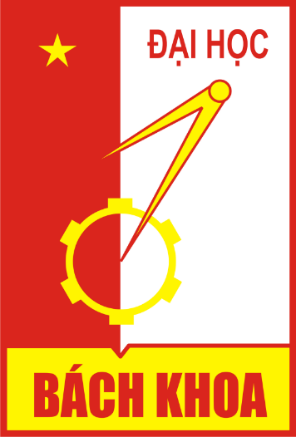
**Trường Đại Học Bách Khoa Hà Nội**

**Viện Điện Tử - Viễn Thông**



**PHÂN TÍCH THIẾT KẾ HƯỚNG ĐỐI TƯỢNG**

**Đề tài: HỆ THỐNG QUẢN LÝ SINH VIÊN**

Giáo viên hướng dẫn: ThS. Nguyễn Thanh Bình

**NHÓM 13**

Tạ Anh Tú 20134509 ĐTTT 04 – K58

Phan Thanh Việt 20134592 ĐTTT 07 – K58

Dương Văn Thông 20133806 ĐTTT 09 – K58

Phạm Thanh Bình 20130349 ĐTTT 08 – K58

Đặng Anh Tú 20134471 ĐTTT 01 – K58

*Hà Nội, 10/2017*

Mục lục

[**1.** **Lập kế hoạch** 3](#_Toc495405431)

[1.1. Khởi tạo dự án 3](#_Toc495405432)

[1.1.1. Giới thiệu đề tài 3](#_Toc495405433)

[1.1.2. Phân tích tính khả thi 3](#_Toc495405434)

[1.2. Quản trị dự án 8](#_Toc495405435)

[**2.** **Phân tích** 9](#_Toc495405436)

[2.1. Xác định rõ yêu cầu 9](#_Toc495405437)

[2.1.1. Yêu cầu chức năng 9](#_Toc495405438)

[2.1.2. Yêu cầu phi chức năng 11](#_Toc495405439)

[2.2. Mô hình hóa chức năng 11](#_Toc495405440)

[2.2.1. Xây dựng biểu đồ hoạt động (Activity diagram) 11](#_Toc495405441)

[2.2.2. Biểu đồ kịch bản sử dụng (use case) 21](#_Toc495405442)

[2.2.3. Mô tả kịch bản sử dụng use case 22](#_Toc495405443)

[2.2.3.1. Đối với người dùng chung 22](#_Toc495405444)

[2.2.3.2. Đối với Giáo vụ 25](#_Toc495405445)

[2.2.3.3. Đối với Giảng viên 33](#_Toc495405446)

[2.2.3.5. Đối với Sinh viên 34](#_Toc495405447)

[2.2.4. Danh sách Use Case 42](#_Toc495405448)

[2.3. Mô hình hóa cấu trúc 43](#_Toc495405449)

[2.3.1. CRC Card 43](#_Toc495405450)

[2.3.2. Biểu đồ lớp (Class diagram) 48](#_Toc495405451)

[2.4. Mô hình hóa hoạt động 48](#_Toc495405452)

[**3.** **Thiết kế** 51](#_Toc495405453)

[3.1. Thiết kế các lớp 51](#_Toc495405454)

[3.2. Thiết kế cơ sở dữ liệu 54](#_Toc495405455)

[3.3. Thiết kế lớp quản lý dữ liệu 54](#_Toc495405456)

[3.4. Thiết kế giao diện 57](#_Toc495405457)

[3.4.1. Giao diện cho Sinh viên 57](#_Toc495405458)

[3.4.2. Giao diện cho giảng viên 62](#_Toc495405459)

[3.4.3. Giao diện cho giáo vụ 62](#_Toc495405460)

[3.5. Thiết kế kiến trúc vật lý 62](#_Toc495405461)

[**4.** **Triển khai** 62](#_Toc495405462)

1. **Lập kế hoạch**

## 1.1. Khởi tạo dự án

### 1.1.1. Giới thiệu đề tài

Trong một trường đại học thì số lượng sinh viên thường là khá lớn, dẫn đến nảy sinh vấn đề làm thế nào để quản lý thông tin của tất cả sinh viên trong trường một cách đơn giản, tối ưu, dễ dàng tra cứu, truy cập, lấy thông tin… khi cần thiết. Từ nhu cầu đó nhóm đã quyết định chọn đề tài phân tích thiết kế hệ thống quản lý sinh viên trong trường Đại học Bách Khoa Hà Nội.

### 1.1.2. Phân tích tính khả thi

a) Khả thi về kỹ thuật

* Về mặt ứng dụng:

Dễ dàng sử dụng, tra cứu nhanh chóng, tiết kiệm thời gian cho cán bộ nhân viên. Có thể ứng dụng trong hầu hết tất cả các trường đại học.

* Về mặt công nghệ:

Nhóm dự tính sẽ sử dụng ngôn ngữ lập trình Java để thiết kế giao diện và hệ quản trị cơ sở dữ liệu MySQL để lưu trữ dữ liệu. Cụ thể, hệ thống sẽ được thiết kế theo mô hình MVC với:

Model: sử dụng hệ quản trị CSDL MySQL để lưu trữ dữ liệu

View: sử dụng HTML, CSS, Javascript để thiết kế giao diện (phía client)

Controller: sử dụng Java Servlet – JSP để điều khiển (phía Server)

* Yêu cầu kĩ thuật:

+ Phần cứng: cấu hình tối thiểu:

* Ram 2Gb
* CPU 1,7Ghz

+ Hệ điều hành:

* Window/Linux
* Hệ quản trị CSDL: MySQL
* Ngôn ngữ lập trình: Java
* IDE: Eclipse/Netbeans
* Trình duyệt: Chrome/Firefox
* Tính tương thích:

Có khả năng tương thích tốt với hệ thống hiện tại, sử dụng cơ sở dữ liệu sẵn có.

b) Khả thi về kinh tế

* Nhận diện về chi phí

|  |  |
| --- | --- |
| Chi phí phát triển | Chi phí vận hành |
| Lương đội ngũ phát triển  Phần cứng và phần mềm  Phí tư vấn | Nâng cấp phần mềm  Lệ phí Cấp phép Phần mềm  Sửa chữa phần cứng  Nâng cấp phần cứng  Lương của Đội Hoạt động  Phí truyền thông  Đào tạo người dùng |

* Nhận diện về doanh thu

|  |  |
| --- | --- |
| Doanh thu hữu hình | Doanh thu vô hình |
| Tăng doanh thu  Giảm nhân viên  Giảm trong Kiểm kê  Giảm chi phí CNTT  Giá nhà cung cấp tốt hơn | Tăng thị phần  Tăng cường công nhận thương hiệu  Sản phẩm chất lượng cao hơn  Dịch vụ khách hàng cải tiến  Mối quan hệ nhà cung cấp tốt hơn |

* Lưu chuyển tiền tệ

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 2017 | 2018 | 2019 | 2020 | 2021 | Tổng |
| Doanh thu |  |  |  |  |  |  |
| Doanh thu bán sản phẩm | 24.000000 | 30.000000 | 35.000000 | 40.000000 | 40.000000 |  |
| Chăm sóc khách hàng | 7.000000 | 7.000000 | 8.000000 | 8.000000 | 8.000000 |  |
| Quảng cáo | 5.000000 | 8.000000 | 10.000000 | 12.000000 | 14.000000 |  |
| **Tổng doanh thu từng năm** | 36.000000 | 45.000000 | 53.000000 | 60.000000 | 62.000000 |  |
| **Tổng doanh thu từng năm PV** | 31.034483 | 38.793103 | 45.689655 | 51.724138 | 53.448276 | 220.689655 |
| **Tổng doanh thu các năm PV** | 31.030000 | 69.823103 | 115.512759 | 167.236897 | 220.685172 |  |
|  |  |  |  |  |  |  |
| Chi phí phát triển |  |  |  |  |  |  |
| Lương đội ngũ phát triển | 8.000000 | 9.000000 | 10.000000 | 11.000000 | 11.000000 |  |
| Phần cứng và phần mềm | 20.000000 | 8.000000 | 6.000000 | 5.000000 | 4.000000 |  |
| Phí tư vấn | 3.000000 | 4.000000 | 5.000000 | 6.000000 | 7.000000 |  |
| **Tổng chi phí phát triển từng năm** | 31.000000 | 21.000000 | 21.000000 | 22.000000 | 22.000000 |  |
| **Tổng chi phí phát triển từng năm PV** | 26.724138 | 18.103448 | 18.103448 | 18.965517 | 18.965517 |  |
| **Tổng chi phí phát triển PV** | 26.720000 | 44.823448 | 62.926897 | 81.892414 | 100.857931 |  |
|  |  |  |  |  |  |  |
| Chi phí vận hành |  |  |  |  |  |  |
| Nâng cấp phần mềm | 2.000000 | 2.000000 | 3.000000 | 3.000000 | 3.000000 |  |
| Lệ phí Cấp phép Phần mềm | 15.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |  |
| Sửa chữa phần cứng | 2.000000 | 2.000000 | 2.000000 | 3.000000 | 3.000000 |  |
| Nâng cấp phần cứng | 1.000000 | 1.000000 | 2.000000 | 2.000000 | 3.000000 |  |
| Lương của Đội Hoạt động | 5.000000 | 5.000000 | 6.000000 | 6.000000 | 7.000000 |  |
| Phí truyền thông | 10.000000 | 5.000000 | 5.000000 | 4.000000 | 4.000000 |  |
| **Tổng chi phí vận hành từng năm** | 35.000000 | 15.000000 | 18.000000 | 18.000000 | 20.000000 |  |
| **Tổng chi phí vận hành từng năm PV** | 30.172414 | 12.931034 | 15.517241 | 15.517241 | 17.241379 |  |
| **Tổng chi phí vận hành PV** | 30.170000 | 43.101034 | 58.618276 | 74.135517 | 91.376897 |  |
|  |  |  |  |  |  |  |
| **Tổng chi phí từng năm** | 66.000000 | 36.000000 | 39.000000 | 40.000000 | 42.000000 |  |
| **Tổng chi phí từng năm PV** | 56.896552 | 31.034483 | 33.620690 | 34.482759 | 36.206897 | 192.241379 |
| **Tổng chi phí PV** | 56.900000 | 87.934483 | 121.555172 | 156.037931 | 192.244828 |  |
|  |  |  |  |  |  |  |
| **Lưu chuyển tiền tệ** | -30.000000 | 9.000000 | 14.000000 | 20.000000 | 20.000000 |  |
| **Giá trị ròng hiện tại** | -25.862069 | 7.758621 | 12.068966 | 17.241379 | 17.241379 | 28.448276 |
| **Giá trị ròng hiện tại tích lũy** | -25.860000 | -18.101379 | -6.032414 | 11.208966 | 28.450345 |  |
|  |  |  |  |  |  |  |
| Tỷ lệ hòa vốn | 14.80% |  |  |  |  |  |
| Điểm hòa vốn | 1.75 năm |  |  |  |  |  |

* Đồ thị điểm hòa vốn

c) Khả thi về tổ chức

* Đánh giá khả năng:

+ Hệ thống cuối cùng sẽ đáp ứng tốt người dùng và tích hợp tốt với hệ thống hiện tại.

+ Phần mềm quản lý sinh viên giúp tự động hóa các quy trình nghiệp vụ, đơn giản hóa công việc. Giúp công tác quản lý khoa học và quy củ.

* Tổ chức con người: Hệ thống được sử dụng bởi cán giáo viên, giáo vụ và sinh viên trong trường đại học Bách Khoa Hà Nội.

## 1.2. Quản trị dự án

* Quy mô dự án:

+ Số người tham gia: nhóm 5 người, teamlead: Tạ Anh Tú

+ Thời gian: 3 tháng cho việc lập kế hoạch, phân tích, thiết kế và triển khai.

+ Các tính năng:

* Đối với sinh viên: cho phép sinh viên tra cứu môn học, xem bảng điểm, tra cứu thời khóa biểu và có thể cập nhật một số thông tin cá nhân, qua từng kì, xem lịch thi, học phí...
* Đối với cán bộ quản lí: theo dõi, cập nhật nhanh thông tin sinh viên, quản lí điểm, lớp học, lịch thi cũng như một số thông tin khác có liên quan đến sinh viên…
* Hệ thống sẽ áp dụng cho sinh viên của trường đại học Bách Khoa Hà Nội
* Xây dựng bản kế hoạch cụ thể: dùng sơ đồ Grantt để thiết kế chi tiết bản kế hoạch, giúp thành viên thấy rõ các công việc cần làm và người đảm nhiệm
* Sau đây là bảng phân công công việc cho từng thành viên

|  |  |  |  |
| --- | --- | --- | --- |
| Task name | Person in charge | Start date | End date |
| Lập nhóm, đề xuất ý tưởng và tổng hợp các đề tài | Cả nhóm | 2017/08/28 | 2017/09/08 |
| Đánh giá, lựa chọn đề tài | Cả nhóm | 2017/08/28 | 2017/09/08 |
| Nhận biết dự án | Bình | 2017/09/04 | 2017/09/08 |
| Phân tích tính khả thi về kinh tế | Tú Đặng | 2017/09/11 | 2017/09/15 |
| Phân tích tính khả thi về kỹ thuật, tổ chức | Việt | 2017/09/11 | 2017/09/16 |
| Xác định quy mô dự án | Bình | 2017/09/16 | 2017/09/18 |
| Xây dựng và tạo bảng kế hoạch | Tú Tạ | 2017/09/18 | 2017/09/19 |
| Phân công công việc | Tú Tạ | 2017/09/20 | 2017/09/20 |
| Phân tích các yêu cầu chức năng, phi chức năng | Bình | 2017/09/18 | 2017/09/20 |
| Xây dựng biểu đồ hoạt động (Activity diagram) | Thông |  |  |
| Xây dựng biểu đồ kịch bản sử dụng (Use case) | Tú Tạ | 2017/09/18 | 2017/09/21 |
| Mô tả kịch bản sử dụng use case | Bình | 2017/09/26 |  |
| Thiết kế CRC cards | Việt | 2017/09/26 |  |
| Xây dựng biểu đồ lớp (Class diagram) | Tú Tạ | 2017/09/21 | 2017/10/05 |
| Xây dựng biểu đồ tuần tự (Sequence diagram) | Thông | 2017/09/18 | 2017/09/26 |
| Xây dựng máy trạng thái (Behavioral State Machine diagrams) | Thông |  |  |
| Đặc tả hoạt động của hệ thống |  |  |  |
| Thiết kế CSDL | Tú Tạ | 2017/09/21 | 2017/09/27 |
| Thiết kế lớp và phương thức |  |  |  |
| Thiết kế lớp quản lý dữ liệu |  |  |  |
| Thiết kế giao diện người dùng |  |  |  |
| Thiết kế kiến trúc vật lý |  |  |  |
| Cài đặt và vận hành |  |  |  |
| Testing, debug |  |  |  |
| Phát hành sản phẩm |  |  |  |

1. **Phân tích**

## 2.1. Xác định rõ yêu cầu

### 2.1.1. Yêu cầu chức năng

Người dùng của hệ thống được chia thành 3 loại: giáo viên, giáo vụ và sinh viên. Với mỗi loại người dùng hệ thống có các yêu cầu chức năng sau:

* Với giáo viên:
  + Đăng nhập, đăng xuất, đổi mật khẩu
  + Quản lý điểm SV : nhập điểm, sửa điểm, tổng kết điểm và submit điểm của sinh viên lên server
  + Quản lý lịch học: xem lịch giảng dạy của mình (việc đăng ký dạy lớp nào sẽ không đề cập đến trong hệ thống này vì việc này giáo viên sẽ làm việc với viện và nhà trường)
* Với giáo vụ:
  + Đăng nhập, đăng xuất, đổi mật khẩu
  + Quản lý sinh viên (SV):
    - Thêm sinh viên
    - Tìm kiếm SV, sửa thông tin của sinh viên
    - Khôi phục tài khoản cho SV nào quên mật khẩu
    - Tạo lớp SV (VD lớp ĐTTT 04 – K58), chỉnh sửa thông tin từng lớp
  + Quản lý lịch thi
    - Tạo và chỉnh sửa lịch thi
  + Quản lý lớp học
    - Tạo danh sách lớp học cho từng học phần, từng học kỳ
    - Tạo bảng phân công giảng dạy cho giáo viên
* Với sinh viên (SV) :
  + Đăng nhập, đăng xuất, đổi mật khẩu
  + Cập nhật hồ sơ SV của chính mình
  + Truy cập thông tin cá nhân
    - Điểm : xem điểm theo môn học, theo học kì ; in bảng điểm
    - Lớp SV : xem danh sách SV trong một lớp nào đó, in danh sách
    - Môn học : xem danh mục học phần, thời khóa biểu, lịch thi
    - Tra cứu học phí
    - Xem phân công giảng dạy của giảng viên, thông tin giảng viên, trang web cá nhân và tài liệu môn học (nếu có)

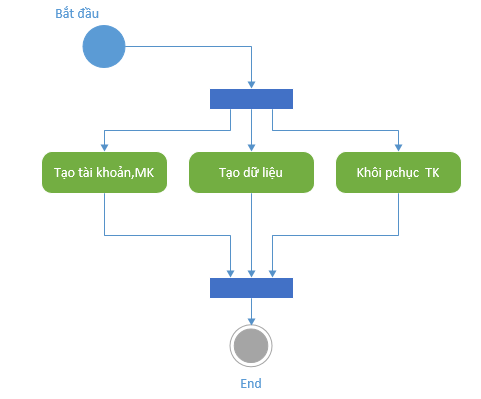
### 2.1.2. Yêu cầu phi chức năng

* Hiệu năng hoạt động
  + Có thể phục vụ nhiều người cùng truy cập 1 lúc mà không bị treo máy
  + Đáp ứng nhanh
* Khả năng tương thích
  + Tương thích với hệ thống và các loại hệ điều hành khác nhau
* Tính khả dụng
  + Hiển thị thông tin rõ ràng, chuẩn xác
  + Dễ dàng cập nhật, chỉnh sửa
  + Giao diện đơn giản, dễ thao tác, sử dụng
  + Dễ dàng bảo trì, sửa đổi khi có lỗi
* Tính tin cậy
  + Ít bug, đảm bảo tính bảo mật
  + Đảm bảo tính toàn vẹn, tính xác thực

## 2.2. Mô hình hóa chức năng

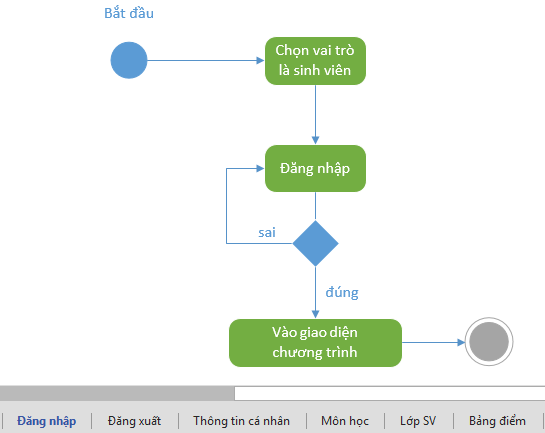
### 2.2.1. Xây dựng biểu đồ hoạt động (Activity diagram)

a) Dành cho admin

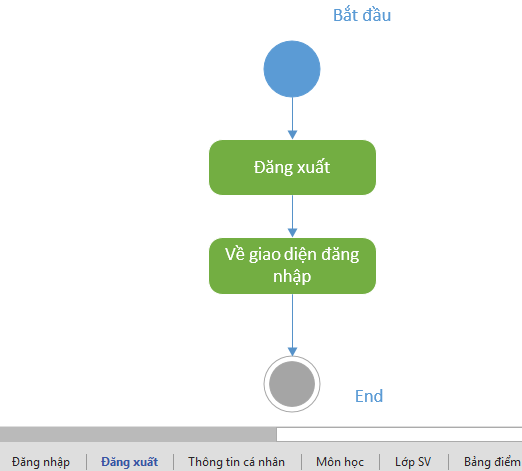


b) Dành cho sinh viên

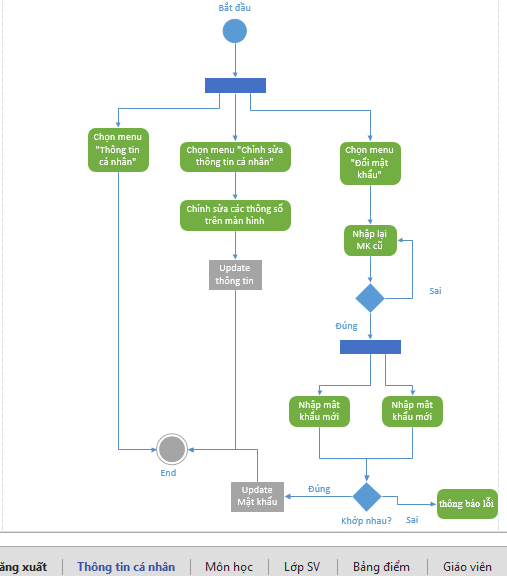
* Đăng nhập



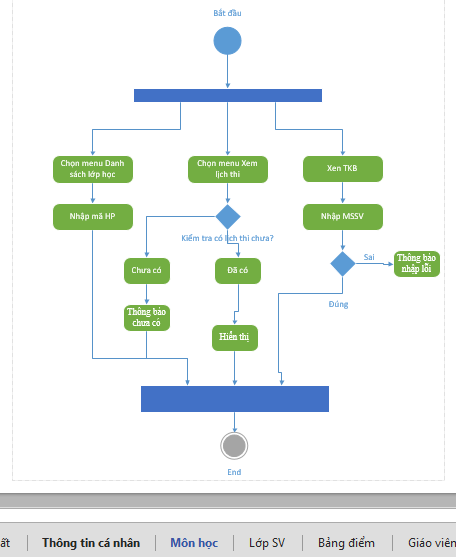
* Đăng xuất



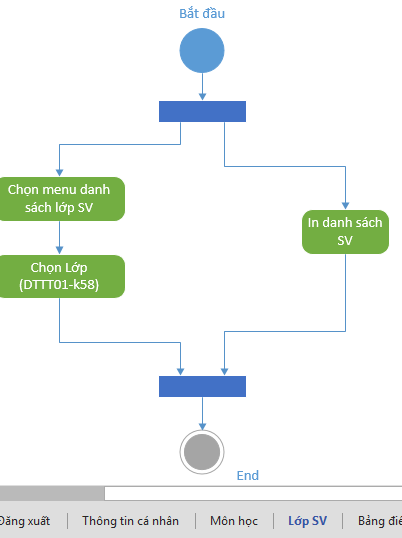
* Thông tin cá nhân



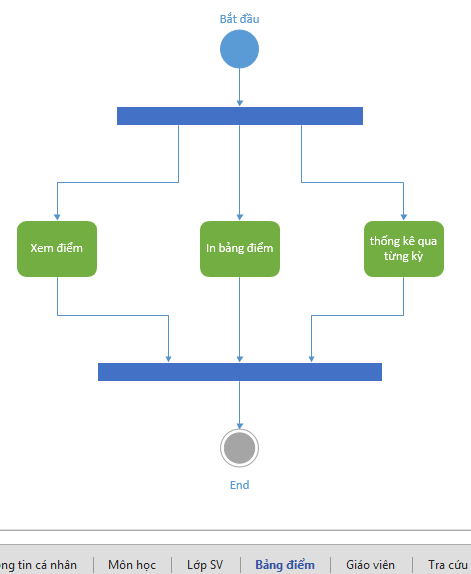
* Môn học



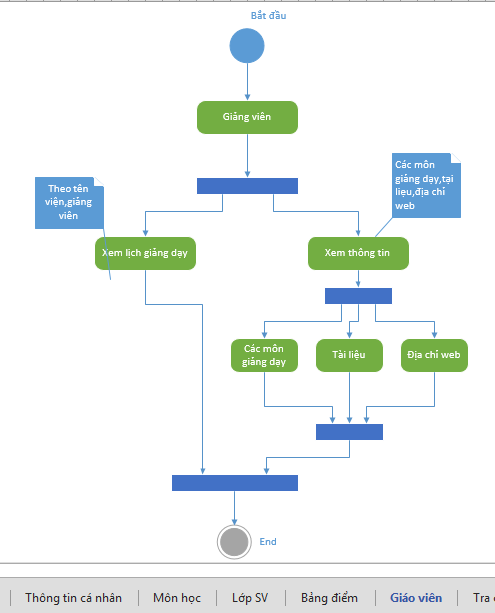
* Lớp sinh viên



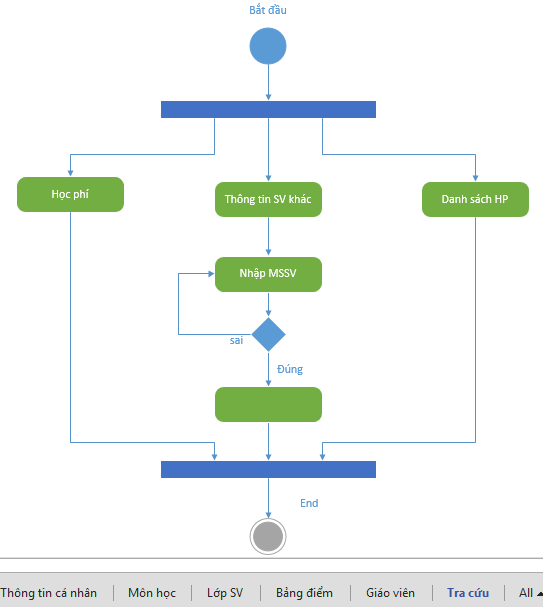
* Bảng điểm



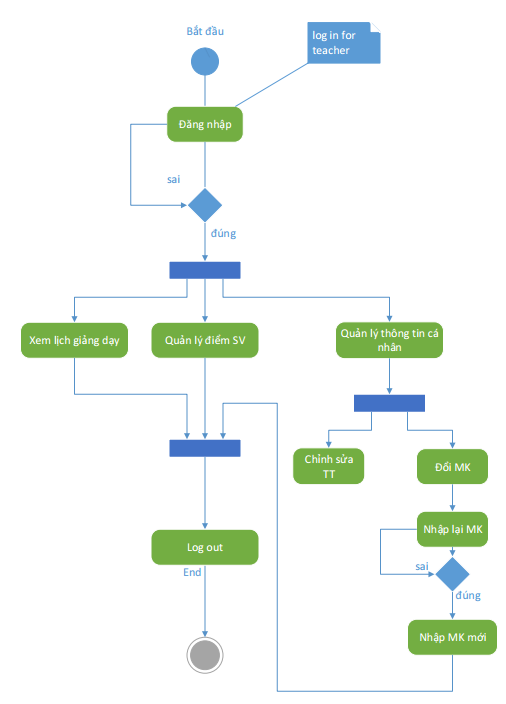
* Giáo viên



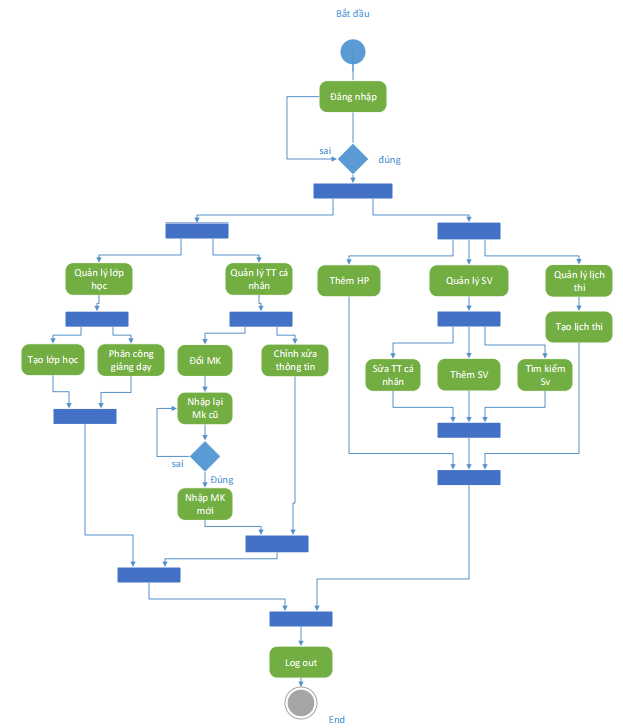
* Tra cứu



c) Dành cho giảng viên

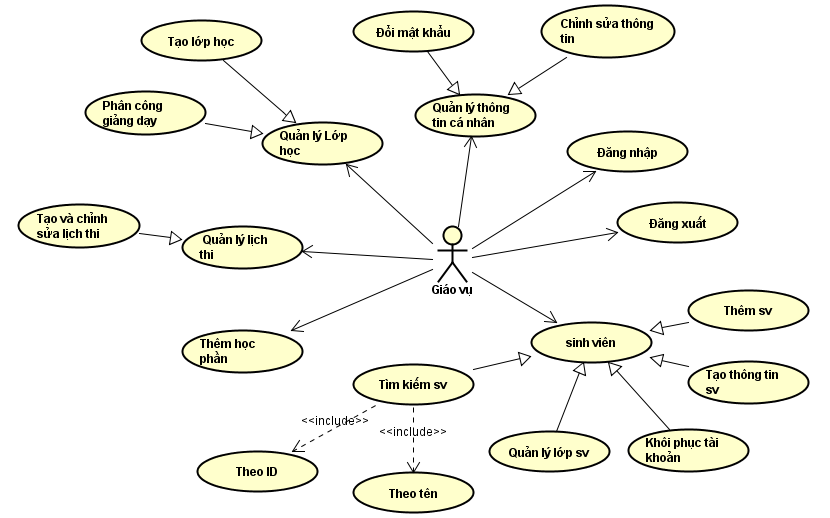


d) Dành cho giáo vụ

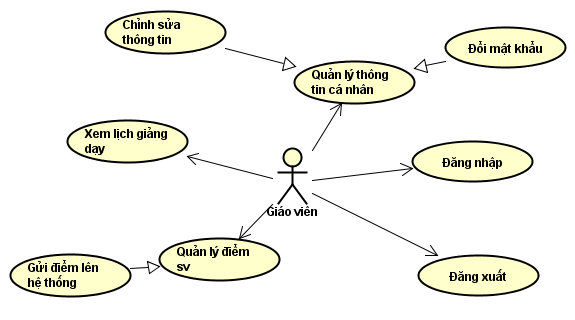


### 2.2.2. Biểu đồ kịch bản sử dụng (use case)

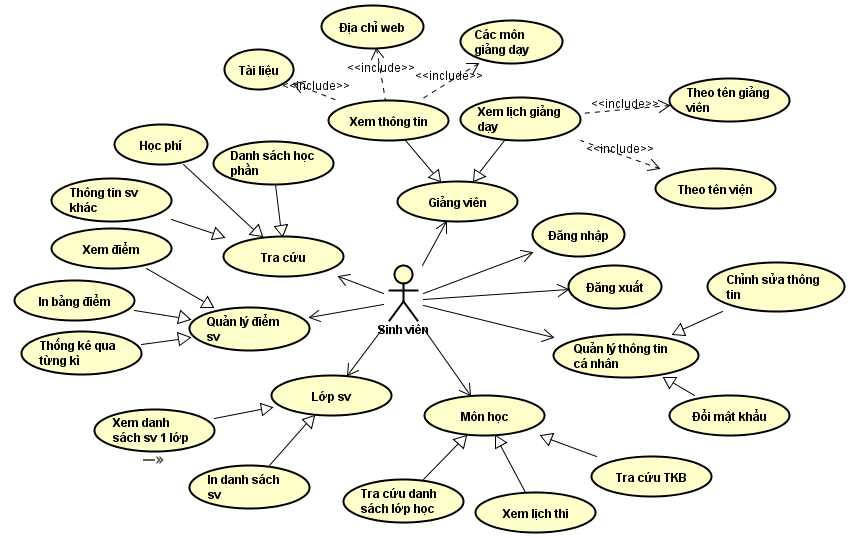
a) Use case đối với tác nhân là giáo vụ



b) Use case đối với tác nhân là giáo viên



c) Use case đối với tác nhân là sinh viên



### 2.2.3. Mô tả kịch bản sử dụng use case

#### 2.2.3.1. Đối với người dùng chung

a) Use case đăng nhập

* **Luồng sự kiện chính:**
* 1: Ban đầu vào hệ thống, người dùng bắt buộc phải đăng nhập để sử dụng hệ thống. Đầu tiên chọn role để đăng nhập (là SV, GV hay giáo vụ)
* 2: Hệ thống hiển thị giao diện đăng nhập, yêu cầu nhập username, password
* 3: Người dùng nhập username và password, nhấn đồng ý đăng nhập
* 4: Hệ thống tiếp nhận, kiểm tra username và password
* 5: Nếu hợp lệ, hệ thống chấp nhận đăng nhập, hệ thống chuyển đến trang chủ
* 6: Kết thúc use case
* **Luồng sự kiện rẽ nhánh:**

*Luồng 1:*

* 1: Tại giao diện đăng nhập, người dùng hủy đăng nhập, chọn Hủy bỏ
* 2: Hệ thống quay lại giao diện trước đó (giao diện chọn role để đăng nhập)
* 3: Kết thúc use case

*Luồng 2:*

* 1: Hệ thống kiểm tra thông tin đăng nhập không chính xác
* 2: Hệ thống hiển thị thông báo rằng đã nhập sai tên tài khoản hoặc mật khẩu
* 3: Kết thúc use case

*Luồng 3:*

* 1: Hệ thống kết nối cơ sở dữ liệu để kiểm tra không thành công, không kiểm tra được
* 2: Hệ thống hiển thị thông báo hệ thống đang bảo trì
* 3: Kết thúc use case

b) Use Case “Đăng xuất”

* **Luồng sự kiện chính:**
* 1: Người dùng chọn Đăng xuất
* 2: Hệ thống chấp nhận đăng xuất người dùng
* 3: Kết thúc use case

c) Use Case “Cập nhật thông tin cá nhân”

* **Luồng sự kiện chính:**
* 1: Người dùng chọn chức năng chỉnh sửa thông tin cá nhân
* 2: Hệ thống hiển thị giao diện chỉnh sửa thông tin cá nhân
* 3: Người dùng cập nhật thông tin cá nhân
* 4: Người dùng chọn Lưu lại thông tin cá nhân
* 5: Hệ thống cập nhật thông tin cá nhân và hiển thị thông báo thành công
* 6: Kết thúc use case
* **Luồng sự kiện rẽ nhánh:**
* 1: Người dùng cập nhật thông tin nhưng không muốn lưu lại, chọn Home
* 2: Hệ thống hủy bỏ việc cập nhật thông tin, trở lại giao diện chính
* 3: Kết thúc use case

d) Use Case “Sửa mật khẩu”:

* **Luồng sự kiện chính:**
* 1: Người dùng chọn chức năng sửa mật khẩu
* 2: Hệ thống hiển thị giao diện sửa mật khẩu
* 3: Người dùng cập nhật mật khẩu mới
* 4: Người dùng chọn Lưu lại mật khẩu
* 5: Hệ thống tiếp nhận mật khẩu mới, kiểm tra mật khẩu có hợp lệ không
* 6: Nếu hợp lệ, hệ thống cập nhật mật khẩu, hiển thị thông báo thành công
* 6: Kết thúc use case
* **Luồng sự kiện rẽ nhánh:**

*Luồng 1:*

* 1: Người dùng cập nhật mật khẩu nhưng không muốn lưu lại, chọn Home
* 2: Hệ thống hủy bỏ việc thay đổi mật khẩu, trở lại giao diện chính
* 3: Kết thúc use case

*Luồng 2:*

* 1: Người dùng nhập mật khẩu mới nhưng không hợp lệ, hệ thống hiển thị thông báo lỗi, yêu cầu nhập lại mật khẩu
* 2: Người dùng nhập lại mật khẩu
* 3: Hệ thống tiếp nhận, kiểm tra mật khẩu vừa nhập. Nếu hợp lệ thì cập nhật mật khẩu, thông báo thành công. Nếu không hợp lệ thì lặp lại bước 1
* 4: Kết thúc use case

#### 2.2.3.2. Đối với Giáo vụ

2.2.3.2.1. Quản lý sinh viên

a) Use Case “Thêm sinh viên”

* **Luồng sự kiện chính:**
* 1: Người dùng chọn chức năng thêm sinh viên mới
* 2: Hệ thống hiển thị giao diện thêm mới
* 3: Người dùng thêm thông tin mong muốn, chọn Lưu lại
* 4: Hệ thống lưu thông tin, hiển thị thông báo thành công
* 5: Kết thúc use case
* **Luồng sự kiện rẽ nhánh:**

*Luồng 1:*

* 1: Người dùng nhập trùng ID
* 2: Hệ thống hiển thị thông báo nhập trùng ID, yêu cầu nhập lại
* 3: Người dùng nhập lại ID
* 4: Hệ thống tiếp nhận, tạo sinh viên theo ID ở trên
* 5: Nếu ID của sinh viên mới không bị trùng nữa, hệ thống hiển thị thêm thành công. Nếu trùng thì lặp lại bước 2
* 6: Kết thúc use case

*Luồng 2:*

* 1: Người dùng muốn hủy bỏ việc thêm sinh viên, chọn Home
* 2: Hệ thống hủy bỏ việc thêm sinh viên, quay lại giao diện chính
* 3: Kết thúc use case

b) Use Case “Chỉnh sửa thông tin sinh viên”

* **Luồng sự kiện chính:**
* 1: Người dùng chọn chức năng sửa thông tin sinh viên
* 2: Hệ thống hiển thị giao diện chỉnh sửa thông tin sinh viên, yêu cầu thực hiện tìm kiếm sinh viên
* 3: Người dùng nhập ID sinh viên, chọn Search
* 4: Hệ thống thực hiện tìm kiếm sinh viên, hiển thị thông tin cần tìm
* 5: Người dùng chỉnh sửa thông tin mong muốn, chọn Lưu lại
* 6: Hệ thống cập nhật thông tin, hiển thị thông báo thành công
* 7: Kết thúc use case
* **Luồng sự kiện rẽ nhánh:**

*Luồng 1:*

* 1: Người dùng nhập sai ID, hệ thống không tìm kiếm được
* 2: Hệ thống hiển thị thông báo nhập sai ID, yêu cầu nhập lại
* 3: Người dùng nhập lại ID
* 4: Hệ thống tiếp nhận, tìm kiếm thông tin
* 5: Nếu tìm được, hệ thống hiển thị thông tin sinh viên lên giao diện chỉnh sửa. Nếu không lặp lại bước 2
* 6: Kết thúc use case

*Luồng 2:*

* 1: Người dùng muốn hủy bỏ việc chỉnh sửa thông tin, chọn Home
* 2: Hệ thống hủy bỏ việc chỉnh sửa thông tin, quay lại giao diện chính
* 3: Kết thúc use case

c) Use Case “Thêm lớp sinh viên”

* **Luồng sự kiện chính:**
* 1: Người dùng chọn chức năng tạo mới lớp sinh viên
* 2: Hệ thống hiển thị giao diện tạo mới lớp sinh viên
* 4: Người dùng nhập thông tin mong muốn, chọn Lưu lại
* 5: Hệ thống lưu thông tin, hiển thị thông báo thành công
* 6: Kết thúc use case
* **Luồng sự kiện rẽ nhánh:**

*Luồng 1:*

* 1: Người dùng nhập trùng ID với lớp khác đã tồn tại
* 2: Hệ thống hiển thị thông báo nhập trùng ID, yêu cầu nhập lại
* 3: Người dùng nhập lại ID
* 4: Hệ thống tiếp nhận, thêm lớp mới vào database
* 5: Nếu ID lớp mới không bị trùng, hệ thống hiển thị thông báo thành công. Nếu trùng ID thì lặp lại bước 2
* 6: Kết thúc use case

*Luồng 2:*

* 1: Người dùng muốn hủy bỏ việc tạo mới lớp SV, chọn Home
* 2: Hệ thống hủy bỏ việc thêm mới lớp SV, quay lại giao diện chính
* 3: Kết thúc use case

d) Use Case “Chỉnh sửa lớp sinh viên”

* **Luồng sự kiện chính:**
* 1: Người dùng chọn chức năng sửa thông tin lớp sinh viên
* 2: Hệ thống hiển thị giao diện chỉnh sửa thông tin lớp sinh viên, yêu cầu thực hiện tìm kiếm lớp
* 3: Người dùng tìm kiếm lớp sinh viên theo ID hoặc tên
* 4: Hệ thống thực hiện tìm kiếm lớp sinh viên, hiển thị thông tin cần tìm
* 5: Người dùng chỉnh sửa thông tin mong muốn, chọn Lưu lại
* 6: Hệ thống cập nhật thông tin, hiển thị thông báo thành công
* 7: Kết thúc use case
* **Luồng sự kiện rẽ nhánh:**

*Luồng 1:*

* 1: Người dùng nhập sai ID, hệ thống không tìm kiếm được
* 2: Hệ thống hiển thị thông báo nhập sai ID, yêu cầu nhập lại
* 3: Người dùng nhập lại ID
* 4: Hệ thống tiếp nhận, tìm kiếm thông tin
* 5: Nếu tìm được, hệ thống hiển thị thông tin lớp sinh viên lên giao diện chỉnh sửa. Nếu không lặp lại bước 2
* 6: Kết thúc use case

*Luồng 2:*

* 1: Người dùng muốn hủy bỏ việc chỉnh sửa thông tin, chọn Home
* 2: Hệ thống hủy bỏ việc chỉnh sửa thông tin, quay lại giao diện chính
* 3: Kết thúc use case

e) Khôi phục tài khoản

* **Luồng sự kiện chính:**
* 1: Người dùng chọn chức năng Khôi phục tài khoản
* 2: Hệ thống hiển thị giao diện Khôi phục tài khoản, yêu cầu nhập ID
* 3: Người dùng nhập ID, chọn Search
* 4: Hệ thống thực hiện tìm kiếm, hiển thị thông tin cần tìm
* 5: Người dùng chọn Khôi phục tài khoản
* 6: Hệ thống thực hiện khôi phục tài khoản bằng cách cập nhật vào CSDL username, password mới.
* 7: Kết thúc use case
* **Luồng sự kiện rẽ nhánh:**

*Luồng 1:*

* 1: Người dùng nhập sai ID, hệ thống không tìm kiếm được
* 2: Hệ thống hiển thị thông báo nhập sai ID, yêu cầu nhập lại
* 3: Người dùng nhập lại ID
* 4: Hệ thống tiếp nhận, tìm kiếm thông tin
* 5: Nếu tìm được, hệ thống hiển thị thông tin. Nếu không lặp lại bước 2
* 6: Kết thúc use case

*Luồng 2:*

* 1: Người dùng muốn hủy bỏ việc khôi phục tài khoản, chọn Home
* 2: Hệ thống hủy bỏ việc khôi phục tài khoản, quay lại giao diện chính
* 3: Kết thúc use case

2.2.3.2.2. Use case thêm học phần

* **Luồng sự kiện chính:**
* 1: Người dùng chọn chức năng sửa Thêm học phần
* 2: Hệ thống hiển thị giao diện Thêm học phần, yêu cầu thực hiện điền thông tin học phần cần thêm
* 3: Người dùng điền thông tin học phần, chọn Lưu lại
* 4: Hệ thống cập nhật thông tin, hiển thị thông báo thành công
* 5: Kết thúc use case
* **Luồng sự kiện rẽ nhánh:**

*Luồng 1:*

* 1: Người dùng muốn hủy bỏ việc thêm học phần, chọn Home
* 2: Hệ thống hủy bỏ việc thêm học phần, quay lại giao diện chính
* 3: Kết thúc use case

*Luồng 2:*

* 1: Người dùng nhập trùng mã học phần
* 2: Hệ thống hiển thị thông báo lỗi
* 3: Người dùng nhập lại. Nếu trùng thì quay lại bước 2, nếu không thì tiếp tục
* 4: Hiển thị thêm thành công

2.2.3.2.3. “Quản lý lớp học”

a) Use case thêm lớp học

* **Luồng sự kiện chính:**
* 1: Người dùng chọn chức năng Quản lý lớp học/Thêm lớp học
* 2: Hệ thống hiển thị giao diện Thêm lớp học
* 3: Người dùng điền thông tin lớp học (bao gồm phân công giảng dạy), chọn Lưu lại
* 4: Hệ thống cập nhật thông tin, hiển thị thông báo thành công
* 5: Kết thúc use case
* **Luồng sự kiện rẽ nhánh:**

*Luồng 1:*

* 1: Người dùng muốn hủy bỏ việc thêm lớp học, chọn Home
* 2: Hệ thống hủy bỏ việc thêm lớp học, quay lại giao diện chính
* 3: Kết thúc use case

b) Use Case “Xóa lớp học” (ít xảy ra)

* **Luồng sự kiện chính:**
* 1: Người dùng chọn chức năng Quản lý lớp học/Xóa lớp học
* 2: Hệ thống hiển thị giao diện Xóa lớp học (danh sách lớp), yêu cầu nhập ID lớp cần xóa
* 3: Người dùng điền ID lớp học, chọn Search
* 4: Hệ thống tìm kiếm thông tin, hiển thị trên danh sách
* 5: Người dùng chọn Xóa lớp
* 6: Hệ thống xóa lớp, cập nhật lại danh sách, thông báo thành công
* 5: Kết thúc use case
* **Luồng sự kiện rẽ nhánh:**

*Luồng 1:*

* 1: Người dùng muốn hủy bỏ việc xóa lớp học, chọn Home
* 2: Hệ thống hủy bỏ việc thêm lơp học, quay lại giao diện chính
* 3: Kết thúc use case

*Luồng 2:*

* 1: Người dùng nhập sai ID, hệ thống không tìm kiếm được
* 2: Hệ thống hiển thị thông báo nhập sai ID, yêu cầu nhập lại
* 3: Người dùng nhập lại ID
* 4: Hệ thống tiếp nhận, tìm kiếm lớp sinh viên
* 5: Nếu tìm được, hệ thống hiển thị lớp sinh viên trên danh sách. Nếu không lặp lại bước 2
* 6: Kết thúc use case

2.2.3.2.4. Use case thêm lịch thi

* **Luồng sự kiện chính:**
* 1: Người dùng chọn chức năng Quản lý lịch thi
* 2: Hệ thống hiển thị giao diện thêm lịch thi
* 3: Người dùng tạo và chỉnh sửa lịch thi
* 4: Hệ thống cập nhật lịch thi, thông báo thành công
* 5: Kết thúc use case
* **Luồng sự kiện rẽ nhánh:**

*Luồng 1:*

* 1: Người dùng muốn hủy bỏ việc quản lý lịch thi, chọn Home
* 2: Hệ thống hủy bỏ việc quản lý lịch thi, quay lại giao diện chính
* 3: Kết thúc use case

#### 2.2.3.3. Đối với Giảng viên

a) Use Case “Xem lịch giảng dạy”

* **Luồng sự kiện chính:**
* 1: Người dùng chọn chức năng Xem lịch giảng dạy
* 2: Hệ thống hiển thị giao diện Xem lịch giảng dạy của chính giảng viên đó
* 3: Hệ thống hiển thị thông tin Lịch giảng dạy
* 4: Kết thúc use case

b) Use case “Submit điểm sinh viên” (gửi điểm của SV lên hệ thống)

* **Luồng sự kiện chính:**
* 1: Người dùng chọn chức năng Submit điểm sinh viên
* 2: Hệ thống hiển thị giao diện Quản lý điểm sinh viên, yêu cầu nhập ID lớp học
* 3: Người dùng nhập ID, chọn Search
* 4: Hệ thống tìm kiếm, hiển thị danh sách lớp học
* 5: Người dùng nhập điểm, chọn Lưu lại
* 6: Hệ thống cập nhật điểm, hiển thị thông báo thành công
* 7: Kết thúc use case
* **Luồng sự kiện rẽ nhánh:**

*Luồng 1:*

* 1: Người dùng muốn hủy bỏ việc quản lý điểm, chọn Home
* 2: Hệ thống hủy bỏ việc quản lý điểm, quay lại giao diện chính
* 3: Kết thúc use case

*Luồng 2:*

* 1: Người dùng nhập sai ID, hệ thống không tìm kiếm được
* 2: Hệ thống hiển thị thông báo nhập sai ID, yêu cầu nhập lại
* 3: Người dùng nhập lại ID
* 4: Hệ thống tiếp nhận, tìm kiếm thông tin
* 5: Nếu tìm được, hệ thống hiển thị thông tin về lớp học. Nếu không lặp lại bước 2
* 6: Kết thúc use case

#### 2.2.3.5. Đối với Sinh viên

a) Use Case “Xem thời khóa biểu”

* **Luồng sự kiện chính:**
* 1: Người dùng chọn chức năng Xem thời khóa biểu
* 2: Hệ thống hiển thị giao diện Thời khóa biểu, yêu cầu nhập ID sinh viên
* 3: Người dùng nhập ID, chọn Search
* 4: Hệ thống tìm kiếm, hiển thị thời khóa biểu
* 5: Kết thúc use case
* **Luồng sự kiện rẽ nhánh:**

*Luồng 1:*

* 1: Người dùng muốn hủy bỏ việc xem thời khóa biểu, chọn Home
* 2: Hệ thống hủy bỏ việc xem thời khóa biểu, quay lại giao diện chính
* 3: Kết thúc use case

*Luồng 2:*

* 1: Người dùng nhập sai ID, hệ thống không tìm kiếm được
* 2: Hệ thống hiển thị thông báo nhập sai ID, yêu cầu nhập lại
* 3: Người dùng nhập lại ID
* 4: Hệ thống tiếp nhận, tìm kiếm thông tin
* 5: Nếu tìm được, hệ thống hiển thị thời khóa biểu . Nếu không lặp lại bước 2
* 6: Kết thúc use case

b) Use Case “Xem lịch thi”

* **Luồng sự kiện chính:**
* 1: Người dùng chọn chức năng Xem lịch thi (theo kì đang học)
* 2: Hệ thống hiển thị lịch thi của chính sinh viên đó
* 3: Kết thúc use case

c) Use Case Xem thông các lớp học đang mở

* **Luồng sự kiện chính:**
* 1: Người dùng chọn chức năng Danh sách lớp học
* 2: Hệ thống hiển thị giao diện tất cả lớp học
* 3: Người dùng có thể nhập thể nhập mã lớp học để Search
* 4: Hệ thống tìm kiếm, hiển thị thông tin lớp đó
* 5: Kết thúc use case
* **Luồng sự kiện rẽ nhánh:**

*Luồng 1:*

* 1: Người dùng muốn hủy bỏ việc xem thông tin lớp học, chọn Home
* 2: Hệ thống hủy bỏ việc xem thông tin lớp học, quay lại giao diện chính
* 3: Kết thúc use case

*Luồng 2:*

* 1: Người dùng nhập sai ID, hệ thống không tìm kiếm được
* 2: Hệ thống nói không có gì để hiển thị
* 3: Kết thúc use case

d) Use Case “Xem danh sách lớp sinh viên”

* **Luồng sự kiện chính:**
* 1: Người dùng chọn chức năng Xem danh sách lớp sinh viên
* 2: Hệ thống hiển thị giao diện Danh sách lớp, yêu cầu nhập ID lớp sinh viên
* 3: Người dùng nhập ID, chọn Search
* 4: Hệ thống tìm kiếm, hiển thị danh sách lớp
* 5: Kết thúc use case
* **Luồng sự kiện rẽ nhánh:**

*Luồng 1:*

* 1: Người dùng muốn hủy bỏ việc xem danh sách lớp, chọn Home
* 2: Hệ thống hủy bỏ việc xem danh sách lớp, quay lại giao diện chính
* 3: Kết thúc use case

*Luồng 2:*

* 1: Người dùng nhập sai ID, hệ thống không tìm kiếm được
* 2: Hệ thống hiển thị thông báo nhập sai ID, yêu cầu nhập lại
* 3: Người dùng nhập lại ID
* 4: Hệ thống tiếp nhận, tìm kiếm thông tin
* 5: Nếu tìm được, hệ thống hiển thị danh sách lớp. Nếu không lặp lại bước 2
* 6: Kết thúc use case

e) Use Case “In danh sach sinh viên” (sau khi đã chọn chức năng “Xem danh sách lớp sinh viên” ở trên)

* **Luồng sự kiện chính:**
* 1: Người dùng chọn chức năng In danh sách lớp sinh viên
* 2: Hệ thống in danh sách lớp
* 3: Kết thúc use case

f) Use Case “Xem bảng điểm”

* **Luồng sự kiện chính:**
* 1: Người dùng chọn chức năng Tra cứu điểm
* 2: Hệ thống hiển thị bảng điểm
* 3: Kết thúc use case

g) Use Case “In bảng điểm” (Sau khi đã chọn “Xem bảng điểm” ở trên)

* **Luồng sự kiện chính:**
* 1: Người dùng chọn chức năng in bảng điểm
* 2: Hệ thống in bảng điểm của sinh viên đó
* 3: Kết thúc use case

h) Use Case “Tra cứu danh sách học phần”

* **Luồng sự kiện chính:**
* 1: Người dùng chọn chức năng Tra cứu học phần
* 2: Hệ thống hiển thị giao diện Tra cứu học phần, yêu cầu nhập ID học phần
* 3: Người dùng nhập ID, chọn Search
* 4: Hệ thống tìm kiếm, hiển thị thông tin học phần
* 5: Kết thúc use case
* **Luồng sự kiện rẽ nhánh:**

*Luồng 1:*

* 1: Người dùng muốn hủy bỏ việc xem thông tin học phần, chọn Home
* 2: Hệ thống hủy bỏ việc xem thông tin học phần, quay lại giao diện chính
* 3: Kết thúc use case

*Luồng 2:*

* 1: Người dùng nhập sai ID, hệ thống không tìm kiếm được
* 2: Hệ thống hiển thị thông báo nhập sai ID, yêu cầu nhập lại
* 3: Người dùng nhập lại ID
* 4: Hệ thống tiếp nhận, tìm kiếm thông tin
* 5: Nếu tìm được, hệ thống hiển thị thông tin học phần. Nếu không lặp lại bước 2
* 6: Kết thúc use case

i) Use Case “Tra cứu học phí”

* **Luồng sự kiện chính:**
* 1: Người dùng chọn chức năng Tra cứu học phí
* 2: Hệ thống hiển thị giao diện Tra cứu học phí, yêu cầu nhập ID sinh viên
* 3: Người dùng nhập ID, chọn Search
* 4: Hệ thống tìm kiếm, hiển thị thông tin học phí (tổng học phí phải nộp, học phí theo tín chỉ học phí, số tín chỉ học phí đang học …)
* 5: Kết thúc use case
* **Luồng sự kiện rẽ nhánh:**

*Luồng 1:*

* 1: Người dùng muốn hủy bỏ việc xem học phí, chọn Home
* 2: Hệ thống hủy bỏ việc xem học phí, quay lại giao diện chính
* 3: Kết thúc use case

*Luồng 2:*

* 1: Người dùng nhập sai ID, hệ thống không tìm kiếm được
* 2: Hệ thống hiển thị thông báo nhập sai ID, yêu cầu nhập lại
* 3: Người dùng nhập lại ID
* 4: Hệ thống tiếp nhận, tìm kiếm thông tin
* 5: Nếu tìm được, hệ thống hiển thị thông tin học phí. Nếu không lặp lại bước 2
* 6: Kết thúc use case

j) Use Case “Tra cứu thông tin sinh viên khác”

* **Luồng sự kiện chính:**
* 1: Người dùng chọn chức năng Tra cứu thông tin sinh viên
* 2: Hệ thống hiển thị giao diện Tra cứu thông tin sinh viên, yêu cầu nhập ID sinh viên
* 3: Người dùng nhập ID, chọn Search
* 4: Hệ thống tìm kiếm, hiển thị thông tin sinh viên
* 5: Kết thúc use case
* **Luồng sự kiện rẽ nhánh:**

*Luồng 1:*

* 1: Người dùng muốn hủy bỏ việc xem thông tin sinh viên, chọn Home
* 2: Hệ thống hủy bỏ việc xem thông tin sinh viên, quay lại giao diện chính
* 3: Kết thúc use case

*Luồng 2:*

* 1: Người dùng nhập sai ID, hệ thống không tìm kiếm được
* 2: Hệ thống hiển thị thông báo nhập sai ID, yêu cầu nhập lại
* 3: Người dùng nhập lại ID
* 4: Hệ thống tiếp nhận, tìm kiếm thông tin
* 5: Nếu tìm được, hệ thống hiển thị thông tin sinh viên. Nếu không lặp lại bước 2
* 6: Kết thúc use case

k) Use Case “Xem thông tin giảng viên”

* **Luồng sự kiện chính:**
* 1: Người dùng chọn chức năng Tra cứu thông tin giảng viên
* 2: Hệ thống hiển thị giao diện Tra cứu thông tin giảng viên, yêu cầu chọn Khoa/Viện và nhập tên giảng viên
* 3: Người dùng điền thông tin, chọn Search
* 4: Hệ thống tìm kiếm, hiển thị danh sách giảng viên tìm được
* 5: Người dùng chọn giảng viên cần tìm
* 6: Hệ thống hiển thị thông tin giảng viên đã chọn (mail, các môn giảng dạy, tài liệu …)
* 7: Kết thúc use case
* **Luồng sự kiện rẽ nhánh:**

*Luồng 1:*

* 1: Người dùng muốn hủy bỏ việc xem thông tin giảng viên, chọn Home
* 2: Hệ thống hủy bỏ việc xem thông tin giảng viên, quay lại giao diện chính
* 3: Kết thúc use case

*Luồng 2:*

* 1: Người dùng nhập sai tên giảng viên, hệ thống không tìm kiếm được
* 2: Hệ thống hiển thị thông báo không tìm thấy, yêu cầu nhập lại
* 3: Người dùng nhập lại tên giảng viên
* 4: Hệ thống tiếp nhận, tìm kiếm thông tin
* 5: Nếu tìm được, hệ thống hiển thị danh sách. Nếu không lặp lại bước 2
* 6: Kết thúc use case

l) Use Case “Xem lịch giảng dạy”

* **Luồng sự kiện chính:**
* 1: Người dùng chọn chức năng Tra cứu lịch giảng dạy
* 2: Hệ thống hiển thị giao diện Tra cứu lịch giảng dạy, yêu cầu nhập tên giảng viên hoặc mã giảng viên
* 3: Người dùng điền thông tin, chọn Search
* 4: Hệ thống tìm kiếm, hiển thị danh sách giảng viên tìm được (nếu tìm theo tên thì có thể trả về nhiều kết quả)
* 5: Người dùng chọn giảng viên cần tìm
* 6: Hệ thống hiển thị lịch giảng dạy của giảng viên đã chọn
* 7: Kết thúc use case
* **Luồng sự kiện rẽ nhánh:**

*Luồng 1:*

* 1: Người dùng muốn hủy bỏ việc xem lịch giảng dạy, chọn Home
* 2: Hệ thống hủy bỏ việc xem lịch giảng dạy, quay lại giao diện chính
* 3: Kết thúc use case

*Luồng 2:*

* 1: Người dùng nhập sai tên giảng viên, hệ thống không tìm kiếm được
* 2: Hệ thống hiển thị thông báo không tìm thấy, yêu cầu nhập lại
* 3: Người dùng nhập lại tên giảng viên
* 4: Hệ thống tiếp nhận, tìm kiếm thông tin
* 5: Nếu tìm được, hệ thống hiển thị danh sách. Nếu không lặp lại bước 2
* 6: Kết thúc use case

### 2.2.4. Danh sách Use Case

|  |  |  |
| --- | --- | --- |
| STT | Use Case | Mô tả |
| 1 | Đăng nhập | Cho phép người dùng đăng nhập vào hệ thống |
| 2 | Đăng xuất | Cho phép người dùng thoát khỏi hệ thống |
| 3 | Quản lý thông tin cá nhân | Người dùng có thể cập nhật thông tin cá nhân của mình, đổi mật khẩu |
| 4 | Quản lý SV | Giáo vụ quản lý thông tin SV, lớp SV … |
| 5 | Quản lý lớp học | Giáo vụ tạo lớp học, phân công giảng dạy |
| 6 | Quản lý lịch thi | Giáo vụ tạo và chỉnh sửa lịch thi |
| 7 | Thêm/Xóa học phần | Giáo vụ thêm học phần mới, xóa học phần đã bỏ |
| 8 | Xem lịch dạy | Giảng viên xem được lịch giảng dạy của mình |
| 9 | Quản lý điểm SV | Giảng viên nhập và gửi điểm lên hệ thống |
| 10 | Xem thông tin lớp học | Xem thời khóa biểu, danh sách lớp học, lịch thi |
| 11 | Xem thông tin lớp SV | Xem danh sách SV 1 lớp, in danh sách SV |
| 12 | Xem điểm SV | Xem điểm, in bảng điểm, thống kê qua từng kì |
| 13 | Tra cứu | Xem danh sách học phần, học phí, thông tin khác … |
| 14 | Xem thông tin giảng viên | Xem hồ sơ giảng viên, lịch giảng dạy |

## 2.3. Mô hình hóa cấu trúc

### 2.3.1. CRC Card

|  |  |
| --- | --- |
| Class name : person | |
| Description : lớp khuôn mẫu cho các tác nhân khác | |
| Responsibolities: :  Login  logout  getInfo  updateInfo | collaborators |
| Attibutes:  ID  Name  Password  Email  Phone | |
| Relationships :  Generalization  Aggregation  Other association : | |

|  |  |
| --- | --- |
| Class name : Student | |
| Description : sinh viên, người có nhu cầu sử dụng hệ thống. | |
| Responsibolities:  showScore  showTimetable  showExamSchedule  getFee  searchTeacher  searchCourse  searchSubject  showSVClass | collaborators |
| Attibutes :  address  photo  SVClass\_id | |
| Relationships :  Generalization  Aggregation  Other association : | |

|  |  |
| --- | --- |
| Class name : AcademicStaff | |
| Description : ban giáo vụ | |
| Responsibilities :  createCourse  createExamSchedule  createTeachingSchedule  createSVClass  addSubject  estoreStudentAccount | Collaborators : |
| Attributes : | |
| Relationships :  Generalization : persons  Aggregation  Other association | |

|  |  |
| --- | --- |
| Class name : Teacher | |
| Description : Giảng viên | |
| Responsibilities :  submitScore  getTeachingSchedule | collaborators |
| Attributes:  Department\_id  Workplace  Website  Photo | |
| Relationships :  Generalization : persons  Aggregation:  Other association : | |

|  |  |
| --- | --- |
| Class name : TimeTable | |
| Description : Thời khóa biểu | |
| Responsibilities : | collaborators |
| Attributes:  Course\_id  Student\_id | |
| Relationships :  Generalization :  Aggregation:  Other association: Student | |

|  |  |
| --- | --- |
| Class name : Fee | |
| Description : Học phý | |
| Responsibilities : | collaborators |
| Attributes:  Student\_id  term  numOfCredit  extraNumOfCredit  creditFee  total | |
| Relationships :  Generalization :  Aggregation:  Other association: | |

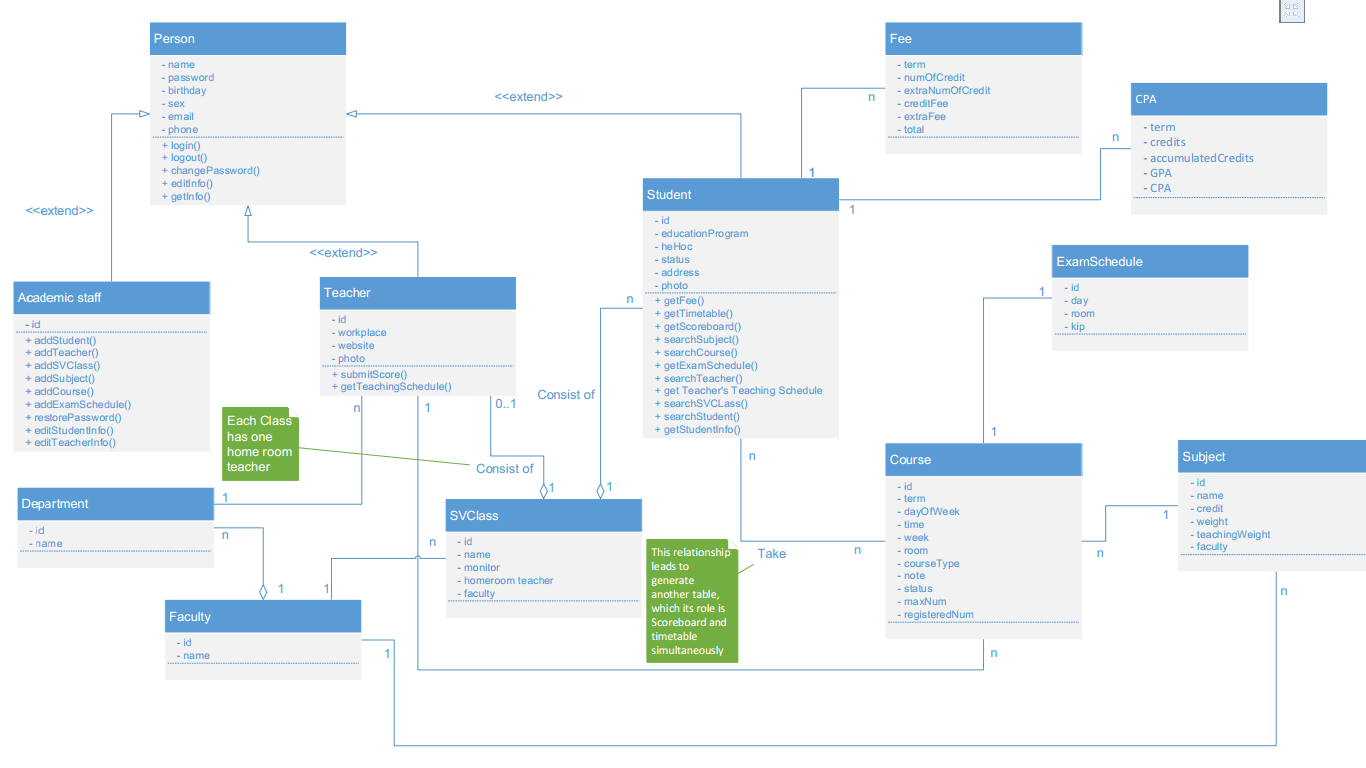
|  |  |
| --- | --- |
| Class name : ScoreBoard | |
| Description : Bảng điểm của sinh viên | |
| Responsibilities : | collaborators |
| Attributes:  Student\_id  term  Subject\_id  processScore  finalScore  overall  letterScore | |
| Relationships :  Generalization :  Aggregation:  Other association: | |

|  |  |
| --- | --- |
| Class name : ExamSchedule | |
| Description : Lịch thi | |
| Responsibilities : | collaborators |
| Attributes:  ID  Course\_id  Room  Kip  Day | |
| Relationships :  Generalization :  Aggregation:  Other association: | |

|  |  |
| --- | --- |
| Class name : Course | |
| Description : khóa học (lớp học) | |
| Responsibilities : | collaborators |
| Attributes:  courseID;  subjectID;  term;  dayOfWeek;  time;  subjectName;  room;  courseType;  week;  note;  status;  maxNum;  registeredNum;  facultyName;  teacherID; | |
| Relationships :  Generalization :  Aggregation:  Other association: | |

|  |  |
| --- | --- |
| Class name : Subject | |
| Description : Học phần | |
| Responsibilities : | collaborators |
| Attributes:  Id  Name  Credit  Weight  teachingWeight  Faculty\_id | |
| Relationships :  Generalization :  Aggregation:  Other association: | |

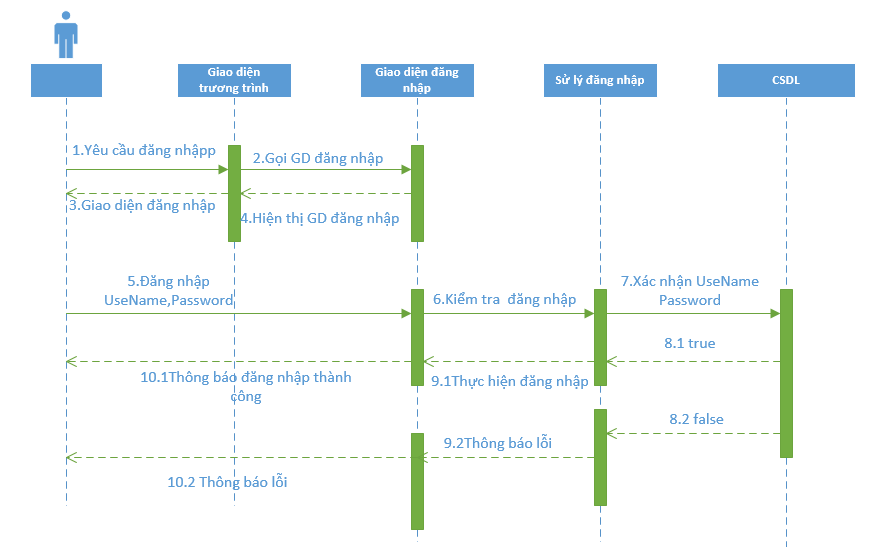
### 2.3.2. Biểu đồ lớp (Class diagram)



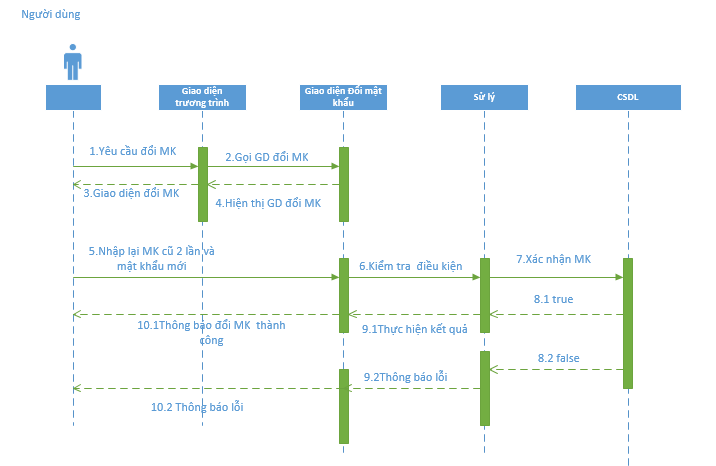
## 2.4. Mô hình hóa hoạt động

Biểu đồ tuần tự (Sequence diagram):

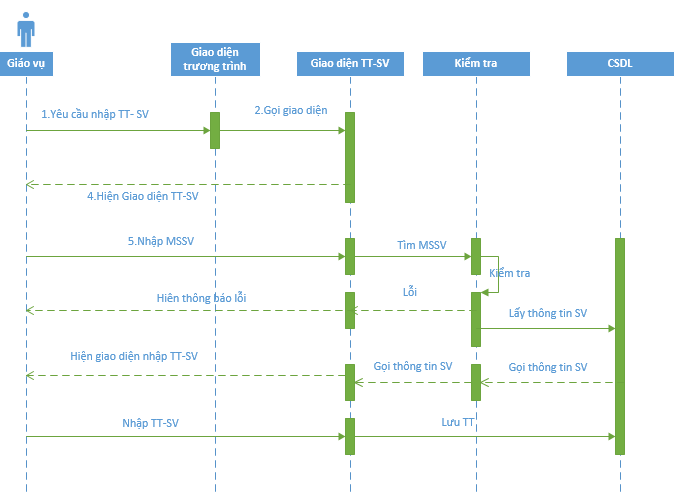
* Quá trình đăng nhập



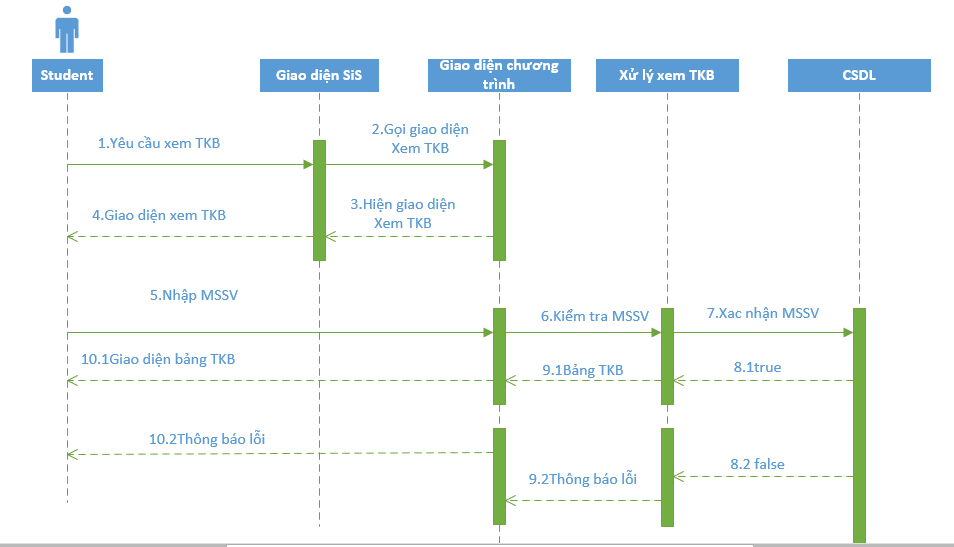
* Đổi mật khẩu



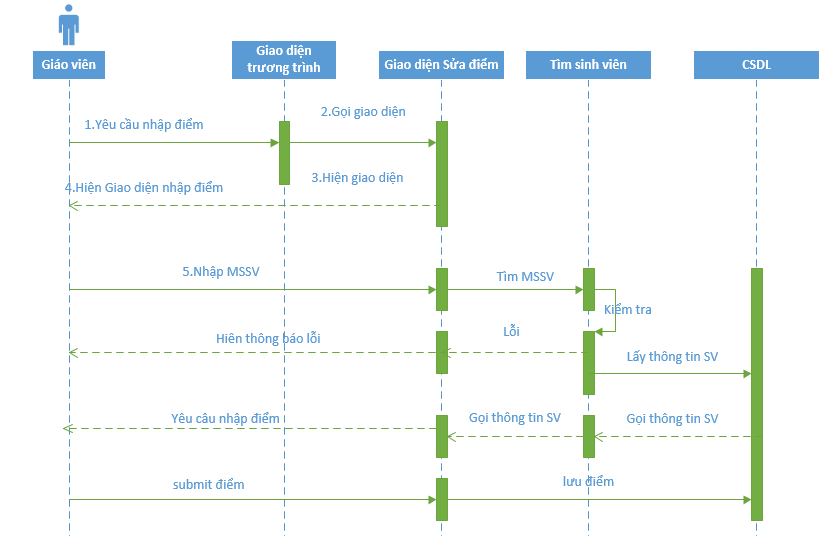
* Thêm sinh viên



* Xem thời khóa biểu



* Giáo viên submit điểm



1. **Thiết kế**

## 3.1. Thiết kế các lớp

* Class Person

**public** **class** **Person** {

**private** **String** name;

**private** **String** password;

**private** **String** birthday;

**private** **String** sex;

**private** **String** email;

**private** **String** phone;

}

* Class Student

**public** **class** **Student** **extends** **Person** {

**private** **int** id;

**private** **String** educationProgram;

**private** **String** heHoc;

**private** **String** status;

**private** **String** address;

**private** **int** SVClass\_id\_st;

**private** **String** SVClassName;

**private** **byte**[] photo;

**private** **float** processScore;

**private** **float** finalScore;

}

* Class Teacher

**public** **class** **Teacher** **extends** **Person** {

**String** id;

**String** department;

**String** faculty;

**String** workplace;

**String** website;

}

* Class Staff

**public** **class** **Staff** **extends** **Person** {

**String** id;

}

* Class Subject

**public** **class** **Subject** {

**String** id;

**String** name;

**int** credit;

**float** feeCredit;

**float** weight;

**String** teachingWeight;

**int** Faculty\_id;

}

* Class Course

**public** **class** **Course** **implements** Comparable<**Course**> {

**int** courseID;

**String** subjectID;

**int** term;

**int** dayOfWeek;

**String** time;

**String** subjectName;

**String** room;

**String** courseType;

**String** week;

**String** note;

**String** status;

**int** maxNum;

**int** registeredNum;

**String** facultyName;

**String** teacherID;

}

* Class SVClass
* Class Fee

**public** **class** **Fee** {

**int** studentID;

**int** term;

**int** numOfCredit;

**int** extraNumOfCredit;

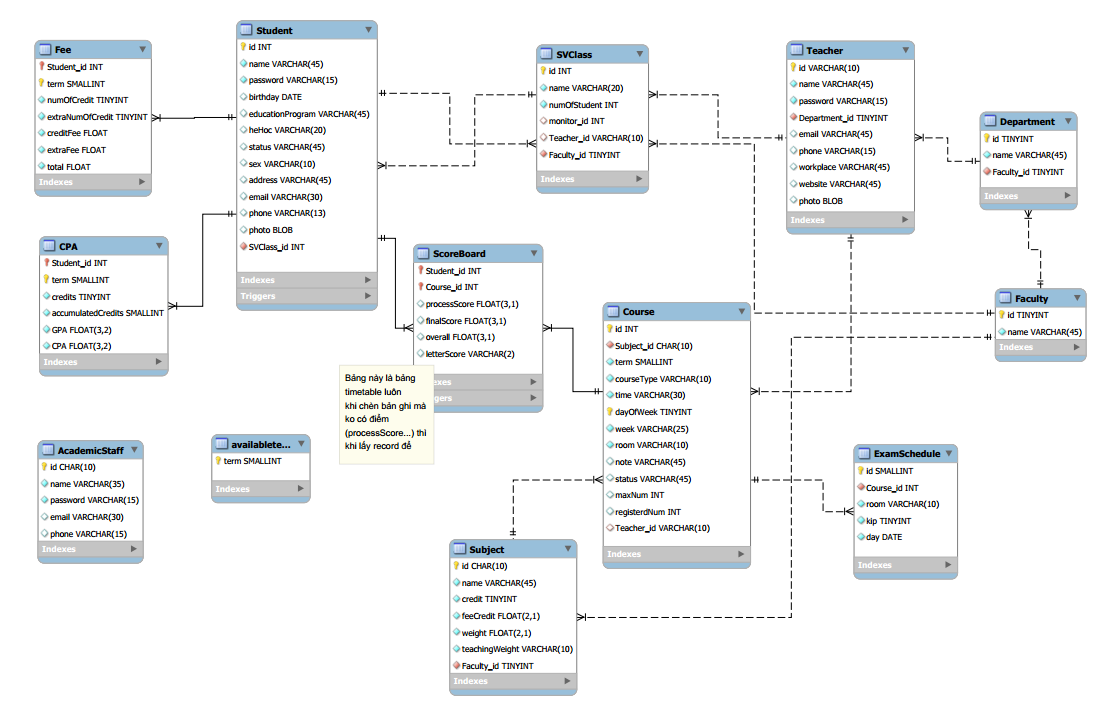
**int** creditFee;

**int** extraFee;

**int** total;

}

## 3.2. Thiết kế cơ sở dữ liệu



## 3.3. Thiết kế lớp quản lý dữ liệu

* DatabaseUtils

**package** database;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.PreparedStatement;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.util.logging.Level;

**import** java.util.logging.Logger;

**public** **class** **DatabaseUtils** {

**public** **final** **String** DATABASE\_NAME = "qlsv\_db";

**public** **final** **String** USERNAME = "root";

**public** **final** **String** PASSWORD = "5555";

**public** **final** **String** URL\_MYSQL = "jdbc:mysql://localhost:3306/"+DATABASE\_NAME;

Connection conn;

PreparedStatement pst;

ResultSet rs;

**public** Connection connect() {

**try** {

**Class**.*forName*("com.mysql.cj.jdbc.Driver");

conn = **DriverManager**.*getConnection*(URL\_MYSQL, USERNAME, PASSWORD);

**System**.***out***.println("Connect successfull");

**return** conn;

} **catch** (**SQLException** ex) {

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

**System**.***out***.println("Error connection! Lỗi kết nối nhé!");

} **catch** (**ClassNotFoundException** ex) {

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

}

**return** **null**;

}

**protected** ResultSet getAllRecords(**String** tableName) {

**try** {

pst = conn.prepareStatement("SELECT \* FROM " + tableName);

rs = pst.executeQuery();

**return** rs;

} **catch** (**SQLException** ex) {

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

}

**return** **null**;

}

**protected** ResultSet getRecordByID(**String** tableName, **int** id) {

**try** {

pst = conn.prepareStatement("SELECT \* FROM " + tableName + " WHERE id = " + id);

rs = pst.executeQuery();

**return** rs;

} **catch** (**SQLException** ex) {

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

}

**return** **null**;

}

**protected** **int** changeUserPassword(**String** tableName, **int** id, **String** newPassword) {

**try** {

pst = conn.prepareStatement("UPDATE " + tableName + " SET password=? WHERE id=?");

pst.setString(1, newPassword);

pst.setInt(2, id);

**return** pst.executeUpdate();

} **catch** (**SQLException** ex) {

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

}

**return** 0;

}

**public** **void** disConnect() {

**try** {

**if**(pst != **null**) {

pst.close();

pst = **null**;

} **else** **System**.***out***.println("[DBUtils] pst already null!");

**if**(rs != **null**) {

rs.close();

rs = **null**;

} **else** **System**.***out***.println("[DBUtils] rs already null!");

**if**(conn != **null**) {

conn.close();

conn = **null**;

} **else** **System**.***out***.println("[DBUtils] conn already null!");

} **catch** (**SQLException** e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

//kiem tra tai khoan khi login

**public** **Object** check(**int** ID, **String** pass) {

**return** **null**;

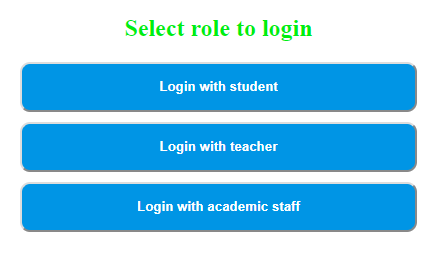
}

}

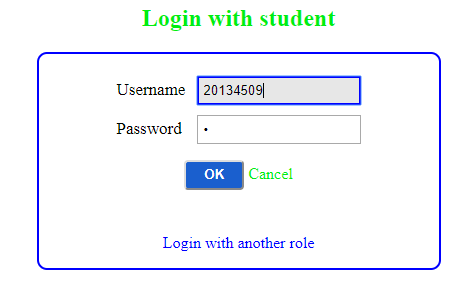
* CourseUtils
* StudentUtils
* SubjectUtils
* TeacherUtils

## 3.4. Thiết kế giao diện

* Chọn vai trò trước khi login

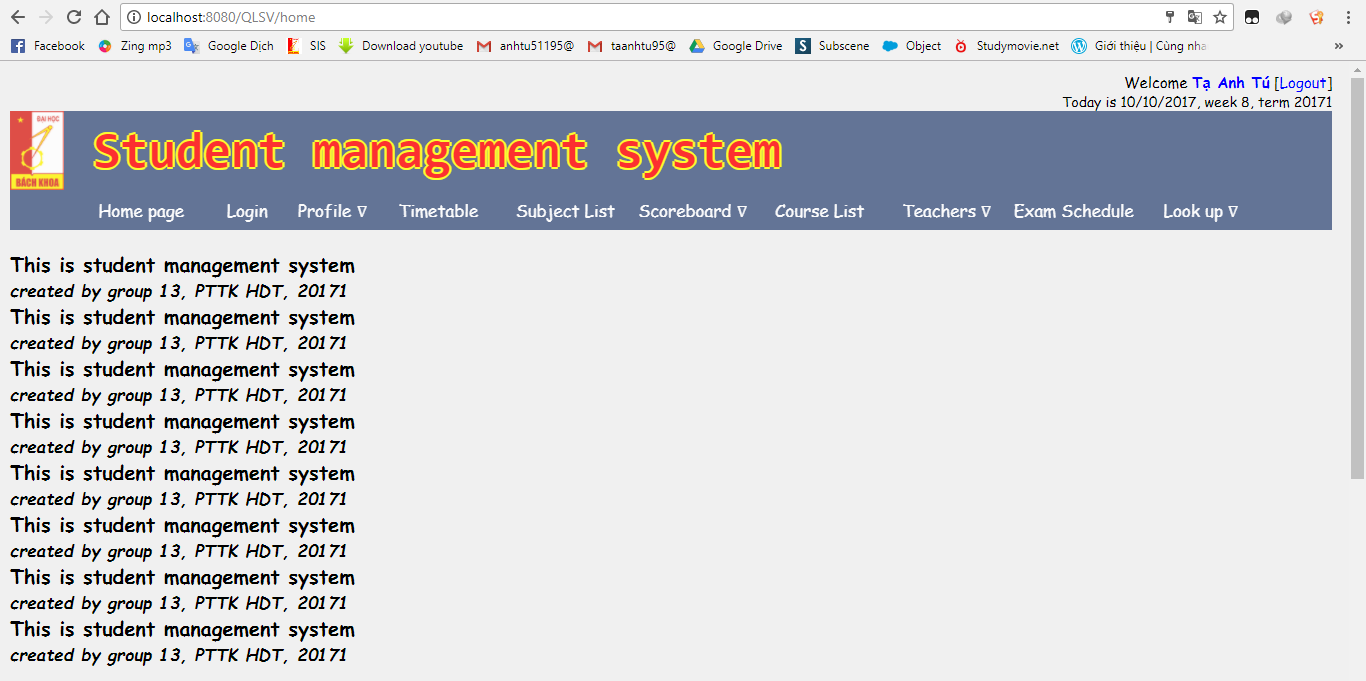


* Login



### 3.4.1. Giao diện cho Sinh viên

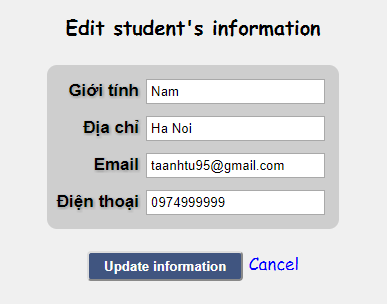
* Home



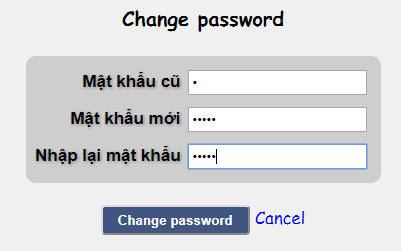
* Thông tin cá nhân



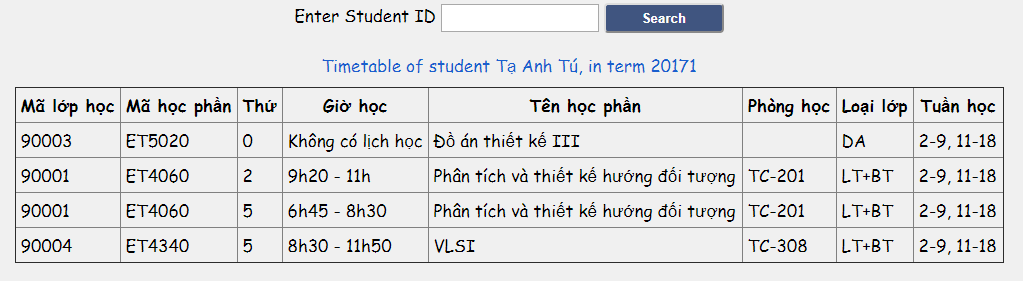
* Chỉnh sửa thông tin cá nhân



* Đổi mật khẩu



* Xem thời khóa biểu



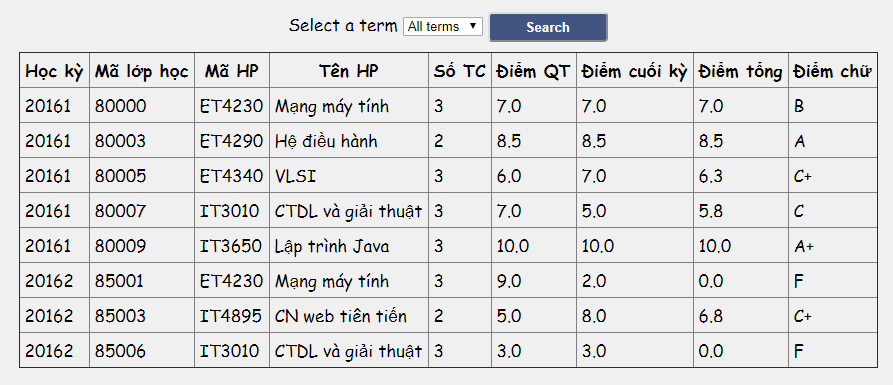
* Xem danh sách học phần



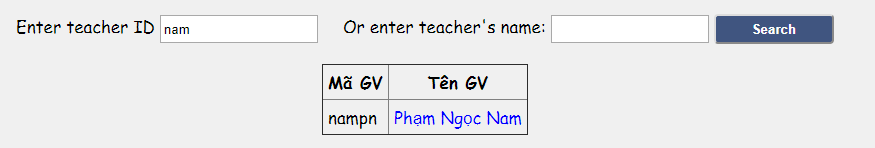
* Xem danh sách lớp học



* Xem bảng điểm



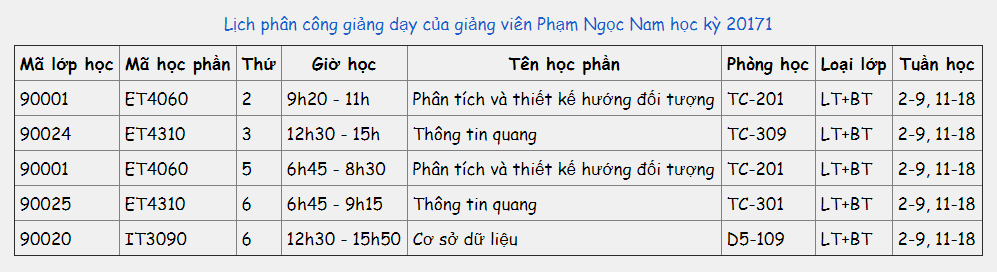
* Tìm kiếm giảng viên



* Xem thông tin giảng viên



* Xem lịch phân công giảng dạy của giảng viên



### 3.4.2. Giao diện cho giảng viên

### 3.4.3. Giao diện cho giáo vụ

## 3.5. Thiết kế kiến trúc vật lý

1. **Triển khai**