
6. Biểu đồ hoạt động chức năng gửi thông báo đến hạn phản hổi
7. Biểu đồ tuần tự
   * 1. Biểu đồ tuần tự cho các chức năng của sinh viên
        1. Biểu đồ tuần tự chức năng đăng ký đề tài



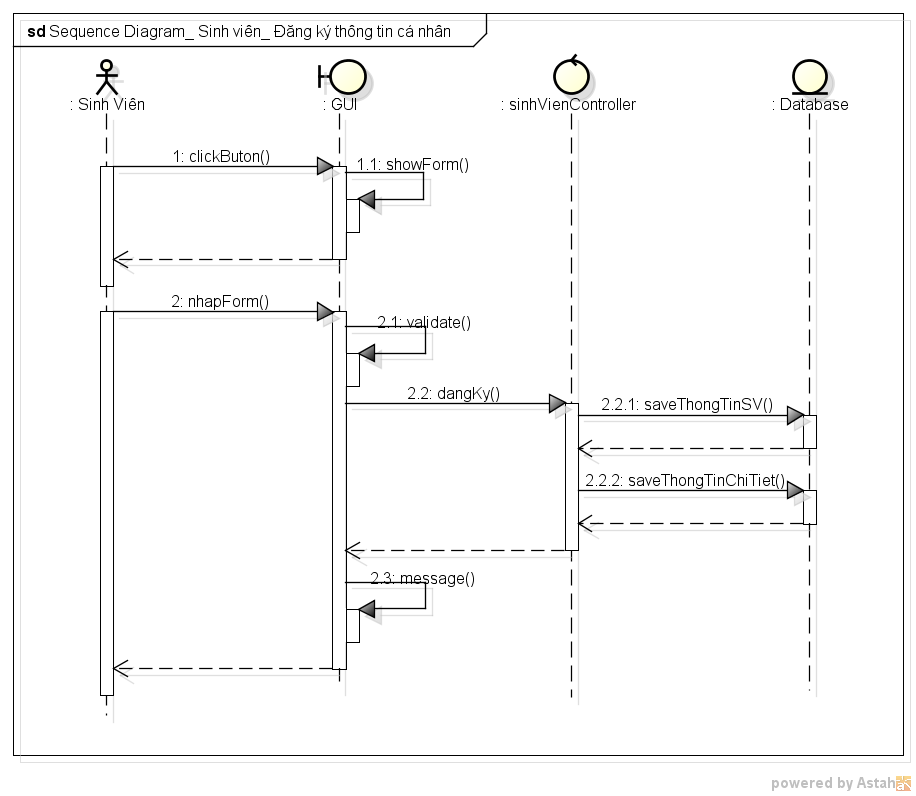
* + - 1. Biểu đồ tuần tự chức năng xem điểm thi



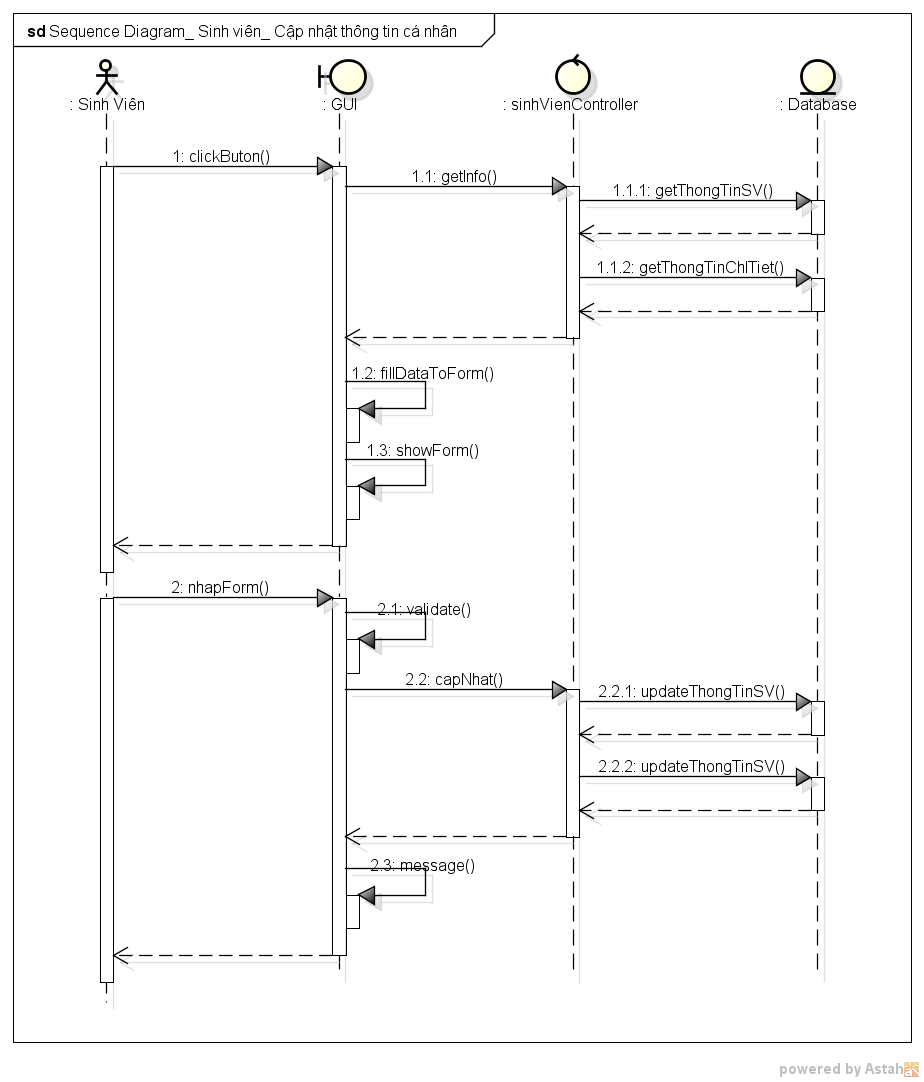
* + - 1. Biểu đồ tuần tự chức năng xem lịch trình



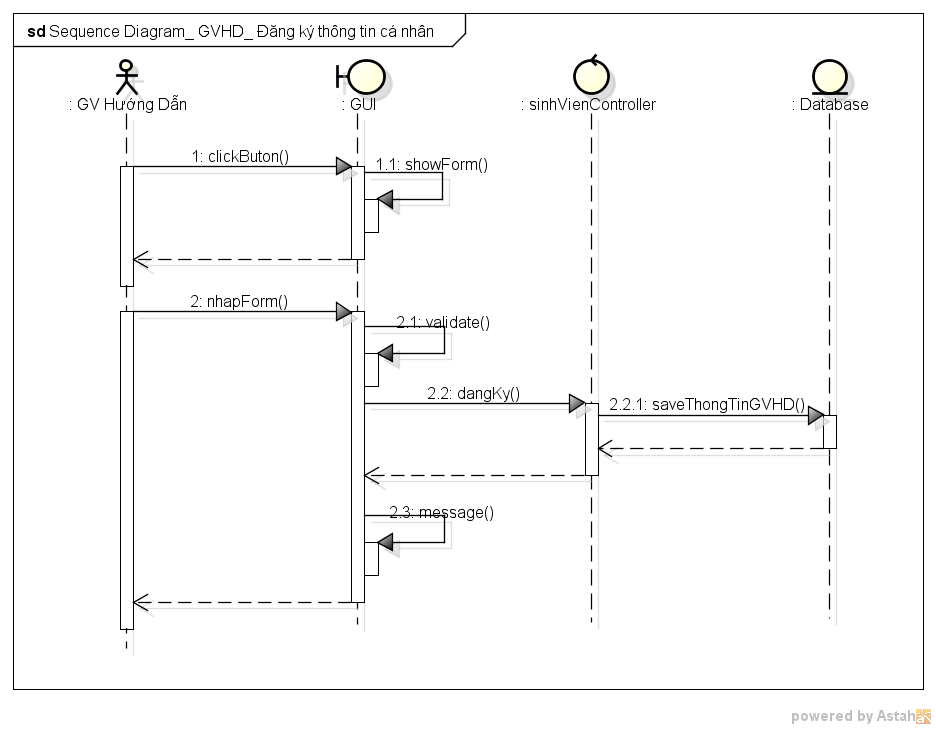
* + - 1. Biểu đồ tuần tự đăng ký thông tin cá nhân



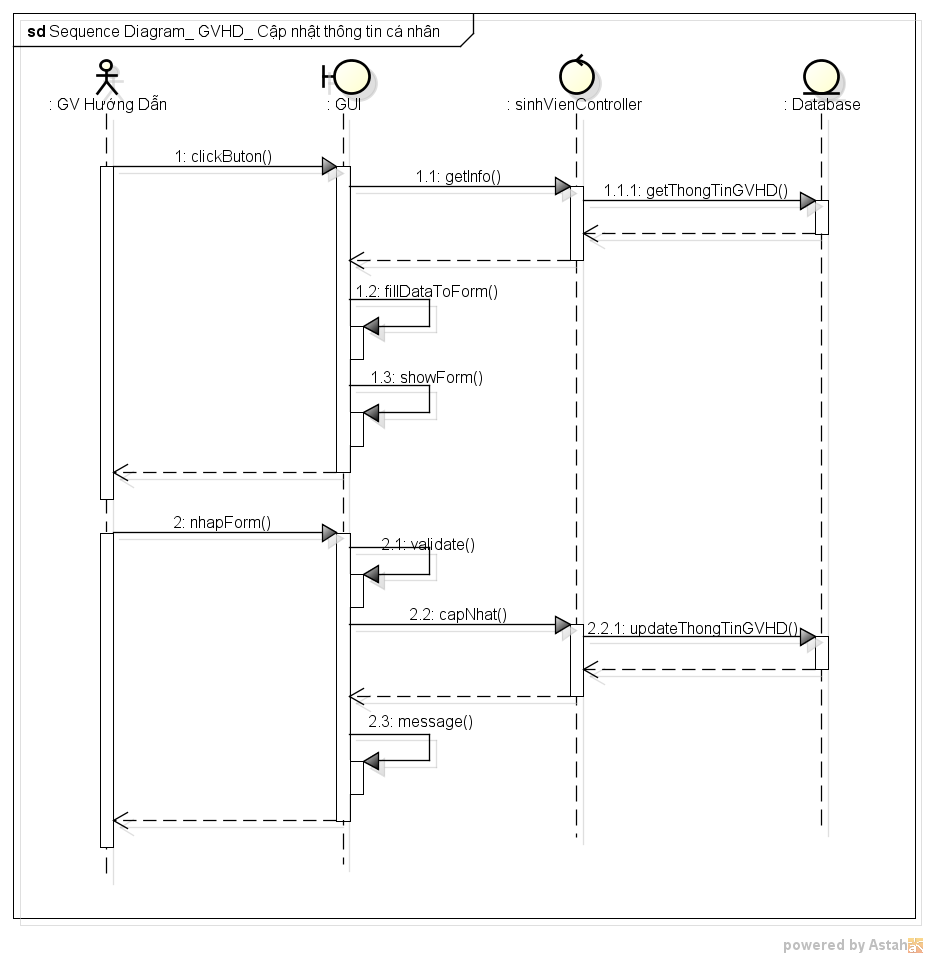
* + - 1. Cập nhật thông tin cá nhân



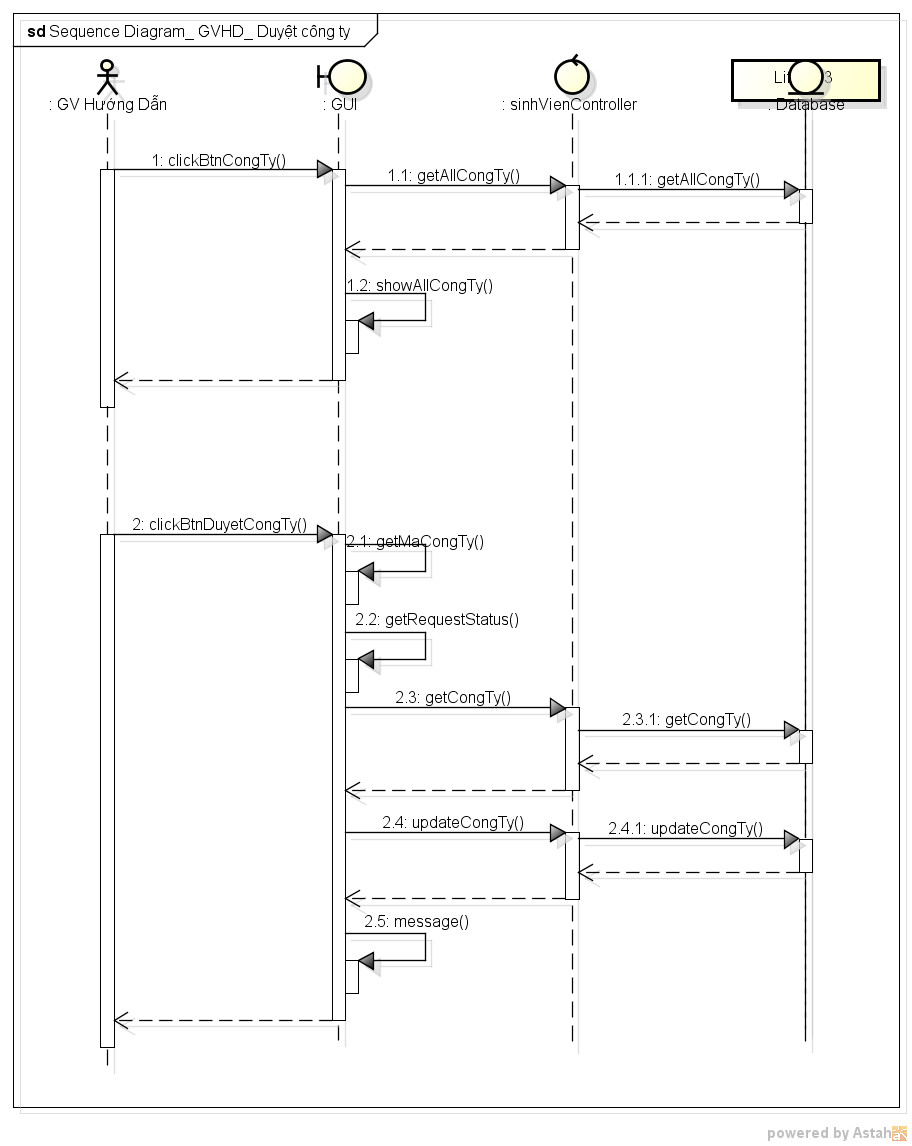
* + - 1. S
    1. Biểu đồ tuần tự giảng viên hướng dẫn
       1. Biểu đồ tuần tự chức năng đăng ký thông tin cá nhân



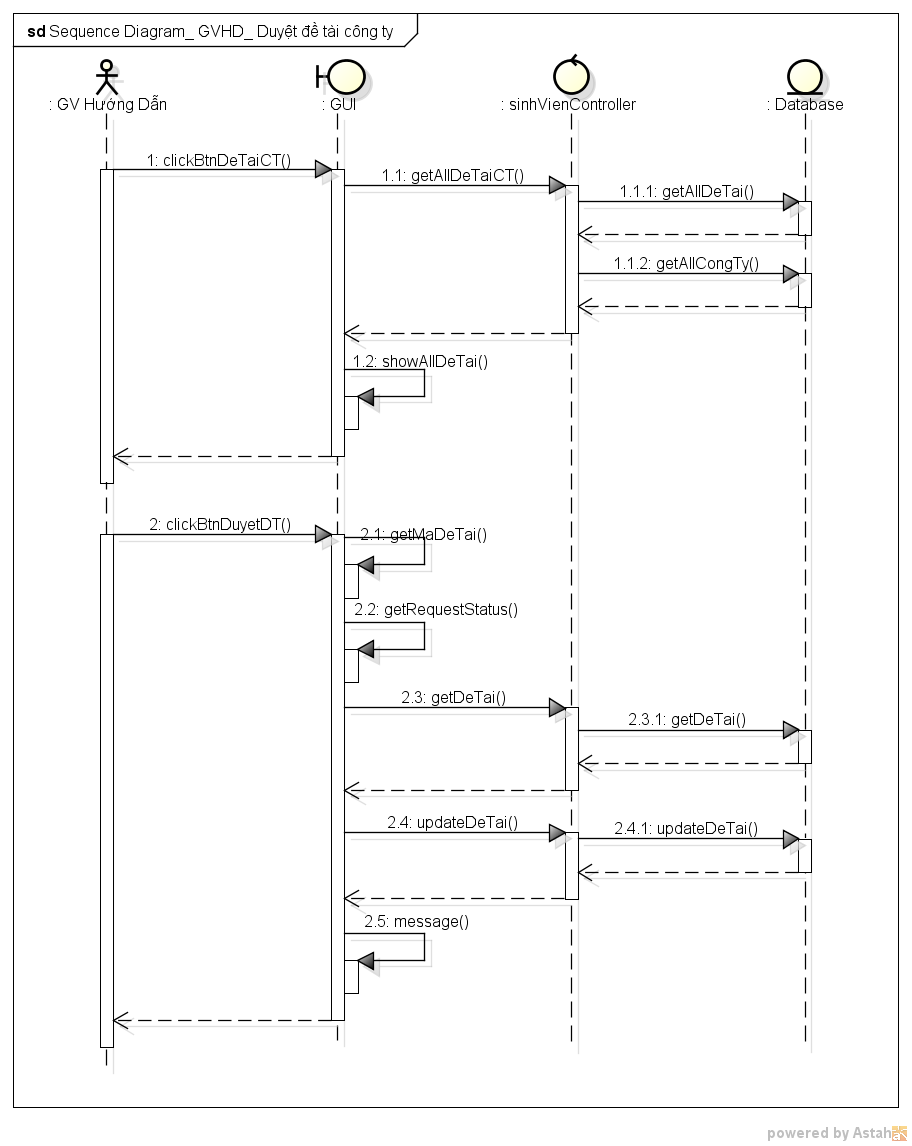
* + - 1. Biểu đồ tuần tự chức năng cập nhật thông tin cá nhân



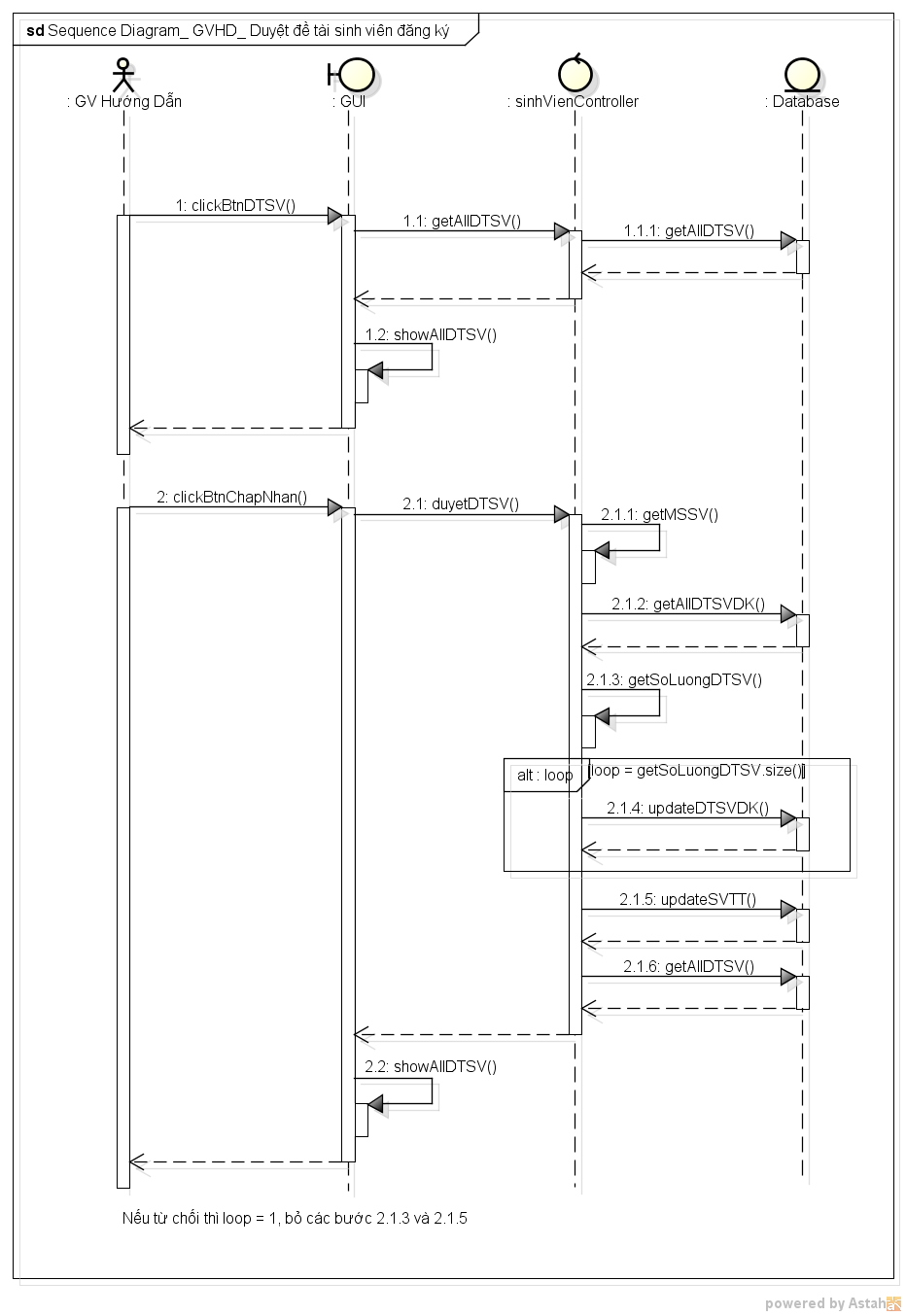
* + - 1. Biểu đồ tuần tự chức năng duyệt công ty



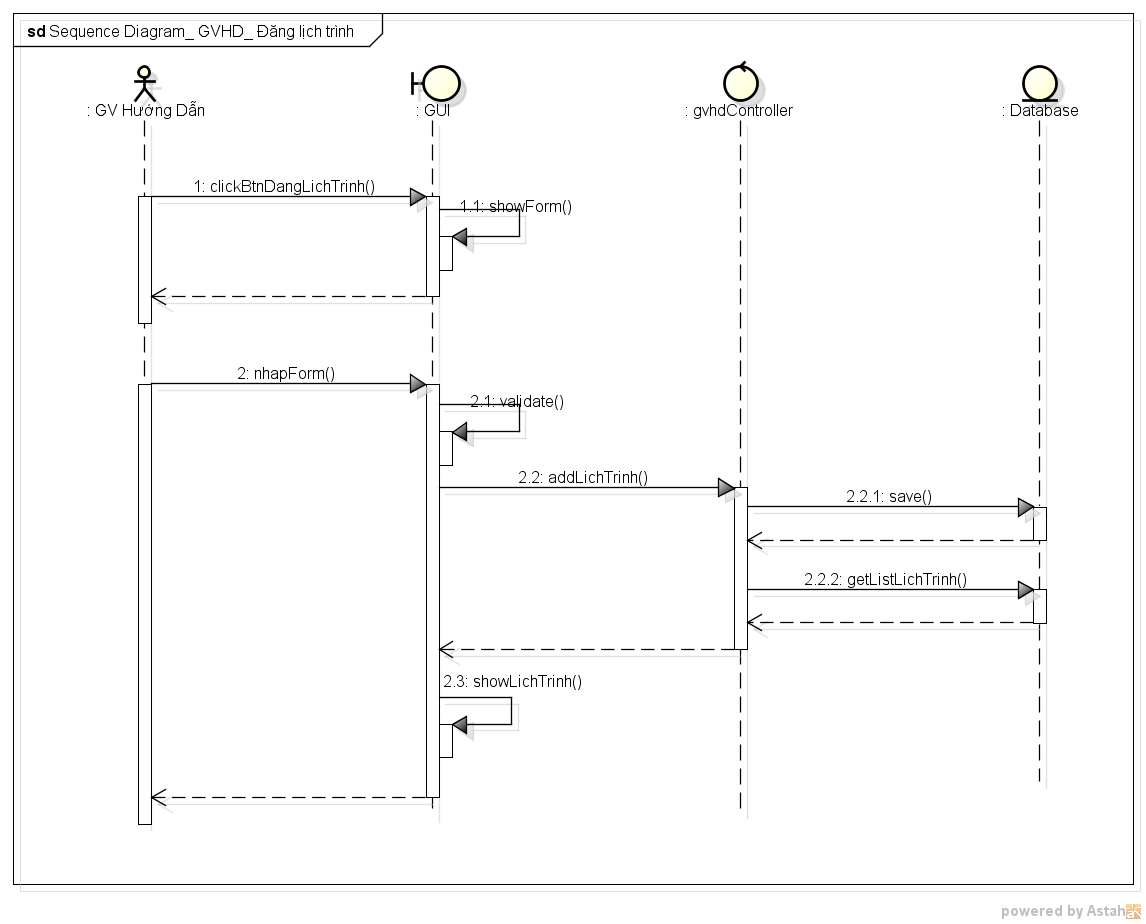
* + - 1. Biểu đồ tuần tự chức năng duyệt đề tài



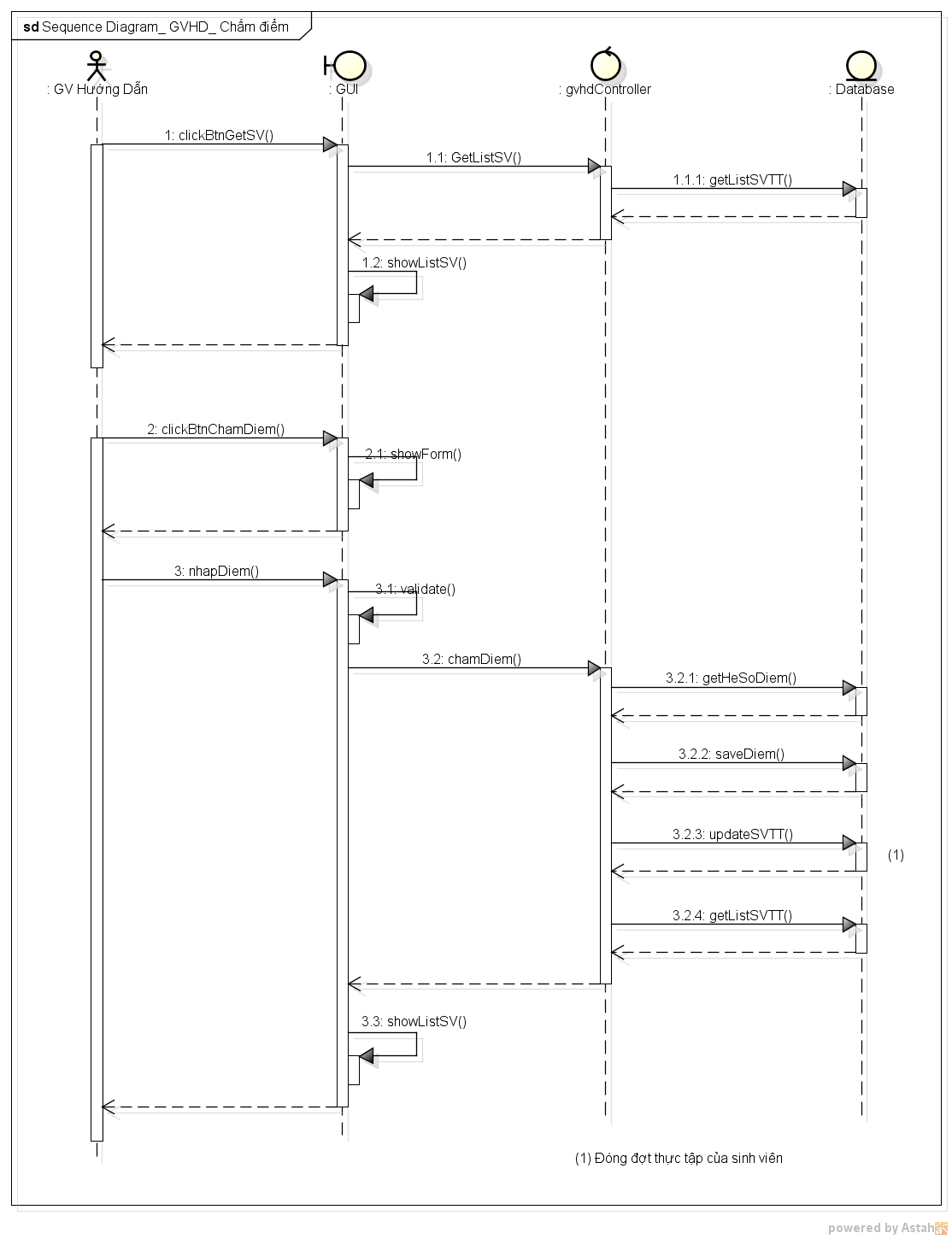
* + - 1. Biểu đồ tuần tự chức năng duyệt đề tài sinh viên đăng ký



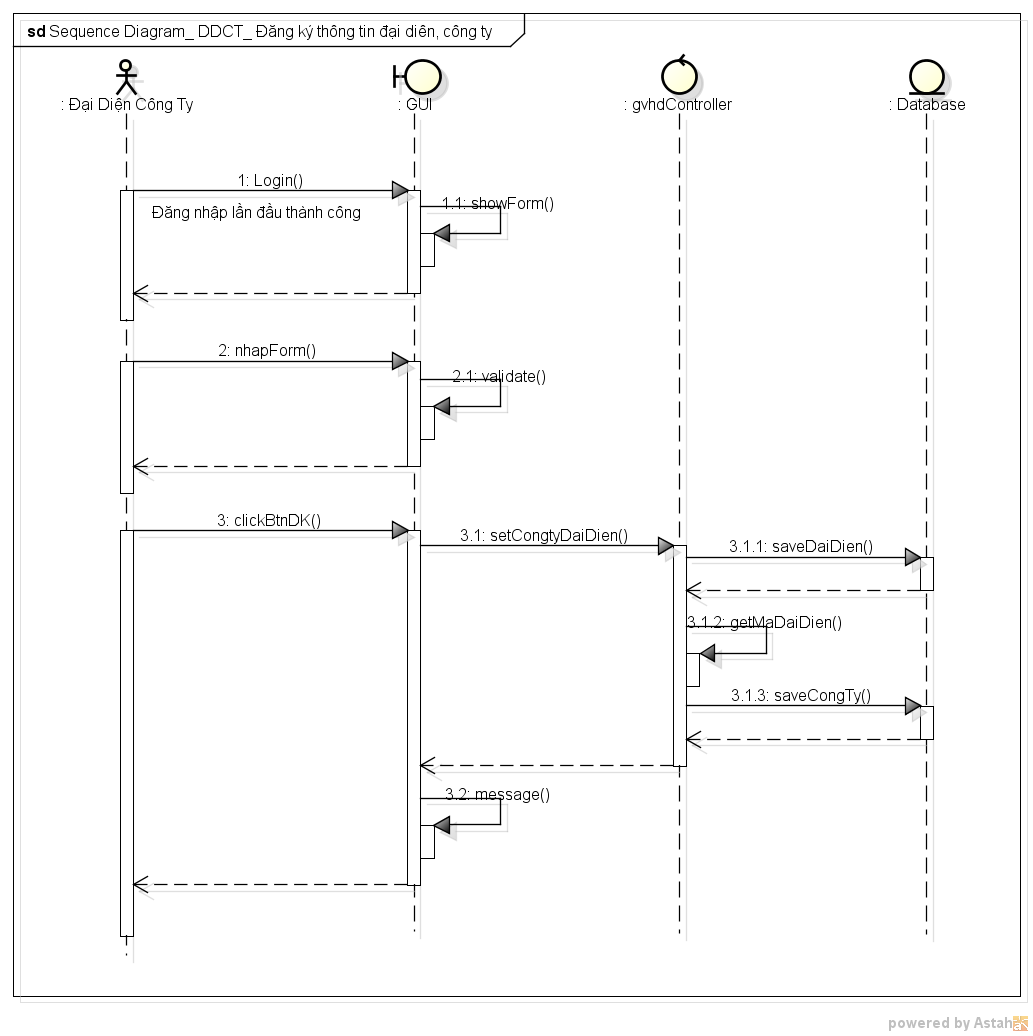
* + - 1. Biểu đồ tuần tự chức năng đăng lịch trình



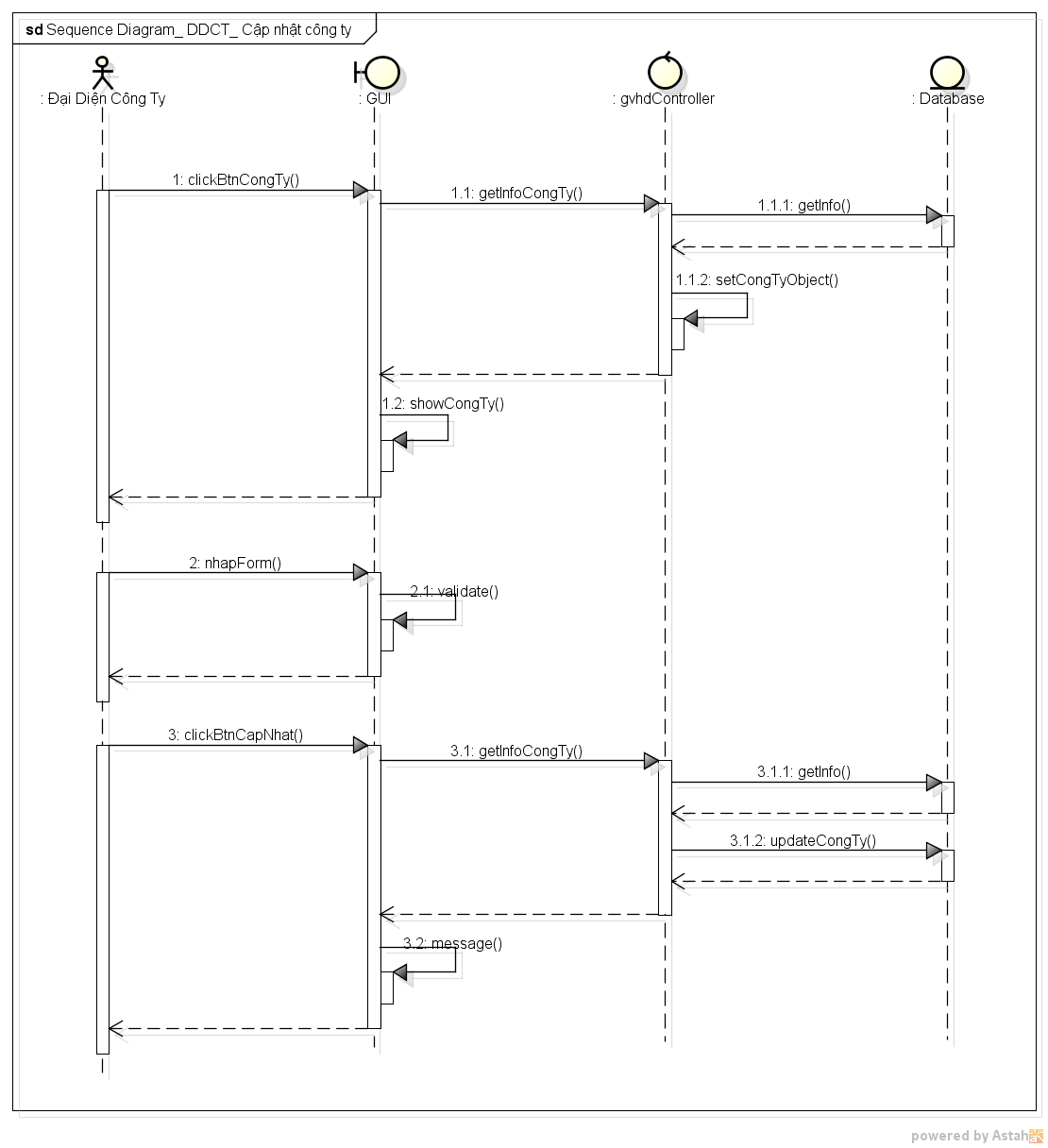
* + - 1. Biểu đồ tuần tự chức năng chấm điểm



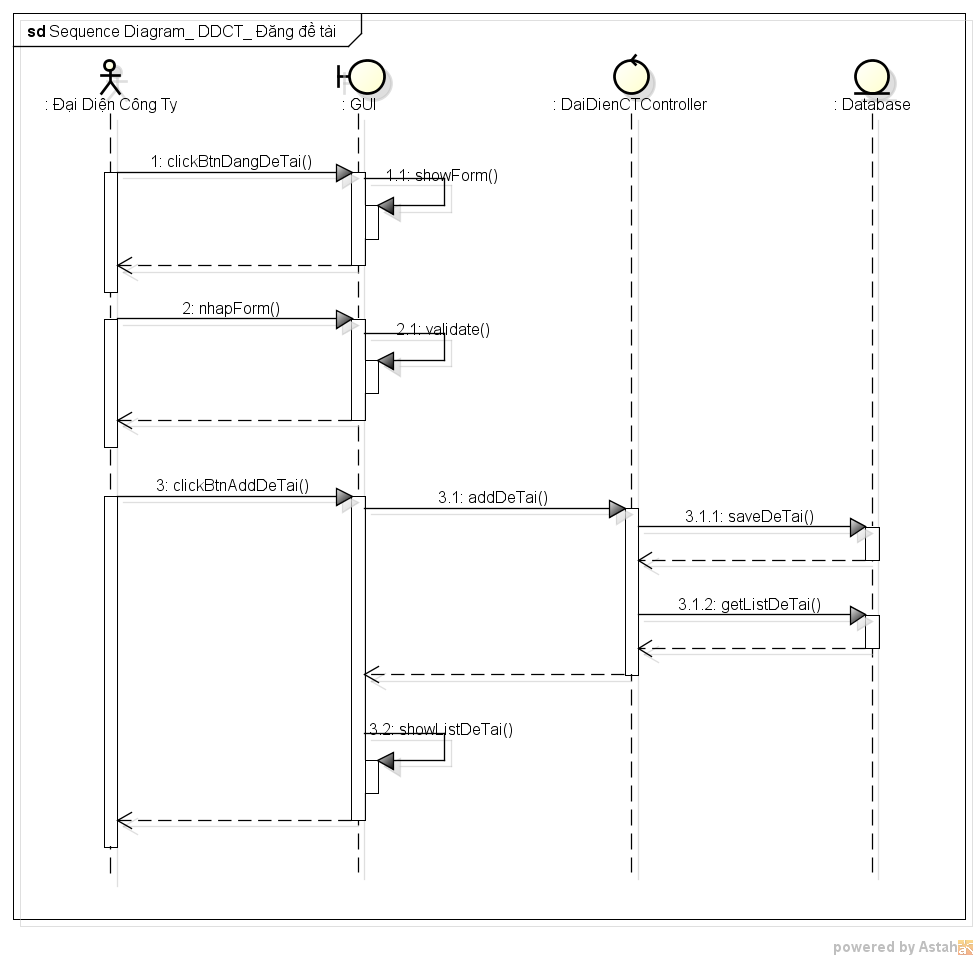
* + - 1. ss
    1. Nhóm biểu đồ tuần tự đại diện công ty
       1. Biểu đồ tuần tự chức năng đăng ký tài khoản, đăng ký công ty



* + - 1. Biểu đồ tuần tự chức năng cập nhật công ty



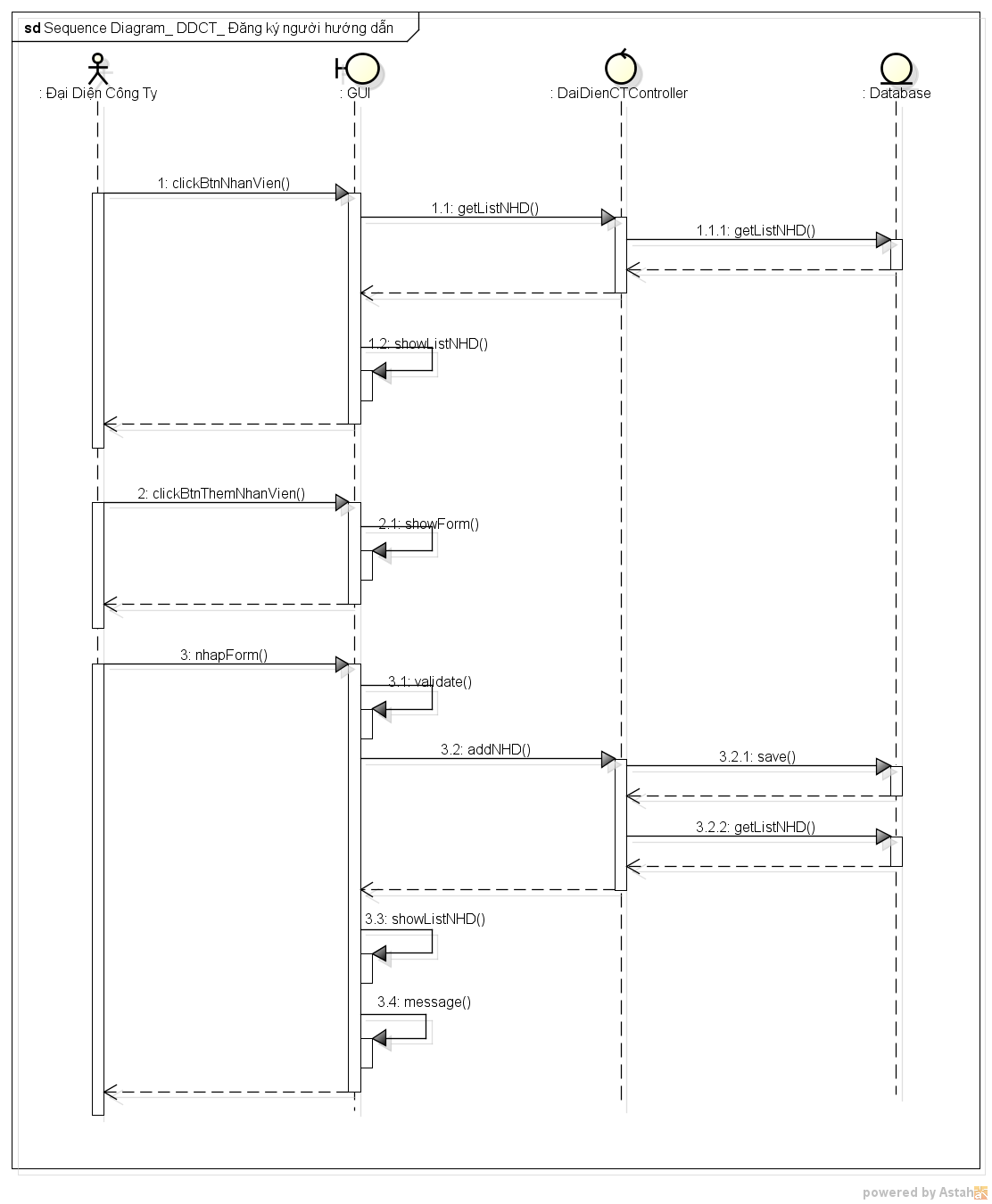
* + - 1. Biểu đồ tuần tự chức năng đăng đề tài



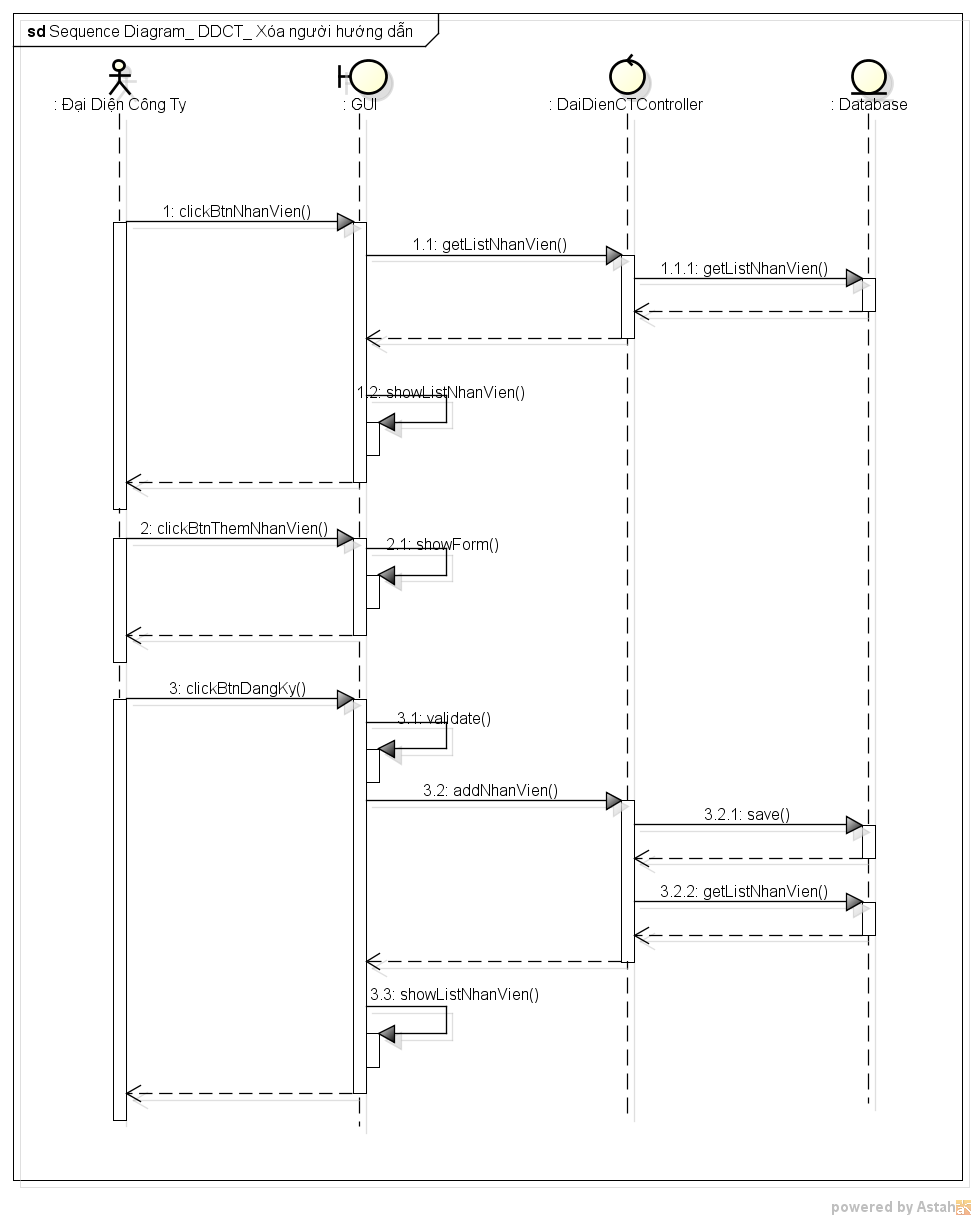
* + - 1. Biểu đồ tuần tự chức năng xóa đề tài



* + - 1. Biểu đồ tuần tự chức năng đăng ký người hướng dẫn

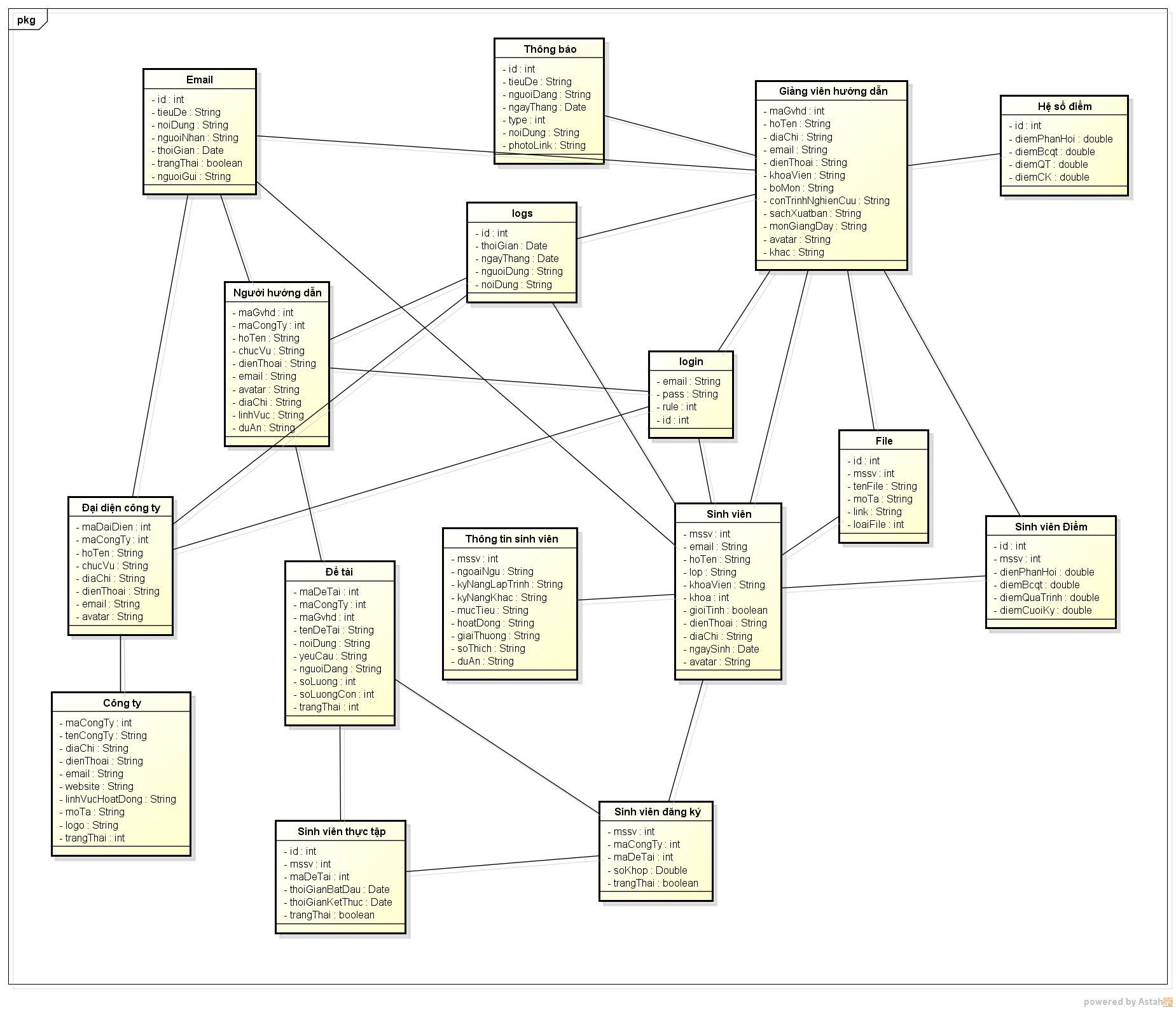


* + - 1. Biểu đồ tuần tự chức năng xóa người hướng dẫn



* + 1. Nhóm biểu đồ tuần tự người hướng dẫn
    2. Nhóm biểu đồ tuần tự admin
       1. Biểu đồ tuần tự chức năng quản lý người dùng (thêm/ sửa/ xóa)
       2. Biểu đồ tuần tự chức năng quản lý công ty
       3. Biểu đồ tuần tự chức năng quản lý đề tài

1. Biểu đồ Class
   * 1. Biểu đồ lớp tổng thể

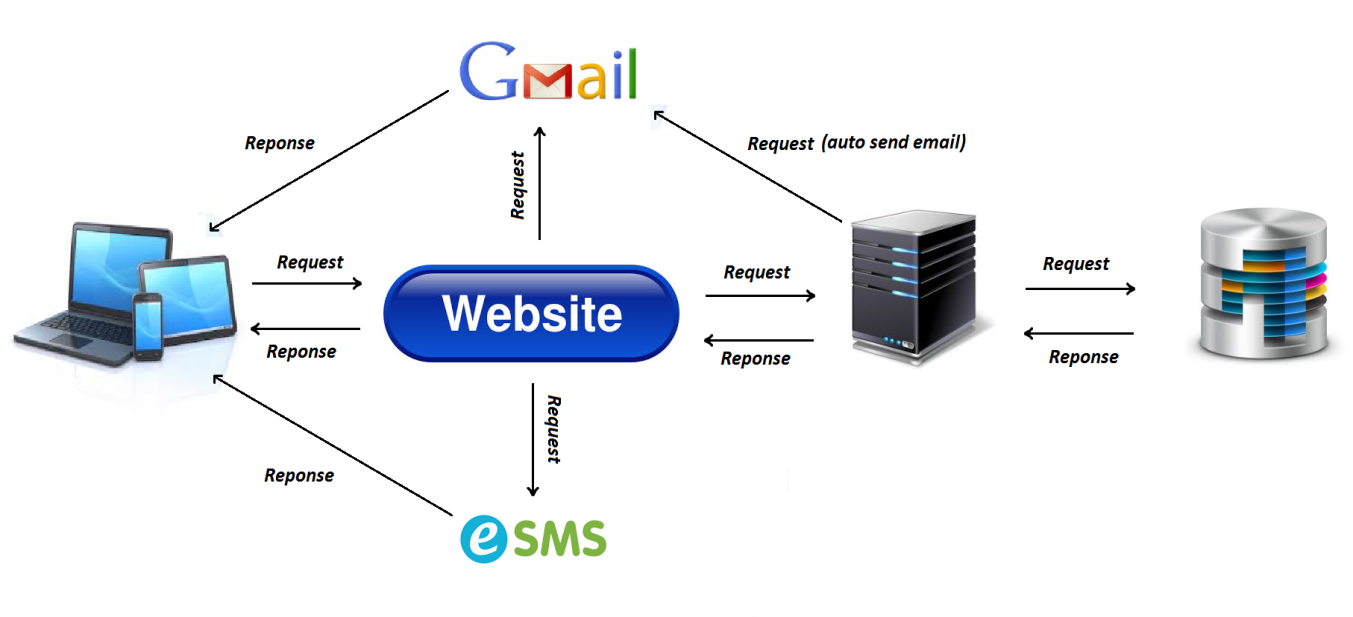


* + 1. Biểu đồ lớp chi tiết

# CHƯƠNG 3: THIẾT KẾ HỆ THỐNG

1. Thiết kế tổng thể

Hệ thống đăng ký và quản lý thực tập trực truyến sẽ bao gồm có: 1 web, 1 database, hệ thống nhắn tin eSMS, hệ thống gửi mail Gmail,… Mô hình của hệ thống như hình dưới:



Tại mức độ về phía người dùng, hệ thống sẽ cũng cấp một giao diện giúp cho người dùng có thể truy cập và sử dụng các chức năng trong hệ thống.

Khi người dùng có bất kỳ sự xác thực thông tin nào hệ thông cung cấp phương thức tựa OTP giúp xác thực người dùng, tránh sự giả mạo thông tin và có hành vi không tốt khi sử dụng hệ thống. Ví dụ: số điện thoại hay emai, hệ thống sẽ nhắn tin qua số điện thoại mà người dùng đã đăng ký hoặc gửi qua tài khoản Gmail.

Về phía cơ sở dữ liệu, hệ thống sử dụng MySQL 5.7.

Về phía server, hệ thống dùng Tomcat 8,

1. Thiết kế chi tiết
   * 1. Thiết kế giao diện
        1. Thiết kế giao diện đăng nhập/ đăng ký
        2. Thiết kế giao diện trang chủ
        3. Thiết kế giao diện mức chi tiết nội dung
        4. Thiết kế giao diện cho admin
     2. Thiết kế logic (thực thể)
        1. Thực thể sinh\_vien (sinh viên):

Thực thể này mô tả thông tin cơ bản về sinh viên thực tập. Thực thể này bao gồm có các thuộc tính sau:

* + mssv: mô tả mã số sinh viên
  + email: mô tả emaiil của sinh viên, bắt buộc theo định dạng: [mssv@student.hust.edu.vn](mailto:mssv@student.hust.edu.vn) . Trong đó, mssv là mã số của sinh viên
  + ho\_ten: mô tả tên của sinh viên thực tập
  + lop: mô tả tên lớp của sinh viên
  + khoa\_vien: mô tả tên khoa hoặc viện mà sinh viên đăng ký học
  + khoa: mô tả khóa học của sinh viên
  + gioitinh: mô tả giới tính của sinh viên
  + dien\_thoai: mô tả số điện thoại của sinh viên
  + dia\_chi: mô tả địa chỉ hiện tại đang ở của sinh viên
  + ngay\_sinh: mô tả ngày sinh của sinh viên
  + avatar: mô tả ảnh thẻ trên hệ thống của sinh viên
  + laptop: mô tả sinh viên có laptop hay không.

Thực thề sinh\_vien có khóa chính là mã số sinh viên: mssv

* + - 1. Thực thể sinh viên đăng ký đề tài
      2. Thực thể sinh viên thực tập
      3. Thực thể file sinh viên
      4. Thực thể điểm thi sinh viên
      5. Thực thể
    1. Thiết kế cơ sở dữ liệu

CREATE TABLE `cong\_ty`

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Key | Null |
| 1 | ma\_cong\_ty | int | PKEY | No |
| 2 | ten\_cong\_ty | varchar(500) | FKEY | No |
| 3 | dia\_chi | varchar(500) |  | No |
| 4 | dien\_thoai | varchar(15) |  | No |
| 5 | email | varchar(125) |  | No |
| 6 | website | varchar(255) |  | No |
| 7 | linh\_vuc\_hoat\_dong | text |  | Yes |
| 8 | mo\_ta | text |  | Yes |
| 9 | logo | varchar(500) |  | Yes |
| 10 | trang\_thai | int(1) |  | Yes |
| 11 | ma\_dai\_dien | int |  | No |

CREATE TABLE `dai\_dien\_cong\_ty`

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Key | Null |
| 1 | ma\_dai\_dien | int | PKEY | No |
| 2 | ho\_ten | varchar(125) |  | No |
| 3 | chuc\_vu | varchar(50) |  | Yes |
| 4 | dia\_chi | varchar(500) |  | Yes |
| 5 | dien\_thoai | varchar(15) |  | No |
| 6 | email | varchar(125) |  | No |
| 7 | avatar | varchar(500) |  | Yes |

CREATE TABLE `de\_tai`

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Key | Null |
| 1 | ma\_de\_tai | int | PKEY | No |
| 2 | ma\_cong\_ty | int | FKEY | No |
| 3 | ma\_nhd | int | FKEY | No |
| 4 | ten\_de\_tai | varchar(500) |  | No |
| 5 | noi\_dung | text |  | No |
| 6 | yeu\_cau\_lap\_trinh | text |  | No |
| 7 | nguoi\_dang | varchar(125) |  | Yes |
| 8 | so\_luong | int(2) |  | No |
| 10 | trang\_thai | int(1) |  | No |
| 11 | ngay\_dang | date |  | No |
| 12 | han\_dang\_ky | date |  | No |
| 13 | yeu\_cau\_khac | text |  | Yes |

CREATE TABLE `dot\_thuc\_tap`

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Key | Null |
| 1 | id | int | PKEY | No |
| 2 | ma\_dot | int(5) |  | No |
| 3 | thoi\_gian\_bat\_dau | date |  | No |
| 4 | hoi\_gian\_ket\_thuc | date |  | No |
| 5 | mo\_ta | text |  | Yes |

CREATE TABLE `email`

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Key | Null |
| 1 | id | int | PKEY | No |
| 2 | tieu\_de | varchar(500) |  | No |
| 3 | noi\_dung | text |  | No |
| 4 | nguoi\_nhan | varchar(125) |  | No |
| 5 | thoi\_gian | datetime |  | Yes |
| 6 | trang\_thai | bit(1) |  | No |
| 7 | nguoi\_gui | varchar(125) |  | No |
| 8 | type | int(1) |  | Yes |
| 9 | mssv | int(8) | FKEY | Yes |
| 10 | dot\_thuc\_tap | int(5) |  | Yes |

CREATE TABLE `file\_all`

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Key | Null |
| 1 | id | int | PKEY | No |
| 2 | email | varchar(125) |  | No |
| 3 | ten\_file | varchar(500) |  | Yes |
| 4 | noi\_dung | varchar(1000) |  | Yes |
| 5 | link | varchar(500) |  | Yes |
| 6 | type | int(1) |  | Yes |
| 7 | ngay\_thang | datetime |  | Yes |

CREATE TABLE `giang\_vien\_huong\_dan`

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Key | Null |
| 1 | ma\_gvpt | int | PKEY | No |
| 2 | ho\_ten | varchar(125) |  | No |
| 3 | dia\_chi | varchar(500) |  | No |
| 4 | email | varchar(125) |  | No |
| 5 | dien\_thoai | varchar(15) |  | No |
| 6 | khoa\_vien | varchar(500) |  | Yes |
| 7 | bo\_mon | varchar(500) |  | Yes |
| 8 | cong\_trinh\_nghien\_cuu | text |  | Yes |
| 9 | sach\_xuat\_ban | text |  | Yes |
| 10 | mon\_giang\_day | text |  | Yes |
| 11 | khac | text |  | Yes |
| 12 | avatar | varchar(500) |  | Yes |

CREATE TABLE `he\_so\_diem`

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Key | Null |
| 1 | id | int |  | No |
| 2 | diem\_phan\_hoi | double(2,2) |  | Yes |
| 3 | diem\_bcqt | double(2,2) |  | Yes |
| 4 | diem\_qt | double(2,2) |  | Yes |
| 5 | diem\_ck | double(2,2) |  | Yes |

CREATE TABLE `login`

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Key | Null |
| 1 | email | varchar(255) | PKEY | No |
| 2 | pass | varchar(255) |  | No |
| 3 | rule | int(1) |  | No |

CREATE TABLE `logs`

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Key | Null |
| 1 | id | int(9) | PKEY | No |
| 2 | thoi\_gian | Time |  | Yes |
| 3 | ngay\_thang | Date |  | Yes |
| 4 | nguoi\_dung | varchar(255) |  | No |
| 5 | noi\_dung | varchar(255) |  | Yes |

CREATE TABLE `nguoi\_huong\_dan`

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Key | Null |
| 1 | ma\_gvhd | int(9) | PKEY | No |
| 2 | ma\_cong\_ty | int(9) | FKEY | No |
| 3 | ho\_ten | varchar(255) |  | No |
| 4 | chucvu | varchar(255) |  | Yes |
| 5 | dien\_thoai | varchar(15) |  | Yes |
| 6 | email | varchar(255) |  | No |
| 7 | avatar | varchar(255) |  | Yes |
| 8 | dia\_chi | varchar(500) |  | Yes |
| 9 | linh\_vuc\_hoat\_dong | text |  | Yes |
| 10 | du\_an | text |  | Yes |

CREATE TABLE `quy\_trinh`

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Key | Null |
| 1 | id | int(11) | PKEY | No |
| 2 | tieu\_de | varchar(500) |  | Yes |
| 3 | noi\_dung | text |  | Yes |
| 4 | ngay\_bat\_dau | date |  | Yes |
| 5 | ngay\_ket\_thuc | date |  | Yes |

CREATE TABLE `sinh\_vien`

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Key | Null |
| 1 | mssv | int(8) | PKEY | No |
| 2 | email | varchar(255) | FKEY | No |
| 3 | ho\_ten | varchar(255) |  | No |
| 4 | lop | varchar(50) |  | No |
| 5 | khoa\_vien | varchar(255) |  | No |
| 6 | khoa | int(3) |  | No |
| 7 | gioi\_tinh | bit(1) |  | No |
| 8 | dien\_thoai | varchar(15) |  | No |
| 9 | dia\_chi | varchar(500) |  | Yes |
| 10 | ngay\_sinh | date |  | Yes |
| 11 | avatar | varchar(500) |  | Yes |
| 12 | laptop | bit(1) |  | Yes |

CREATE TABLE `sinh\_vien\_dang\_ky

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Key | Null |
| 1 | id | int(9) | PKEY | No |
| 2 | mssv | int(8) | FKEY | No |
| 3 | ma\_de\_tai | int(9) | FKEY | No |
| 4 | trang\_thai | int(1) |  | No |
| 5 | so\_khop | double |  | No |
| 6 | ngay\_dang\_ky | date |  | Yes |
| 7 | dot\_thuc\_tap | int(11) |  | Yes |

CREATE TABLE `sinh\_vien\_diem`

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Key | Null |
| 1 | id | int(9) | PKEY | No |
| 2 | mssv | int(8) | FKEY | No |
| 3 | diem\_phan\_hoi | double |  | Yes |
| 4 | diem\_bcqt | double |  | Yes |
| 5 | diem\_qua\_trinh | double |  | Yes |
| 6 | diem\_cuoi\_ky | double |  | Yes |
| 7 | dot\_thuc\_tap | int(11) |  | Yes |

CREATE TABLE `sinh\_vien\_file`

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Key | Null |
| 1 | id | int(9) | PKEY | No |
| 2 | mssv | int(8) | FKEY | No |
| 3 | ten\_file | varchar(500) |  | Yes |
| 4 | mo\_ta | varchar(1000) |  | Yes |
| 5 | link | varchar(500) |  | Yes |
| 6 | loai\_file | int(1) |  | Yes |
| 7 | ngay\_thang | date |  | Yes |

CREATE TABLE `sinh\_vien\_info

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Key | Null |
| 1 | mssv | int(8) | PKEY | No |
| 2 | ngoai\_ngu | varchar(500) |  | yes |
| 3 | ky\_nang\_lt | text |  | Yes |
| 4 | ky\_nang\_khac | text |  | Yes |
| 5 | muc\_tieu | text |  | Yes |
| 6 | hoat\_dong | text |  | Yes |
| 7 | giai\_thuong | text |  | Yes |
| 8 | so\_thich | text |  | Yes |
| 9 | du\_an | text |  | Yes |

CREATE TABLE `sinh\_vien\_thuc\_tap`

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Key | Null |
| 1 | id | int(9) | PKEY | No |
| 2 | mssv | int(8) | FKEY | No |
| 3 | ma\_de\_tai | int(9) | FKEY | Yes |
| 4 | thoi\_gian\_bat\_dau | date |  | Yes |
| 5 | thoi\_gian\_ket\_thuc | date |  | Yes |
| 6 | trang\_thai | bit(1) |  | No |
| 7 | dot\_thuc\_tap | int(11) |  | No |

CREATE TABLE `thong\_bao`

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Key | Null |
| 1 | id | int(9) | PKEY | No |
| 2 | tieu\_de | varchar(1000) |  | Yes |
| 3 | noi\_dung | text |  | Yes |
| 4 | nguoi\_dang | varchar(255) |  | No |
| 5 | ngay\_thang | datetime |  | Yes |
| 6 | type | int(11) |  | Yes |
| 7 | photo\_link | varchar(500) |  | Yes |

* + 1. Thuật toán so khớp

Thuật toán so khớp được hệ thống xác định từ chỉ số yêu cầu của đề tài với chỉ số yêu cầu đáp ứng về kỹ năng lập trình của sinh viên.

Khi đăng các đề tài, người dùng cần phải xác định theo đúng cú pháp quy định trên hệ thống. Điều này là bắt buộc đối với mỗi đề tài.

Cú pháp được quy định như sau: [abc-xyz]

Trong đó, ‘[’ và ‘]’ là các khóa mở của cú pháp, bắt buộc phải khai báo.

* abc là các ngôn ngữ lập trình, viết bằng chữ thường
* xyz là các yêu cầu về trình độ của mỗi ngôn ngữ lập trình, viết bằng chữ thường. Trong đó:
  + thành thạo: ký hiệu là thanhthao với giá trị là 100%
  + tốt: ký hiệu là tot với giá trị là 75%
  + khá: ký hiệu là kha với giá trị là 50%
  + biết: ký hiệu là biet với giá trị là 25%
  + không biết: ký hiệu là khongbiet với giá trị là 0%

Ví dụ: [java-tot][c-kha][sql-tot][html-tot]

Đối với yêu cầu của sinh viên khi đăng ký thông tin cá nhân cũng cần xác định đúng yêu cầu về cú pháp đăng ký như của người đăng đề tài.

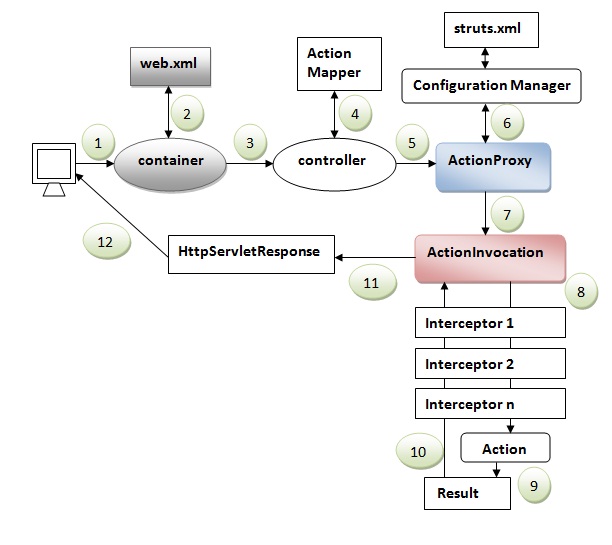
Về việc thực hiện so khớp: hệ thống sẽ lấy thông tin yêu cầu về đề tài và thông tin về kỹ năng năng lập trình của sinh viên. Hệ thống sẽ duyệt các yêu cầu của đề tài theo từng kỹ năng lập trình của sinh viên có. Điểm trung bình so khớp sẽ bằng số lượng so khớp trùng chia cho tổng số yêu cầu của đề tài.

# CHƯƠNG 4: CÀI ĐẶT THỰC NGHIỆM

* 1. Công cụ lập trình và và các Framework
     1. Công cụ lập trình
* Chương trình được viết bằng ngôn ngữ lập trình Java.
* Công cụ lập trình:
  + NetBeans IDE 8.2
  + JDK 8
  + MySQL server 5.7
  + MySQL Workbeans 6.3 CE
  + Phát triển trên hệ điều hành Windows 7
    1. Các ngôn ngữ lập trình sử dụng
       1. HTML và CSS
       2. Javascript
       3. JSP
       4. Java
    2. Strust 2 Framework
       1. Giới thiệu về Strust 2 Framework

Struts 2 Framework được sử dụng để phát triển các ứng dụng web dựa trên MVC. Struts 2 là sự kết hợp của Webwork Framework và Struts 1. Struts 2 không phải là phiên bản tiếp theo của Struts 1. Struts 2 cung cấp rất nhiều đặc điểm mà không có trong Struts 1. Dưới đây là các đặc điểm quan trọng nhất của Struts 2:

* Các thành phần MVC có thể cấu hình
* Các Action dựa trên POJO (Plain Old Java Object)
* Hỗ trợ AJAX
* Dễ dàng tích hợp
* Kiểu kết quả (Result Type) đa dạng
* Hỗ trợ tag
* Hỗ trợ Theme và Template
  + - 1. Cách thức hoạt động



Quá trình xử lý request trong Struts 2 Framework

* Người dùng gửi một yêu cầu (request) cho Action
* Container ánh xạ request này trong web.xml file và lấy tên lớp của Controller
* Container triệu hồi Controller (StrutsPrepareAndExecuteFilter hoặc FilterDispatcher). Bắt đầu từ Struts 2.1, đó là lớp StrutsPrepareAndExecuteFilter
* Controller lấy thông tin cho Action từ ActionMapper
* Controller triệu hồi ActionProxy
* ActionProxy lấy thông tin của Action và Interceptor Stack từ Configuration Manager mà lấy thông tin từ struts.xml file
* ActionProxy chuyển tiếp request tới ActionInvocation
* ActionInvocation triệu hồi mỗi Interceptor và Action
* Một kết quả được tạo ra
* Một kết quả được tạo ra
* Một HttpServletResponse được tạo ra
* Phản hồi (response) được gửi tới người dùng
  + - 1. Action và result trong Struts 2 Framework
* Action: Khi một request được gửi từ phía người dùng đến server thì nó sẽ được ánh xạ thành một action tương ứng (lập trình viên tự cấu hình). Các action này sẽ được controller điều hướng cho một classAction tương ứng thức thi (lập trình viên tự cấu hình).
  + ActionClass: thuộc tầng model, là đơn cị làm việc cơ bản. Lớp ActionClass sẽ được kế thừa từ lớp ActionSupport
  + ActionSupport: được implement từ lớp Action, LocaleProvider, TextProvider, validationAwave. Nó cũng có một số thuộc tính final kế thừa từ lớp Action và được xem như là kết quả trả về từ action như: ERROR, INPUT, SUCCESS, LOGIN, NONE. Chúng ta cũng có thể thêm các action trả về bằng kiểu dữ liệu String.
* Result: không giống như Struts1, kết quả của một action là một ActionForward. Trong Struts2 khi một phương thức thực thi một action hoàn thành thì kết quả của nó trả về là một chuỗi và tùy vào chuỗi nàu mà ta có thể cấu hình để Struts2 mapping với một tập các action kết quả trả về cho người dùng. Trong Struts2 có các loại result sau:
  + Depatcher result: thực hiện include hoặc là forward đến một view thường là một trang jsp. Nếu trong quá trình khai báo mà ta không khai báo bất kỳ loại result nào thì mặc định Struts sẽ hiểu trả về theo kiểu depatcher
  + Redirect action result: kết quả trả sau khi thực hiện action sẽ sử dụng actionmapper để chuyển hướng để thực hiện một action khác. Trong đó:
    - actionName: là tên action se redirect tới
    - namespace: là namespace chứa action Name. Nếu namespace là null thì Struts2 xem actionName nằm trong cùng namespace với action gọi nó
    - Ví dụ:
  + Chain Result: Result này sẽ call đến toàn bộ action khác với toàn bộ interceptor stack và result của chính bản thân nó. Cách khai báo hoàn toàn giống với redirect action result chỉ thay chỗ type là chain.
    - actionName: là action sẽ chuyển hướng đến.
    - namespace: là namespace chứa actionName, nếu là null thì là trong cùng namespace với action hiện tại.
    - method: là phương thức sẽ được gọi là class Action của action đích. Nếu không khai báo thì sử dụng phương thức execute.
    - skipAction: danh sách các action cách nhau bởi dấu “,” là những action có thể được chuyển hướng đến.
  + Redirect result: Result này sẽ thực hiện redirect browser tới một vị trí/ trang khác, tương tự như HttpServletSponse.sendRedirect(url).
  + FreeMarker result Sẽ render một view sử dụng template FreeMaker. Tham số:
    - Location: nơi lưu template.
    - contentType: loại nội dung. Mặc định là “text/html”.
    - writeIfCompleted – (mặc định là false), write vào stream khi và chỉ khi không có lỗi nào trong quá trình xử dụng mẫu. Cài đặt template\_exception\_handler=rethrow trong freemarker.properties sẽ có hiệu quả tương tự.
  + Velocity result: Giả lập một môi trường thực thi jsp và hiển thị theo mẫu Velocity. Tham số Location: là vị trí mẫu để thực hiện xử lý.
  + Plain Text Result: Result này thực hiện trả về nội dung theo định dạng text bình thường. Thường được sử dụng khi muốn thể hiện một trang web dưới dạng text bình thường. Tham số:
    - Location: vị trí file cần show nội dung.
    - charSet: tập ký tự được sử dụng.
  + HttpHeader Result: Là một custom Result cài đặt trạng thái và header của HTTP bằng cách tuỳ chọn giá trị của valueStack. Thường được dùng để thông báo lỗi cho người dùng. Tham số:
    - status – http servlet trả về mã trạng thái mà sẽ được thiết lập trên một response.
    - headers – giá trị header.
    - error – http servlet trả về mã lỗi mà sẽ được thiếp lập trên một response.
    - errorMessage – thông điệp lỗi để thiết lập trên response nếu tham số ‘error’ được thiết lập.
  + Ngoài ra còn có: Tiles result, stream result, XSL result, global result, custom result.
    - 1. Cấu hình Struts 2 Framework

Struts 2 cung cấp cho chúng ta 2 cách cấu hình để thực hiện mapping một action từ người dùng với lớp Action thực hiện xử lý yêu cầu đó là cấu hình trong file cấu hình struts.xml và cấu hình theo Annotation. Trong đồ án này, tác giả chỉ đề cập tới cấu hình bằng cách dùng file Struts.xml

1. Action

* Trong cách khai báo trên thì ứng với mỗi action ta sẽ khai báo trong thẻ “action” trong thẻ action thì có các thuộc tính như
  + Name: là tên của action mà ta sẽ gọi.
  + Class: là ActionClass thực hiện xử lý action trên.
  + Method: nếu người dùng không khai báo attribute method này thì đồng nghĩa với việc Struts sẽ chọn phương thức *execute* trong ActionClass mà thực hiện. Còn nếu người dùng khai báo chỗ này thì Struts sẽ lấy phương thức được khai báo để thực hiện xử lý cho action đó.
* Ngoài method như trên còn có một dạng gọi là WildCard method.
* Đôi khi ta thao tác trên một đối tượng có một tập các action ví dụ như SinhVien có các action mà người dùng thường sử dụng như là delete, insert, update,... thì ta có một các đơn giản để khai báo mapping cho action là dùng wildcare method.
* Cách khai báo

1. Result

Trong thẻ action còn có thẻ con result

* Khi 1 Action được thực hiện thành công, nó sẽ trả về 1 chuỗi.
* Giá trị của chuỗi được sử dụng để chọn thành phần kết quả.
* 1 action mapping sẽ thường có 1 tập thể hiện các kết quả khác nhau có thể xảy ra.
* Có các tên kết quả đã được xác định trước (tokens).
* Ứng dụng có thể định nghĩa các tên kết quả khác(token) để khớp với 1 số trường hợp mong muốn.

Result định nghĩa trước gồm:

* String SUCCESS = “success”;
* String NONE = “none”;
* String ERROR = “error”;
* String INPUT = “input”;
* String LOGIN = “login”;

Result Element:

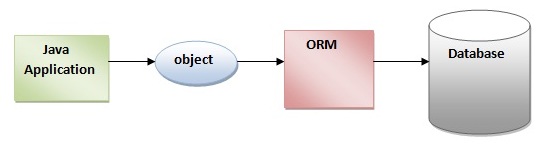
* Name: tên result là giá trị trả về khi action thực hiện, nó cso thể là tokens có sẵn hoặc một tokens do mình định nghĩa.
* Type: kiểu result quy định kiểu result: depatcher, chain... Nếu người dùng không khai báo thì struts sẽ lấy mặc định là depatcher.
* Có 3 cách để khai báo result:

Khai báo không dùng type là default

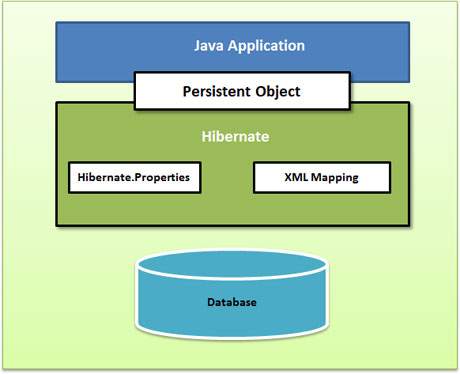
Khai báo sử dụng type defaults

Khai báo sử dụng type default, param cũng default

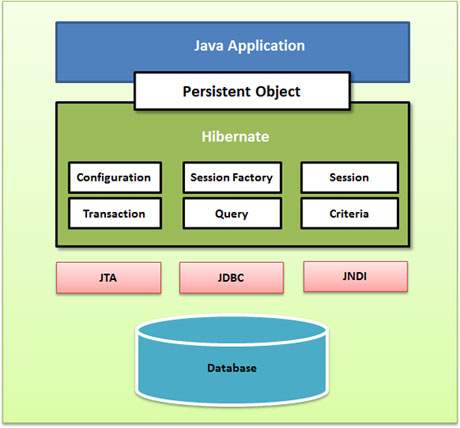
* + 1. Hibernate Framework
       1. Giới thiệu về Hibernate famework
* Hibernate framework là một giải pháp ORM (Object Relational Mapping) mã nguồn mở, gọn nhẹ. Hibernate giúp đơn giản hoá sự phát triển của ứng dụng java để tương tác với cơ sở dữ liệu.
* Tool ORM giúp đơn giản hoá việc tạo ra dữ liệu, thao tác dữ liệu và truy cập dữ liệu. Đó là một kỹ thuật lập trình để ánh xạ đối tượng vào dữ liệu được lưu trữ trong cơ sở dữ liệu.



* Những lợi ích của Hibernate framework:
  + Mã nguồn mở và nhẹ: Hibernate Framework là mã nguồn mở có giấy phép LGPL và nhẹ.
  + Hiệu suất nhanh: Hiệu suất của Hibernate Framework là nhanh bởi vì bộ nhớ cache được sử dụng trong nội bộ Hibernate Framework. Có hai loại bộ nhớ cache trong Hibernate Framework, gồm bộ nhớ cache cấp một và bộ nhớ cache cấp hai. Bộ nhớ cache cấp một được bật bằng lệnh mặc định.
  + Truy vấn cơ sở dữ liệu độc lập: HQL (Hibernate Query Language) là phiên bản hướng đối tượng của SQL. Nó tạo ra các truy vấn cơ sở dữ liệu độc lập. Vì vậy, bạn không cần phải viết các truy vấn cơ sở dữ liệu cụ thể. Trước Hibernate, nếu dự án có cơ sở dữ liệu bị thay đổi, chúng ta cần phải thay đổi truy vấn SQL dẫn đến sự cố bảo trì.
  + Tạo bảng tự động: Hibernate framework cung cấp phương tiện để tạo ra các bảng cơ sở dữ liệu tự động. Vì vậy, không cần phải tạo ra các bảng trong cơ sở dữ liệu bằng tay.
  + Đơn giản lệnh join phức tạp: Có thể lấy dữ liệu từ nhiều bảng một cách dễ dàng với Hibernate framework.
  + Cung cấp thống kê truy vấn và trạng thái cơ sở dữ liệu: Hibernate hỗ trợ bộ nhớ cache truy vấn và cung cấp số liệu thống kê về truy vấn và trạng thái cơ sở dữ liệu.
* Các hệ quản trị CSDL được hỗ trợ: Hibernate hỗ trợ hầu hết tất các RDBMS chính. Dưới đây là danh sách các cơ sở dữ liệu quan hệ được hỗ trợ bởi Hibernate.
  + HSQL Database Engine
  + DB2/NT
  + MySQL
  + PostgreSQL
  + FrontBase
  + Oracle
  + Microsoft SQL Server Database
  + Sybase SQL Server
  + Informix Dynamic Server
    - 1. Kiến trúc của Hibernate framework
* Kiến trúc Hibernate bao gồm nhiều đối tượng như đối tượng persistent, session factory, transaction factory, connection factory, session, transaction,…
* Có 4 tầng trong kiến trúc hibernate đó là tầng ứng dụng java, tầng hibernate framework, tầng backhand api và tầng database. Dưới đây là sơ đồ kiến trúc hibernate:



* Dưới đây là một hình ảnh chi tiết về Kiến trúc ứng dụng Hibernate với vài lớp core quan trọng.



* Hibernate sử dụng các API Java hiện có, như JDBC, Java Transaction API (JTA) và Java Naming and Directory Interface (JNDI). JDBC cung cấp một mức độ trừu tượng cơ bản của cơ sở dữ liệu quan hệ, cho phép hầu hết các cơ sở dữ liệu với trình điều khiển JDBC được hỗ trợ bởi Hibernate. JNDI và JTA cho phép Hibernate được tích hợp với các máy chủ ứng dụng J2EE.
* Các thành phần trong kiến trúc Hibernate
  + Đối tượng Configuration: Đối tượng Configuration là đối tượng Hibernate đầu tiên bạn tạo trong bất kỳ ứng dụng Hibernate nào và chỉ cần tạo một lần trong quá trình khởi tạo ứng dụng. Nó đại diện cho một tập tin cấu hình hoặc thuộc tính yêu cầu của Hibernate. Đối tượng Configuration cung cấp hai thành phần chính:
    - Database Connection: Thao tác này được xử lý thông qua một hoặc nhiều tệp cấu hình được Hibernate hỗ trợ. Các tệp này là hibernate.properties và hibernate.cfg.xml.
    - Class Mapping Setup: Thành phần này tạo ra kết nối giữa các lớp Java và các bảng cơ sở dữ liệu.
  + Đối tượng SessionFactory:
    - Đối tượng Configuration được sử dụng để tạo ra một đối tượng SessionFactory mà cấu hình Hibernate cho ứng dụng bằng cách sử dụng tệp tin cấu hình được cung cấp và cho phép một đối tượng Session được khởi tạo. SessionFactory là một đối tượng luồng an toàn và được sử dụng bởi tất cả các luồng của một ứng dụng.
    - SessionFactory là đối tượng nặng nên thường nó được tạo ra trong quá trình khởi động ứng dụng và lưu giữ để sử dụng sau này. Chúng sẽ cần một đối tượng SessionFactory cho mỗi cơ sở dữ liệu bằng cách sử dụng một tập tin cấu hình riêng biệt. Vì vậy, nếu chúng ta đang sử dụng nhiều cơ sở dữ liệu thì chúng ta sẽ phải tạo nhiều đối tượng SessionFactory.
  + Đối tượng Session
    - Một session được sử dụng để có được một kết nối vật lý với một cơ sở dữ liệu. Đối tượng Session là nhẹ và được thiết kế để được tạo ra thể hiện mỗi khi tương tác với cơ sở dữ liệu. Các đối tượng liên tục được lưu và truy xuất thông qua một đối tượng Session.
    - Các đối tượng Session không nên được mở trong một thời gian dài bởi vì chúng thường không phải là luồng an toàn và chúng cần được tạo ra và được đóng khi cần thiết.
  + Đối tượng Transaction
    - Một Transaction đại diện cho một đơn vị làm việc với cơ sở dữ liệu và hầu hết các RDBMS hỗ trợ chức năng transaction. Các transaction trong Hibernate được xử lý bởi trình quản lý transaction và transaction (từ JDBC hoặc JTA).
    - Đây là một đối tượng tùy chọn và các ứng dụng Hibernate có thể chọn không sử dụng interface này, thay vào đó quản lý transaction trong code ứng dụng riêng.
  + Đối tượng Query: Các đối tượng Query sử dụng chuỗi truy vấn SQL hoặc Hibernate Query Language (HQL) để lấy dữ liệu từ cơ sở dữ liệu và tạo các đối tượng. Ví dụ truy vấn được sử dụng để ràng buộc các tham số truy vấn, giới hạn số lượng kết quả được trả về bởi truy vấn và cuối cùng thực hiện truy vấn.
  + Đối tượng Criteria: Đối tượng Criteria được sử dụng để tạo và thực hiện truy vấn các tiêu chí định hướng đối tượng để lấy các đối tượng.
    - 1. Cấu hình XML kết nối Database

Hibernate yêu cầu phải thiết định thông tin các để lớp Java ánh xạ đến các bảng cơ sở dữ liệu. Hibernate cũng yêu cầu một tập hợp các thiết lập cấu hình liên quan đến cơ sở dữ liệu và các tham số liên quan khác. Tất cả các thông tin như vậy thường được cung cấp dưới dạng file thuộc tính Java tiêu chuẩn có tên gọi là hibernate.properties, hoặc dưới dạng file XML có tên hibernate.cfg.xml.

Chúng ta sử dụng file XML hibernate.cfg.xml để chỉ định các thuộc tính Hibernate bắt buộc trong các ví dụ trong các bài học Hibernate. Hầu hết các thuộc tính lấy giá trị mặc định của chúng và không bắt buộc phải chỉ định chúng trong file properties trừ khi nó thực sự là bắt buộc. Tập tin này được lưu trong thư mục gốc của ứng dụng.

Cấu hình kết nối Hibernate với MySQL Database:

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE hibernate-configuration PUBLIC "-//Hibernate/Hibernate Configuration DTD 3.0//EN" "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">

<hibernate-configuration>

<session-factory>

<property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>

<property name="hibernate.connection.driver\_class">com.mysql.jdbc.Driver</property>

<property name="hibernate.connection.url">jdbc:mysql://localhost:3306/doantotnghiep?zeroDateTimeBehavior=convertToNull</property>

<property name="hibernate.connection.username">root</property>

<property name="hibernate.connection.password">sonnc</property>

<property name="hibernate.current\_session\_context\_class">thread</property>

<property name="hibernate.jdbc.fetch\_size">100000</property>

<property name="hibernate.show\_sql">true</property>

<!—Các file mapping sẽ được viết ở đây -->

<mapping resource="hust/sie/inpg12/sonnc/entities/login.hbm.xml"/>

</session-factory>

</hibernate-configuration>

* + - 1. Mapping POJOs

Để có thể mapping POJO, trước tiên, chúng ta cần phải có một hệ cơ sở dữ liệu. Giả sử chúng ta có một bảng dữ liệu như sau:

DROP TABLE IF EXISTS `login`;

CREATE TABLE `login` (

`email` varchar(255) CHARACTER SET utf8 NOT NULL,

`pass` varchar(255) CHARACTER SET utf8 NOT NULL,

`rule` int(1) NOT NULL,

`status` varchar(50) CHARACTER SET utf8 NOT NULL,

PRIMARY KEY (`email`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci;

Chúng ta sẽ tạo một class chứa các đối tượng :

public class Employee {

private String email;

private String pass;

private int rule;

private String status;

// Viết các contructor ở đây

// Viết các setter và getter ở đây

}

Tiếp theo, chúng ta tạ một file <classname>.hbm.xml:

<?xml version="1.0"?>

<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD 3.0//EN"

"http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">

<!-- Generated Mar 25, 2018 2:40:18 PM by Hibernate Tools 4.3.1 -->

<hibernate-mapping>

<class name="hust.sie.inpg12.sonnc.entities.Login" table="login" catalog="doantotnghiep" optimistic-lock="version">

<id name="email" type="string">

<column name="email" />

<generator class="assigned" />

</id>

<property name="pass" type="string">

<column name="pass" not-null="true" />

</property>

<property name="rule" type="int">

<column name="rule" not-null="true" />

</property>

<property name="status" type="string">

<column name="status" not-null="true" />

</property>

</class>

</hibernate-mapping>

Các thẻ trong file mapping bao gồm:

* Tài liệu mapping là một tài liệu XML có <hibernate-mapping> là phần tử gốc chứa tất cả các phần tử <class>.
* Các phần tử <class> được sử dụng để định nghĩa ánh xạ cụ thể từ các lớp Java sang các bảng cơ sở dữ liệu. Tên lớp Java được chỉ định sử dụng thuộc tính name của phần tử lớp và tên bảng cơ sở dữ liệu được chỉ định sử dụng thuộc tính table.
* Phần tử <meta> là thành phần tùy chọn và có thể được sử dụng để tạo ra mô tả lớp.
* Phần tử <id> ánh xạ thuộc tính ID duy nhất trong lớp tới khóa chính của bảng cơ sở dữ liệu. Thuộc tính name của id id đề cập đến thuộc tính trong lớp và thuộc tính column đề cập đến cột trong bảng cơ sở dữ liệu. Thuộc tính type giữ kiểu ánh xạ hibernate, các kiểu mapping này sẽ chuyển đổi từ kiểu dữ liệu Java sang SQL.
* Phần tử <generator> bên trong phần tử id được sử dụng để tự động tạo giá trị cho khóa chính. Thiết lập thuộc tính class của phần tử generator được đặt là native để cho phép hibernate chọn identity, sequence hoặc hilo để tạo khoá chính tùy thuộc vào khả năng của cơ sở dữ liệu.
* Phần tử <property> được sử dụng để ánh xạ một thuộc tính của lớp Java vào một cột trong bảng cơ sở dữ liệu. Thuộc tính name của phần tử đề cập đến thuộc tính trong lớp và thuộc tính column đề cập đến cột trong bảng cơ sở dữ liệu. Thuộc tính type giữ kiểu ánh xạ hibernate, các kiểu mapping này sẽ chuyển đổi từ kiểu dữ liệu Java sang SQL.
  1. Cấu trúc hệ thống
  2. Triển khai trên Tomcat server 8

Để có thể triển khai và chạy ứng dụng trên Tomcat Server 8, chúng ta cần phải có một máy chủ được cài JDK và cài Tomcat

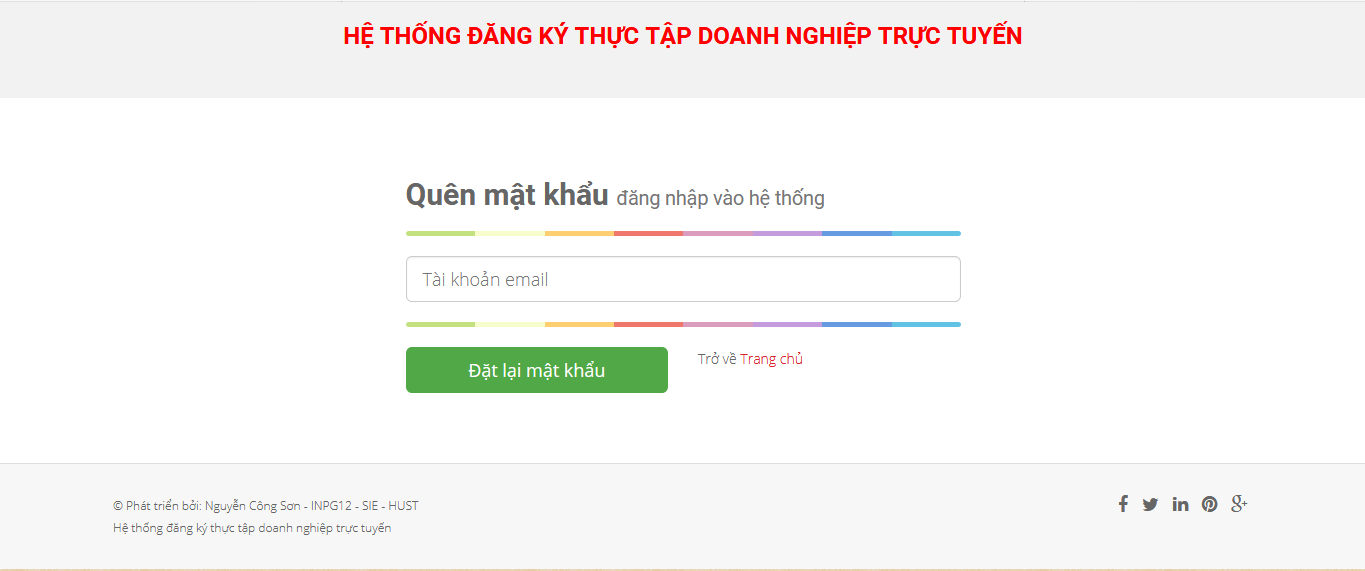
* 1. Kết quả
     1. Kết quả chương trình dành chung cho các đối tượng
        1. Đăng nhập/ đăng ký



Người dùng sẽ phải nhập đầy đủ về tài khoản và mật khẩu. Tài khoản là email mà người dùng đã đăng ký trước đó.

* Đối với sinh viên, tài khoản là email có định dạng: [mssv@student.hust.edu.vn](mailto:mssv@student.hust.edu.vn), trong đó: mssv là mã số sinh viên
* Đối với giảng viên hướng dẫn, tài khoản là email có định dạng: [abc@hust.edu.vn](mailto:abc@hust.edu.vn) hoạc [abc@soict.hust.edu.vn](mailto:abc@soict.hust.edu.vn), trong đó: abc là định danh do người dùng đặt

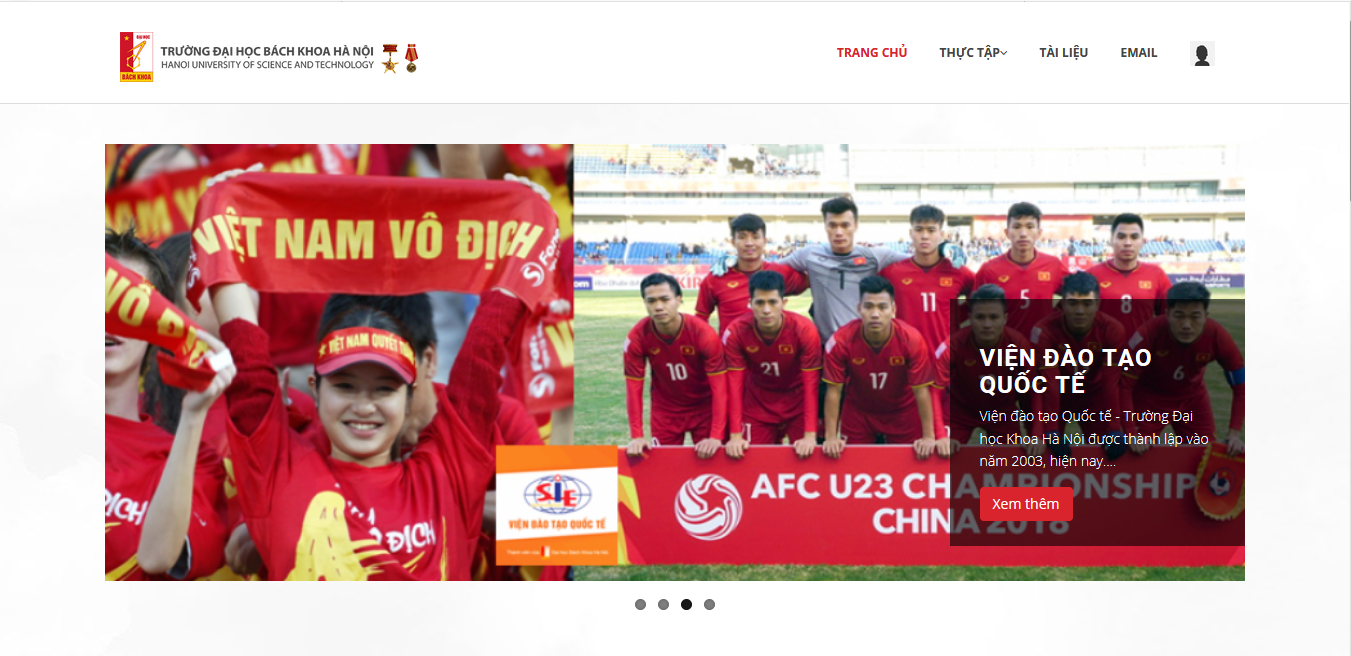
Đối với giao diện quên mật khẩu, người dùng sẽ cần nhập email, hệ thống sẽ gửi mã xác thực qua email và qua tin nhắn SMS



Đối với giao diện đăng ký, người dùng cần nhập email và mật khẩu, hệ thống sẽ xác thực qua email. Sau khi đăng ký xong, người dùng phải đăng nhập lại mới có hiệu lực



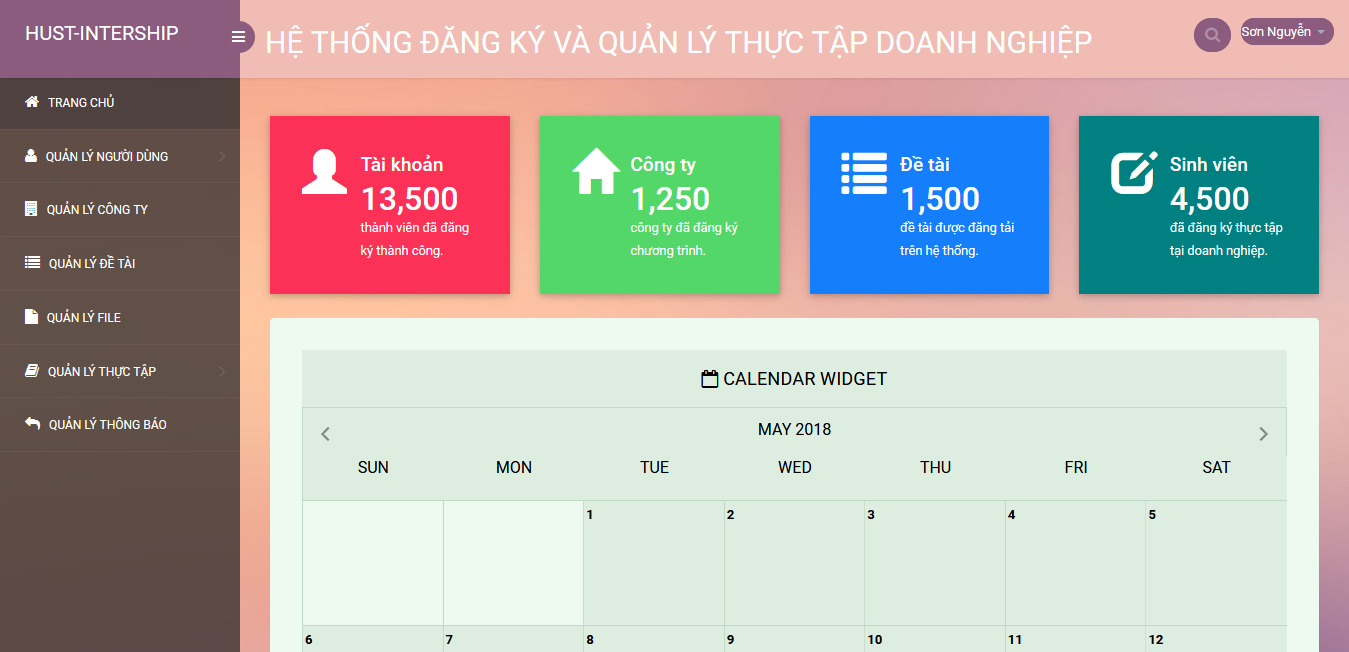
* + - 1. Giao diện trang chủ
    1. Kết quả chương trình cho người dùng là sinh viên
       1. Giao diện trang chủ



Tại giao diện trang chủ, sinh viên sẽ xem được các thông báo, xem được 5 đến 10 công ty mới nhất và các file tài liệu cần thiết.

Sinh viên thực hiện được các chức năng của hệ thống gồm có xem danh sách công ty, xem đề tài, đăng ký đề tài, xem đề tài đã đăng ký, xem điểm thi, xem lịch trình thực tập, đăng – tải tài liệu, email, cập nhật thông tin cá nhân.

* + - 1. Giao diện xem danh sách công ty và chi tiết công ty
      2. Giao diện xem đề tài, danh sách đề tài, đăng ký đề tài
      3. Giao diện xem danh sách đề tài đã đăng ký và trạng thái đăng ký
      4. Giao diện xem điểm thi
      5. Giao diện xem lịch trình thực tập
      6. Giao diện xem, đăng, tải tài liệu
      7. Giao diện đăng ký, cập nhật thông tin cá nhân
    1. Kết quả chương trình cho người dùng là giảng viên hướng dẫn
       1. Giao diện trang chủ
    2. Kết quả chương trình cho người dùng là đại diện công ty
    3. Kết quả chương trình cho người dùng là người hướng dẫn
    4. Kết quả chương trình cho người dùng là admin
       1. Giao diện trang chủ



Tại đây, admin có thể thực hiện các chức năng quản lý và xem các thông số cơ bản của hệ thống về số lượng tài khoản, số lượng công ty, số lượng đề tài và số lượng sinh viên đăng ký thực tập. Các hành động chức năng quản lý sẽ được đặt dọc theo bên trái của màn hình.

* + - 1. Giao diện quản lý tài khoản

Đối với giao diện quản lý tài khoản, hệ thống phân cấp nhỏ hơn thành 5 giao diện quản lý bao gồm: quản lý tài khoản sinh viên, giảng viên hướng dẫn, đại diện công ty, người đại diện và admin (áp dụng cho tài khoản admin cấp cao nhất trong hệ thống).

<hình ảnh>

Đối với tài khoản admin đăc biệt cao nhất sẽ có quyền quản lý thêm về tài khoản admin thấp hơn.

Trong giao diện này, admin sẽ thêm được các tài khoản sinh viên và giảng viên hướng dẫn, không thể thêm tài khoản đại diện công ty và người hướng dẫn. Nhưng admin có thể khóa hoặc mở khóa cho tất cả các tài khoản.

* + - 1. Giao diện quản lý công ty

<hình ảnh>

Tại giao diện quản lý công ty, admin sẽ xem được thông tin của công ty cũng như người đại diện. Admin không thể thêm công ty do công ty phải được đăng ký từ đối tác.

Admin có quyền xóa công ty trên mức logic nghiệp vụ, đối với trường hợp đặc biệt mới có quyền xóa công ty trên mặt dữ liệu.

* + - 1. Giao diện quản lý đề tài

**<hình ảnh>**

Tại giao diện quản lý đề tài, admin xem được tên đề tài, đề tài thuộc công ty nào, admin không thể thêm đề tài do đề tài được đăng tải là do yêu cầu nghiệp vụ của một công ty muốn tuyển thực tập.

Admin có quyền xóa đề tài trên mức logic nghiệp vụ, đối với trường hợp đặc biệt mới có quyền xóa đề tài trên mặt dữ liệu.

* + - 1. Giao diện quản lý File

**<hình ảnh>**

Tại giao diện quản lý file, admin đăng được các file lên hệ thống, các file này sẽ được công khai trên mọi giao diện trang chủ của các người dùng.

Admin có quyền xóa các file ra khỏi hệ thống trên mặt dữ liệu.

* + - 1. Giao diện quản lý sinh viên thực tập

Đối với quản lý thực tập, hệ thống sẽ phân cấp ra thành các giao diện quản lý nhỏ hơn bao gồm: quản lý sinh viên thực tập, quản lý đề tài sinh viên thực tập, quản lý điểm thi, quản lý file báo cáo của sinh viên, quản lý, quản lý kỳ thực tập,…

**<hình ảnh giao diện quản lý sinh viên thực tập>**

Tại giao diện quản lý sinh viên thực tập, admin sẽ xem được thông tin của sinh viên, và thay đổi trạng thái thực tập của sinh viên. Admin có thể tìm kiếm sinh viên theo các trường thông tin có trên màn hình.

**<hình ảnh giao diện quản lý đề tài sinh viên>**

Tại giao diện quản lý đề tài thực tập của sinh viên, admin sẽ xem được thông tin sinh viên đăng ký thực tập tại công ty nào, đề tài gì và ai hướng dẫn đề tài đó.

Admin có thể thay đổi trạng thái về đề tài sinh viên đăng ký với trường hợp sinh viên đã hoàn thành thực tập.

<hình ảnh giao diện quản lý điểm thi của sinh viên>

Tại giao diện quản lý điểm thi sinh viên, admin xem được điểm thực tập của sinh viên. Admin có thể sửa điểm thi của sinh viên, nhưng chỉ có thể sửa điểm báo cáo quá trình và báo cáo cuối kỳ, không thể sửa điểm phản hồi và điểm quá trình vì đây là điểm được tính tự động từ hệ thống.

<hình ảnh giao diện quản lý kỳ thực tập>

Tại giao diện quản lý kỳ thực tập, admin có quyền mở các đề tài thực tập mới hoặc đóng các kỳ thực tập đã hết khoảng thời gian thực tập.

<hình ảnh giao diện quản lý báo cáo thực tập của sinh viên>

Tại giao diện này ……………….

* + - 1. Giao diện quản lý thông báo

<hình ảnh giao diện quản lý thông báo>

Tại giao diện này, admin sẽ có quyền đăng các thông báo, các thông báo này sẽ được hiển thị trên các trang chủ của người dùng. Admin cũng có quyền xem hoặc xóa thông báo.

# KẾT LUẬN

1. Các kết quả đạt được

* Nắm bắt cơ bản nghiệp vụ của ứng dụng
* Hoàn thành hệ thống
* Có thể triển khai mở rộng

1. Những tồn tại trong quá trình phát triển

Trong quá trình phát triển đề tài: “Hệ thống đăng ký và quản lý thực tập tại doanh nghiệp” vẫn còn tồn tại một số những vấn đề chưa thể giải quyết ngay được và cần phải tìm hiểu thêm về kiến thức cũng như là các yêu cầu về nghiệp vụ của ứng dụng. Các tồn tại trong quá trình phát triển bao gồm:

* Chưa phân tích được nghiệp vụ của ứng dụng một cách rõ ràng, vẫn còn thiếu nhiều thông tin cần phải được phân tích một cách chi tiết hơn nữa.
* Chưa

1. Hướng phát triển

Để khắc phục các vấn đề còn tồn tại trong quá trình phát triển và tối ưu thì hướng phát triển của đề tài sẽ được đưa ra như sau:

* Tìm hiểu thêm về yêu cầu và phân tích thêm về nghiệp vụ của từng người dùng.
* Tối ưu hóa giao diện
* Tối ưu hóa chức năng sao cho có sự đáp ứng và phản hồi của hệ thống tới người dùng là nhanh nhất.
* Tối ưu hóa về dữ liệu và đảm bảo sự đồng nhất dữ liệu nếu phát triển phân tán.
* Triển khai lên host/ cloudserver và đưa vào thử nghiệm nếu có thể.

# TÀI LIỆU THAM KHẢO