**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP.HCM**



**ĐỀ TÀI QUẢN LÝ TRƯỜNG TRUNG HỌC PHỔ THÔNG**

**Nhóm 08 - Si nh viên thực hiện:**

1. Lê Tuấn Kiệt – 16016991
2. Lê Thành Kỷ – 16011051
3. Lý Đông Cảnh – 16038081

# MỤC LỤC

[MỤC LỤC 1](#_Toc21875699)

[DANH MỤC CÁC HÌNH VẼ 3](#_Toc21875700)

[DANH MỤC CÁC BẢNG BIỂU 4](#_Toc21875701)

[CHƯƠNG 1 : GIỚI THIỆU 5](#_Toc21875702)

[1.1 Tổng quan 5](#_Toc21875703)

[1.2 Mục tiêu đề tài 6](#_Toc21875704)

[1.3 Phạm vi đề tài 6](#_Toc21875705)

[1.4 Mô tả yêu cầu chức năng 6](#_Toc21875706)

[1.4.1 Yêu cầu chức năng 6](#_Toc21875707)

[1.4.1.1 Người quản lý( admin ) 6](#_Toc21875708)

[1.4.1.2 Giáo viên 7](#_Toc21875709)

[1.4.1.3 Học sinh 7](#_Toc21875710)

[1.4.2 Yêu cầu phi chức năng 7](#_Toc21875711)

[CHƯƠNG 2 : CƠ SỞ LÝ THUYẾT 8](#_Toc21875712)

[2.1 Cloud Computing với AWS 8](#_Toc21875713)

[2.1.1 Điện toán đám mây là gì? 8](#_Toc21875714)

[2.1.2 Điện toán đám mây hoạt động như thế nào? 8](#_Toc21875715)

[2.2 Node.js 8](#_Toc21875716)

[- Phần Core bên dưới của Node.js được viết hầu hết bằng C++ nên cho tốc độ xử lý và hiệu năng khá cao. 8](#_Toc21875719)

[- Node.js tạo ra được các ứng dụng có tốc độ xử lý nhanh, realtime thời gian thực. 8](#_Toc21875720)

[- Node.js áp dụng cho các sản phẩm có lượng truy cập lớn, cần mở rộng nhanh, cần đổi mới công nghệ, hoặc tạo ra các dự án Startup nhanh nhất có thể. 8](#_Toc21875721)

[2.3 DynamoDB 9](#_Toc21875722)

[2.4 React 10](#_Toc21875723)

[2.5 Restful API 10](#_Toc21875724)

[2.6 Simple Page Application 11](#_Toc21875725)

[CHƯƠNG 3 : PHÂN TÍCH VÀ THIẾT KẾ 12](#_Toc21875726)

[3.1 Phân tích yêu cầu bằng UML 12](#_Toc21875727)

[3.1.1 Usecase tổng quát 12](#_Toc21875728)

[3.1.2 Danh sách tác nhân và mô tả 12](#_Toc21875729)

[3.1.3 Danh sách các tình huống hoạt động (Use cases) 13](#_Toc21875730)

[3.1.4 Tình huống hoạt động 13](#_Toc21875731)

[3.2 Class diagram 30](#_Toc21875732)

[3.3 Deployment diagram 31](#_Toc21875733)

[CHƯƠNG 4 : HIỆN THỰC 32](#_Toc21875734)

[4.1 Cấu hình phần cứng, phần mềm 32](#_Toc21875735)

[4.2 Giao diện của hệ thống 32](#_Toc21875736)

[4.3 Kế hoạch và hiện thực kiểm thử hệ thống 32](#_Toc21875737)

[4.3.1 Kế hoạch kiểm thử 32](#_Toc21875738)

[4.3.2 Kiểm thử hệ thống 32](#_Toc21875739)

[CHƯƠNG 5 : KẾT LUẬN 33](#_Toc21875740)

[5.1 Kết quả đạt được 33](#_Toc21875741)

[5.2 Hạn chế của đồ án 33](#_Toc21875742)

[5.3 Hướng phát triển 33](#_Toc21875743)

[TÀI LIỆU THAM KHẢO 34](#_Toc21875744)

# DANH MỤC CÁC HÌNH VẼ

[Hình 2‑1 Node.js development 9](#_Toc21876433)

[Hình 3‑1-1 UseCase tổng quát 12](#_Toc21876434)

[Hình 3‑1-1 : Activity Đăng nhập 14](#_Toc21876435)

[Hình 3‑1-2 : Activity Xem điểm 15](#_Toc21876436)

[Hình 3‑1-3 : Activity Xem thông tin học sinh 16](#_Toc21876437)

[Hình 3‑1-4 : Activity Xem lịch học 17](#_Toc21876438)

[Hình 3‑1-5 : Activity Nhập điểm 18](#_Toc21876439)

[Hình 3‑1-6 : Activity Xem lịch dạy 19](#_Toc21876440)

[Hình 3‑1-7 : Activity Xem thông tin giáo viên 20](#_Toc21876441)

[Hình 3‑1-8 : Activity Thêm học sinh 21](#_Toc21876442)

[Hình 3‑1-9 : Activity Sửa thông tin học sinh 23](#_Toc21876443)

[Hình 3‑1-10 : Activity Xoá học sinh 24](#_Toc21876444)

[Hình 3‑1-11 : Activity Thêm giáo viên 26](#_Toc21876445)

[Hình 3‑1-12 : Activity Sửa giáo viên 27](#_Toc21876446)

[Hình 3‑1-13 : Activity Xoá giáo viên 29](#_Toc21876447)

[Hình 3‑1-14 : Activity Thêm lớp 30](#_Toc21876448)

[Hình 3‑1-15 : Activity Sửa lớp 32](#_Toc21876449)

[Hình 3‑1-16 : Activity Xoá lớp 33](#_Toc21876450)

[Hình 3‑1-17 : Activity Thêm lịch 34](#_Toc21876451)

[Hình 3‑1-18 : Activity Sửa lịch 36](#_Toc21876452)

[Hình 3‑1-19 : Activity Xoá lịch 37](#_Toc21876453)

[Hình 3‑1-20 : ClassDiagram 38](#_Toc21876454)

# DANH MỤC CÁC BẢNG BIỂU

[Bảng 3‑1 Danh sách tác nhân và mô tả 13](#_Toc21874839)

[Bảng 3‑2 Danh sách các tình huống trong hệ thống 13](#_Toc21874840)

[Bảng 3-3-1: Usecase Đăng nhập 14](#_Toc21874841)

[Bảng 3-3-2: Usecase Xem điểm 14](#_Toc21874842)

[Bảng 3-3-3: Usecase Xem thông tin cá nhân 15](#_Toc21874843)

[Bảng 3-3-4: Usecase Xem lịch học 15](#_Toc21874844)

[Bảng 3-3-5: Usecase Nhập điểm bộ môn 16](#_Toc21874845)

[Bảng 3-3-6: Usecase Xem lịch dạy 16](#_Toc21874846)

[Bảng 3-3-7: Usecase Xem thông tin giáo viên 17](#_Toc21874847)

[Bảng 3-3-8: Usecase Thêm học sinh 17](#_Toc21874848)

[Bảng 3-3-9: Usecase Sửa thông tin học sinh 18](#_Toc21874849)

[Bảng 3-3-10:Usecase Xóa học sinh 19](#_Toc21874850)

[Bảng 3-3-11: Usecase Thêm giáo viên 20](#_Toc21874851)

[Bảng 3-3-12: Usecase Sửa giáo viên 21](#_Toc21874852)

[Bảng 3-3-13: Usecase Xóa giáo viên 21](#_Toc21874853)

[Bảng 3-3-14: Usecase Thêm lớp 22](#_Toc21874854)

[Bảng 3-3-15: Usecase Sửa lớp 23](#_Toc21874855)

[Bảng 3-3-16: Usecase Xóa lớp 23](#_Toc21874856)

[Bảng 3-3-17: Usecase Thêm tiết dạy 24](#_Toc21874857)

[Bảng 3-3-18: Usecase Sửa tiết dạy 25](#_Toc21874858)

[Bảng 3-3-19: Usecase Xóa tiết dạy 25](#_Toc21874859)

# : GIỚI THIỆU

## Tổng quan

* Trong những năm gần đây, với sự phát triển vượt trội của khoa học kĩ thuật đặc biệt là công nghệ thông tin con người đã tạo ra nhiều thành quả và ứng dụng chúng vào nhiều lĩnh vực đã góp một phần to lớn cho sự nghiệp phát triển của con người. Trong những đóng góp đó không thể không kể đến internet, internet đang dần trở thành một phần không thể thiếu của cuộc sống. Lợi ích của nó ngày càng được ứng dụng rộng rãi, việc sử dụng một website làm công cụ truyền thông đã quá phổ biến trên phạm vi toàn thế giới.
* Ngày nay, sự bùng nổ của hệ thống mạng internet trên toàn cầu và những lợi ích vô cùng to lớn mà nó mang lại đã thực sự thu hút và chinh phục được con người. Tận dụng những tính năng mạnh mẽ của Internet, những công nghệ mới trong lĩnh vực về công nghệ thông tin và truyền thông mà còn người đã xây dựng được các hệ thống ngày càng tiên tiến có thể đưa vào đời sống thực tiễn làm thay đổi cuộc sống của xã hội chúng ta theo chiều hướng tiện lợi, văn minh, tiến bộ hơn.
* Cổng thông tin điện tử trên internet ra đời cùng với việc internet đang lan nhanh một cách chóng mặt trên toàn cầu, nó sẽ trở thành công cụ chủ yếu và đắc lực cho việc trao đổi, tìm kiếm thông tin trên phạm vi toàn cầu. Bây giờ, chỉ cần một chiếc máy vi tính có kết nối mạng internet bạn có thể kết nối với toàn thế giới bao gồm việc tìm kiếm thông tin, truy cập vào các website của công ty, báo điện tử cũng như nhiều trang web khác.
* Hiện tại, hầu như toàn bộ các trường học, các công ty, cửa hàng, cơ sở kinh doanh đều đã có website riêng để phục vụ cho việc quản lý các lĩnh vực trở nên dễ dàng và chuyên nghiệp hơn. Website đã dần trở thành một nhu cầu thiết yếu, vì vậy việc đáp ứng được nhu cầu đó đã thu hút được nhiều người tìm hiểu và phát triển các trang web khiến thiết kế website nói riêng và công nghệ thông tin nói chung trở thành một ngành nghề thu hút nhiều sinh viên tại các trường đại học.
* Trước đây để có thể triển khai một ứng dụng website, các công ty doanh nghiệp phải đi thuê hoặc mua một hay nhiều máy chủ sau đó đặt máy chủ tại các trung tâm dữ liệu thì nay điện toán đám mây (Cloud Computing) cho phép chúng ra giản lược quá trình mua\thuê đi tối thiểu. Ta chỉ cần nêu ra yêu cầu của mình, hệ thống điện toán đám mây sễ tự động gom nhặt các tài nguyên để đáp ứng nhu cầu của ta. Điện toán đám mây mang lại cho người dùng rất nhiều lợi ích như: tiết kiệm chi phí đầu tư cơ sở hạ tầng; tiện lợi cho người dùng có thể nhanh chóng truy cập, sử dụng tài nguyên thông qua internet mà không cần phải cài đặt phức tạp; mọi dữ liệu được đồng bộ hóa trên đám mây làm cho dữ liệu có tính an toàn và liên tục; chỉ với một vài thao tác đơn giản để triển khai chúng nhanh chóng ở bất kì đâu.
* Với các lợi ích nêu trên của điện toán đám mây cùng với nhu cầu cần một website để quản lý trường trung học phổ thông nhóm 8 chúng em đã xây dựng một website quản lý trường trung học phổ thông với các chức năng: quản lý học sinh, quản lý giáo viên, quản lý điểm, quản lý lớp, quản lý lịch học, lịch dạy.

## Mục tiêu đề tài

* Xây dựng website phục vụ cho việc quản lý trường trung học phổ thông: quản lý học sinh, quản lý giáo viên, quản lý lớp, xem điểm, xem lịch học, lịch dạy.
* Xây dựng hệ thống website thân thiện, dễ sử dụng đối với người dùng.
* Hệ thống xây dựng các chức năng được phân quyền theo loại người dùng: Người quản lý có thể thực hiện các chức năng quản lý học sinh, quản lý giáo viên, quản lý lớp, quản lý tiết dạy; giáo viên có thể thực hiện các chức năng nhập điểm, xem tiết dạy, xem thông tin cá nhân; học sinh xem điểm, xem xem lịch học, xem thông tin cá nhân.

## Phạm vi đề tài

* Thực hiện các chức năng trên nền tảng node.js
* Sử dụng cơ sở dữ liệu dynamoDB
* Website thực hiện được các chức năng: xem điểm, xem lịch học, xem thông tin cá nhân, quản lý học sinh, quản lý giáo viên, quản lý lớp học, quản lý tiết dạy.

## Mô tả yêu cầu chức năng

* Hệ thống website quản lý trường trung học phổ thông gồm 3 actor chính là: Học sinh, Giáo viên và Quản lý(admin). Hệ thống website gồm các yêu cầu chức năng và phi chức năng như sau:

### Yêu cầu chức năng

* Đăng nhập: giáo viên và admin bắt buột phải đăng nhập để thực hiện các chức năng của mình.
* Quản lý học sinh: là chức năng thêm, sửa, học sinh và sửa điểm của học sinh trên hệ thống.
* Quản lý giáo viên: là chức năng thêm, sửa, xoá giáo viên trên hệ thống.
* Quản lý lớp: là chức năng thêm, sửa, xoá lớp học.
* Quản lý tiết dạy: là chức năng thêm , sửa, xoá tiết dạy cho giáo viên.
* Nhập điểm bộ môn: là chức năng cho phép giáo viên thực hiện việc nhập điểm bộ môn của các học sinh.
* Xem tiết dạy: cho phép giáo viên xem tiết dạy của bản thân
* Xem thông tin giáo viên: cho phép giáo viên xem các thông tin của bản thân.
* Xem điểm: là chức năng cho phép học sinh cũng như phụ huynh theo dõi điểm số.
* Xem thời khóa biếu: là chức năng giúp học sinh có thể biết được lịch học của mình.
* Xem thông tin học sinh: cho phép người dung xem các thông tin của học sinh.

### Yêu cầu phi chức năng

* Hệ thống sẽ gửi email thông báo kết quả học tập hoặc các thông tin quan trọng đến phụ huynh học sinh.

# : CƠ SỞ LÝ THUYẾT

## Cloud Computing với AWS

### Điện toán đám mây là gì?

* Điện toán đám mây là việc cung cấp sức mạnh tính toán, cơ sở dữ liệu, khả năng lưu trữ, ứng dụng và các tài nguyên CNTT khác theo yêu cầu qua internet với chính sách thanh toán theo mức sử dụng.
* Cho dù bạn đang sử dụng nền tảng này để chạy các ứng dụng chia sẻ ảnh với hàng triệu người dùng thiết bị di động hay để hỗ trợ các hoạt động quan trọng của doanh nghiệp, nền tảng dịch vụ đám mây cho phép truy cập nhanh vào các tài nguyên CNTT linh hoạt và chi phí thấp. Với điện toán đám mây, bạn không cần đầu tư ngay một khoản tiền lớn vào phần cứng rồi dành nhiều thời gian vất vả để quản lý phần cứng đó. Thay vào đó, bạn có thể cung cấp chính xác loại và quy mô tài nguyên điện toán phù hợp mà bạn cần để tiếp sức cho ý tưởng mới nhất của mình hoặc điều hành bộ phận CNTT. Bạn có thể truy cập bao nhiêu tài nguyên tùy thích, gần như ngay lập tức và chỉ phải trả cho những gì bạn sử dụng.

### Điện toán đám mây hoạt động như thế nào?

* Điện toán đám mây cho bạn quyền truy cập vào máy chủ, bộ nhớ, cơ sở dữ liệu và hàng loạt dịch vụ ứng dụng trên Internet. Một nhà cung cấp dịch vụ đám mây như Amazon Web Services sở hữu và duy trì phần cứng được kết nối mạng cần thiết cho các dịch vụ ứng dụng này, trong khi bạn cung cấp và sử dụng những gì bạn cần thông qua một ứng dụng web.

## Node.js

## Node.js là một nền tảng (Platform) phát triển độc lập được xây dựng ở trên Javascript Runtime của Chrome mà chúng ta có thể xây dựng được các ứng dụng mạng một cách nhanh chóng và dễ dàng mở rộng.

## Node.js được xây dựng và phát triển từ năm 2009, bảo trợ bởi công ty Joyent, trụ sở tại California, Hoa Kỳ.

## Phần Core bên dưới của Node.js được viết hầu hết bằng C++ nên cho tốc độ xử lý và hiệu năng khá cao.

## Node.js tạo ra được các ứng dụng có tốc độ xử lý nhanh, realtime thời gian thực.

## Node.js áp dụng cho các sản phẩm có lượng truy cập lớn, cần mở rộng nhanh, cần đổi mới công nghệ, hoặc tạo ra các dự án Startup nhanh nhất có thể.



Hình 2‑1 Node.js development

## DynamoDB

* **Amazon DynamoDB** là một dịch vụ [cơ sở dữ liệu](https://vi.wikipedia.org/wiki/C%C6%A1_s%E1%BB%9F_d%E1%BB%AF_li%E1%BB%87u) độc quyền quản lý hoàn toàn sử dụng [NoSQL](https://vi.wikipedia.org/wiki/NoSQL) có hỗ trợ giá trị khóa và cấu trúc dữ liệu tài liệu và được cung cấp bởi [Amazon.com](https://vi.wikipedia.org/wiki/Amazon.com) như một phần của Amazon Web Services portfolio. DynamoDB trưng bày một mô hình dữ liệu tương tự và lấy tên của nó từ Dynamo, nhưng có một triển khai cơ bản khác. Dynamo có thiết kế đa chủ yêu cầu khách hàng giải quyết xung đột phiên bản và DynamoDB sử dụng sao chép đồng bộ trên nhiều trung tâm dữ liệu để có độ bền và tính sẵn sàng cao. DynamoDB được Amazon CTO [Werner Vogels](https://vi.wikipedia.org/w/index.php?title=Werner_Vogels&action=edit&redlink=1) công bố vào ngày 18 tháng 1 năm 2012 và được trình bày dưới dạng một sự phát triển của giải pháp [Amazon SimpleDB](https://vi.wikipedia.org/w/index.php?title=Amazon_SimpleDB&action=edit&redlink=1).
* DynamoDB khác với các dịch vụ khác của Amazon bằng cách cho phép các nhà phát triển mua dịch vụ dựa trên thông lượng, thay vì [lưu trữ](https://vi.wikipedia.org/wiki/B%E1%BB%99_nh%E1%BB%9B). Nếu Tự động mở rộng quy mô, thì cơ sở dữ liệu sẽ tự động mở rộng quy mô. Ngoài ra, quản trị viên có thể yêu cầu thay đổi thông lượng và DynamoDB sẽ phân tán dữ liệu và lưu lượng truy cập trên một số máy chủ sử dụng [ổ đĩa trạng thái rắn](https://vi.wikipedia.org/wiki/SSD), cho phép hiệu suất có thể dự đoán được. Nó cung cấp tích hợp với [Hadoop](https://vi.wikipedia.org/wiki/Apache_Hadoop) thông qua [đàn hồi MapReduce](https://vi.wikipedia.org/wiki/Apache_Hadoop).
* Vào tháng 9 năm 2013, Amazon đã cung cấp phiên bản phát triển cục bộ của DynamoDB để các nhà phát triển có thể kiểm tra các ứng dụng được hỗ trợ bởi DynamoDB cục bộ.
* Amazon DynamoDB được "xây dựng dựa trên các nguyên tắc của công nghệ lưu trữ Amazon Dynamo ", một hệ thống lưu trữ nội bộ ban đầu được sử dụng cho trang web Amazon. Các ngôn ngữ và khung với liên kết DynamoDB bao gồm [Java](https://vi.wikipedia.org/wiki/Java_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)), [Node.js](https://vi.wikipedia.org/wiki/Node.js), [Go](https://vi.wikipedia.org/wiki/Go_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)), [C #](https://vi.wikipedia.org/wiki/C_th%C4%83ng_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)) [.](https://vi.wikipedia.org/wiki/.NET_Framework) [NET](https://vi.wikipedia.org/wiki/.NET_Framework), [Perl](https://vi.wikipedia.org/wiki/Perl), [PHP](https://vi.wikipedia.org/wiki/PHP), [Python](https://vi.wikipedia.org/wiki/Python_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)), [Ruby](https://vi.wikipedia.org/wiki/Ruby_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)), [Haskell](https://vi.wikipedia.org/wiki/Haskell_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)) và [Erlang](https://vi.wikipedia.org/w/index.php?title=Erlang_(programming_language)&action=edit&redlink=1).

## React

* **React** mở ra cho chúng ra những khả năng mới như: render dữ liệu ở tầng server, cập nhật dữ liệu thời gian thực, hiển thị dữ liệu đa nền tảng… Vậy React (còn gọi là ReactJS hay React.js) là một thư viện javascript được tạo ra bởi sự cộng tác giữa Facebook và Instagram. Nó cho phép những nhà phát triển web tạo ra giao diện người dùng nhanh chóng. Phần view của React thường được hiển thị bằng việc chủ yếu dùng các component mà chứa các component cụ thể hoặc các thẻ HTML. Một trong những đặc trưng duy nhất của React là việc render dữ liệu không những có thể thực hiện ở tầng server mà còn ở tầng client.
* Nó cũng sử dụng khái niệm là Virtual DOM tạo ra bản cache cấu trúc dữ liệu của ứng dụng trên bộ nhớ. Sau đó, ở mỗi vòng lặp nó liệt kê những thay đổi và sau đó cập nhập lại sự thay đổi trên DOM của trình duyệt một cách hiệu quả. Điều này cho phép ta viết các đoạn code như thể toàn bộ trang được render lại dù thực tế là React chỉ render những component hay subcomponent nào thực sự thay đổi.
* React cực kì hiệu quả khi tạo ra cho chính nó DOM ảo – nơi mà các component thực sự tồn tại trên đó. Điều này giúp cải thiện hiệu suất rất nhiều. React cũng tính toán những thay đổi nào cần cập nhật lên DOM và chỉ thực hiện chúng, điều này giúp tránh những thao tác cần trên DOM tốn nhiều chi phí. React cũng giúp viết các đoạn code dễ dàng hơn khi sử dụng cú pháp đặc biệt là JSX (JavaScript mở rộng) cho phép ta trộn giữa code Javascript và HTML: ta có thể thêm các đoạn HTML vào trong hàm render mà không cần phải nối chuỗi. Một trong những vấn đề với các ứng dụng đơn trang là tối ưu SEO và thời gian tải trang, React có thể vừa render ở ngoài trình duyệt sử dụng DOM và cũng có thể render bằng các chuỗi HTML mà server trả về. Ngoài ra, React giúp việc thiết kế testcase giao diện trở nên dễ dàng hơn vì virtual DOM được cài đặt hoàn toàn bằng Javascript; Hiệu năng khi sử dụng React cao đối với các ứng dụng có dữ liệu thay đổi liên tục; bên cạnh đó, React giúp việc bảo trì và sửa lỗi trở nên dễ dàng hơn.
* React không phải là một MVC Framework như những Framework khác, nó chỉ là thư viện giúp render ra phần view. Vì thế, React sễ không có phần Model và Controller mà phải kết hợp giữa các thư viện khác. React khá nặng so với các Framework khác và đối với người mới tiếp cận đến việc thiết kế web rất khó tiếp cận nó.

## Restful API

* **REST** là viết tắt của RepresentationalStateTransfer là một cách cung cấp khả năng tương tác giữa các hệ thống trên mạng Internet. Nghe có vẻ phức tạp nhỉ nhưng thực ra đây là một thuật ngữ rất đơn giản. Là tập hợp của các hướng dẫn và các kiểu cấu trúc được sử dụng trong truyền dữ liệu được áp dụng cho ứng dụng web nhưng có thể sử dụng trong cả truyền dữ liệu giữa các phần mềm. Thuật ngữ API là viết tắt củaApplicationProgrammingInterface, là các phương thức giúp kết nối các thư viện hoặc ứng dụng. Tất cả các ứng dụng lớn đều có rất nhiều các API, ví dụ:
* **RESTful API** là một tiêu chuẩn dùng trong việc thết kế các thiết kế API cho các ứng dụng web để quản lý các resource. RESTful là một trong những kiểu thiết kế API được sử dụng phổ biến nhất ngày nay. Trọng tâm của REST quy định cách sử dụng các HTTP method (như GET, POST, PUT, DELETE...) và cách định dạng các URL cho ứng dụng web để quản các resource. RESTful không quy định logic code ứng dụng và không giới hạn bởi ngôn ngữ lập trình ứng dụng.
* Có 4 hoạt động chủ yếu ta có thể kể đến khi làm việc với server: lấy dữ liệu ở một định dạng nào đó (JSON), tạo dữ liệu mới, cập nhật dữ liệu, xóa dữ liệu. REST hoạt động chủ yếu dựa vào giao thức HTTP. Các hoạt động cơ bản nêu trên sẽ sử dụng những phương thức HTTP riêng.

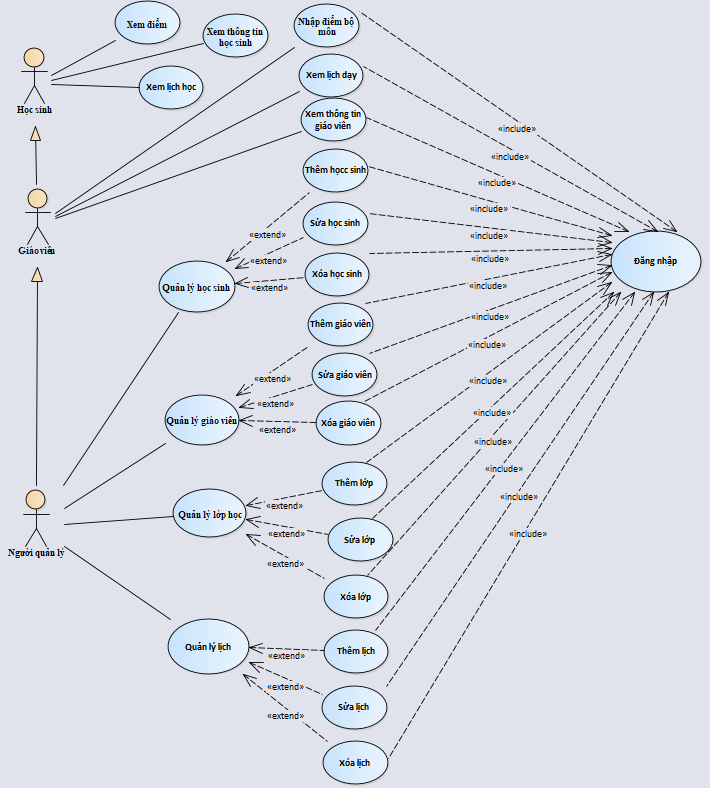
## Simple Page Application

* **Single-page Application** là mọt ứng dụng hoạt động trên browser và không cần bắt buộc phải reload lại khi sử dụng. Những trang web là SPA hàng ngày, ví dụ như Facebook, Google, hay Gmail ....Trang web chỉ cần load trang một lần, sau đó sẽ sử dụng javascript để load những dữ liệu mà bạn cần dùng (Đây có thể coi vừa là điểm mạnh cũng như điểm yếu của một ứng dụng SPA)
* Với SPA, chúng ta sẽ được tiếp cận một cụm từ khác là **Client Side Rendering** khi Client mới là người render ra từ những dòng HTML, CSS để hiển thị. Nhờ đó mà chúng ta cũng tách biệt được giữ nhưng công việc mà Sever phải làm khi lượng người dùng Internet ngày càng đông, khiến Sever phải chịu tải rất nặng khi vừa phải xử lý dữ liệu lại phải render ra để trả lại cho người dùng.

# : PHÂN TÍCH VÀ THIẾT KẾ

## Phân tích yêu cầu bằng UML

### Usecase tổng quát



Hình 3‑1-1 UseCase tổng quát

### Danh sách tác nhân và mô tả

Danh sách tác nhân và mô tả xem ở Bảng 3-1.

|  |  |
| --- | --- |
| Tác nhân | Mô tả tác nhân |
| Học Sinh | Truy cập vào website để xem thông tin, lịch học , điểm. |
| Giáo Viên | Đăng nhập vào website để xem thông tin, lịch dạy, và nhập điểm cho học sinh. |
| Admin | Quản lý học sinh, giáo viên và lịch . |

Bảng 3‑1 Danh sách tác nhân và mô tả

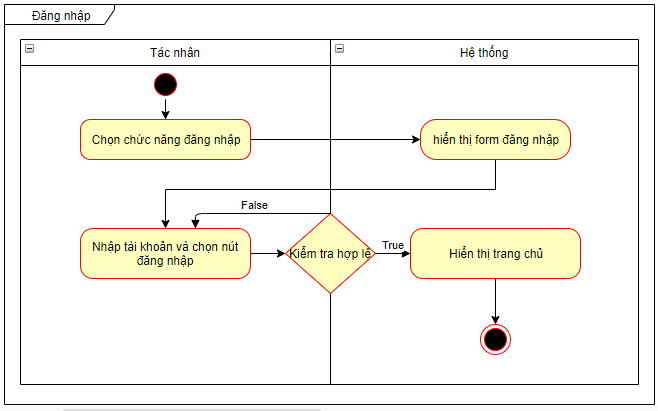
### Danh sách các tình huống hoạt động (Use cases)

|  |  |
| --- | --- |
| ID | Usecase website quản lý trường trung học phổ thông |
| UC01 | Đăng nhập |
| UC02 | Xem điểm |
| UC03 | Xem thông tin học sinh |
| UC04 | Xem lịch học |
| UC05 | Nhập điểm bộ môn |
| UC06 | Xem lịch dạy |
| UC07 | Xem thông tin giáo viên |
| UC08 | Thêm học sinh |
| UC09 | Sửa thông tin học sinh |
| UC10 | Xoá học sinh |
| UC11 | Thêm giáo viên |
| UC12 | Sửa giáo viên |
| UC13 | Xoá giáo viên |
| UC14 | Thêm lớp |
| UC15 | Sửa lớp |
| UC16 | Xoá lớp |
| UC17 | Thêm lịch |
| UC18 | Sửa lịch |
| UC19 | Xoá lịch |

### Tình huống hoạt động

|  |  |  |
| --- | --- | --- |
| UC01 : Đăng nhập | | |
| Mục đích: | Đăng nhập vào website | |
| Mô tả: | Tác nhân đăng nhập vào website để truy cập hệ thống. | |
| Tác nhân: | Admin, Giáo Viên | |
| Điều kiện trước: | Đã được cấp tài khoản. | |
| Điều kiện sau: | Đăng nhập thành công vào hệ thông | |
| Luồng sự kiện chính | Tác nhân | Hệ thống |
| 1) Tác nhân chọn chức năng đăng nhập  3 ) Tác nhân nhập tài khoản và nhấn nút “đăng nhập” | 2 )Hệ thống hiển thị form đăng nhập  4) Nếu đăng nhập thành công hệ thống hiện thị trang chủ website |
| Luồng sự kiện phụ | 4.1 Nếu thông tin không hợp lệ thông báo lỗi, quay lại bước 3 | |

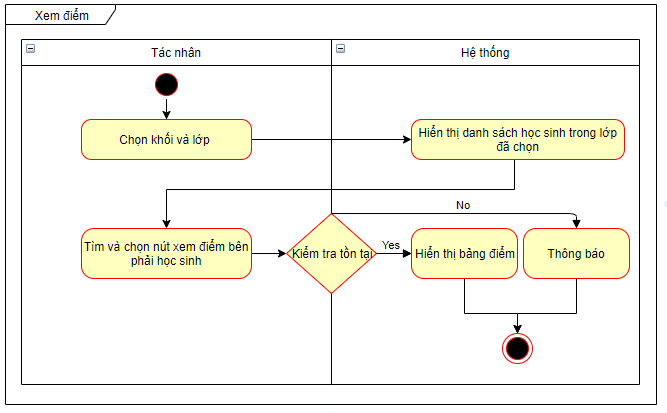
Bảng -3-1: Usecase Đăng nhập



Hình 3‑1-1 : Activity Đăng nhập

|  |  |  |
| --- | --- | --- |
| UC02 : Xem điểm | | |
| Mục đích: | Xem điểm cá nhân | |
| Mô tả: | Người dùng truy cập vào website để xem điểm | |
| Tác nhân: | Học sinh | |
| Điều kiện trước: | Truy cập vào trang chủ | |
| Điều kiện sau: | Xem được bảng điểm của học sinh | |
| Luồng sự kiện chính | Tác nhân | Hệ thống |
| 1) Tác nhân chọn khối và lớp của học sinh cần xem.  3) Tìm và nhấn chọn chức năng xem điểm bên phải học sinh . | 2) Hệ thống hiển thị danh sách học sinh trong lớp đã chọn  4) Hiển thị bảng điểm gồm tất cả các môn của học sinh. |
| Luồng sự kiện phụ | 4.1) Nếu chưa có điểm hệ thống hiển thị thông báo. | |

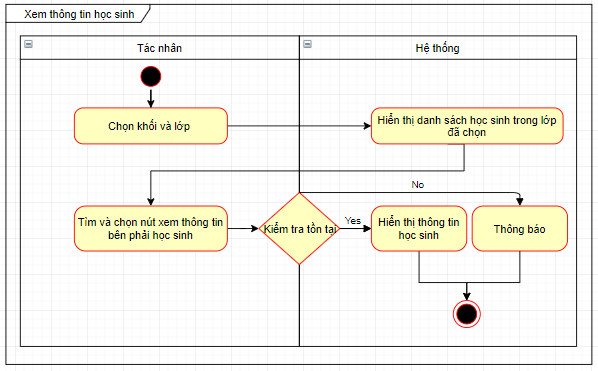
Bảng 3-3-2: Usecase Xem điểm



Hình 3‑1-2 : Activity Xem điểm

|  |  |  |
| --- | --- | --- |
| UC03 : Xem thông tin học sinh | | |
| Mục đích: | Xem thông tin của học sinh | |
| Mô tả: | Người dùng truy cập vào website để thông tin cá nhân | |
| Tác nhân: | Học sinh | |
| Điều kiện trước: | Truy cập vào hệ thống | |
| Điều kiện sau: | Xem được thông tin cá nhân của học sinh | |
| Luồng sự kiện chính | Tác nhân | Hệ thống |
| 1) Tác nhân chọn khối và lớp của học sinh cần xem.  3) Tìm và nhấn chọn chức năng xem thông tin bên phải học sinh . | 2) Hệ thống hiển thị danh sách học sinh trong lớp đã chọn  4) Hiển thị thông tin của học sinh . |
| Luồng sự kiện phụ | 4.1) Nếu chưa có thông hệ thống thông báo. | |

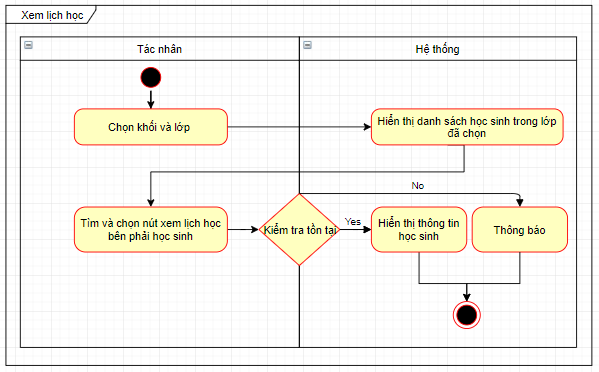
Bảng 3-3-3: Usecase Xem thông tin cá nhân



Hình 3‑1-3 : Activity Xem thông tin học sinh

|  |  |  |
| --- | --- | --- |
| UC04 : Xem lịch học | | |
| Mục đích: | Xem lịch học của học sinh | |
| Mô tả: | Tác nhân vào xem lịch học của học sinh | |
| Tác nhân: | Học sinh | |
| Điều kiện trước: | Đăng nhập vào hệ thông | |
| Điều kiện sau: | Xem lịch học thành công | |
| Luồng sự kiện chính | Tác nhân | Hệ thống |
| 1) Tác nhân chọn khối và lớp của học sinh cần xem.  3) Tìm và nhấn chọn chức năng xem lịch học bên phải học sinh . | 2) Hệ thống hiển thị danh sách học sinh trong lớp đã chọn  4) Hiển thị lịch học của học sinh. |
| Luồng sự kiện phụ | 4.1) Nếu chưa có lịch học hệ thống thông báo. | |

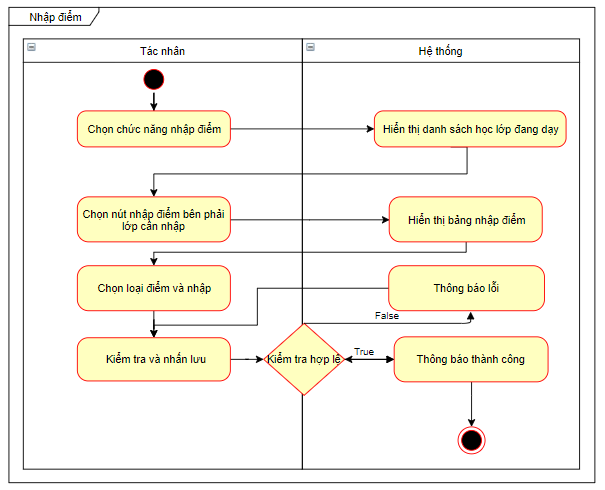
Bảng 3-3-4: Usecase Xem lịch học



Hình 3‑1-4 : Activity Xem lịch học

|  |  |  |
| --- | --- | --- |
| UC05 : Nhập điểm bộ môn | | |
| Mục đích: | Nhập điểm cho học sinh | |
| Mô tả: | Tác nhân nhập điểm bộ môn cho học sinh các lớp dạy. | |
| Tác nhân: | Giáo viên | |
| Điều kiện trước: | Đăng nhập vào hệ thông | |
| Điều kiện sau: | Nhập thành công điểm cho học sinh | |
| Luồng sự kiện chính | Tác nhân | Hệ thống |
| 1) Chọn chức năng “nhập điểm”.  3 ) Chọn nút nhập điểm bên phải lớp muốn nhập.  5) Chọn loại điểm và nhập cho điểm cho từng học sinh  6) Kiểm tra và nhấn lưu | 2 )Hiển thị danh sách các lớp đang dạy.  4) Hiển thị bảng nhập điểm gồm các học sinh trong lớp.  7) Nếu điểm hợp lệ hệ thống thông báo thành công và lưu. |
| Luồng sự kiện phụ | 7.1) Nếu không hợp lê hệ thống thông báo lỗi. | |

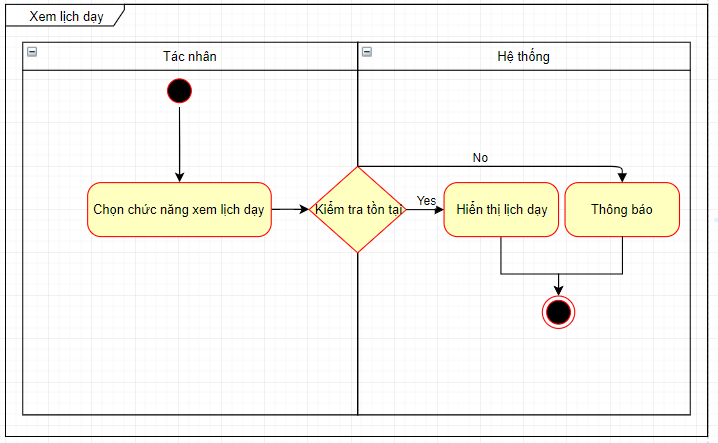
Bảng 3-3-5: Usecase Nhập điểm bộ môn



Hình 3‑1-5 : Activity Nhập điểm

|  |  |  |
| --- | --- | --- |
| UC06 : Xem lịch dạy | | |
| Mục đích: | Xem lịch dạy của giáo viên | |
| Mô tả: | Tác nhân vào xem lịch dạy của mình. | |
| Tác nhân: | Giáo viên | |
| Điều kiện trước: | Đăng nhập vào hệ thông | |
| Điều kiện sau: | Xem lịch dạy thành công | |
| Luồng sự kiện chính | Tác nhân | Hệ thống |
| 1) Tác nhân chọn chức năng xem lịch dạy | 2)Hệ thống hiển thị lịch dạy. |
| Luồng sự kiện phụ | 2.1) Nếu không có lịch dạy hệ thống thông báo. | |

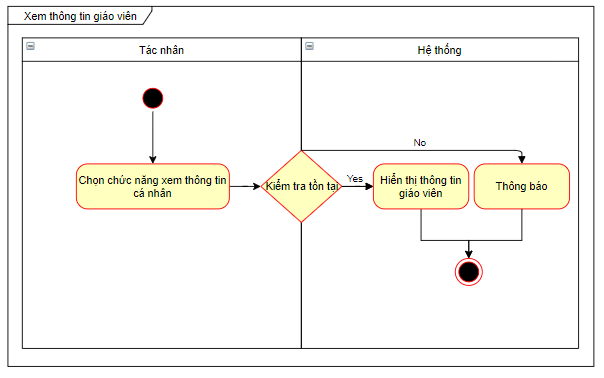
Bảng 3-3-6: Usecase Xem lịch dạy



Hình 3‑1-6 : Activity Xem lịch dạy

|  |  |  |
| --- | --- | --- |
| UC07 : Xem thông tin giáo viên | | |
| Mục đích: | Xem thông tin của giáo viên | |
| Mô tả: | Tác nhân vào xem thông tin cá nhân của bản thân. | |
| Tác nhân: | Giáo viên | |
| Điều kiện trước: | Đăng nhập vào hệ thông | |
| Điều kiện sau: | Xem thông tin thành công | |
| Luồng sự kiện chính | Tác nhân | Hệ thống |
| 1) Tác nhân chọn chức năng xem thông tin cá nhân | 2 )Hệ thống hiển thị trang xem thông tin giáo viên |
| Luồng sự kiện phụ | 2.1) Nếu chưa có thông tin hệ thống thông báo. | |

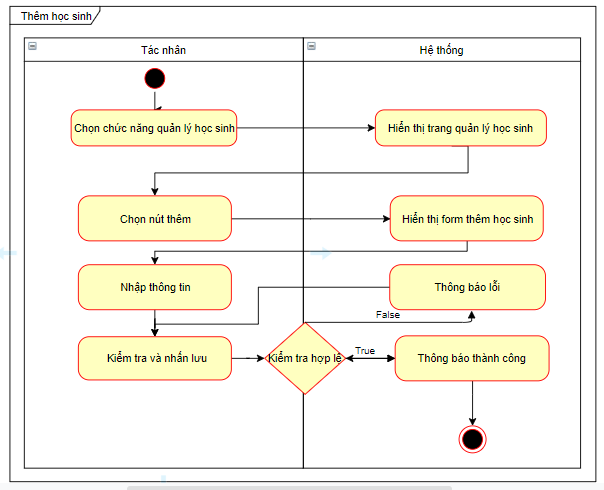
Bảng 3-3-7: Usecase Xem thông tin giáo viên



Hình 3‑1-7 : Activity Xem thông tin giáo viên

|  |  |  |
| --- | --- | --- |
| UC08 : Thêm học sinh | | |
| Mục đích: | Thêm mới học sinh | |
| Mô tả: | Tác nhân thêm học sinh mới vào hệ thống | |
| Tác nhân: | Quản lý | |
| Điều kiện trước: | Đăng nhập vào với quyền quản lý. | |
| Điều kiện sau: | Thêm thành công học sinh | |
| Luồng sự kiện chính | Tác nhân | Hệ thống |
| 1) Chọn chức năng quản lý học sinh.  3 ) Chọn nút thêm.  5 ) Tác nhân nhập thông tin  6) Nhấn lưu | 2 )Hiển thị trang quản lý học sinh.  4) Hiển thị form thêm học sinh  7) Nếu thông tin hợp lệ hệ thống thông báo thành công và lưu học sinh. |
| Luồng sự kiện phụ | 7.1) Nếu không hợp lệ hệ thống thông báo lỗi, quay lai bước 6 | |

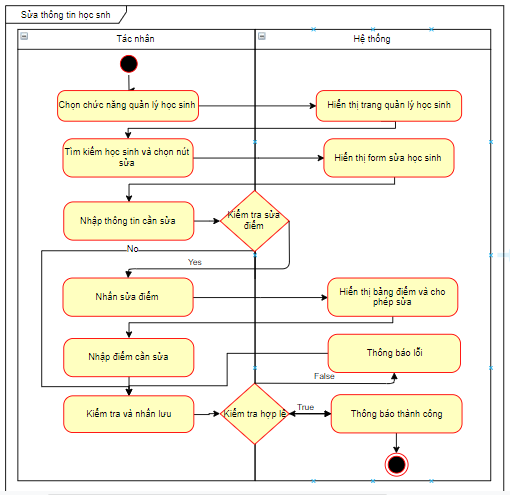
Bảng 3-3-8: Usecase Thêm học sinh



Hình 3‑1-8 : Activity Thêm học sinh

|  |  |  |
| --- | --- | --- |
| UC09 : Sửa thông tin học sinh | | |
| Mục đích: | Sửa thông tin của học sinh | |
| Mô tả: | Sửa các thông tin của học sinh và điểm. | |
| Tác nhân: | Quản lý | |
| Điều kiện trước: | Đăng nhập với quyền quản lý | |
| Điều kiện sau: | Sửa thông tin học sinh thành công | |
| Luồng sự kiện chính | Tác nhân | Hệ thống |
| 1) Tác nhân chọn chức năng quản lý học sinh  3 ) Tìm kiếm học sinh và chọn nút “sửa học sinh”  5) Tác nhân nhập thông tin cần sửa  6) Nếu sửa điểm tác nhân nhấn sửa điểm.  8) Sửa lại điểm  9) Nhấn lưu | 2 ) Hiển thị trang quản lý học sinh.  4) Hệ thông hiển thị form sửa học sinh  7) Hệ thống hiển thị bảng điểm của học sinh và cho phép sửa.  10) Nếu thông tin hợp lệ hệ thống thông báo thành công và lưu. |
| Luồng sự kiện phụ | 6.1) Nếu không sửa điểm tác nhân nhấn lưu.  10.1) Nếu thông tin không hợp lệ hệ thống thông báo lỗi và quay lại bước 5. | |

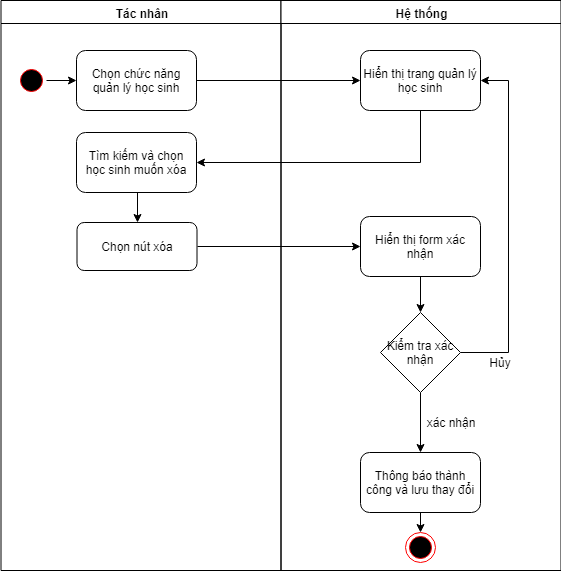
Bảng 3-3-9: Usecase Sửa thông tin học sinh



Hình 3‑1-9 : Activity Sửa thông tin học sinh

|  |  |  |
| --- | --- | --- |
| UC10 : Xoá học sinh | | |
| Mục đích: | Xoá học sinh | |
| Mô tả: | Tác nhân xoá học sinh khỏi hệ thống. | |
| Tác nhân: | Quản lý | |
| Điều kiện trước: | Đăng nhập với quyền quản lý. | |
| Điều kiện sau: | Xoá thành công học sinh | |
| Luồng sự kiện chính | Tác nhân | Hệ thống |
| 1) Tác nhân chọn chức năng quản lý học sinh  3 ) Tìm kiếm học sinh và chọn nút xoá  5) Tác nhân chọn nút “xác nhân” | 2 ) Hiển thị trang quản lý học sinh.  4) Hệ thông hiển thị form xác nhận  6) Hệ thống thông báo thành công và lưu thay đổi. |
| Luồng sự kiện phụ | 5.1) Tác nhân chọn “huỷ” quay lại bước 2. | |

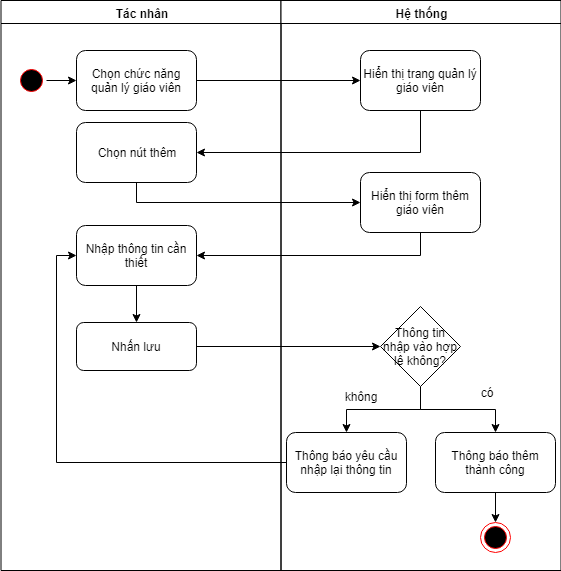
Bảng 3-3-10:Usecase Xóa học sinh



Hình 3‑1-10 : Activity Xoá học sinh

|  |  |  |
| --- | --- | --- |
| UC11 : Thêm giáo viên | | |
| Mục đích: | Thêm giáo viên mới vào hệ thống | |
| Mô tả: | Người quản lý thực hiện việc thêm một giáo viên mới vào hệ thống | |
| Tác nhân: | Người quản lý | |
| Điều kiện trước: | Đăng nhập vào hệ thống với tài khoản của Người quản lý | |
| Điều kiện sau: | Thêm thành công giáo viên vào hệ thống | |
| Luồng sự kiện chính | Tác nhân | Hệ thống |
| 1) Chọn chức năng quản lý giáo viên  3 ) Chọn nút thêm.  5 ) Tác nhân nhập thông tin  6) Nhấn lưu | 2 )Hiển thị trang quản lý giáo viên  4) Hiển thị form thêm giáo viên  7) Nếu thông tin hợp lệ hệ thống thông báo thành công và lưu giáo viên. |
| Luồng sự kiện phụ | 7.1)Nếu không hợp lệ thì hiển thị thông báo yêu cầu nhập lại thông tin. | |

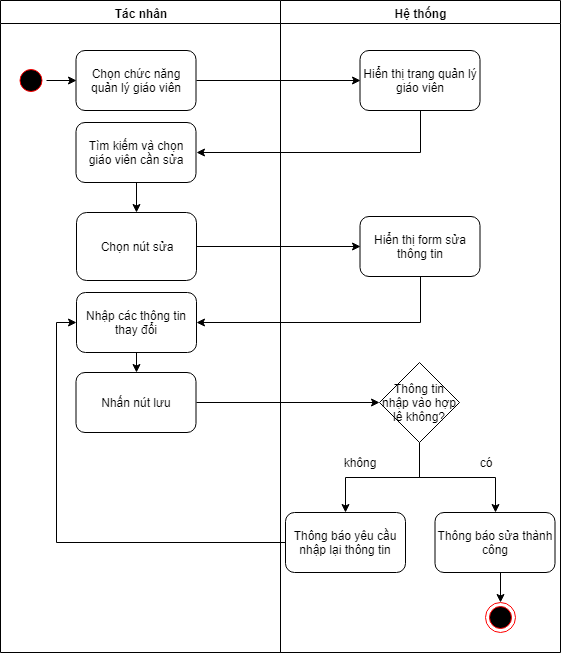
Bảng 3-3-11: Usecase Thêm giáo viên



Hình 3‑1-11 : Activity Thêm giáo viên

|  |  |  |
| --- | --- | --- |
| UC12 : Sửa giáo viên | | |
| Mục đích: | Sửa thông tin của giáo viên trong hệ thống | |
| Mô tả: | Người quản lý thực hiện việc sửa thông tin của một giáo viên trong hệ thống | |
| Tác nhân: | Người quản lý | |
| Điều kiện trước: | Đăng nhập vào hệ thống với tài khoản của Người quản lý | |
| Điều kiện sau: | Sửa thành công thông tin của giáo viên trong hệ thống | |
| Luồng sự kiện chính | Tác nhân | Hệ thống |
| 1) Chọn chức năng quản lý giáo viên  3) Tác nhân tìm kiếm giáo viên và chọn nút sửa  5)Tác nhân nhập các thông tin thay đổi của giáo viên  6)Tác nhân nhấn nút lưu | 2 )Hiển thị trang quản lý giáo viên  4) Hiển thị form sửa thông tin.  7)Nếu dữ liệu nhập vào hợp lệ thì hiển thị thông báo thành công và lưu thay đổi. |
| Luồng sự kiện phụ | 7.1)Nếu không hợp lệ thì hiển thị thông báo yêu cầu nhập lại thông tin. | |

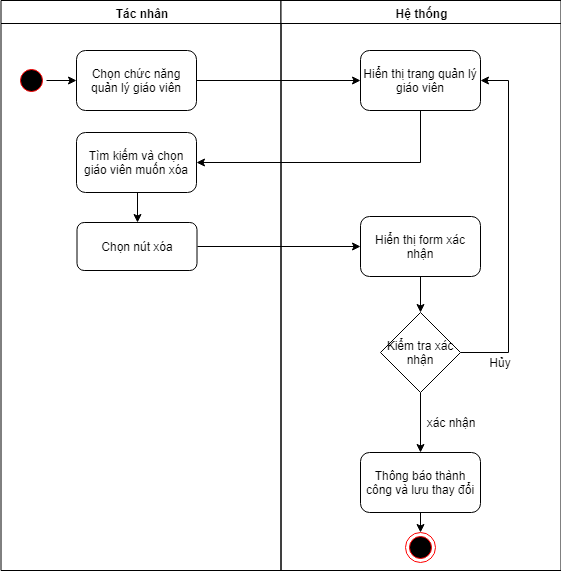
Bảng 3-3-12: Usecase Sửa giáo viên

\

Hình 3‑1-12 : Activity Sửa giáo viên

|  |  |  |
| --- | --- | --- |
| UC13 : Xóa giáo viên | | |
| Mục đích: | Xóa giáo viên khỏi hệ thống | |
| Mô tả: | Người quản lý thực hiện việc xóa một giáo viên khỏi hệ thống | |
| Tác nhân: | Người quản lý | |
| Điều kiện trước: | Đăng nhập vào hệ thống với tài khoản của Người quản lý | |
| Điều kiện sau: | Xóa thành công giáo viên khỏi hệ thống | |
| Luồng sự kiện chính | Tác nhân | Hệ thống |
| 1) Tác nhân chọn chức năng quản lý giáo viên  3 ) Tìm kiếm giáo viên và chọn nút xoá  5) Tác nhân chọn nút “xác nhân” | 2 ) Hiển thị trang quản lý giáo viên  4) Hệ thông hiển thị form xác nhận  6) Hệ thống thông báo thành công và lưu thay đổi. |
| Luồng sự kiện phụ | 5.1) Tác nhân chọn “huỷ” quay lại bước 2. | |

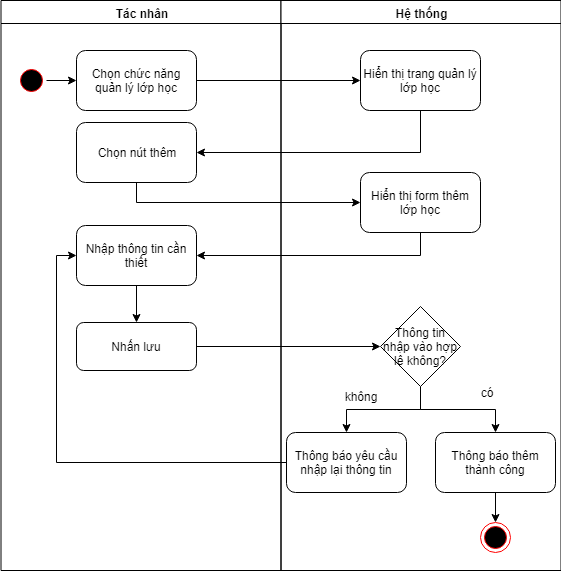
Bảng 3-3-13: Usecase Xóa giáo viên



Hình 3‑1-13 : Activity Xoá giáo viên

|  |  |  |
| --- | --- | --- |
| UC14 : Thêm lớp | | |
| Mục đích: | Thêm lớp mới vào hệ thống | |
| Mô tả: | Người quản lý thực hiện việc thêm một lớp mới vào hệ thống | |
| Tác nhân: | Người quản lý | |
| Điều kiện trước: | Đăng nhập vào hệ thống với tài khoản của Người quản lý | |
| Điều kiện sau: | Thêm thành công lớp vào hệ thống | |
| Luồng sự kiện chính | Tác nhân | Hệ thống |
| 1) Chọn chức năng quản lý lớp học  3 ) Chọn nút thêm.  5 ) Tác nhân nhập thông tin  6) Nhấn lưu | 2 )Hiển thị trang quản lý lớp học  4) Hiển thị form thêm lớp học  7) Nếu thông tin hợp lệ hệ thống thông báo thành công và lưu |
| Luồng sự kiện phụ | 7.1)Nếu không hợp lệ thì hiển thị thông báo yêu cầu nhập lại. | |

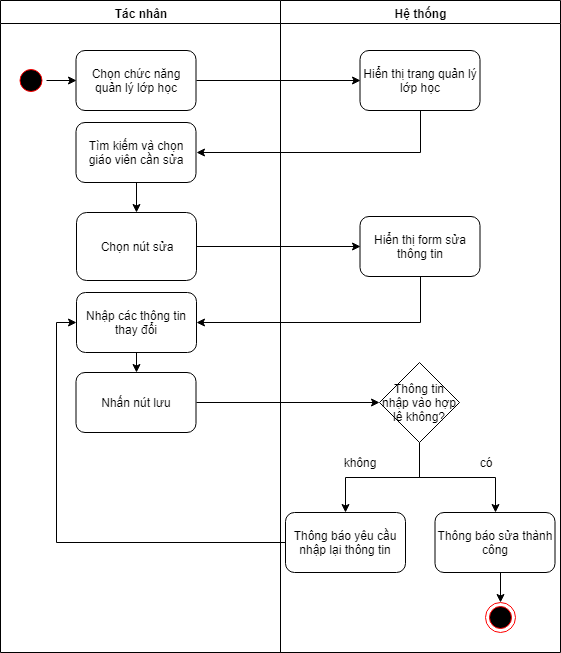
Bảng 3-3-14: Usecase Thêm lớp



Hình 3‑1-14 : Activity Thêm lớp

|  |  |  |
| --- | --- | --- |
| UC15 : Sửa lớp | | |
| Mục đích: | Sửa thông tin của lớp trong hệ thống | |
| Mô tả: | Người quản lý thực hiện việc sửa thông tin của một lớp trong hệ thống | |
| Tác nhân: | Người quản lý | |
| Điều kiện trước: | Đăng nhập vào hệ thống với tài khoản của Người quản lý | |
| Điều kiện sau: | Sửa thành công thông tin của lớp trong hệ thống | |
| Luồng sự kiện chính | Tác nhân | Hệ thống |
| 1) Tác nhân chọn chức năng quản lý lớp học  3) Tác nhân tìm kiếm lớp và chọn nút sửa  6)Tác nhân nhập các thông tin thay đổi của lớp  7)Tác nhân nhấn nút lưu | 2) Hiển thị trang quản lý lớp học  5) Hiển thị form sửa lớp  7)Nếu dữ liệu nhập vào hợp lệ thì hiển thị thông báo sửa lớp thành công |
| Luồng sự kiện phụ | 7.1)Nếu không hợp lệ thì hiển thị thông báo yêu cầu nhập lại thông tin. | |

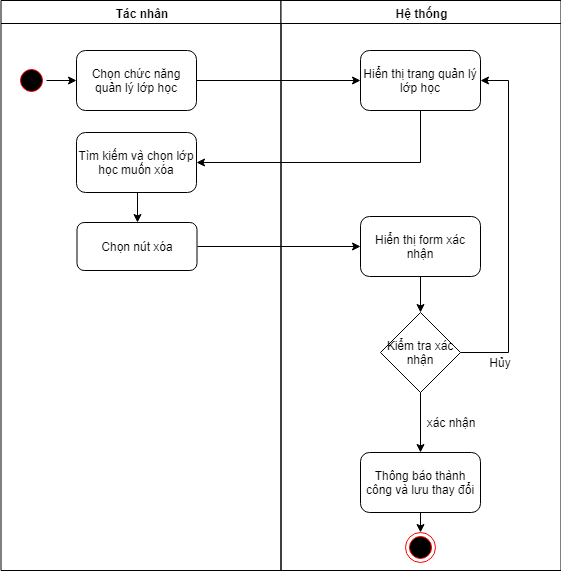
Bảng 3-3-15: Usecase Sửa lớp



Hình 3‑1-15 : Activity Sửa lớp

|  |  |  |
| --- | --- | --- |
| UC16 : Xóa lớp | | |
| Mục đích: | Xóa lớp khỏi hệ thống | |
| Mô tả: | Người quản lý thực hiện việc xóa một lớp khỏi hệ thống | |
| Tác nhân: | Người quản lý | |
| Điều kiện trước: | Đăng nhập vào hệ thống với tài khoản của Người quản lý | |
| Điều kiện sau: | Xóa thành công lớp khỏi hệ thống | |
| Luồng sự kiện chính | Tác nhân | Hệ thống |
| 1) Tác nhân chọn chức năng quản lý lớp học  3 ) Tìm kiếm lớp học và chọn nút xoá  5) Tác nhân chọn nút “xác nhân” | 2 ) Hiển thị trang quản lớp học  4) Hệ thông hiển thị form xác nhận  6) Hệ thống thông báo thành công và lưu thay đổi. |
| Luồng sự kiện phụ | 5.1) Tác nhân chọn “huỷ” quay lại bước 2. | |

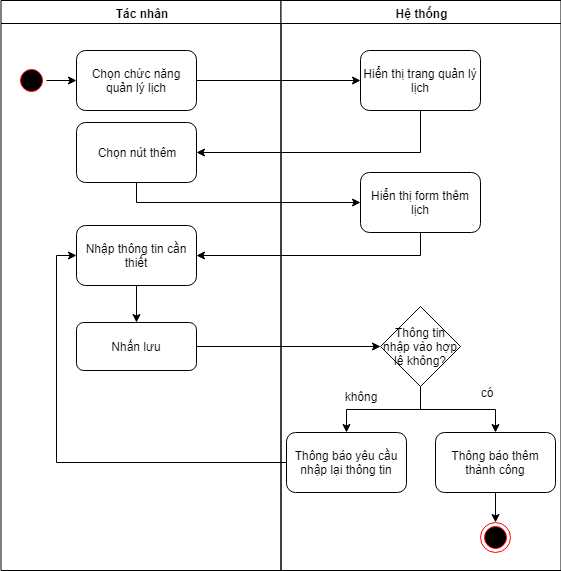
Bảng 3-3-16: Usecase Xóa lớp



Hình 3‑1-16 : Activity Xoá lớp

|  |  |  |
| --- | --- | --- |
| UC17 : Thêm lịch | | |
| Mục đích: | Thêm lịch | |
| Mô tả: | Người quản lý thực hiện việc thêm một lịch mới vào hệ thống | |
| Tác nhân: | Người quản lý | |
| Điều kiện trước: | Đăng nhập vào hệ thống với tài khoản của Người quản lý | |
| Điều kiện sau: | Thêm thành công lịch vào hệ thống | |
| Luồng sự kiện chính | Tác nhân | Hệ thống |
| 1) Chọn chức năng quản lý lịch  3 ) Chọn nút thêm.  5 ) Tác nhân nhập thông tin  6) Nhấn lưu | 2 )Hiển thị trang quản lý lịch  4) Hiển thị form thêm lịch  7) Nếu thông tin hợp lệ hệ thống thông báo thành công và lưu |
| Luồng sự kiện phụ | 7.1)Nếu không hợp lệ thì hiển thị thông báo yêu cầu nhập lại thông tin. | |

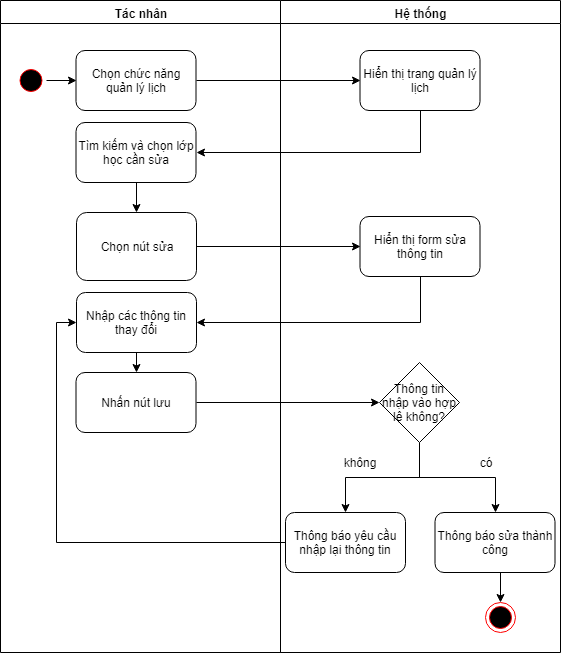
Bảng 3-3-17: Usecase Thêm lịch



Hình 3‑1-17 : Activity Thêm lịch

|  |  |  |
| --- | --- | --- |
| UC18 : Sửa lịch | | |
| Mục đích: | Sửa thông tin của lịch trong hệ thống | |
| Mô tả: | Người quản lý thực hiện việc sửa thông tin của lịch trong hệ thống | |
| Tác nhân: | Người quản lý | |
| Điều kiện trước: | Đăng nhập vào hệ thống với tài khoản của Người quản lý | |
| Điều kiện sau: | Sửa thành công thông tin của lịch trong hệ thống | |
| Luồng sự kiện chính | Tác nhân | Hệ thống |
| 1) Tác nhân chọn chức năng quản lý lịch  3) Tác nhân tìm kiếm lịch và chọn sửa  6)Tác nhân nhập các thông tin thay đổi của lịch  7)Nhấn nút lưu | 2) Hiển thị trang quản lý lịch  5) Hiển thị form sửa lịch  7)Nếu dữ liệu nhập vào hợp lệ thì hiển thị thông báo sửa lịch thành công |
| Luồng sự kiện phụ | 7.1)Nếu không hợp lệ thì hiển thị thông báo yêu cầu nhập lại thông tin. | |

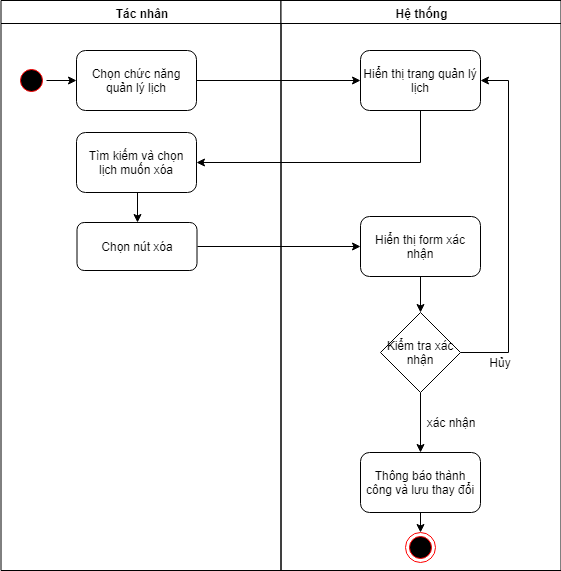
Bảng 3-3-18: Usecase Sửa lịch



Hình 3‑1-18 : Activity Sửa lịch

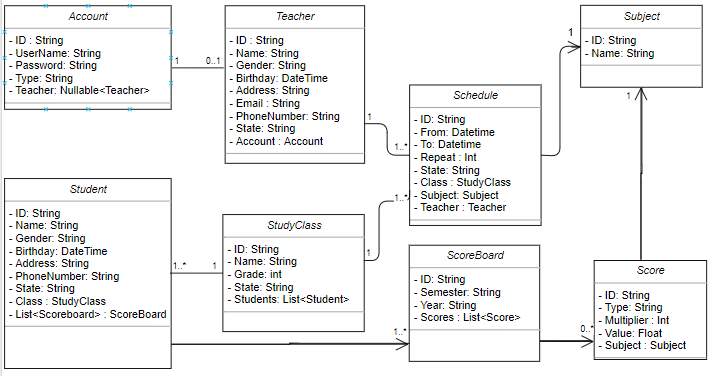
|  |  |  |
| --- | --- | --- |
| UC19 : Xóa lịch | | |
| Mục đích: | Xóa lịch khỏi hệ thống | |
| Mô tả: | Người quản lý thực hiện việc xóa lịch khỏi hệ thống | |
| Tác nhân: | Người quản lý | |
| Điều kiện trước: | Đăng nhập vào hệ thống với tài khoản của Người quản lý | |
| Điều kiện sau: | Xóa thành công lịch khỏi hệ thống | |
| Luồng sự kiện chính | Tác nhân | Hệ thống |
| 1) Tác nhân chọn chức năng quản lý lịch  3 ) Tìm kiếm lịc và chọn nút xoá  5) Tác nhân chọn nút “xác nhân” | 2 ) Hiển thị trang quản lịch  4) Hệ thống hiển thị form xác nhận  6) Hệ thống thông báo thành công và lưu thay đổi. |
| Luồng sự kiện phụ | 5.1) Tác nhân chọn “huỷ” quay lại bước 2. | |

Bảng 3-3-19: Usecase Xóa tiết dạy



Hình 3‑1-19 : Activity Xoá lịch

## Class diagram



Hình 3‑1-20 : ClassDiagram

## Deployment diagram

# : HIỆN THỰC

## Cấu hình phần cứng, phần mềm

Ghi rõ thông tin sử dụng các dịch vụ nào của AWS để thực hiện đồ án.

Cấu hình phần cứng, phần mềm của Server (nếu có).

Cấu hình phần cứng, phần mềm của Client trong hệ thống (nếu có).

## Giao diện của hệ thống

## Kế hoạch và hiện thực kiểm thử hệ thống

### Kế hoạch kiểm thử

### Kiểm thử hệ thống

# : KẾT LUẬN

## Kết quả đạt được

## Hạn chế của đồ án

## Hướng phát triển

# TÀI LIỆU THAM KHẢO

Các tài liệu Tiếng Anh

1. Kevin Faaborg & Sandro Pasquali - Mastering Node.js Second Edition – December 2017
2. David Herron, Diogo Resende & Valentin Bojinov - Node.js Complete Reference Guide – December 2018

Các tài liệu từ Internet

1. <https://nodejs.org/en/docs/>
2. <https://expressjs.com/en/guide/routing.html>