**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](#_Toc22590604)

[DANH MỤC CÁC HÌNH VẼ 5](#_Toc22590605)

[DANH MỤC CÁC BẢNG BIỂU 7](#_Toc22590606)

[CHƯƠNG 1 : GIỚI THIỆU 8](#_Toc22590607)

[1.1 Tổng quan 8](#_Toc22590608)

[1.2 Mục tiêu đề tài 9](#_Toc22590609)

[1.3 Phạm vi đề tài 9](#_Toc22590610)

[1.4 Mô tả yêu cầu chức năng 9](#_Toc22590611)

[CHƯƠNG 2 : CƠ SỞ LÝ THUYẾT 11](#_Toc22590612)

[2.1 Cloud Computing với AWS 11](#_Toc22590613)

[2.1.1 Điện toán đám mây là gì? 11](#_Toc22590614)

[2.1.2 Điện toán đám mây hoạt động như thế nào? 11](#_Toc22590615)

[2.1.3 AWS là gì? 13](#_Toc22590616)

[2.1.4 DynamoDB 13](#_Toc22590617)

[2.1.5 EC2 15](#_Toc22590618)

[2.1.6 S3 16](#_Toc22590619)

[2.2 Single Page Application 18](#_Toc22590620)

[2.3 Server side 19](#_Toc22590621)

[2.3.1 Node js 19](#_Toc22590622)

[2.3.2 Express 20](#_Toc22590628)

[2.3.2.1 Request & Response trong Expres 20](#_Toc22590630)

[2.3.2.2 Route cở bản trong Express 20](#_Toc22590631)

[2.3.2.3 File tĩnh trong Express 21](#_Toc22590632)

[2.4 DynamoDB 22](#_Toc22590633)

[2.5 Client Sid 22](#_Toc22590634)

[2.5.1 React 22](#_Toc22590635)

[2.5.1.1 DOM ảo (Virtual DOM) là gì ? 23](#_Toc22590636)

[*2.5.1.2* JSX là gì? 25](#_Toc22590637)

[*2.5.1.3* Component là gì *?* 26](#_Toc22590638)

[2.5.1.4 Props là gì *?* 27](#_Toc22590639)

[*2.5.1.5* State là gì*?* 28](#_Toc22590640)

[2.5.1.6 React xử lý style như thế nào? 29](#_Toc22590641)

[*2.5.1.7* Lifecycle Methods là gì *?* 30](#_Toc22590642)

[2.5.2 React- Router 31](#_Toc22590643)

[2.5.2.1 BrowserRouter vs HashRouter 31](#_Toc22590644)

[2.5.2.2 Route 31](#_Toc22590645)

[2.5.2.3 Link 32](#_Toc22590646)

[2.5.2.4 NavLink 32](#_Toc22590647)

[2.5.2.5 Custom Link 33](#_Toc22590648)

[2.5.2.6 Đối tượng Match 33](#_Toc22590649)

[2.5.2.7 Đối tượng prompt - Xác nhận trước khi chuyển trang 33](#_Toc22590650)

[2.5.2.8 Redirect 34](#_Toc22590651)

[2.5.3 Ant Design 34](#_Toc22590652)

[2.5.4 Axios 35](#_Toc22590653)

[2.5.4.1 Thực hiện request với phương thức Get 35](#_Toc22590654)

[2.5.4.2 Thực hiện request với phương thức Post 35](#_Toc22590655)

[2.5.4.3 Thực hiện nhiều truy cập đồng thời 36](#_Toc22590656)

[2.5.4.4 Các alias cho việc tạo ra các request với phương thức tương ứng 36](#_Toc22590657)

[2.5.4.5 Tạo ra một đối tượng Axios 36](#_Toc22590658)

[2.5.4.6 Cấu trúc dữ liệu trả về 37](#_Toc22590659)

[2.6 Restful API 37](#_Toc22590660)

[CHƯƠNG 3 : PHÂN TÍCH VÀ THIẾT KẾ 39](#_Toc22590661)

[3.1 Phân tích yêu cầu bằng UML 39](#_Toc22590662)

[3.1.1 Usecase tổng quát 39](#_Toc22590663)

[3.1.2 Danh sách tác nhân và mô tả 39](#_Toc22590664)

[3.1.3 Danh sách các tình huống hoạt động (Use cases) 40](#_Toc22590665)

[3.1.4 Tình huống hoạt động 40](#_Toc22590666)

[3.2 Class diagram 65](#_Toc22590667)

[3.3 Deployment diagram 65](#_Toc22590668)

[3.4 Giao diện Mockup 66](#_Toc22590670)

[CHƯƠNG 4 : HIỆN THỰC 74](#_Toc22590671)

[4.1 Cấu hình phần cứng, phần mềm 74](#_Toc22590672)

[4.2 Kế hoạch và hiện thực kiểm thử hệ thống 74](#_Toc22590673)

[4.2.1 Kế hoạch kiểm thử 74](#_Toc22590674)

[4.2.2 Kiểm thử hệ thống 74](#_Toc22590675)

[CHƯƠNG 5 : KẾT LUẬN 75](#_Toc22590676)

[5.1 Kết quả đạt được 75](#_Toc22590677)

[5.2 Hạn chế của đồ án 75](#_Toc22590678)

[5.3 Hướng phát triển 75](#_Toc22590679)

[TÀI LIỆU THAM KHẢO 76](#_Toc22590680)

# DANH MỤC CÁC HÌNH VẼ

[Hình 2‑1 Dịch vụ trên AWS 12](#_Toc22589906)

[Hình 2‑1-1 AWS DynamoDB 14](#_Toc22589907)

[Hình 2‑1-2 AWS EC2 Intance 15](#_Toc22589908)

[Hình 2‑1-3 Single page Application 17](#_Toc22589909)

[Hình 2‑1-4 Node.js development 18](#_Toc22589910)

[Hình 2‑1-5 DOM 22](#_Toc22589911)

[Hình 2‑1-6 Virtual DOM 23](#_Toc22589912)

[Hình 3‑1-1 UseCase tổng quát 38](#_Toc22589913)

[Hình 3‑1-1 : Activity Đăng nhập 40](#_Toc22589914)

[Hình 3‑1-2 : Activity Xem điểm 41](#_Toc22589915)

[Hình 3‑1-3 : Activity Xem thông tin học sinh 42](#_Toc22589916)

[Hình 3‑1-4 : Activity Xem lịch học 43](#_Toc22589917)

[Hình 3‑1-5 : Activity Nhập điểm 44](#_Toc22589918)

[Hình 3‑1-6 : Activity Xem lịch dạy 45](#_Toc22589919)

[Hình 3‑1-7 : Activity Xem thông tin giáo viên 46](#_Toc22589920)

[Hình 3‑1-8 : Activity Thêm học sinh 47](#_Toc22589921)

[Hình 3‑1-9 : Activity Sửa thông tin học sinh 49](#_Toc22589922)

[Hình 3‑1-10 : Activity Xoá học sinh 50](#_Toc22589923)

[Hình 3‑1-11 : Activity Thêm giáo viên 52](#_Toc22589924)

[Hình 3‑1-12 : Activity Sửa giáo viên 53](#_Toc22589925)

[Hình 3‑1-13 : Activity Xoá giáo viên 55](#_Toc22589926)

[Hình 3‑1-14 : Activity Thêm lớp 56](#_Toc22589927)

[Hình 3‑1-15 : Activity Sửa lớp 58](#_Toc22589928)

[Hình 3‑1-16 : Activity Xoá lớp 59](#_Toc22589929)

[Hình 3‑1-17 : Activity Thêm lịch 60](#_Toc22589930)

[Hình 3‑1-18 : Activity Sửa lịch 62](#_Toc22589931)

[Hình 3‑1-19 : Activity Xoá lịch 63](#_Toc22589932)

[Hình 3‑1-20 : ClassDiagram 64](#_Toc22589933)

[Hình 3‑1-21 : Deployment Diagram 64](#_Toc22589934)

[Hình 3‑1-21 : Trang chủ 65](#_Toc22589935)

[Hình 3‑1-22 : Đăng nhập 65](#_Toc22589936)

[Hình 3‑1-23 : Xem lịch học 66](#_Toc22589937)

[Hình 3‑1-24 : Xem thông tin giáo viên, học sinh 66](#_Toc22589938)

[Hình 3‑1-25 : Xem điểm 67](#_Toc22589939)

[Hình 3‑1-26 : Xem lịch dạy 67](#_Toc22589940)

[Hình 3‑1-27 : Nhập điểm 68](#_Toc22589941)

[Hình 3‑1-27 : Quản lý giáo viên 68](#_Toc22589942)

[Hình 3‑1-28 : Quản lý học sinh 69](#_Toc22589943)

[Hình 3‑1-29 : Quản lý lớp 69](#_Toc22589944)

[Hình 3‑1-29 : Quản lý lịch 70](#_Toc22589945)

[Hình 3‑1-30 : Thêm, sửa học sinh 70](#_Toc22589946)

[Hình 3‑1-31 : Sửa điểm học sinh 71](#_Toc22589947)

[Hình 3‑1-32 : Thêm lịch 71](#_Toc22589948)

[Hình 3‑1-33 : Sửa lịch 72](#_Toc22589949)

# DANH MỤC CÁC BẢNG BIỂU

[Bảng 3‑1 Danh sách tác nhân và mô tả 39](#_Toc22589950)

[Bảng 3-3-1: Usecase Đăng nhập 40](#_Toc22589951)

[Bảng 3-3-2: Usecase Xem điểm 40](#_Toc22589952)

[Bảng 3-3-3: Usecase Xem thông tin cá nhân 41](#_Toc22589953)

[Bảng 3-3-4: Usecase Xem lịch học 42](#_Toc22589954)

[Bảng 3-3-5: Usecase Nhập điểm bộ môn 44](#_Toc22589955)

[Bảng 3-3-6: Usecase Xem lịch dạy 45](#_Toc22589956)

[Bảng 3-3-7: Usecase Xem thông tin giáo viên 45](#_Toc22589957)

[Bảng 3-3-8: Usecase Thêm học sinh 47](#_Toc22589958)

[Bảng 3-3-9: Usecase Sửa thông tin học sinh 48](#_Toc22589959)

[Bảng 3-3-10:Usecase Xóa học sinh 50](#_Toc22589960)

[Bảng 3-3-11: Usecase Thêm giáo viên 51](#_Toc22589961)

[Bảng 3-3-12: Usecase Sửa giáo viên 53](#_Toc22589962)

[Bảng 3-3-13: Usecase Xóa giáo viên 54](#_Toc22589963)

[Bảng 3-3-14: Usecase Thêm lớp 56](#_Toc22589964)

[Bảng 3-3-15: Usecase Sửa lớp 57](#_Toc22589965)

[Bảng 3-3-16: Usecase Xóa lớp 59](#_Toc22589966)

[Bảng 3-3-17: Usecase Thêm lịch 60](#_Toc22589967)

[Bảng 3-3-18: Usecase Sửa lịch 61](#_Toc22589968)

[Bảng 3-3-19: Usecase Xóa tiết dạy 63](#_Toc22589969)

# : 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

* Dựa trên nhu cầu về một trang web hỗ trợ việc quản lý trường trung học phổ thông, nhóm chúng em đã tìm hiểu về các công nghệ mới để xây dựng website: Cloud computing, NodeJS, DynamoDB, React, Restful API, Simple Page Application.
* Từ những thông tin mà nhóm chúng em tìm hiểu được về các công nghệ nêu trên chúng em đã thực hiện việc xây dựng website thực hiện 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

* Website cung cấp các chức năng cơ bản trong việc quản lý trường trung học phổ thông như:
  + Quản lý học sinh, giáo viên, lớp , lịch .
  + Xem lịch học, lịch dạy, nhập, chỉnh sửa điểm, xem thông tin cá nhân, kết quả học tập..
* Thực hiện các chức năng trên nền tảng Node.js với Express Framework.
* Sử dụng các dịch vụ của AWS : DynamoDB, EC2, S3
* Phía giao diện sử dụng Reactjs cùng thư viện hỗ trợ ant-design, axios.

## 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:
* Khi học sinh hoặc phụ huynh muốn xem kết quả học tập hoặc các thông tin khác của học sinh thì chỉ cần truy cập vào tìm kiếm và xem không cần phải đăng nhập.
* Để đảm bảo tính bảo mật các thông tin quan trọng của nhà trường thì giáo viên và người quản lý cần phải đăng nhập bằng tài khoản đã được cấp các quyền cụ thể để thực hiện các chức năng của mình.
* Giáo viên có thể thực hiện các chức năng: nhập điểm bộ môn, xem thông tin các nhân, xem lịch dạy khi đã đăng nhập thành công vào hệ thống.
  + Chức năng nhập điểm: giáo viên chọn chức năng nhập điểm bộ môn sau đó chọn lớp mà mình cần nhập điểm, chọn loại điểm cần nhập và tiến hành nhập điểm cho các học sinh.
  + Chức năng xem thông tin: giáo viên có thể xem các thông tin cá nhân của mình bằng cách chọn chức năng xem thông tin cá nhân để kiểm tra thông tin cá nhân của bản thân có chính xác chưa.
  + Chức năng xem lịch dạy: giáo viên có thể xem lịch dạy cụ thể của mình khi chọn chức năng xem lịch dạy.
* Người quản lý có thể thực hiện các chức năng về: quản lý học sinh, quản lý giáo viên, quản lý lớp học và quản lý lịch khi đăng nhập thành công vào hệ thống.
  + Quản lý học sinh: người quản lý có thể thực hiện các chức năng để quản lý học sinh trong trường bao gồm các chức năng: thêm, xóa, sửa các học sinh.
  + Quản lý giáo viên: người quản lý có thể thực hiện các chức năng để quản lý giáo viên trong trường bao gồm các chức năng: thêm, xóa, sửa các giáo viên.
  + Quản lý lớp học: người quản lý có thể thực hiện các chức năng để quản lý lớp học bao gồm chức năng: thêm một lớp học mới, chỉnh sửa thông tin của lớp học hiện có và xóa một lớp học khỏi hệ thống.
  + Quản lý lịch: người quản lý có thể thực hiện các chức năng để quản lý lịch học cũng như lịch dạy của học sinh và giáo viên bào gồm các chức năng: thêm lịch, sửa thông tin lịch đã có và xóa lịch khỏi hệ thống

# : CƠ SỞ LÝ THUYẾT

## Cloud Computing với AWS

### Điện toán đám mây là gì?

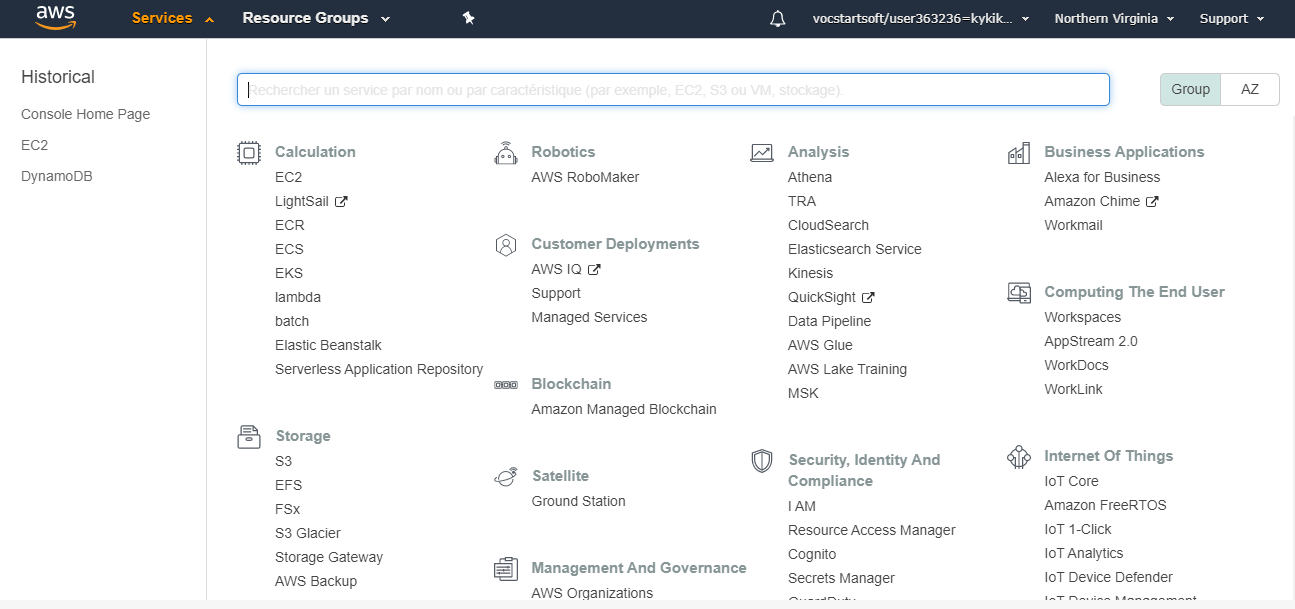
* Điện toán đám mây là việc cung cấp sức mạnh tính toán, cơ sở dữ liệu, khả năng lưu trữ, ứng dụng và các tài nguyên CNTT khác theo yêu cầu qua internet với chính sách thanh toán theo mức sử dụng.
* Cho dù bạn đang sử dụng nền tảng này để chạy các ứng dụng chia sẻ ảnh với hàng triệu người dùng thiết 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.
* Đại bộ phận hạ tầng cơ sở của điện toán đám mây hiện nay là sự kết hợp của những dịch vụ đáng tin cậy được phân phối thông qua các [trung tâm dữ liệu](https://vi.wikipedia.org/w/index.php?title=Trung_t%C3%A2m_d%E1%BB%AF_li%E1%BB%87u&action=edit&redlink=1) (data center) được xây dựng trên những máy chủ với những cấp độ khác nhau của các công nghệ [ảo hóa](https://vi.wikipedia.org/w/index.php?title=%E1%BA%A2o_h%C3%B3a&action=edit&redlink=1). Những dịch vụ này có thể được truy cập từ bất kỳ đâu trên thế giới, trong đó Đám mây là một điểm truy cập duy nhất cho tất cả các máy tính có nhu cầu của khách hàng. Các dịch vụ thương mại cần đáp ứng yêu cầu [chất lượng dịch vụ](https://vi.wikipedia.org/wiki/Ch%E1%BA%A5t_l%C6%B0%E1%BB%A3ng_d%E1%BB%8Bch_v%E1%BB%A5) từ phía khách hàng và thông thường đều đưa ra các [mức thỏa thuận dịch vụ](https://vi.wikipedia.org/w/index.php?title=M%E1%BB%A9c_th%E1%BB%8Fa_thu%E1%BA%ADn_d%E1%BB%8Bch_v%E1%BB%A5&action=edit&redlink=1) (Service level agreement). Các [tiêu chuẩn mở](https://vi.wikipedia.org/w/index.php?title=Ti%C3%AAu_chu%E1%BA%A9n_m%E1%BB%9F&action=edit&redlink=1) (Open standard) và [phần mềm mã nguồn mở](https://vi.wikipedia.org/wiki/Ph%E1%BA%A7n_m%E1%BB%81m_m%C3%A3_ngu%E1%BB%93n_m%E1%BB%9F) (open source software) cũng góp phần phát triển điện toán máy chủ ảo
* Như vậy, trước đây để có thể triển khai một ứng dụng (ví dụ một [trang Web](https://vi.wikipedia.org/wiki/Trang_Web)), bạn phải đi mua/thuê một hay nhiều máy chủ ([server](https://vi.wikipedia.org/wiki/Server)), sau đó đặt máy chủ tại các trung tâm dữ liệu (data center) thì nay điện toán đám mây cho phép bạn giản lược quá trình mua/thuê đi. Bạn chỉ cần nêu ra yêu cầu của mình, hệ thống sẽ tự động gom nhặt các tài nguyên rỗi (free) để đáp ứng yêu cầu của bạn. Chính vì vậy, có thể kể đến một vài lợi ích cơ bản của điện toán đám mây như sau:
  + Sử dụng các [tài nguyên tính toán động](https://vi.wikipedia.org/w/index.php?title=T%C3%A0i_nguy%C3%AAn_t%C3%ADnh_to%C3%A1n_%C4%91%E1%BB%99ng&action=edit&redlink=1) (Dynamic computing resources): Các tài nguyên được cấp phát cho doanh nghiệp đúng như những gì doanh nghiệp muốn một cách tức thời. Thay vì việc doanh nghiệp phải tính toán xem có nên mở rộng hay không, phải đầu tư bao nhiêu máy chủ thì nay doanh nghiệp chỉ cần yêu cầu "Hey, đám mây, chúng tôi cần thêm tài nguyên tương đương với 1 CPU 3.0 GHz, 128GB RAM…" và đám mây sẽ tự tìm kiếm tài nguyên rỗi để cung cấp cho bạn.
  + Giảm chi phí: Doanh nghiệp sẽ có khả năng cắt giảm chi phí để mua bán, cài đặt và bảo trì tài nguyên. Rõ ràng thay vì việc phải cử một chuyên gia đi mua máy chủ, cài đặt máy chủ, bảo trì máy chủ thì nay bạn chẳng cần phải làm gì ngoài việc xác định chính xác tài nguyên mình cần và yêu cầu. Quá tiện!.
  + Giảm độ phức tạp trong cơ cấu của [doanh nghiệp](https://vi.wikipedia.org/wiki/Doanh_nghi%E1%BB%87p): [Doanh nghiệp](https://vi.wikipedia.org/wiki/Doanh_nghi%E1%BB%87p) sản xuất hàng hóa mà lại phải có cả một [chuyên gia](https://vi.wikipedia.org/wiki/Chuy%C3%AAn_gia) [IT](https://vi.wikipedia.org/wiki/IT) để vận hành, bảo trì máy chủ thì quá tốn kém. Nếu [khoán ngoài](https://vi.wikipedia.org/w/index.php?title=Kho%C3%A1n_ngo%C3%A0i&action=edit&redlink=1) được quá trình này thì [doanh nghiệp](https://vi.wikipedia.org/wiki/Doanh_nghi%E1%BB%87p) sẽ chỉ tập trung vào việc sản xuất hàng hóa chuyên môn của mình và giảm bớt được độ phức tạp trong cơ cấu.
  + Tăng khả năng sử dụng tài nguyên tính toán: Một trong những câu hỏi đau đầu của việc đầu tư tài nguyên (ví dụ máy chủ) là bao lâu thì nó sẽ hết khấu hao, tôi đầu tư như thế có lãi hay không, có bị outdate về [công nghệ](https://vi.wikipedia.org/wiki/C%C3%B4ng_ngh%E1%BB%87) hay không. Khi sử dụng tài nguyên trên đám mây thì bạn không còn phải quan tâm tới điều này nữa.
* Các nhà cung cấp dịch vụ điện toán đám mây cung cấp các dịch vụ của họ theo ba mô hình cơ bản:
  + [Cơ sở hạ tầng như một dịch vụ](https://vi.wikipedia.org/wiki/C%C6%A1_s%E1%BB%9F_h%E1%BA%A1_t%E1%BA%A7ng_nh%C6%B0_m%E1%BB%99t_d%E1%BB%8Bch_v%E1%BB%A5) ([IaaS](https://vi.wikipedia.org/w/index.php?title=IaaS&action=edit&redlink=1) - Infrastructure as a Service)
  + [Nền tảng như một dịch vụ](https://vi.wikipedia.org/w/index.php?title=N%E1%BB%81n_t%E1%BA%A3ng_nh%C6%B0_m%E1%BB%99t_d%E1%BB%8Bch_v%E1%BB%A5&action=edit&redlink=1) ([PaaS](https://vi.wikipedia.org/w/index.php?title=PaaS&action=edit&redlink=1) - Platform as a Service)
  + [Phần mềm như một dịch vụ](https://vi.wikipedia.org/wiki/Ph%E1%BA%A7n_m%E1%BB%81m_nh%C6%B0_m%E1%BB%99t_d%E1%BB%8Bch_v%E1%BB%A5) ([SaaS](https://vi.wikipedia.org/wiki/SaaS) - Software as a Service)
* Trong đó IaaS là cơ bản nhất và mỗi mô hình cao hơn tóm tắt từ các chi tiết của những mô hình thấp hơn. Trong năm 2012, [mạng lưới như một dịch vụ](https://vi.wikipedia.org/w/index.php?title=M%E1%BA%A1ng_l%C6%B0%E1%BB%9Bi_nh%C6%B0_m%E1%BB%99t_d%E1%BB%8Bch_v%E1%BB%A5&action=edit&redlink=1) ([NaaS](https://vi.wikipedia.org/w/index.php?title=NaaS&action=edit&redlink=1) - Network as a Service) và [giao tiếp như một dịch vụ](https://vi.wikipedia.org/w/index.php?title=Giao_ti%E1%BA%BFp_nh%C6%B0_m%E1%BB%99t_d%E1%BB%8Bch_v%E1%BB%A5&action=edit&redlink=1) ([CaaS -](https://vi.wikipedia.org/w/index.php?title=CaaS&action=edit&redlink=1) Communications as a Service) đã chính thức thêm vào bởi Hiệp hội viễn thông quốc tế (International Telecommunication Union) như là một phần của các mô hình điện toán đám mây cơ bản, các loại hình dịch vụ được công nhận của hệ sinh thái đám mây viễn thông[1].

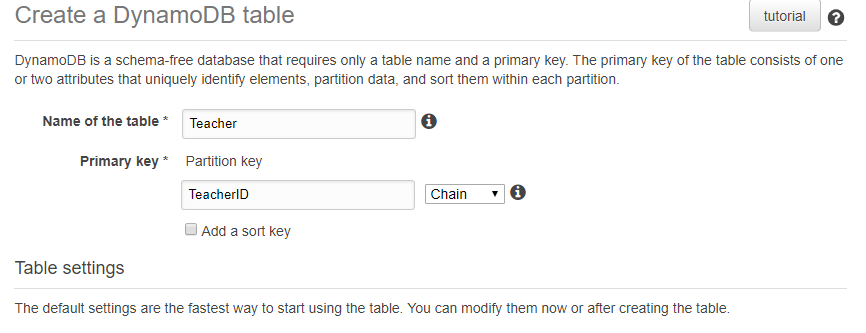
### AWS là gì?

* Amazon Web Services (AWS) là nền tảng đám mây toàn diện và được sử dụng rộng rãi nhất, cung cấp trên 165 dịch vụ đầy đủ tính năng từ các trung tâm dữ liệu trên toàn thế giới. Hàng triệu khách hàng—bao gồm các công ty khởi nghiệp tăng trưởng nhanh nhất, các tập đoàn lớn nhất cũng như các cơ quan hàng đầu của chính phủ—đều tin tưởng vào AWS để phát triển cơ sở hạ tầng, trở nên linh hoạt hơn và giảm chi phí.
* Dưới đây là một số dịch vụ của AWS :



Hình 2‑1 Dịch vụ trên AWS

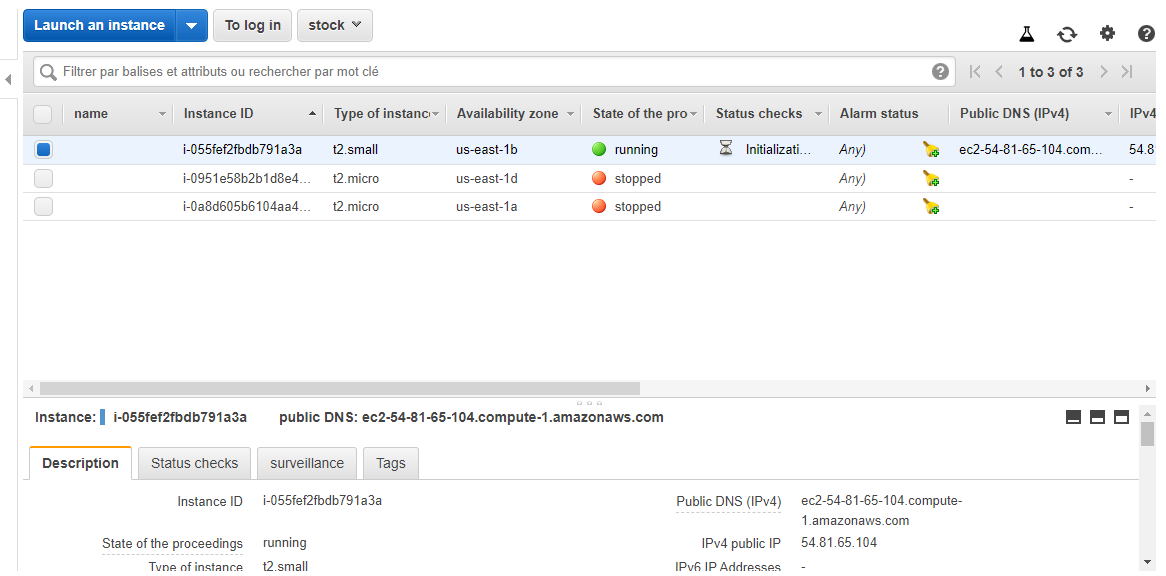
### DynamoDB

* Khi tạo một bảng trong Amazon DynamoBD, ta cần cung cấp tên bảng, primary key và giá trị read write throughput. Mỗi item trong bảng có thể có bao nhiêu attributes tùy ý, tuy nhiên có giới hạn 400KB cho dung lượng một item.
* Khi tạo một bảng thì ngoài việc khai báo tên bảng ta cần phải chỉ rõ Primary key. Primary key để đảm bảo rằng không có hai items nào trong bảng có cùng primary key. DynamoDB hỗ trợ hai loại primary key:
  + Hash Primary Key: Chỉ cần một attribute để tạo ra primary key. DynamoDB xây dựng hash index không được sắp xếp cho primary key attribute.
  + Hash và Range Primary Key: Cần 2 attributes để tạo ra primary key. Attribute đầu tiền là hash attribute và attribute còn lại là range attribute. DynamoDB xây dựng hash index không được sắp xếp cho hash attribute và range index được sắp xếp cho range attribute. Mỗi item được định danh bởi cặp primary key này. Điều này cho phép 2 items có cùng hash attribute nhưng khác range attribute cùng tồn tại trong một bảng.
* Trong quá trình tạo bảng ta phải chỉ rõ mức tiêu thụ (throughput) yêu cầu theo đơn vị là capacity unit. Ta có thể thay đổi chỉ số này qua UpdateTable request:
  + Strongly consistent: đảm bảo dữ liệu lấy ra là dữ liệu mới nhất.
  + Eventually consistent: đảm bảo rằng sau này dữ liệu sẽ đồng bộ hết trên phân vùng nhưng không đảm bảo dữ liệu trả về vào một thời điểm nhất định là mới nhất.
  + Read capacity unit: con số phản ánh số lượng đọc strongly consistent của các item không quá 4KB.
  + Write capacity unit: số lượng những lần ghi có dung lượng 1KB mỗi giây.
* DynamoDB sử dụng những capacity unit này để cung cấp tài nguyên đầy đủ cho lượng tiêu thụ yêu cầu.
* Các lợi ích của DynamoDB:
  + Hiệu năng trên quy mô lớn: DynamoDB hỗ trợ một số ứng dụng có quy mô lớn nhất thế giới bằng cách luôn đáp ứng thời gian phản hồi chỉ trong vài mili giây ở mọi quy mô. Chúng ta có thể xây dựng các ứng dụng gần như là không giới hạn về thông lượng và dung lượng lưu trữ.
  + Không cần phải quản lý máy chủ: DynamoDB là dịch vụ serverless khi ta không cần phải cung cấp, vá hoặc quản lý máy chủ, cũng như không cần phải cài đặt, duy trì hoặc vận hành phần mềm. DynamoDB tự động thay đổi quy mô các bảng để điều chỉnh dung lượng và duy trì hiệu năng. Tích hợp dung sai cao và độ sẵn sàng hoạt động, nhờ đó ứng dụng của chúng ta không cần phải thiết kế các tính năng này. DynamoDB cung cấp cả chế độ dung lượng được cung cấp và chế độ dung lượng theo yêu cầu để ta có thể tối ưu hóa chi phí bằng cách chỉ định công suất theo khối lượng công việc hoặc chỉ chi trả cho các tài nguyên mà chúng ta sử dụng.
  + Phù hợp sử dụng cho các doanh nghiệp: DynamoDB hỗ trợ các giao tác ACID để ta có thể xây dựng các ứng dụng quan trọng với doanh nghiệp trên quy mô lớn. DynamoDB mã hóa tất cả dữ liệu theo mặc định và cung cấp danh tính chi tiết cũng như quyền kiểm soát truy cập đối với mọi bảng của chúng ta[2].
  + 

Hình 2‑1-1 AWS DynamoDB

### EC2

* Amazon Elastic Compute Cloud (Amazon EC2) là dịch vụ web cung cấp năng lực điện toán bảo mật và có kích cỡ linh hoạt trên đám mây. Dịch vụ này được thiết kế để giúp các nhà phát triển dễ sử dụng điện toán đám mây ở quy mô web hơn.
* Giao diện dịch vụ web đơn giản của Amazon EC2 cho phép bạn có được và cấu hình năng lực với sự va chạm tối thiểu. Dịch vụ này cung cấp cho bạn khả năng kiểm soát toàn phần các tài nguyên điện toán của bạn và giúp bạn chạy trên môi trường điện toán đã được kiểm chứng của Amazon. Amazon EC2 giảm thời gian cần thiết để lấy và khởi động các phiên bản máy chủ mới xuống còn vài phút, cho phép bạn nhanh chóng tăng hoặc giảm quy mô năng lực theo sự thay đổi trong yêu cầu điện toán. Amazon EC2 thay đổi bài toán kinh tế của điện toán bằng cách cho phép bạn chỉ chi trả cho phần năng lực bạn sử dụng thực tế. Amazon EC2 cung cấp cho các nhà phát triển nhiều công cụ để dựng các ứng dụng có khả năng chống chịu lỗi và không để chúng rơi vào các tình huống lỗi phổ biến[3].
* Amazon EC2 mang lại các lợi ích:
  + Cho phép sử dụng hàng trăm thậm chí hàng nghìn máy chủ đồng thời, bạn cũng có thể dùng Auto Scaling để tăng hoặc giảm quy mô giúp tiết kiệm chi phí và phù hợp với nhu cầu sử dụng.
  + Bạn sở hữu khả năng kiểm soát toàn phần các phiên bản của mình, kể cả quyền truy cập root và khả năng tương tác với chúng như khi thực hiện với bất kỳ chiếc máy nào. Bạn có thể vừa dừng bất kỳ phiên bản nào vừa giữ lại dữ liệu trên phân hoạch khởi động, rồi sau đó khởi động lại cùng phiên bản đó bằng cách sử dụng các API dịch vụ web. Có thể khởi động lại từ xa các phiên bản bằng cách sử dụng các API dịch vụ web, đồng thời, bạn cũng có quyền truy cập đầu ra bảng điều khiển của các phiên bản này.
  + Amazon EC2 cho phép bạn lựa chọn cấu hình bộ nhớ, CPU, bộ nhớ phiên bản và kích cỡ phân hoạch khởi động một cách tối ưu cho lựa chọn hệ điều hành và ứng dụng của bạn.
  + Amazon EC2 được tích hợp với hầu hết các dịch vụ AWS.
  + Amazon EC2 cung cấp môi trường rất tin cậy mà trong đó, các phiên bản thay thế có thể được đưa vào sử dụng một cách nhanh chóng và có thể dự báo trước.
  + Bảo mật đám mây là ưu tiên cao nhất của AWS. Amazon EC2 phối hợp với Amazon VPC to để cung cấp chức năng mạng bảo mật và ổn định cho các tài nguyên điện toán của bạn.
  + Nhờ quy mô của AWS chi phí phải trả rất thấp cho năng lực điện toán sử dụng thực tế từ đó sẽ có được lợi ích kinh tế.
  + Amazon EC2 dễ dàng làm quen bằng nhiều cách khác nhau.

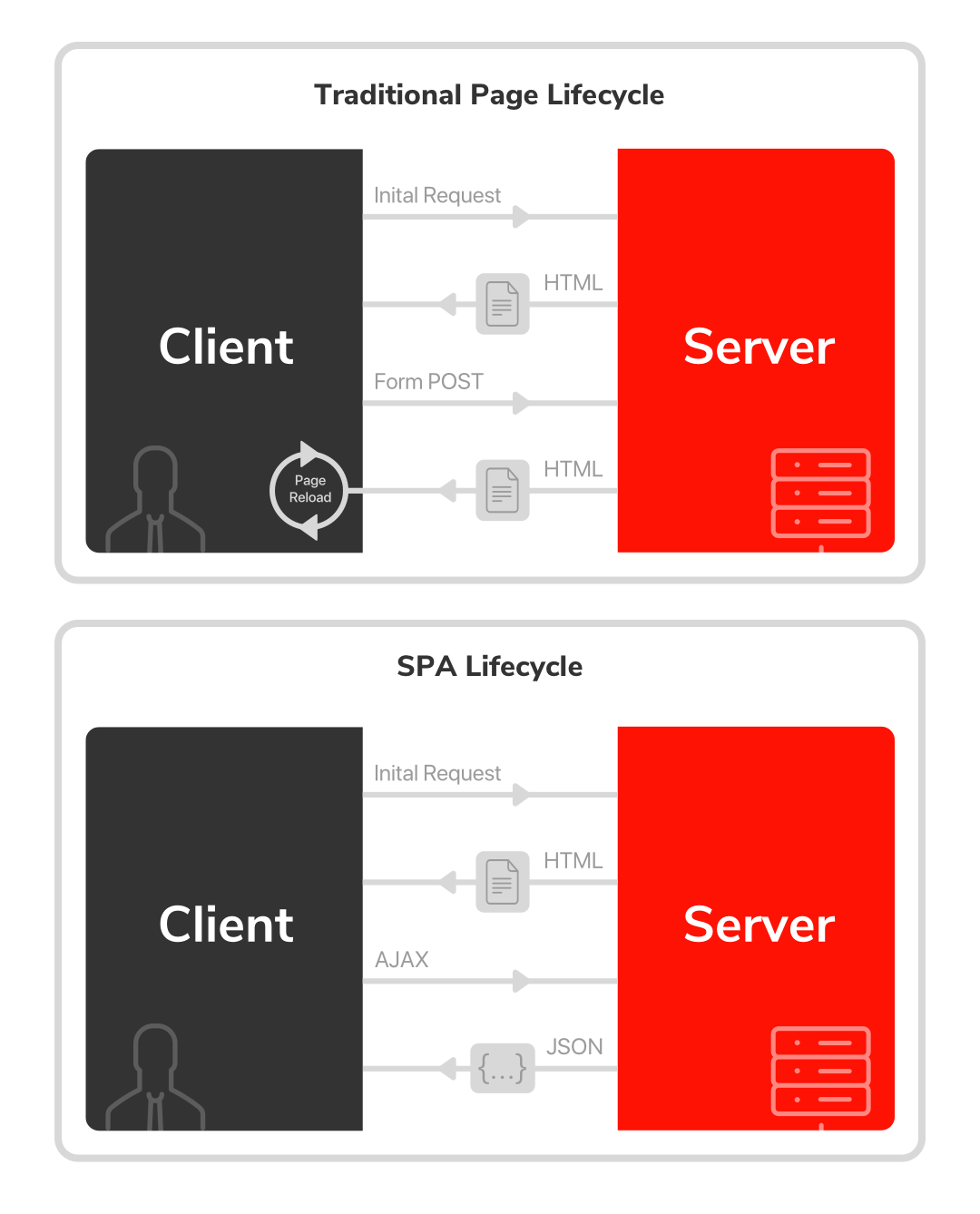


Hình 2‑1-2 AWS EC2 Intance

### S3

* Amazon S3 là dịch vụ lưu trữ đối tượng được xây dựng để lưu trữ và truy xuất dữ liệu với khối lượng bất kỳ từ bất cứ nơi nào trên Internet. Đây là dịch vụ lưu trữ đơn giản cung cấp hạ tầng lưu trữ dữ liệu có độ bền cực cao, độ khả dụng cao và quy mô vô cùng linh hoạt với chi phí rất thấp.
* Amazon S3 cung cấp một giao diện web đơn giản, cho phép bạn sử dụng để lưu trữ và truy xuất dữ liệu với khối lượng bất kỳ, vào bất cứ thời điểm nào, từ bất cứ nơi nào trên web. Sử dụng dịch vụ web này, bạn có thể dễ dàng xây dựng những ứng dụng tận dụng được khả năng lưu trữ trên Internet. Do Amazon S3 có khả năng mở rộng cao và bạn chỉ phải trả tiền cho những gì bạn dùng, bạn có thể bắt đầu từ quy mô nhỏ và phát triển các ứng dụng bạn muốn mà không phải hy sinh về hiệu năng hoặc độ tin cậy.
* Amazon S3 cung cấp một giao diện dịch vụ web REST dựa trên tiêu chuẩn, đơn giản, nó được thiết kế để làm việc với bất kỳ bộ công cụ phát triển Internet nào. Các hoạt động được đơn giản hóa có chủ đích để thuận tiện cho việc thêm các giao thức phân phối và các lớp chức năng mới.
* Amazon S3 cũng được thiết kế có độ linh hoạt cao. Lưu bất kỳ loại và khối lượng dữ liệu nào bạn muốn; đọc cùng một đoạn dữ liệu cả triệu lần hoặc chỉ để khắc phục sự cố khẩn cấp; xây dựng một ứng dụng FTP đơn giản hoặc một ứng dụng web phức tạp chẳng hạn như trang web bán lẻ Amazon.com. Amazon S3 cho phép các nhà phát triển tự do tập trung vào đổi mới thay vì phải lo tìm cách lưu trữ dữ liệu của mình.
* Cung cấp nhiều lớp lưu trữ khác nhau được thiết kế cho các tình huống sử dụng khác nhau. Trong đó có S3 Tiêu chuẩn dành cho dữ liệu truy cập thường xuyên thông dụng; S3 Phân bậc thông minh dành cho dữ liệu có cấu trúc truy cập không xác định hoặc hay thay đổi; S3 Tiêu chuẩn - Truy cập không thường xuyên và S3 Một vùng - Truy cập không thường xuyên dành cho dữ liệu lâu dài nhưng truy cập không thường xuyên; và Amazon S3 Glacier (S3 Glacier) và Amazon S3 Glacier Deep Archive (S3 Glacier Deep Archive) cho lưu trữ dài hạn và bảo tồn kỹ thuật số.  Bạn có thể tìm hiểu thêm về các lớp lưu trữ này trên [trang Lớp lưu trữ của Amazon S3](https://aws.amazon.com/vi/s3/storage-classes/).
* Bạn chỉ định một Khu vực AWS khi tạo bộ chứa Amazon S3. Với các lớp lưu trữ S3 Tiêu chuẩn, S3 Tiêu chuẩn – Truy cập không thường xuyên và S3 Glacier, các đối tượng của bạn được lưu trữ tự động trên nhiều thiết bị trải rộng trên tối thiểu ba Vùng sẵn sàng, mỗi vùng này cách nhau nhiều dặm trên Khu vực AWS. Các đối tượng được lưu trữ trong lớp lưu trữ S3 Một vùng – Truy cập không thường xuyên được lưu trữ dư thừa trong một Vùng sẵn sàng đơn trong Khu vực AWS bạn chọn. Vui lòng tham khảo [Sản phẩm và dịch vụ theo khu vực](https://aws.amazon.com/vi/about-aws/global-infrastructure/regional-product-services/) để biết chi tiết về tính khả dụng của dịch vụ Amazon S3 theo Khu vực AWS[4].

## Single Page Application

****

Hình 2‑1-3 Single page Application

* Single page Application là một ứng dụng web giúp nâng cao trải nghiệm người dùng bằng cách sử dụng HTML5 và AJAX. Đầu tiên khi tải một trang web bất kỳ, SPA sẽ tải một trang HTML đơn, sau đó dựa trên request của người dùng, SPA sẽ tiếp tục tải các HTML khác trong cùng một trang đó, SPA có thể sử dụng một vài thư viện JavaScript như React, Angular,…
* Hiểu một cách đơn giản, thì toàn bộ resource của web bao gồm các file CSS, Javascript, master layout hay cấu trúc web page sẽ được load lần đầu tiên khi chúng ta bắt đầu duyệt môt website A nào đó. Ở những lần sau, khi chuyển trang khác, client sẽ gửi những ajax request để get dữ liệu cần thiết (thường là phần nội dung). Việc này mang đến trải nghiệm cho người dùng web tốt hơn, giảm thời gian phải load lại toàn bộ trang web cồng kềnh, tiết kiệm băng thông cũng như thời gian chờ đợi. Việc này là trái ngược hoàn toàn với trang web truyền thống khi toàn bộ trang web phải load lại mỗi khi chuyển trang.

## Server side

### 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 2‑1-4 Node.js development

### Express

## Express là một web application framework cho nodejs, nó cung cấp cho chúng những rất nhiều tính năng mạnh mẽ trên nền tảng web. Express rất dễ dàng để phát triển các ứng dụng nhanh dựa trên Node.js cho các ứng dụng Web. Express hỗ trợ các phương thức HTTP và middleware tạo ra 1 API rất mạnh mẽ và sử dụng dễ dàng hơn. Khi mới tiếp cận với Express mình thực sự bị cuốn hút bởi các API của nó, từ cách sử dụng route, template, đều khá dễ tùy chọn và làm việc. Các tính năng của Express framework phải kể đến như:

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

#### Request & Response trong Expres

* Express sử dụng một hàm callback có các tham số là các đối tượng request và response.

****

* Request - Biểu diễn một HTTP request và có các thuộc tính cho các request như các chuỗi truy vấn, tham số, body, HTTP header và những phần khác.
* Response - Biểu diễn một HTTP response được ứng dụng Express gửi đi khi nó nhận về một HTTP request.

#### Route cở bản trong Express

* Trong Express, router được tích hợp sẵn và dễ dàng sử dụng, dưới đây là ví dụ cơ bản về route của Express:

****

#### File tĩnh trong Express

* Express cung cấp tiện ích **express.static** để phục vụ cho các file tĩnh như hình ảnh, css, js, ...Về cơ bản, bạn chỉ cần truyền tên thư mục nơi bạn giữ các file này, express.static sẽ sử dụng file đó một cách trực tiếp. Giả sử ứng dụng của bạn có cấu trúc như sau:



* Sử dụng Express static:

****

## DynamoDB

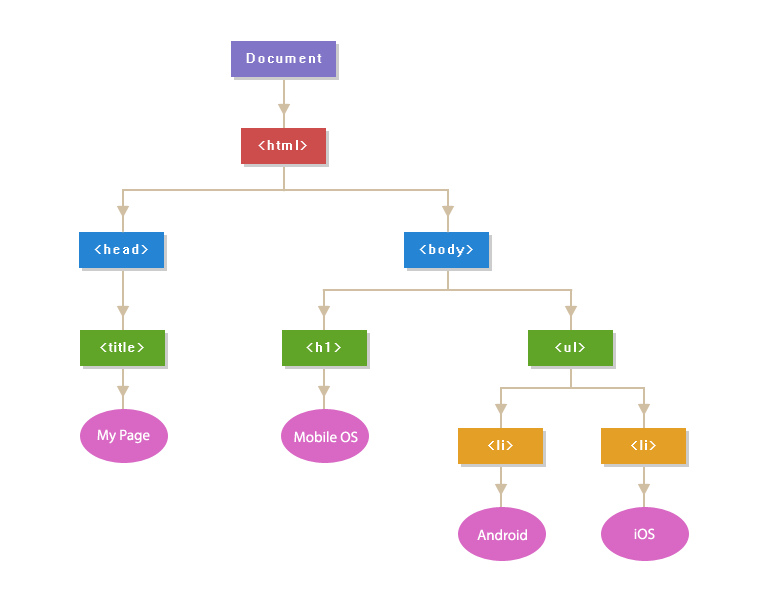
* **Amazon DynamoDB** là một dịch vụ [cơ sở dữ liệu](https://vi.wikipedia.org/wiki/C%C6%A1_s%E1%BB%9F_d%E1%BB%AF_li%E1%BB%87u) độc quyền quản lý hoàn toàn sử dụng [NoSQL](https://vi.wikipedia.org/wiki/NoSQL) có hỗ trợ giá trị khóa và cấu trúc dữ liệu tài liệu và được cung cấp bởi [Amazon.com](https://vi.wikipedia.org/wiki/Amazon.com) 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).

## Client Sid

### React

* ReactJS là một thư viện của Javascript gần đây đang nổi lên với xu hướng Single Page Application. React nổi lên với sự đơn giản và dễ dàng phối hợp với những thư viện Javascript khác. Một trong những điểm hấp dẫn của React là thư viện này không chỉ hoạt động trên phía client, mà còn được render trên server và có thể kết nối với nhau. 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… 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.

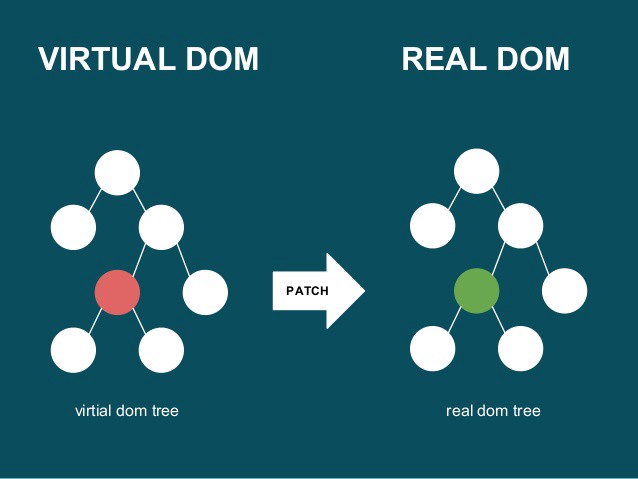
#### DOM ảo (Virtual DOM) là gì ?



Hình 2‑1-5 DOM

* DOM(Document Object Model) là một thư viện giao diện người dùng giống như các phần tử, thuộc tính.
* DOM sẽ được tạo ra khi trang web vừa tải xong DOM và tồn tại dưới dạng tree nodes được dùng để quản lí, truy xuất, chỉnh sửa tới bất kì phần tử nào thông qua đối tượng gốc là document [5]. Ví dụ một phần tử thẻ HTML <p></p>có id là “house” ta sẽ truy xuất đến phần tử đó như sau:





Hình 2‑1-6 Virtual DOM

* Bất cứ khi nào phương thức setState () được gọi, ReactJS reset DOM ảo từ đầu. Việc tạo lại rất nhanh nên nó không ảnh hưởng đến hiệu suất. Tại bất kỳ thời điểm nào, ReactJS duy trì hai DOM ảo, một với DOM được cập nhật trạng thái ảo và một với DOM DOM trạng thái trước đó gần đây nhấ để so sánh và cập nhật phần tử thay đổi.
* Vì sao lại dùng DOM ảo mà không dùng DOM thực? Vì khi ở DOM thực mỗi khi có sự thay đổi, vì cấu trúc của DOM là tree nodes, khi muốn thay đổi các element và các thẻ con của nó, nó phải thông qua các bước sau theo như ví dụ ở trên:

1. Trình duyệt phân tích ra cú pháp HTML
2. Loại bỏ phần tử con của elementId
3. Cập nhật giá trị mới cho phần tử DOM
4. Cập nhật lại CSS cho các phần tử liên quan
5. Cập nhật lại layout và hiển thị lại ra màn hình

Nên rất mất thời gian nếu việc cập nhật một phần nhỏ trong DOM khiến cho tất cả các phần tử DOM khác phải cập nhật lại mặc dù không thay đổi.

* Câu lệnh ReactDOM.render() bạn thấy trong file index.js khi khởi tạo project ReactJS có ý nghĩa cập nhận những phần tử DOM thay đổi.
* ReactDOM là một thư viện Javascript để hỗ trợ cho việc cập nhật đó, được triển khai như sau:

* 

#### JSX là gì?

* JSX là một phần quan trọng trong ReactJS. JSX giúp code trở lên dễ đọc hơn. JSX là sự kết hợp giữa Javascript và XML. Nó chuyển đổi cú pháp dạng gần như XML về thành Javascript. Giúp người lập trình có thể code ReactJS bằng cú pháp của XML thay vì sử dụng Javascript. Các XML elements, attributes và children được chuyển đổi thành các đối số truyền vào React.createElement[8].
* Phần tử JSX có thể được lưu dưới dạng biến, được chuyển đến hàm hoặc lưu trữ trong một mảng hay là đối tượng.
* Mỗi một biểu thức JSX cần được bao bọc bởi một phần tử ngoài cùng, thông thường chúng ta thường dùng thẻ <div></div> để bao bọc:



* Chúng ta có thể thêm mã Javascript bằng cách bao bọc nó trong cặp dấu ngoặc nhọn “{}”:



* Việc sử dụng JSX trong ReactJS là không bắt buộc. Bạn có thể sử dụng chỉ JS thuần. Nhưng tôi không khuyến khích vì thứ nhất: JSX cú pháp giống XML theo cấu trúc cây, điều đó giúp ta dễ dàng định nghĩa, quản lý được các component phức tạp, JSX không làm thay đổi ngữ nghĩa của Javascript
* Trong JSX có một số cú pháp đặc trưng cần lưu ý như hàm map(), arrow function, .v..v..

#### Component là gì *?*

* Là một đoạn mã ngắn có ý nghĩa thể hiện một chức năng hay đối tượng nhất định, có thể tái sử dụng
* Chúng ta có thể sử dụng nhiều mã JSX trong một component
* Một ứng dụng React có thể có hàng chục, hoặc hàng trăm các component liên kết, tương tác với nhau
* Để hiểu đơn giản hơn về component tôi sẽ đề cập cụ thể: Ví dụ bạn xây một ứng dụng “Nhà” bạn có thể chia ra thành các component nhỏ hơn như “Tầng 1”, “Tầng 2”, sau đó có thể chia nhỏ tiếp thành “Phòng Ngủ”, “Phòng Ăn”….. Cứ như vậy ứng dụng của chúng ta sẽ được chia ra một cách cụ thể rõ ràng, mạch lạc thuận lợi cho việc phát triển. Ví dụ:

****

Sau đó ta sẽ import file trên vào Component mẹ:

****

#### Props là gì *?*

* Thông tin được truyền từ component này sang component khác được gọi là props
* Props của một component là một object. Nó chứa thông tin về thành phần đó. Để xem props của component đó chứa gì ta sẽ sử dụng đoạn câu lệnh this.props
* Component ở phía trên có component con là tầng 1. Vậy bây giờ chúng ta cần biết số phòng của component đó (tạm gọi là x). Mà biến “x” lại đang ở trên component mẹ, Bài toán được đặt ra và chúng ra giải quyết như sau:
* Đây là component tang\_1 đang trạng chờ nhận được props từ component mẹ:

****

* Triển khai truyền props từ component mẹ:

****

* Ngoài ra ta có thể truyền props là một hàm hoặc phương thức nhứ sau:

****

Sau đó import component Button:

****

* Trong props của mỗi component có một thuộc tính là children, this.props.children sẽ trả về mọi thứ nằm giữa các thẻ JSX mở và đóng của component.

#### State là gì*?*

* Một coponent có thể truy cập thông tin theo hai cách là : props và state.
* Không giống như props, state của component không thể truyền được ra ngoài. Component sẽ quản lí state của chính nó.
* Để tạo ra một Component có state, chúng ta sẽ thêm một thuộc tính state, thuộc tính phải được khai báo bên trong phương thức khởi tạo như sau:

****

* this.state phải là một đối tượng.
* Để truy cập đến state ta sử dụng cú pháp this.state.name-of-property
* Cách phổ biến để thay đổi state trong một component ta sử sụng hàm this.setState() như sau:

****

* Một component con có thể thay đổi được state của component mẹ bằng cách từ mẹ truyền xuống cho con một hàm callback để nhận dữ liệu cập nhật từ component con.

#### React xử lý style như thế nào?

* Tương tự như HTML , biến CSS ở react tuy khác biệt nhưng không quá lớn. Thay vì được viết bằng chữ thường và được nối bởi dấu “-” thì bây giờ đã thay thế được viết bằng kiểu camelCase. Đây là một số cách triển khai CSS trong React:

****

****

#### Lifecycle Methods là gì *?*

* Một thành phần đặc trưng không kém Props, State của của React là Lifecycle
* Lifecycle là những phương thức được viết sẵn truy cập đến từng thời điểm trong một vòng đời của component, từ đó chúng ta có thể triển khai ý tưởng một cách dễ dàng linh hoạt, thuận lời cho quá trình fix bug sau này
* Chi tiết về vòng đời như sau:
* componentWillMount: đây là method sẽ được thực thi trước khi 1 component được render trên cả server side và client side.
* render: Hàm hiển thị
* componentDidMount: method này được thực thi khi 1 component được render trên client side. Đây là nơi các hàm AJAX requests, DOM or update state được thực thi. Method này cũng đucợ sử dụng để kết nối với các JS Framework khác. Đây là nơi thích hợp để gọi đến API.
* Có 5 phương thức được định nghĩa để thực hiện khi một phần tử được cập nhật:
  + componentWillReceiveProps: sẽ được thực thi ngay khi thuộc tính props được update và trước khi component được render lại.
  + shouldComponentUpdate sẽ trả về kết quả true or false. Phương thức này sẽ xác định 1 component có được update hay không. Mặc định giá trị này là true. Nếu bạn không muốn component render lại sau khi update state hay props thì return giá trị thành false.
  + componentWillUpdate được gọi khi chúng ta update state của component trước khi nó render lại.
  + render: render component.
  + componentDidUpdate sau khi componentWillUpdate ở trên được gọi xong thì đến lượt thằng này được gọi.

Cuối cùng là:

* **componentWillUnmount**: được gọi trước khi một component được xóa khỏi DOM

### React- Router

* React-Router là một thư viện định tuyến (routing) tiêu chuẩn trong React. Nó giữ cho giao diện của ứng dụng đồng bộ với URL trên trình duyệt. React-Router cho phép bạn định tuyến "luồng dữ liệu" (data flow) trong ứng dụng của bạn một cách rõ ràng. Nó tương đương với sự khẳng định, nếu bạn có URL này, nó sẽ tương đương với Route này, và giao diện tương ứng [9].

#### ****BrowserRouter vs HashRouter****

* React-Router cung cấp cho chúng 2 thành phần hay sử dụng đó là BrowserRouter & HashRouter. Hai thành phần này khác nhau ở kiểu URL mà chúng sẽ tạo ra và đồng bộ.
* BrowserRouter: Được sử dụng phổ biến hơn, nó sử dụng History API có trong HTML5 để theo dõi lịch sử bộ định tuyến của bạn.
* HashRouter: Sử dụng hash của URL (window.location.hash) để ghi nhớ mọi thứ.



#### Route

* Route: Định nghĩa một ánh xạ (mapping) giữa một URL và một Component. Điều đó có nghĩa là khi người dùng truy cập theo một URL trên trình duyệt, một Component tương ứng sẽ được render trên giao diện.



Trong đó:

* path: Là đường dẫn trên URL.
* exact: Liúp cho route này này chỉ hoạt động nếu URL trên trình duyệt phù hợp tuyệt đối với giá trị của thuộc tính path của nó.
* component: Là component sẽ đươc load ra tương ứng với Route đó.

#### Link

Trong HTML thì cặp thẻ để chuyển hướng đó là thẻ <a></a> thì trong react chúng ta sẽ dử dụng cặp thẻ <Link></Link> được import từ React-Router.



Trong đó:

* to: Giống như thuộc tính href trong thẻ a.

#### NavLink

* NavLink thì rất giống với Link về cách sử dụng, nhưng **NavLink** tốt hơn vì nó hỗ trợ thêm một số thuộc tính như là **activeClassName** và **activeStyle** 2 thuộc tính này giúp cho khi mà nó trùng khớp thì nó sẽ được active lên và chúng ta có thể style cho nó.



#### Custom Link

* Ở trên ta có thẻ NavLink giúp chúng ta có thêm một thuộc tính nhưng giả sử khi bạn không muốn activeClassName hoặc activeStyle tại thẻ NavLink mà nó lại nằm ở một thẻ bao nó ví dụ như thẻ div hay thẻ li thì sao? sau đây mình sẽ custom lại để có thể sử dụng các class hoặc style ở thẻ bao ngoài của nó.



#### Đối tượng Match

* Khi bạn muốn lấy một số thông tin ở trên URL thì bạn có thể dùng đối tượng match để lấy dữ liệu về. Tại cấu hình Router ta chỉ cần truyền thêm đối tượng match vào component mà cần sử dụng đối tượng match:



* Trong đối tượng params sẽ chứa các tham số mà ta truyền trên URL

#### ****Đối tượng prompt - Xác nhận trước khi chuyển trang****

* Giả sử khi bạn đang nhập liệu ở form nào đó mà không may click nút back hay chuyển trang thì thôi xong dữ liệu bạn nhập sẽ mất hết để khác phục điều đó ta có đối tượng prompt nó sẽ giúp chúng ta trước khi back hay chuyển trang nó sẽ xác nhận xem là chúng ta có chắc chắn muốn back hay chuyển trang không!
* Khi muốn sử dụng đối tượng prompt thì chúng ta chỉ cần import nó từ react-router



#### Redirect

* Chức năng dùng để chuyển trang.
* Có thể truy xuất thông tin trang trước đó thông qua đối tượng location. Để sử dụng Redirect ta chỉ cần import nó từ react-router.



* Khi bạn muốn sử dụng location thì tại cấu hình Router ta chỉ cần truyền thêm đối tượng location vào component mà cần sử dụng đối tượng location.

****

### Ant Design

* Ant là tập hợp các components của React được xây dựng theo chuẩn thiết kế của Ant UED Team. Tương tự như chuẩn Material Design, Ant cung cấp hầu hết các component thông dụng trong ứng dụng web hiện đại, như Layout, Button, Icon, DatePicket, v.v…Bên cạnh đó Ant cũng có những component riêng thú vị, như LocaleProvider cho phép bạn thay đổi ngôn ngữ trên toàn ứng dụng [6]. Có thể coi Ant Design cho React là tập hợp của hầu hết các thư viện về React. Nó đáp ứng được hầu hết các yêu cầu của project của bạn mà ban không phải cài thêm bất cứ thư viện nào nữa. Dưới đây là danh sách các component mà nó cung cấp:
* General: Button, Icon
* Layout: Grid, Layout
* Navigation: Affix, Breadcrumb, Dropdown, Menu, Pagination, Steps
* Data Entry: AutoComplete, Checkbox, Cascader, DatePicker, Form, InputNumber, Input, Mention, Rate, Radio, Switch, Slider, Select, TreeSelect, Transfer, TimePicker, Upload
* Data Display: Avatar, Badge, Collapse, Carousel, Card, Calendar, List, Popover, Tree, Tooltip, Timeline, Tag, Tabs, Table
* Feedback: Alert, Drawer, Modal, Message, Notification, Progress, Popconfirm, Spin, Skeleton
* Other: Anchor, BackTop, Divider, LocaleProvider

### Axios

* Axios là một thư viện HTTP Client dựa trên Promise. Cơ bản thì nó cung cấp một API cho việc xử lý XHR (XMLHttpRequests).

#### Thực hiện request với phương thức Get



#### Thực hiện request với phương thức Post

****

#### Thực hiện nhiều truy cập đồng thời

****

#### Các alias cho việc tạo ra các request với phương thức tương ứng

* axios.request(config)
* axios.get(url[, config])
* axios.delete(url[, config])
* axios.head(url[, config])
* axios.options(url[, config])
* axios.post(url[, data[, config]])
* axios.put(url[, data[, config]])
* axios.patch(url[, data[, config]])
* Khi sử dụng các alias này thì url, method, và data không phải cần phải chỉ định trong config nữa.

#### Tạo ra một đối tượng Axios

* Có thể tạo ra một đối tượng axios với các config của mình bằng hàm axios.create([config]) Ví dụ:



* Khi đó, đối tượng này có thể sử dụng các phương thức ở mục 4.4.4 để thực hiện việc gửi request.

#### Cấu trúc dữ liệu trả về

* Dữ liệu trả về cho một request sẽ mang các thông tin sau:

****

* response sẽ là tham số của hàm được truyền vào then.

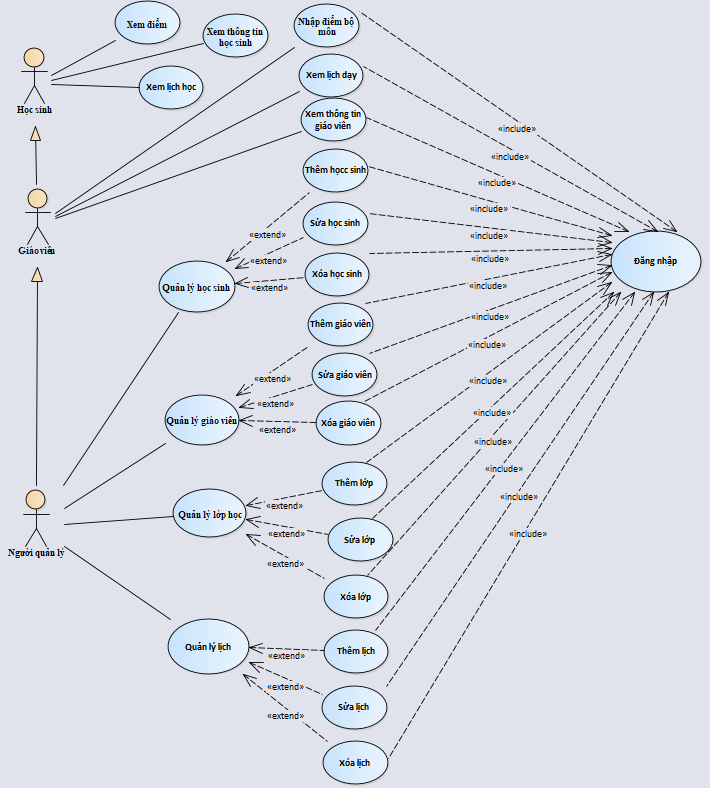
## Restful API

* **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[7]. 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.

# : 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

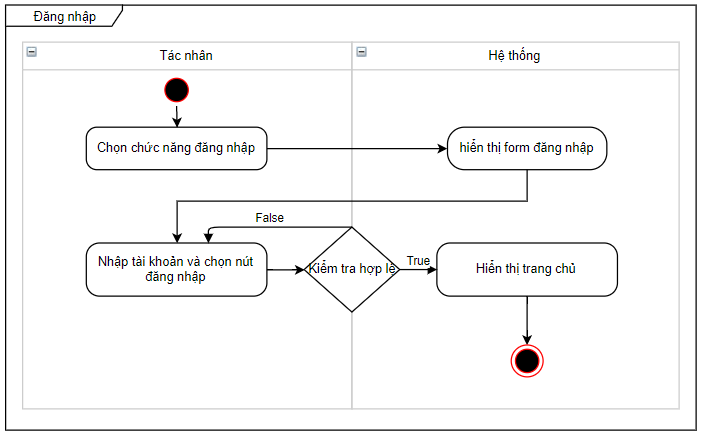
### Danh sách các tình huống hoạt động (Use cases)

|  |  |
| --- | --- |
| 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 | |

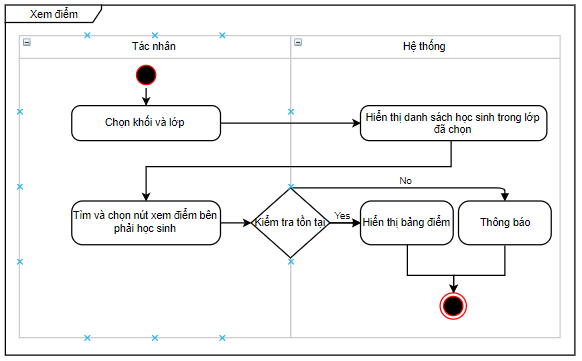
Bảng 3-3-1: Usecase Đăng nhập



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. | |

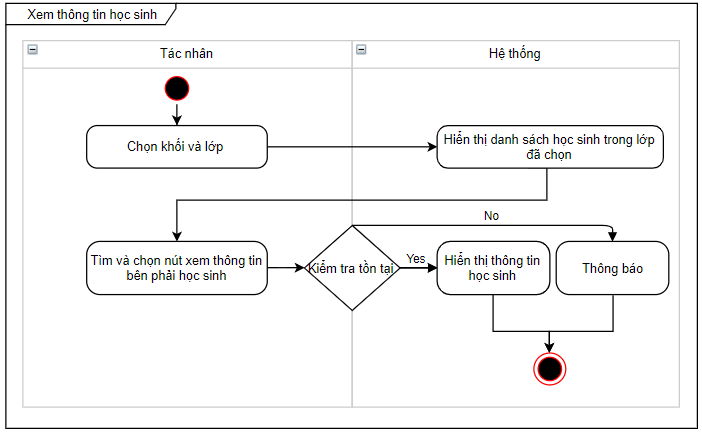
Bảng 3-3-2: Usecase Xem điểm



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. | |

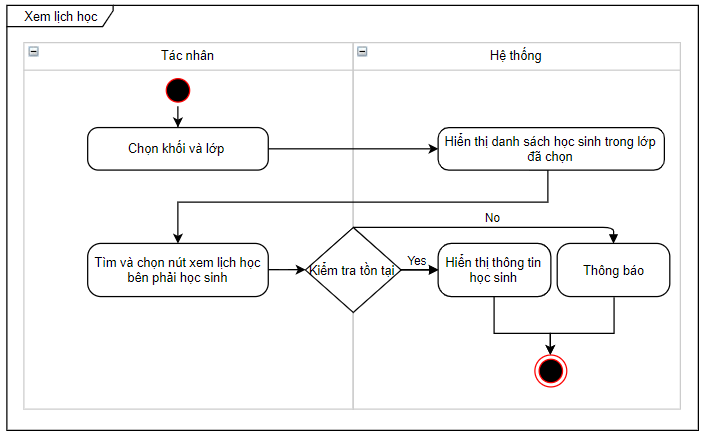
Bảng 3-3-3: Usecase Xem thông tin cá nhân



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. | |

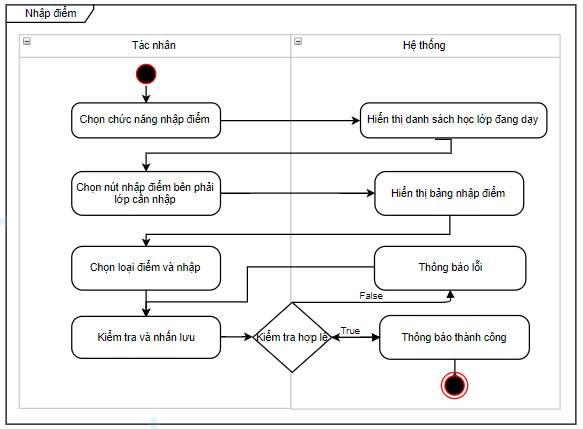
Bảng 3-3-4: Usecase Xem lịch học



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. | |

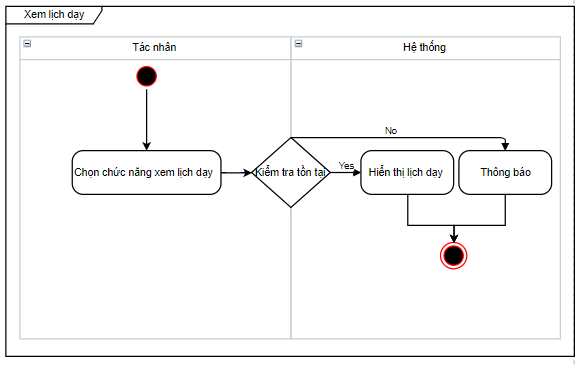
Bảng 3-3-5: Usecase Nhập điểm bộ môn



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. | |

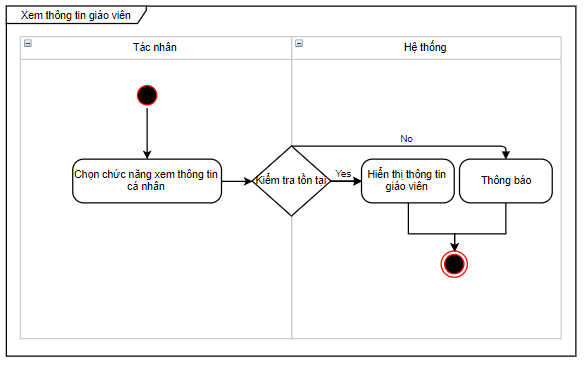
Bảng 3-3-6: Usecase Xem lịch dạy



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. | |

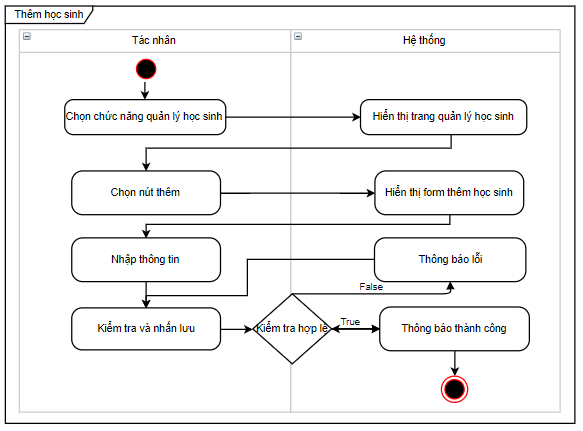
Bảng 3-3-7: Usecase Xem thông tin giáo viên



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 | |

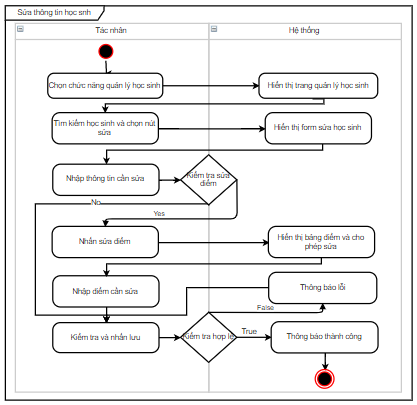
Bảng 3-3-8: Usecase Thêm học sinh



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. | |

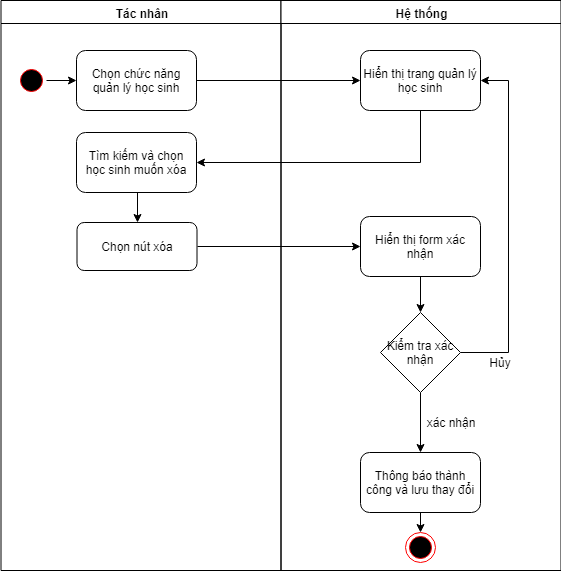
Bảng 3-3-9: Usecase Sửa thông tin học sinh



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. | |

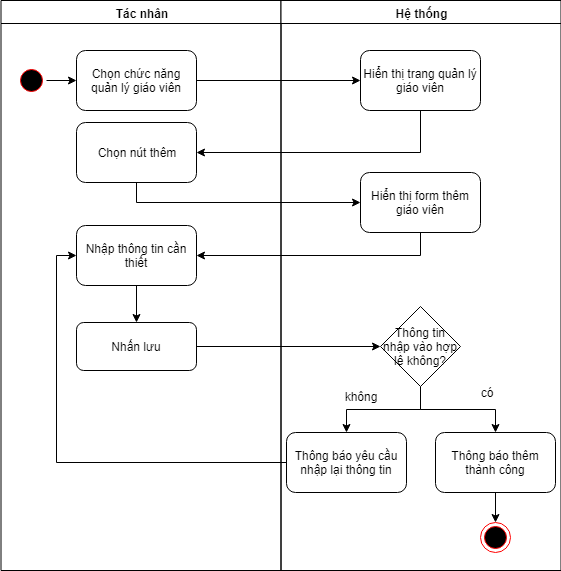
Bảng 3-3-10:Usecase Xóa học sinh



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. | |

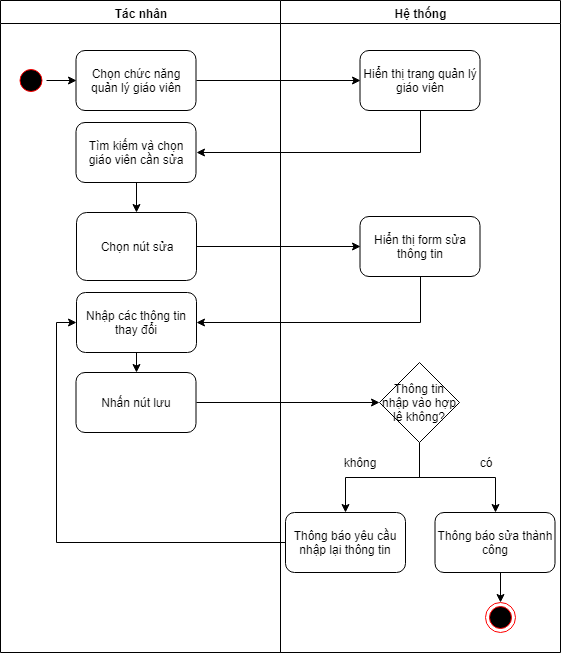
Bảng 3-3-11: Usecase Thêm giáo viên



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. | |

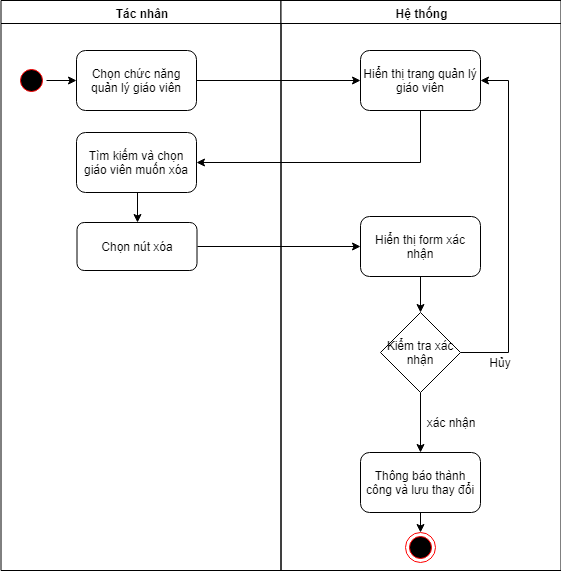
Bảng 3-3-12: Usecase Sửa giáo viên

\

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. | |

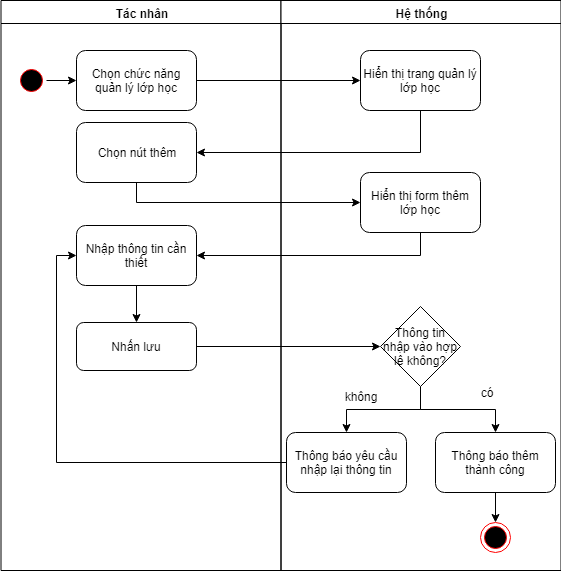
Bảng 3-3-13: Usecase Xóa giáo viên



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. | |

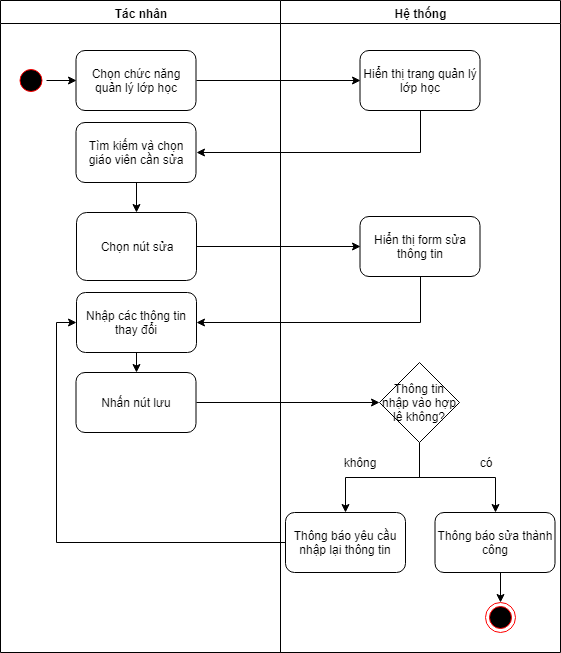
Bảng 3-3-14: Usecase Thêm lớp



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. | |

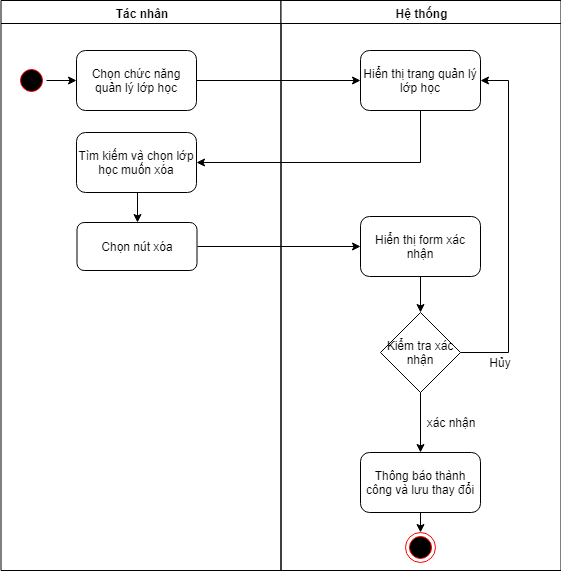
Bảng 3-3-15: Usecase Sửa lớp



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. | |

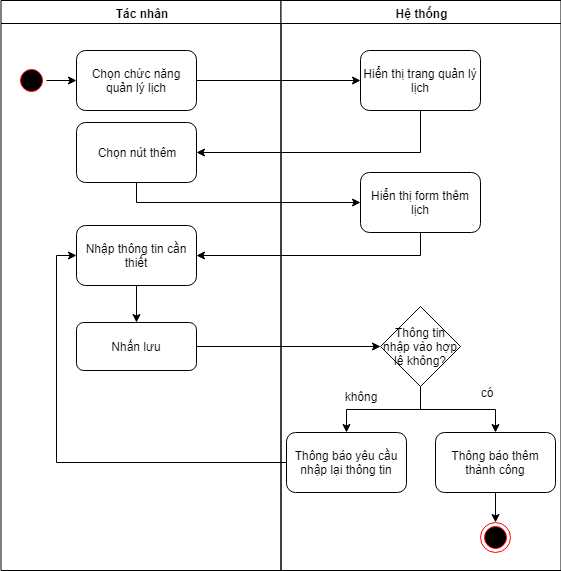
Bảng 3-3-16: Usecase Xóa lớp



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. | |

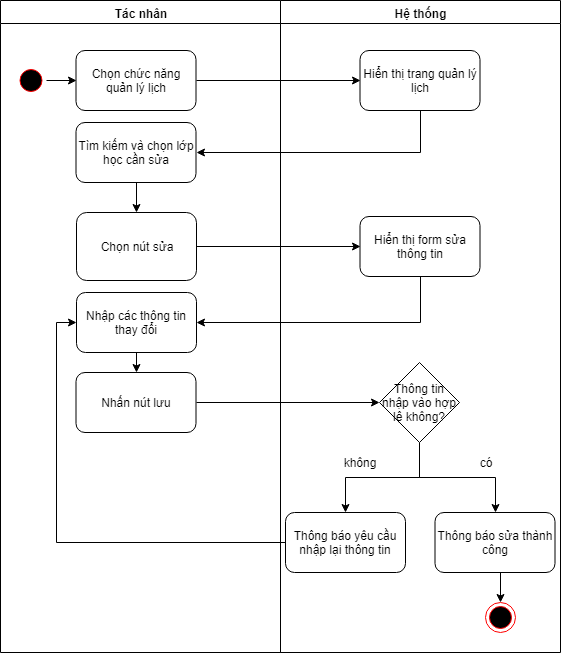
Bảng 3-3-17: Usecase Thêm lịch



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. | |

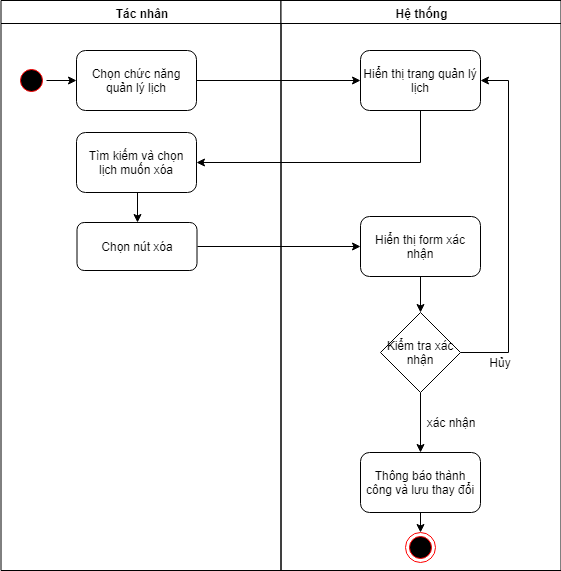
Bảng 3-3-18: Usecase Sửa lịch



Hình 3‑1-18 : Activity Sửa lịch

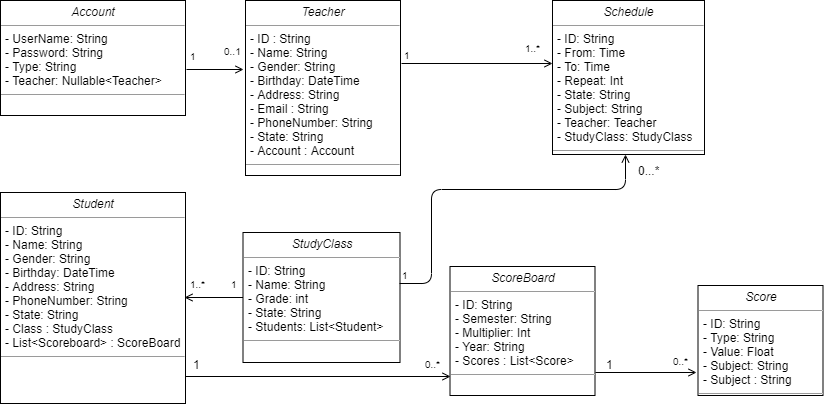
|  |  |  |
| --- | --- | --- |
| UC19 : Xóa lịch | | |
| Mục đích: | Xóa lịch khỏi hệ thống | |
| Mô tả: | Người quản lý thực hiện việc xóa lịch khỏi hệ thống | |
| Tác nhân: | Người quản lý | |
| Điều kiện trước: | Đăng nhập vào hệ thống với tài khoản của Người quản lý | |
| Điều kiện sau: | Xóa thành 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



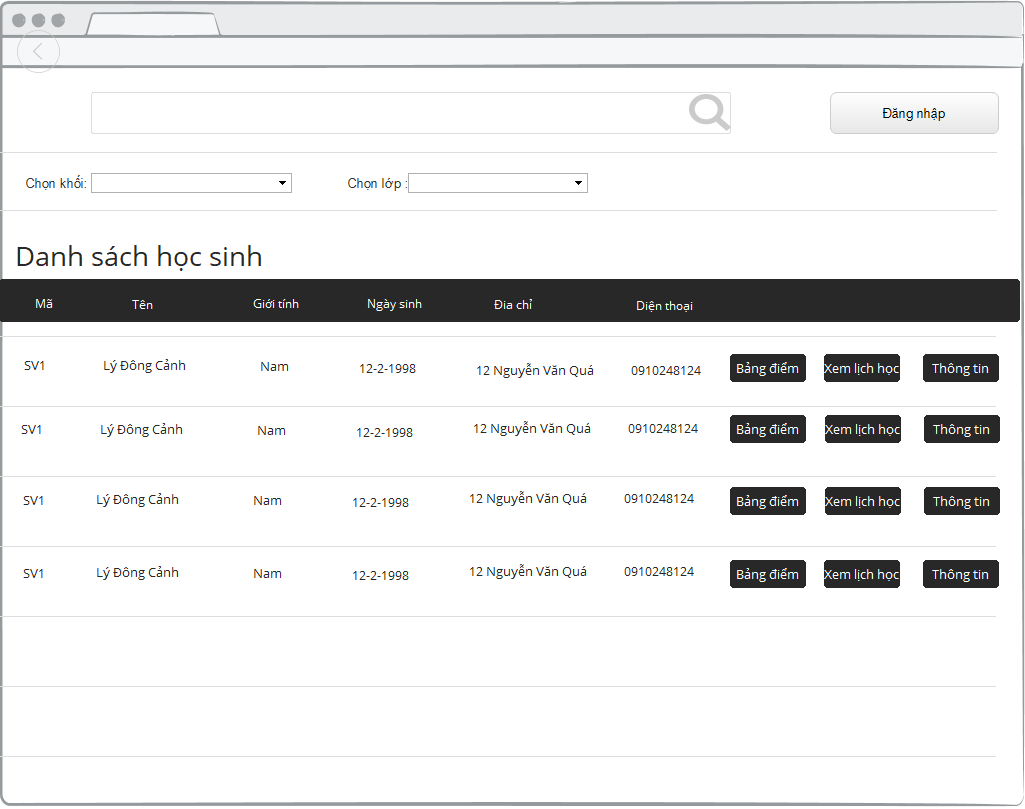
Hình 3‑1-20 : ClassDiagram

## Deployment diagram

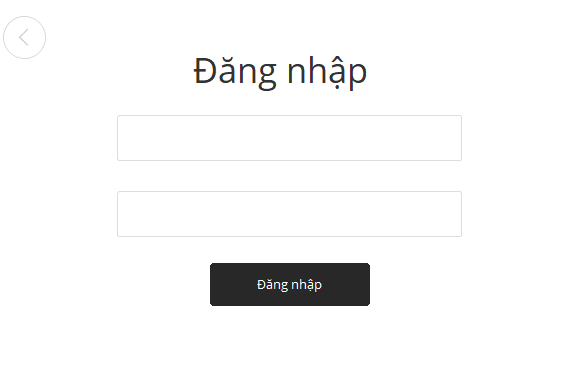
## 

Hình 3‑1-21 : Deployment Diagram

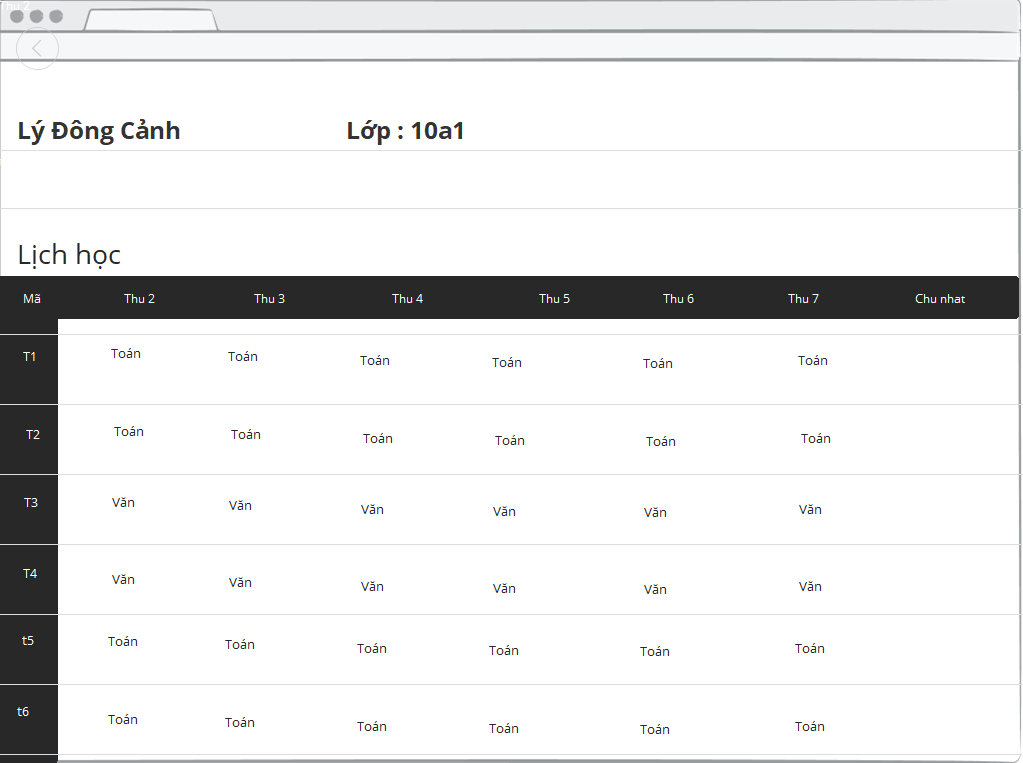
## Giao diện Mockup



Hình 3‑1-21 : Trang chủ



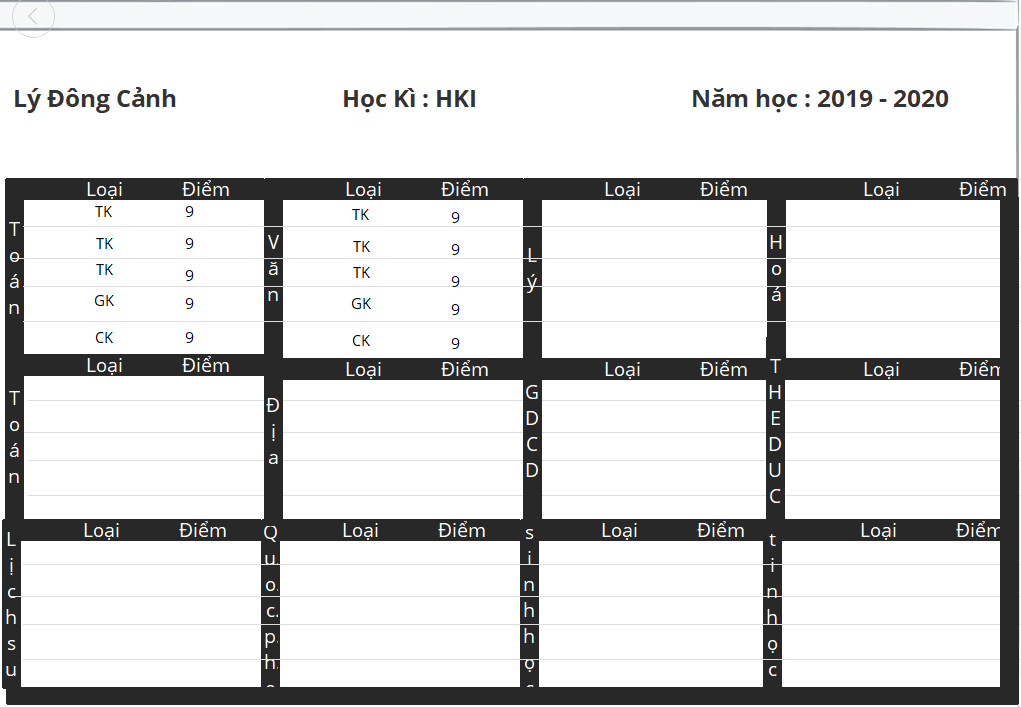
Hình 3‑1-22 : Đăng nhập



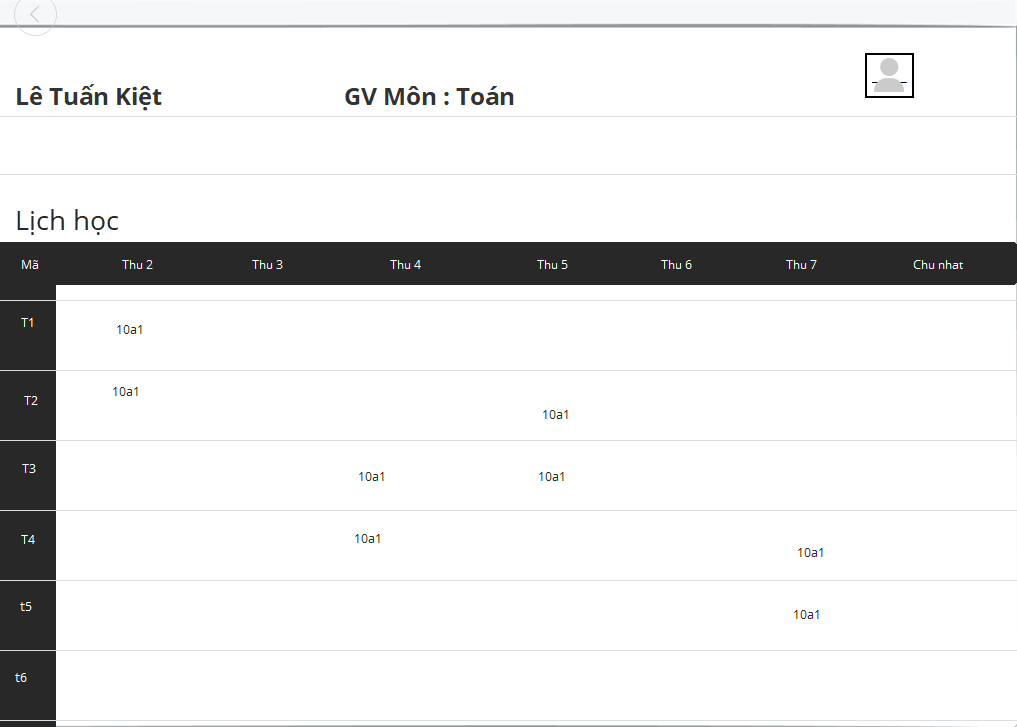
Hình 3‑1-23 : Xem lịch học



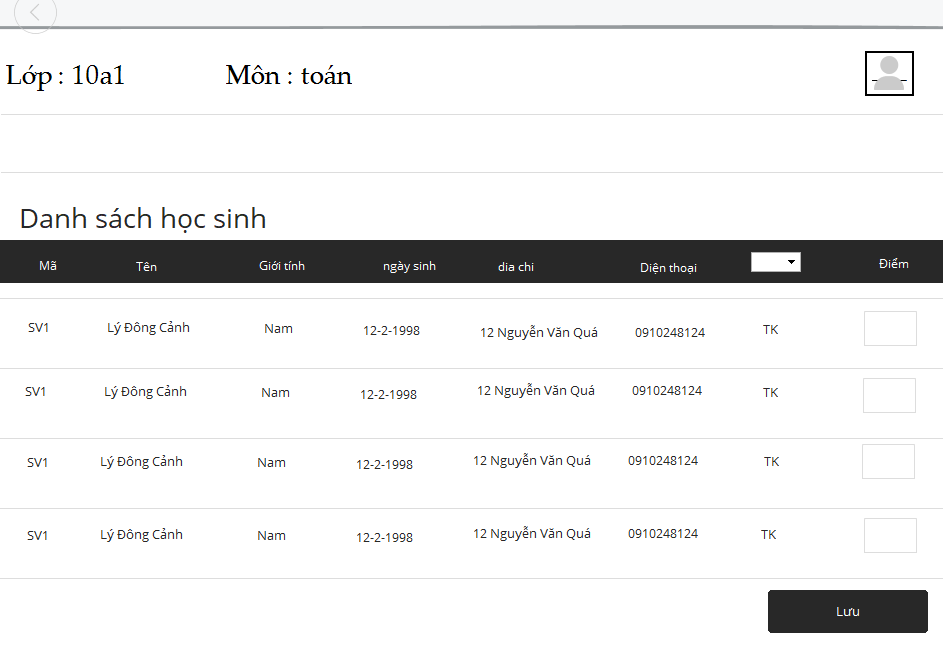
Hình 3‑1-24 : Xem thông tin giáo viên, học sinh



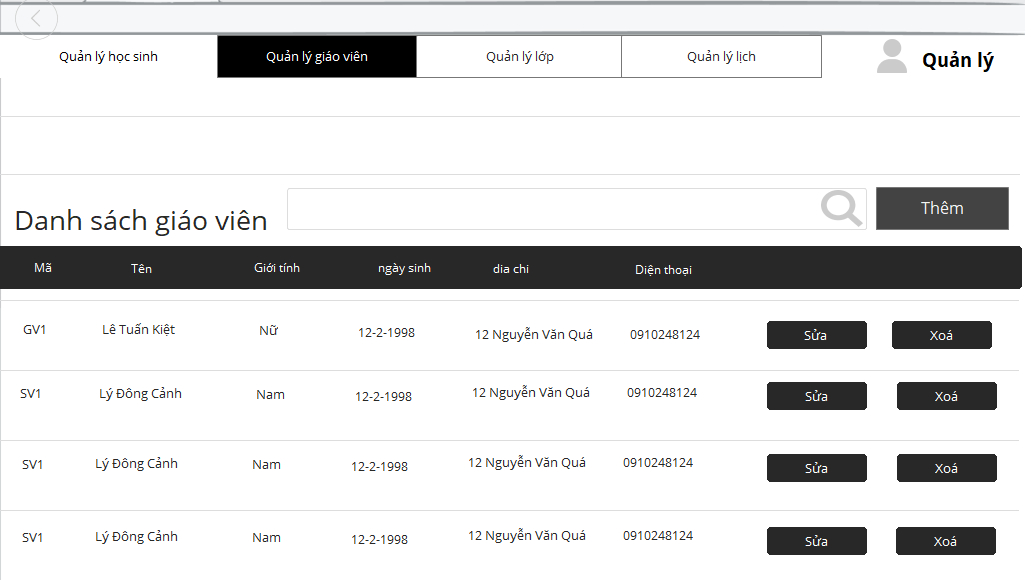
Hình 3‑1-25 : Xem điểm



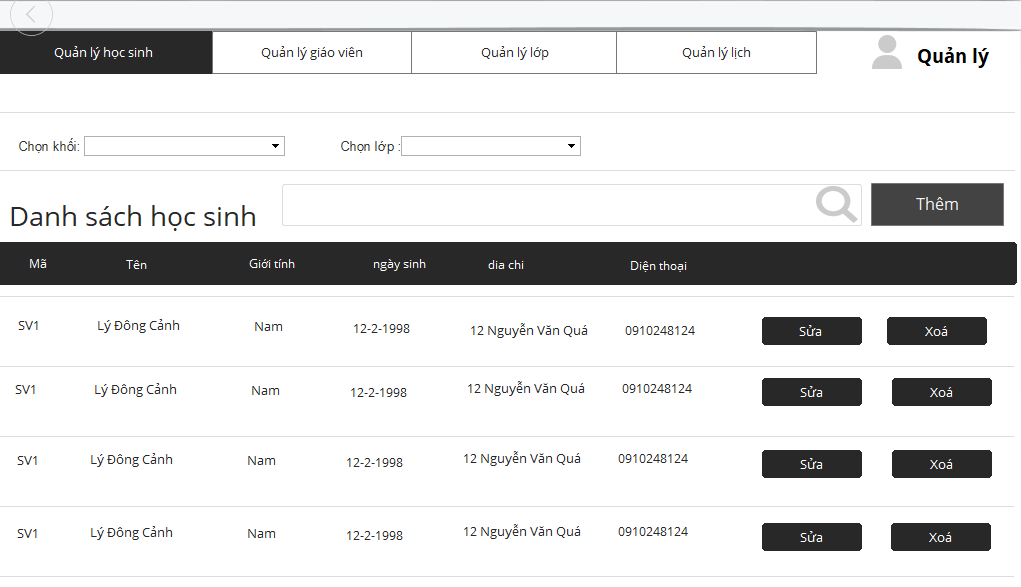
Hình 3‑1-26 : Xem lịch dạy



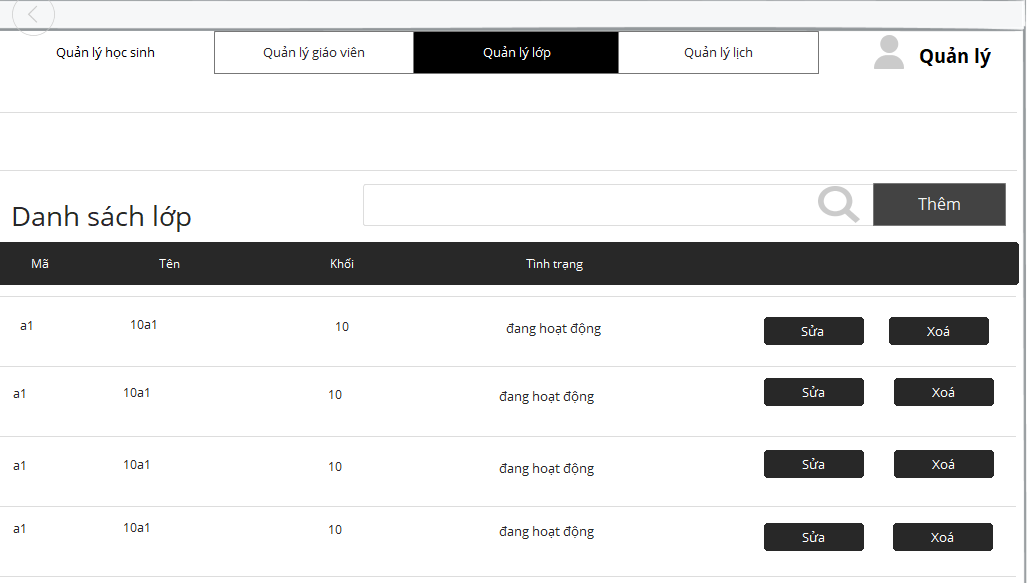
Hình 3‑1-27 : Nhập điểm



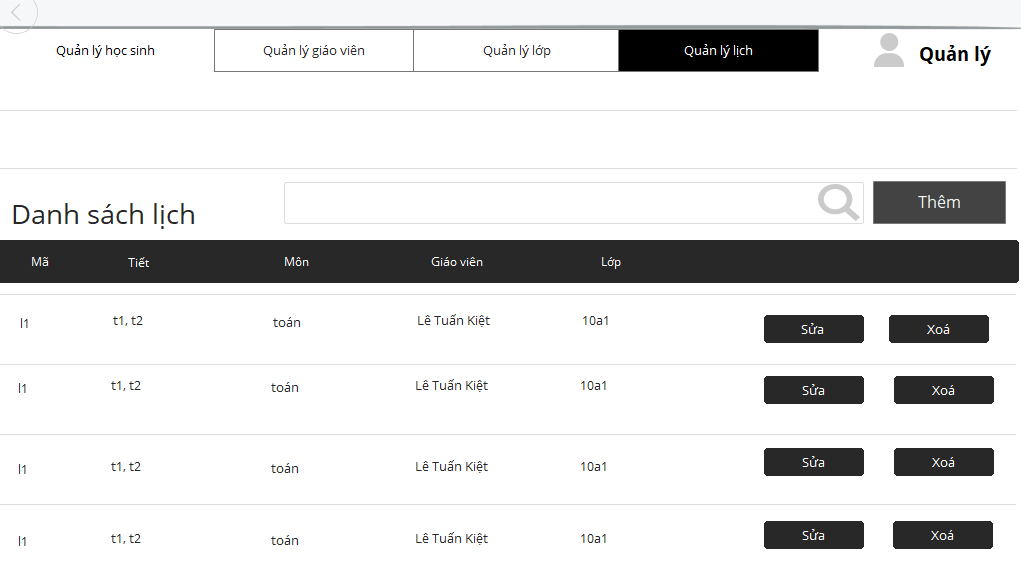
Hình 3‑1-27 : Quản lý giáo viên



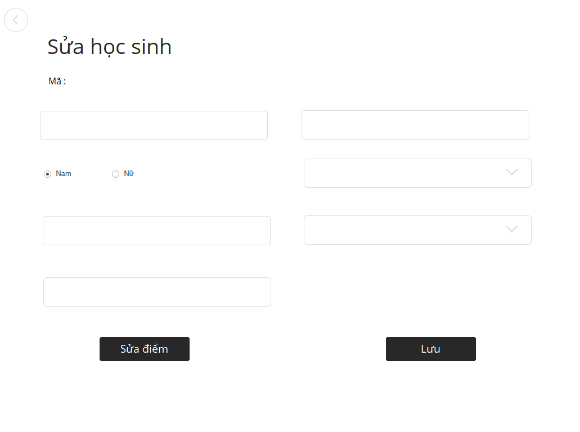
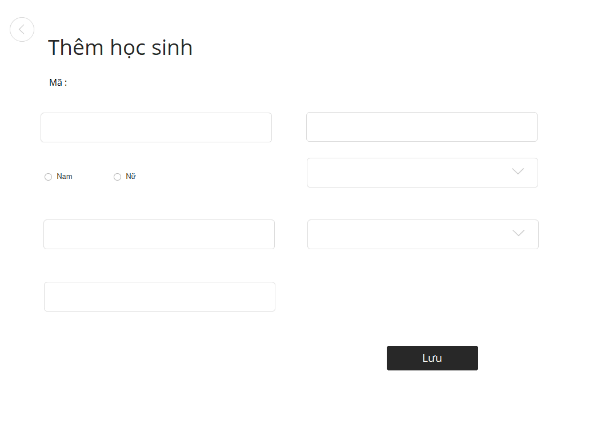
Hình 3‑1-28 : Quản lý học sinh



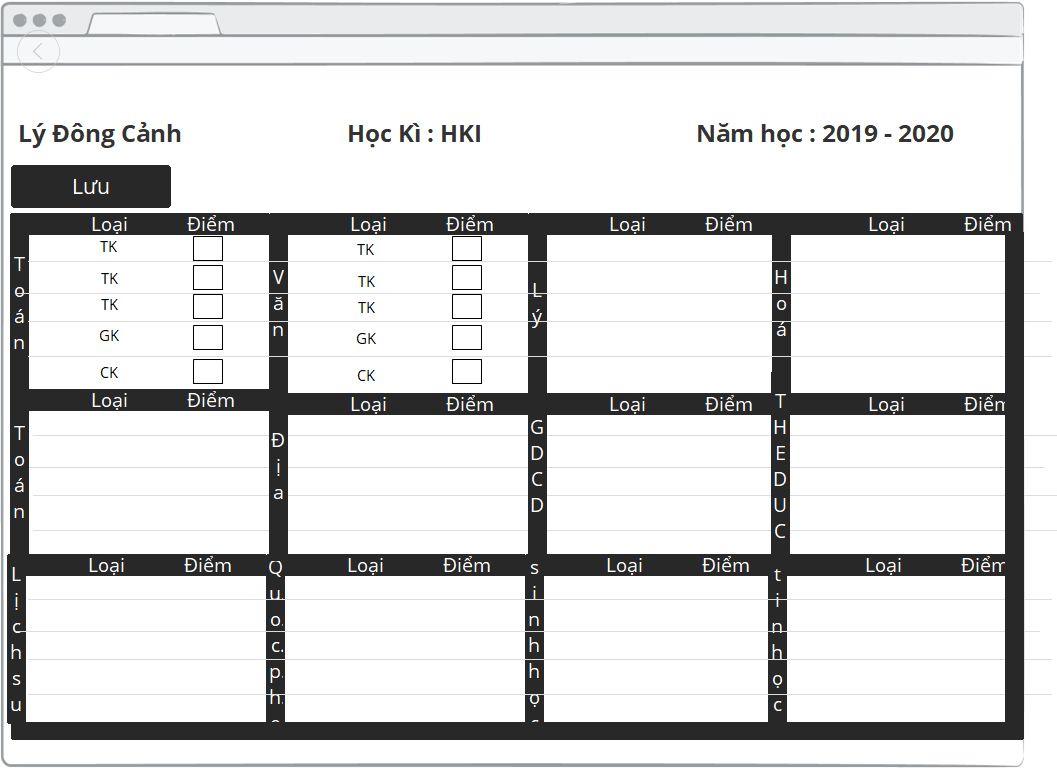
Hình 3‑1-29 : Quản lý lớp



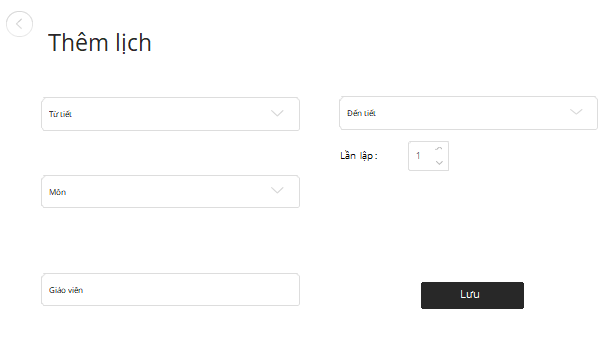
Hình 3‑1-29 : Quản lý lịch



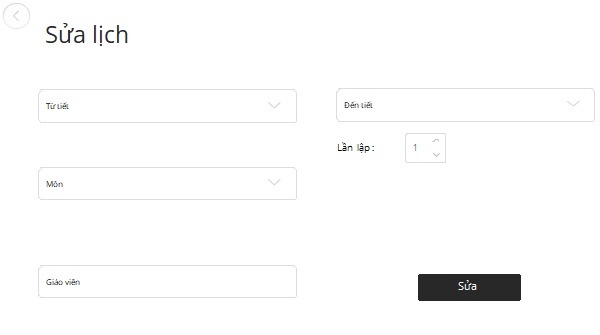
Hình 3‑1-30 : Thêm, sửa học sinh



Hình 3‑1-31 : Sửa điểm học sinh



Hình 3‑1-32 : Thêm lịch



Hình 3‑1-33 : Sửa lịch

# : HIỆN THỰC

## Cấu hình phần cứng, phần mềm

Phía Server:

Phía Client:

* Yêu cầu phần cứng:
  + Bộ vi xử lý: Intel Dual Core trở lên
  + Máy tính : Máy tính cá nhân có kết nối mạng LAN.
  + Bộ nhớ RAM: Tối thiểu 4GB.
  + Ổ cứng: 20GB ổ cứng.
  + Màn hình: Có độ phân giải 1366x768 trở lên.
* Yêu cầu phần mềm:
  + Hệ điều hành Microsoft Windows 7 trở lên.
  + Visual Studio Code.
  + Trình duyệt web (Chrome).

## 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 từ Internet

1. <https://vi.wikipedia.org/wiki>
2. <https://aws.amazon.com/vi/dynamodb/>
3. <https://aws.amazon.com/vi/ec2/>
4. <https://aws.amazon.com/vi/s3/>
5. <https://techtalk.vn/virtual-dom-trong-reactjs.html>
6. <https://viblo.asia/p/gioi-thieu-ant-design-L4x5xwzblBM>
7. <https://techblog.vn/co-ban-ve-rest-va-restfull-api>
8. <https://viblo.asia/p/tong-quan-ve-jsx-Qbq5QqBL5D8>