

AcadIUH: CÔNG CỤ HỖ TRỢ SINH VIÊN TẠI TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THÀNH PHỐ HỒ CHÍ MINH

TÔN LONG PHƯỚC, ĐỖ ANH BÔN, TẠ MINH HIẾU

Khoa Công nghệ thông tin, Trường Đại học Công nghiệp thành phố Hồ Chí Minh

tonlongphuoc@iuh.edu.vn

Tóm tắt.

Trường đại học Công nghiệp Tp.HCM là trường thuộc top 10 trường đại học hàng đầu của Việt Nam trong bảng xếp hạng QS năm 2021. Nơi đào tạo đa ngành nghề với cơ sở hạ tầng đạt chuẩn và đội ngũ giảng viên có chuyên môn cao. Trường hiện nay có hơn 30,000 sinh viên đang học tập và nghiên cứu. Vì vậy, việc hỗ trợ các công tác học vụ cho sinh viên đang học tập và nghiên cứu tại đây là cấp thiết. Chúng tôi, đề xuất một công cụ có tên là AcadIUH nhằm hỗ trợ cho sinh viên thực hiện các nghiệp vụ học vụ tại trường. Công cụ này chúng tôi tập trung vào 2 phân hệ: Ứng dụng web và ứng dụng cho thiết bị di động. Hai phân hệ này chúng tôi triển khai đồng bộ với nhau nhằm tăng cường khả năng xử lý và truyền tải thông tin kịp thời đến sinh viên. Cả hai phân hệ này chúng tôi tập trung giải quyết các vấn đề như: Đăng ký học phần của sinh viên; Quản lý thời khóa biểu của sinh viên; Kế hoạch học tập của sinh viên; Quản lý kết quả học tập của sinh viên cũng như quản lý công nợ của sinh viên. Chúng tôi chọn lựa công nghệ để triển khai cho cả hai phân hệ trên là: Ngôn ngữ lập trình Java và Javascript, framework React Native, ReactJS, Nodejs và NPM, Spring Boot và Cloud Computing, cơ sở hạ tầng dưới dạng code bằng Terraform (HCL), các công cụ trong CI/CD. AcadIUH sẽ được triển khai cho các sinh viên thuộc bộ môn SE khoa CNTT của trường nhằm đánh giá mức độ hiệu quả của công cụ. Trong tương lai, chúng tôi cũng sẽ phát triển công cụ này với việc tích hợp chúng vào hệ thống PMT và mở rộng cho phạm vi đối tượng sử dụng của công cụ.

Từ khóa. Công cụ hỗ trợ, Học vụ, Ứng dụng web, Ứng dụng trên di động.

AcadIUH: ACADEMIC SUPPORT TOOL FOR STUDENTS AT INDUSTRIAL UNIVERSITY OF HO CHI MINH CITY

Abstract. Industrial University of Ho Chi Minh City is one of the top 10 universities in Vietnam in the QS rankings in 2021. The university also trains students for multiple industries. It has full standard infrastructure and highly qualified teaching staff. The school currently has more than 30,000 students studying and researching. Because of that, it is urgent to support academic work for students who are studying and doing research here. We are proposing a tool called *AcadIUH* to assist students in carrying out academic careers on campus. In *AcadIUH*, we focus on 2 modules: web applications and mobile applications. These two modules we deploy synchronously to enhance the ability to process and transmit information promptly to students. In both of these modules we focus on managing academic such as student enrollment; student timetable; student study plan; student performance and student debt. We choose the technology to deploy for both modules above: Java and Javascript programming language, framework React Native, ReactJS, Nodejs and NPM, Spring Boot and Cloud Computing, infrastructure as code by Terraform (HCL), tools CI/CD. *AcadIUH* will be deployed to students in the Information Technology department to assess the effectiveness of the tool. In the future, we will also develop this tool by integrating them into the PMT system and expanding the scope of the tool's audience.

Keywords. Support Tools, Academic, Web Application, Mobile Application

1 GIỚI THIỆU TỔNG QUAN

Trường ĐH Công nghiệp TP.HCM (IUH) là một trong những trường đại học hàng đầu của Bộ Công thương, cũng là trường có qui mô đào tạo lớn với hơn 30000 sinh viên học tập. Trường cũng đào tạo đa ngành, đa

bậc học và nhiều chương trình liên kết quốc tế¹. Trường được đầu tư cơ sở vật chất tốt, đáp ứng nhu cầu của người học và các tiêu chuẩn quốc tế. Bên cạnh, đội ngũ giảng viên có trình độ cao cùng với nhiều chương trình đào tạo đạt chuẩn quốc tế như AUN. Chính vì lẽ đó, nhà trường cần có một hệ thống website hỗ trợ đào tạo một cách hiệu quả và kịp thời cho người học. Tuy nhiên, hệ thống website hỗ trợ cho người học hiện nay còn nhiều hạn chế, điển hình là khả năng đáp ứng truy cập đồng thời của sinh viên trong một số thời điểm nhất định không được ổn định. Ví dụ như vào thời điểm nhà trường mở đăng ký lớp học phần cho học kì mới, mở đợt xét tuyển đầu cho các thí sinh thì nhu cầu truy cập website tăng cao một cách đột ngột. Lúc này hiện tượng trang không phản hồi khi truy cập website của nhà trường xảy ra thường xuyên trong nhiều ngày hoặc không thực hiện các chức năng trên hệ thống. Sinh viên không thể làm gì được trong vòng nhiều giờ dù đã tốn công sức và thời gian để đăng ký học phần chẳng hạn. Để xảy ra tình trạng này cũng có nhiều nguyên nhân. Nguyên nhân đầu tiên là số lượng sinh viên truy cập website của trường tăng đột biến, từ đó dẫn đến việc số lượng yêu cầu (request) gửi lên máy chủ tăng cao. Thứ hai, xử lý quá nhiều yêu cầu vượt quá giới hạn của máy chủ gây hiện tượng trang không phản hồi trong vào nhiều giờ, nhiều ngày liền.

Ngoài ra, cũng còn có những nguyên nhân khác quan khác như đường truyền và hệ thống hạ tầng của nhà trường chưa đáp ứng đủ mạnh. Vì vậy, giải pháp dễ thấy nhất là nâng cấp cấu hình của máy chủ, thậm chí là tăng số lượng máy chủ lên để xử lý được nhiều yêu cầu hơn. Đồng thời, nếu như việc đầu tư này đã đáp ứng, chúng ta cũng cần phải đưa ra các giải pháp này rõ ràng hơn như việc kết nối máy chủ mới vào hệ thống xử lý cũ cũng là một bài toán lớn. Tương tự như với việc nâng cấp cấu hình cũng có nhiều câu hỏi đặt ra: Nâng cấp trong bao lâu? Có xảy ra rủi ro trong lúc nâng cấp không? Nếu phát sinh những vấn đề ngoài ý muốn lúc nâng cấp thì phải ứng cứu kịp thời như thế nào? Và còn rất nhiều vấn đề cần giải quyết nữa. Trong đó một vấn đề chúng ta cũng cần lưu ý, đó là một năm chỉ xảy ra hai đến ba đợt mở lớp đăng ký học phần. Nghĩa là chỉ có tại thời điểm đó thì nhu cầu mới vượt quá ngưỡng xử lý của máy chủ. Vậy nhà trường có nên tốn kém hay không khi đầu tư một hệ thống với chi phí lớn cho một số sự cố xảy ra không thường xuyên?

Để giải quyết các vấn đề đó, chúng tôi đã đưa ra các giải pháp như dùng điện toán đám mây (cloud computing) nhằm tổ chức lưu trữ và xử lý các nghiệp vụ; xây dựng hệ thống website và ứng dụng trên di động (app) cho sinh viên dễ dàng xử lý và giảm tải cho máy chủ. Nói về giải pháp sử dụng các dịch vụ trên điện toán đám mây như cho thuê máy chủ, dùng các dịch vụ web... nổi tiếng trên thế giới hiện nay đó là các dịch vụ cung cấp bởi các công ty như Amazon với AWS, Microsoft với Azure, trong nước có các tổ chức như Viettel, FPT, Sao Bắc Đẩu... Điện toán đám mây là một giải pháp có tính xu hướng toàn cầu cho các cá nhân, tổ chức, doanh nghiệp vừa và nhỏ [1]. Hiện nay, nhu cầu tạo ra các hệ thống dịch vụ trực tuyến phục vụ cho nhu cầu cá nhân, kinh doanh ngày càng nhiều. Những hệ thống này khi triển khai trên điện toán đám mây sẽ có chi phí rẻ, cấu hình đơn giản nhưng vẫn an toàn, bảo mật tốt, khả năng nâng cấp bảo trì hệ thống dễ dàng. Hơn hết, trong các ưu điểm trên là khả năng cân bằng tải, đó là ưu điểm tuyệt vời của điện toán đám mây. Trong đó, Scale là dịch vụ có khả năng tự động điều chỉnh số lượng máy chủ để đáp ứng nhu cầu truy cập của khách hàng (tăng khi yêu cầu nhiều và giảm khi yêu cầu ít). Do đó nhà phát triển hệ thống không còn phải đầu tư quá nhiều chi phí chỉ để giải quyết nhu cầu cho việc sử dụng tài nguyên của hệ thống trong một quãng thời gian ngắn trong năm. Đó cũng là lý do chúng tôi chọn giải pháp này cho bài toán xây dựng hệ thống hỗ trợ đào tạo cho sinh viên của IUH.

Riêng giải pháp xây dựng website và ứng dụng trên di động cho sinh viên sử dụng nhằm giải quyết các nguyên nhân được liệt kê sau đây. Nguyên nhân thứ nhất là website của trường IUH thực hiện cơ chế kết xuất máy chủ Server-Side Rendering (SSR). Cơ chế SSR có nhiều ưu điểm tuyệt vời như khả năng Search Engine Optimization (SEO) hiệu quả, chạy được trên phần lớn các trình duyệt thông dụng (kể cả bị vô hiệu hóa JavaScript thì vẫn chạy tốt), hiệu năng cho lần tải trang đầu tiên tốt... Nhưng cơ chế SSR lại có một nhược điểm lớn chính là máy chủ phải đảm nhiệm gánh nặng kết xuất trang web (thuật toán kết xuất lại tốn nhiều chi phí), gánh nặng trả về nội dung trang web (tốn nhiều băng thông) và vô tình làm tăng số lượng như cầu phải xử lý cho máy chủ [2]. Nguyên nhân thứ hai, đặt trường hợp khi nhu cầu truy cập website tăng cao đột biến, máy chủ phải thực hiện một số lượng yêu cầu cực lớn và còn phải phụ trách gánh nặng kết xuất trang web và gửi trả về nó càng làm tình hình xấu đi. Hơn nữa, các trang web như trang đăng ký

¹ www.iuh.edu.vn accessed 3/4/2021

học phần, quản lý thông tin cho sinh viên là những trang web có tính cá nhân nên cần rất ít khả năng SEO, tính chất tương thích với trình duyệt hoàn toàn có thể khắc phục được bởi lập trình viên. Đó cũng là lý do chúng tôi xây dựng ứng dụng cho thiết bị di động nhằm hỗ trợ cá nhân hóa cũng như giảm tải cho máy chủ trong hệ thống.

Để triển khai giải pháp trên, chúng tôi đã triển khai hệ thống web với kiến trúc micro-services.

Micro-service là một kỹ thuật phát triển phần mềm, một biến thể của kiến trúc hướng dịch vụ (SOA), trong đó cấu trúc một ứng dụng như một tập hợp các dịch vụ được ghép với nhau [3]. Trong kiến trúc micro-service, các dịch vụ được xử lý tốt và các giao thức rất nhẹ. Lợi ích của việc phân tách một ứng dụng thành các dịch vụ nhỏ hơn là nó cải thiện tính mô đun. Điều này làm cho ứng dụng trở nên gọn nhẹ, dễ hiểu, dễ phát triển và trở nên linh hoạt hơn. Cụ thể chúng tôi chia hệ thống làm hai phần chính: Front-end và Back-end. Front-end của chúng tôi ở đây chính là ứng dụng di động AcadIUH cho sinh viên, ứng dụng website AcadIUH cho quản trị viên và giảng viên của nhà trường. Front-end cũng sẽ đảm nhận trách nhiệm nhận dữ liệu, hiển thị dữ liệu cũng như các tương tác cho người dùng. Riêng website dành cho quản trị viên, giảng viên của nhà trường, chúng tôi sử dụng cơ chế kết xuất máy khách Client-Side Rendering (CSR) thay vì dùng cơ chế SSR trước đây. Chúng tôi chọn hướng tiếp cận này vì website cho quản trị viên vốn dĩ không cần quảng cáo (SEO) trên các công cụ tìm kiếm nên cơ chế CSR vô cùng thích hợp. Back-end chúng tôi xây dựng hệ thống máy chủ không cần phải đảm nhiệm vai trò kết xuất trang web mà chỉ cần xử lý, lấy dữ liệu từ hệ cơ sở dữ liệu và trả về dưới định dạng JSON. Ngoài ra, khả năng phát triển song song và tương đối độc lập giữa Front-end và Back-end cũng là ưu điểm cho việc bảo trì và nâng cấp.

Nhằm hỗ trợ cho giải pháp trên, việc bảo trì, sửa lỗi, nâng cấp, triển khai...cũng là vấn đề quan trọng cần phải giải quyết. Chúng tôi ứng dụng Jenkins và Docker để tối ưu hóa việc này. Jenkins nói riêng và các hệ thống CI/CD nói chung cho phép lập trình viên viết kịch bản cho quy trình bằng ngôn ngữ chuyên dụng và tự động hóa mọi việc chỉ bằng một thao tác đơn giản [4]. Jenkins sẽ tự động thực hiện kiểm tra lỗi, kiểm tra hiệu năng ứng dụng, tự động xây dựng, triển khai phiên bản mới...Quá trình triển khai hệ thống cũng tốn nhiều công sức, lập trình viên phải cài đặt môi trường thích hợp để chạy được chương trình. Trong đó, cài đặt càng nhiều thì càng phát sinh nhiều vấn đề như phiên bản không tương thích, xung đột giữa các phần mềm được cài đặt... Docker là công nghệ cho phép tạo ra các máy ảo, lập trình viên chỉ cần khai báo những thành phần cần thiết cho cấu hình thông qua Dockerfile và Docker sẽ tự động cài đặt nó. Máy chủ không cần phải cài đặt quá nhiều thứ để chạy ứng dụng nữa mà chỉ cần cài đặt Docker.

Bài báo của chúng tôi chia làm 5 phần: *Phần 1 Giới thiệu tổng quan về hệ thống; Phần 2 Các công nghệ liên quan; Phần 3 Công cụ AcadIUH; Phần 4 Kiểm nghiệm thực tế và Phần 5 Kết luận và hướng phát triển.* Trong đó *Phần 3* là phần đóng góp chính của bài báo.

2 CÁC CÔNG NGHỆ LIÊN QUAN

2.1 Ngôn ngữ phát triển ứng dụng

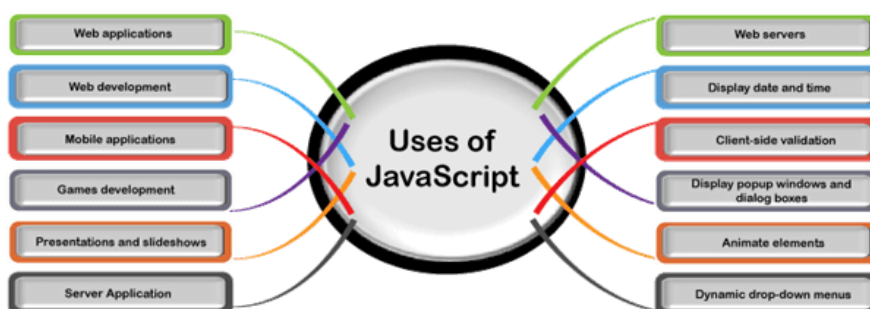
2.1.1 Ngôn ngữ lập trình Java

Java là ngôn ngữ lập trình hướng đối tượng đang được duy trì và phát triển bởi Oracle. Java được sử dụng trong phát triển phần mềm ứng dụng desktop, website, game hay ứng dụng trên các thiết bị di động. Java cũng là một ngôn ngữ cho phép phát triển các ứng dụng chạy trên mọi nền tảng. Hiện nay, đội ngũ phát triển ứng dụng dùng ngôn ngữ Java ngày càng hùng hậu. Đội ngũ này cũng cung cấp một tập hợp các chương trình giúp lập trình viên phát triển và chạy các ứng dụng một cách hiệu quả. Java được tổ chức bao gồm một công cụ thực thi, một trình biên dịch và một bộ thư viện. James Gosling đã phát triển nền tảng Java tại Sun Microsystems và tập đoàn Oracle sau đó đã mua lại nó và phát triển đến ngày hôm nay [5].

2.1.2 Ngôn ngữ lập trình Javascript

Javascript ra đời năm 1995, ban đầu Javascript không được xem là một ngôn ngữ lập trình mà chỉ được xem là một ngôn ngữ kịch bản (scripting language). Nguyên nhân là thời bấy giờ Javascript chỉ được sử dụng để xây dựng các tương tác với trang Web. Bản thân nó phải thực thi trong một chương trình không phải hệ điều hành (cụ thể là trình duyệt Web) [6]. Nhưng sau này, khi vai trò và sự phổ biến của websites ngày càng lớn, cộng với đó là sự ra đời và lớn mạnh thần tốc của NodeJS (2009) giúp Javascript không còn là một ngôn ngữ phải thực thi trong trình duyệt. Hiện Javascript chính thức được công nhận là một ngôn ngữ lập trình chính thống. Javascript hoạt động được trên nhiều nền tảng (Web, Server) và là ngôn ngữ đơn giản, dễ học cho người mới sử dụng. Hiện nay, nhiều công nghệ xây dựng dựa trên Javascript như Nodejs,

React, VueJS...Đồng thời Javascript cũng được dùng cho việc xây dựng nhiều ứng dụng với vai trò khác nhau từ Web, ứng dụng Mobile và một số ứng dụng khác (Hình 2.1.1).

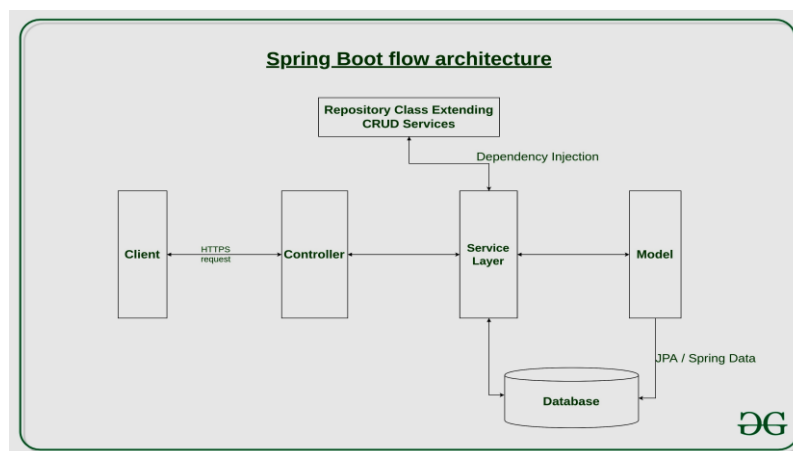


Hình 2.21 Những ứng dụng được phát triển bởi JavaScript [6]

2.2 Những framework hỗ trợ phát triển ứng dụng

2.2.1 Framework Spring Boot

Spring Boot là khuôn mẫu (framework) phát triển dựa trên ngôn ngữ Java trong hệ sinh thái Spring framework [7]. Nó giúp cho các lập trình viên đơn giản hóa quá trình lập trình một ứng dụng với Spring. Trong Spring Boot, người lập trình chỉ tập trung vào việc phát triển các nghiệp vụ cho ứng dụng mà không quan tâm nhiều về kiến trúc và mã nguồn của ứng dụng. Một kiến trúc webserver xây dựng dựa trên spring boot gồm các thành phần cơ bản như sau: Repository Class Extending, Service Layer, Model, Controller (Hình 2.2.2)



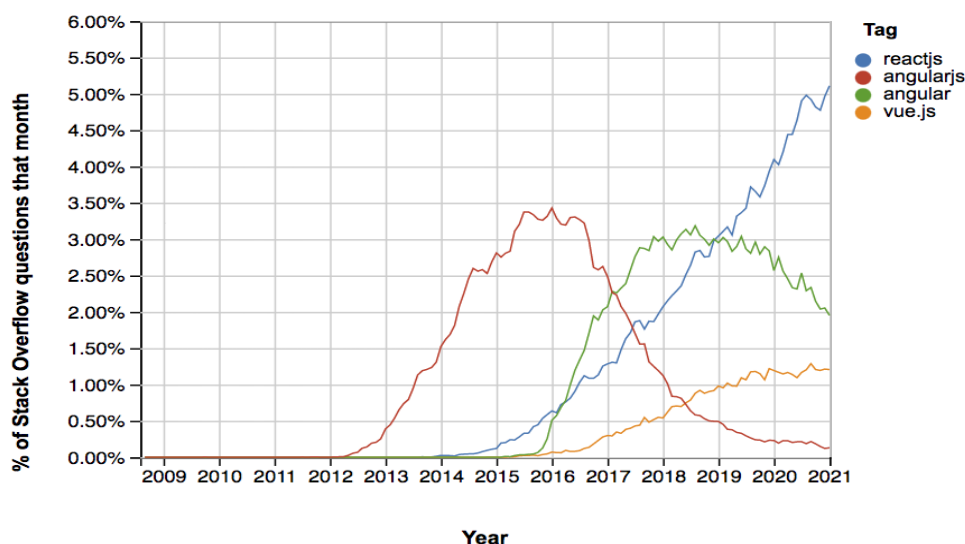
Hình 2.2.2 Kiến trúc của luồng hoạt động của Spring Boot [7]

Repository Class Extending là các thư viện hoặc framework được phát triển và duy trì bởi cộng đồng các lập trình viên Java để hỗ trợ quá trình xây dựng hệ thống một cách nhanh chóng hoặc cũng có thể là các lớp chứa các phương thức xử lý, truy xuất dữ liệu (CRUD) do lập trình viên viết ra; Service Layer là lớp dịch vụ chứa các phương thức (hàm) xử lý các yêu cầu như chỉnh sửa, lớp Service thường sử dụng các phương thức của các lớp trong Repository Class Extending (bằng cách gọi tới một hay kết hợp nhiều phương thức với nhau) để truy xuất, thêm, xóa dữ liệu trong cơ sở dữ liệu. Nó là nơi tập trung các xử lý nghiệp vụ của hệ thống Back-end; Model là thành phần được đính kèm theo các truy vấn của Services Layer xuống cơ sở dữ liệu. Model đóng nhiều vai trò quan trọng như phát sinh hệ cơ sở dữ liệu theo liên kết các lớp của lập trình viên, chuyển đổi kết quả truy xuất từ cơ sở dữ liệu thành các thực thể lớp; Controller là thành phần giao tiếp trực tiếp với phía client, tồn tại dưới dạng các API. Model có trách nhiệm xử lý yêu cầu của client, xác minh các yêu cầu đó. Model truy cập đến phương thức phù hợp trong Service Layer để đạt được kết quả mong muốn, trả kết quả về cho phía client.

2.2.2 React Native, React.js

React Native là một thư viện mở dùng xây dựng ứng dụng di động dựa trên JavaScript. React Native được phát triển và duy trì bởi Facebook [8]. React Native giúp phát triển ứng dụng di động một cách nhanh chóng. Nó hỗ trợ cho các ứng dụng chạy được trên hai nền tảng iOS và Android. React Native giúp lập trình viên chỉ cần phát triển ứng dụng trên một mã nguồn duy nhất mà vẫn có thể tương thích với cả hai hệ điều hành Android và iOS (thậm chí là cả các hệ điều hành cho Ipad và các dòng tivi).

Ngoài ra, React.js nói chung và React Native nói riêng là công nghệ được yêu thích hiện nay. Phong cách xây dựng mã nguồn trong React Native là phân chia các khối chức năng giúp dễ dàng trong việc quản lý, bảo trì, sửa lỗi, nâng cấp ứng dụng. Tương tự React Native, ReactJS là công nghệ sử dụng việc xây dựng ứng dụng web dựa trên JavaScript được duy trì và phát triển bởi Facebook. ReactJS được nhận xét là dễ học, dễ hiểu, dễ làm quen nhưng hiệu quả lại cao, phù hợp với nhiều ứng dụng web, kể cả những ứng dụng có quy mô cực lớn (như Facebook). Tương tự React Native, ReactJS cũng có cộng đồng lớn mạnh, nhiều thư viện và framework hỗ trợ [9]. Đội ngũ phát triển React cũng đang không ngừng làm việc hằng ngày để phát triển công nghệ này ngày một tốt hơn. Theo khảo sát của Stack Overflow, ReactJS là công nghệ Front-end được nhiều người quan tâm nhất trong vòng những năm từ 2019 trở lại đây (Hình 2.2.3). Cụ thể trong năm 2020, React.js dẫn đầu bảng khảo sát ở mức độ phổ biến và yêu thích trong danh sách các framework Front-end [10].

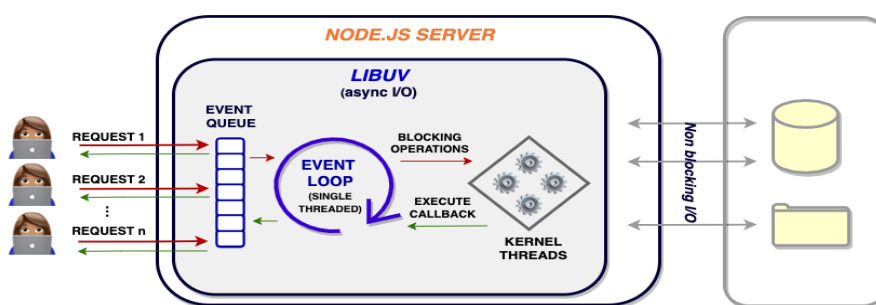


Hình 2.2.3 Khảo sát sự phổ biến của các framework Frontend [10]

Node JS

Node JS là một môi trường máy chủ thực thi Javascript mã nguồn mở, hoạt động trên nhiều nền tảng như Windows, MacOS, Linux...NodeJs là 1 công nghệ có cộng đồng hỗ trợ và phát triển lớn mạnh. Năm 2019 Node JS bổ sung thêm được hơn 50% số lượng lập trình viên sử dụng trên thế giới. Đến nay vẫn chiếm vị thế là công nghệ được ưa chuộng và sử dụng nhiều nhất [11]. NodeJS không cần phải cấu hình phức tạp, xây dựng dựa trên Javascript, cho nên NodeJS được nhiều lập trình viên ưa chuộng bởi tính đơn giản và tiện dụng của nó. Node JS được sử dụng làm webserver, làm môi trường phát triển web (ReactJS), ứng dụng di động (React Native),...Webserver được phát triển bằng Node JS có hiệu năng, tốc độ xử lý cao nhưng vẫn có khả năng bảo trì nâng cấp tốt.

Hệ thống web xây dựng từ Node JS có khả năng hoạt động ổn định, khắc phục nhiều điểm yếu của các webserver xây dựng dựa trên các công nghệ đa luồng (các bài toán chống ách tắc luồng, xử lý đồng bộ và bất đồng bộ khi cùng truy cập đến một hoặc nhiều tài nguyên,... là những bài toán lớn đối với các lập trình viên). Bên cạnh đó, NodeJS có số lượng rất lớn các thư viện được phát triển bởi cộng đồng lập trình viên. Hiện nay số lượng thư viện đã vượt qua con số 1,3 triệu là minh chứng cho thấy sức ảnh hưởng của công nghệ này.



Hình 2.2.4 Kiến trúc của luồng hoạt động của Node JS [11]

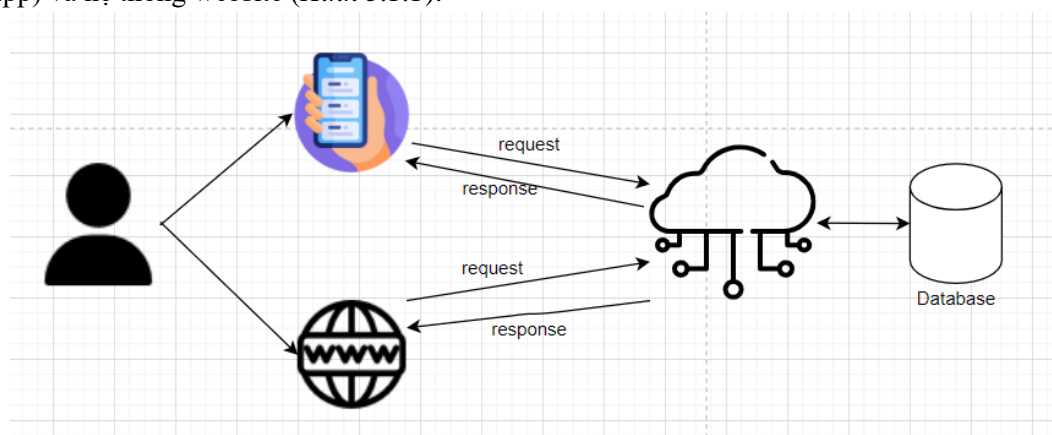
2.3 Cloud Computing

Điện toán đám mây (Cloud Computing) là việc ảo hóa các tài nguyên tính toán và các ứng dụng. Thay vì việc sử dụng một hoặc nhiều máy chủ thật (ngay trước mắt, có thể sờ được, có thể tự bạn ấn nút bật tắt được) thì nay sẽ sử dụng các tài nguyên được ảo hóa (virtualized) thông qua môi trường internet [12]. Việc phân phối các tài nguyên máy tính theo nhu cầu qua internet với chính sách thanh toán theo mức sử dụng. Thay vì mua, sở hữu và bảo trì các trung tâm dữ liệu và máy chủ vật lý, có thể tiếp cận các dịch vụ công nghệ, như năng lượng điện toán, lưu trữ và cơ sở dữ liệu, khi cần thiết [13]. Hiện có nhiều nhà cung cấp dịch vụ đám mây như Amazon Web Services (AWS), Google Cloud, Microsoft có Azure. Lợi ích của Cloud Computing mang lại cho người dùng rất nhiều như tiết kiệm chi phí, khả năng mở rộng linh hoạt về quy mô, hiệu năng cao, bảo mật, tốc độ truy xuất dữ liệu, năng suất của một hệ thống [14]. Hiện nay, Cloud cung cấp dịch vụ theo ba mô hình cơ bản: Cơ sở hạ tầng như một dịch vụ (IaaS – Infrastructure as a Service), nền tảng như một dịch vụ (PaaS – Platform as a Service), phần mềm như một dịch vụ (SaaS – Software as a Service) [15]. Việc cung cấp nhiều mô hình sẽ đáp ứng được nhu cầu đa dạng của các doanh nghiệp hiện nay. Ngoài ra, điện toán đám mây còn có mô hình triển khai như public/private cloud.

3. ACADIUH: CÔNG CỤ HỖ TRỢ HỌC TẬP CHO SINH VIÊN ĐẠI HỌC CÔNG NGHIỆP TP.HCM

2.4 Giới thiệu

Để triển khai các giải pháp mà chúng tôi nêu ra trong phần giới thiệu, chúng tôi đề xuất xây dựng ứng dụng AcadIUH. Ứng dụng này chúng tôi xây dựng dựa vào các công nghệ như Spring Boot, ReactJS, React Native. Đồng thời, để triển khai ứng dụng AcadIUH đến thiết bị người dùng (sinh viên, giảng viên và quản trị), chúng tôi cũng dùng các dịch vụ như AWS và CI/CD. Hệ thống AcadIUH gồm ứng dụng trên điện thoại (app) và hệ thống website (Hình 3.1.1).



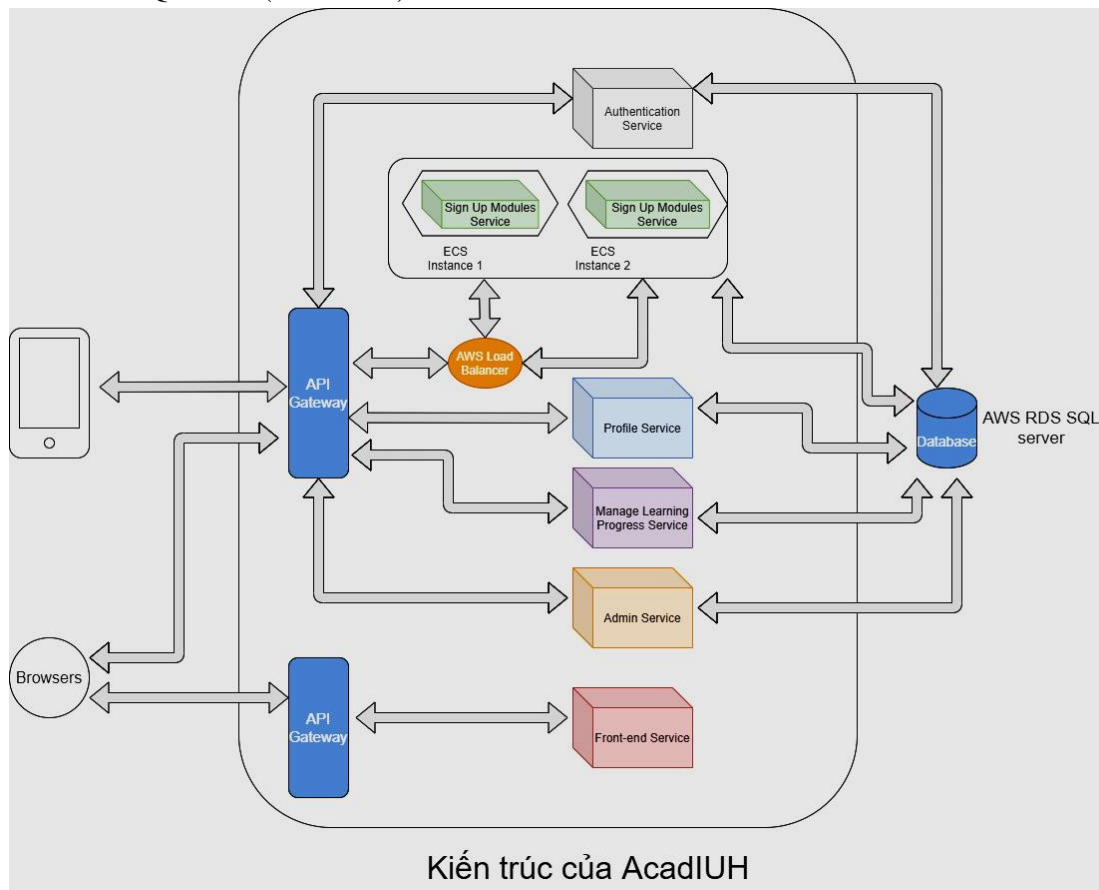
Hình 3.1.1 Mô hình Micro-Services của AcadIUH

Trong đó, React Native tạo ứng dụng trên điện thoại hệ điều hành Android/iOS, phục vụ cho các chức năng của sinh viên. ReactJS tạo website cho admin, các chức năng cho admin như tuyển sinh, quản lý đào tạo... Spring Boot chịu trách nhiệm tạo ra các API, xử lý các request. SQLServer dùng lưu trữ dữ liệu của toàn

hệ thống như dữ liệu sinh viên, dữ liệu học phần... Các dịch vụ điện toán đám mây sử dụng trong AWS gồm: Elastic Container Service (ECS); RDS; AutoScaling; Load Balancer; Elastic Container Registry (ECR). Cụ thể các dịch vụ này đảm nhiệm các công việc như sau: ECS điều phối container hiệu suất cao có hỗ trợ Docker; RDS lưu trữ dữ liệu quan hệ của hệ thống trên cloud; AutoScaling giúp tạo hoặc tắt số lượng server tùy vào lượng truy cập vào web/app trong một thời điểm; Load Balancer giúp điều hướng lượng truy cập, giúp cân bằng tải cho số lượng server hiện có và ECR, nơi lưu trữ các hình ảnh trên AWS.

3.2 Kiến trúc phần mềm

Micro-service là một kỹ thuật phát triển phần mềm, một biến thể của kiến trúc hướng dịch vụ (SOA), trong đó cấu trúc một ứng dụng như một tập hợp các dịch vụ được ghép với nhau [3]. Trong kiến trúc micro-service, các dịch vụ được xử lý tốt và các giao thức rất nhẹ. Đó cũng là lý do chúng tôi đề xuất cho kiến trúc của công cụ AcadIUH ở theo mô hình Microservice. Trong kiến trúc này chúng tôi tách các service thành Profile Service, Manage Learning Service, Sign Up Modules Service, Authentication Service và Admin Service. Đồng thời, chúng tôi cũng đề xuất phần AWS Load Balancer cho việc triển khai hệ thống trên nền tảng điện toán đám mây. Ngoài ra, dữ liệu chúng tôi cũng đề xuất tổ chức lưu trữ trên AWS RDS SQL sever (Hình 3.2.1).



Hình 3.2.1 Kiến trúc microservice của công cụ AcadIUH đề xuất

2.3 Phân tích hệ thống công cụ AcadIUH

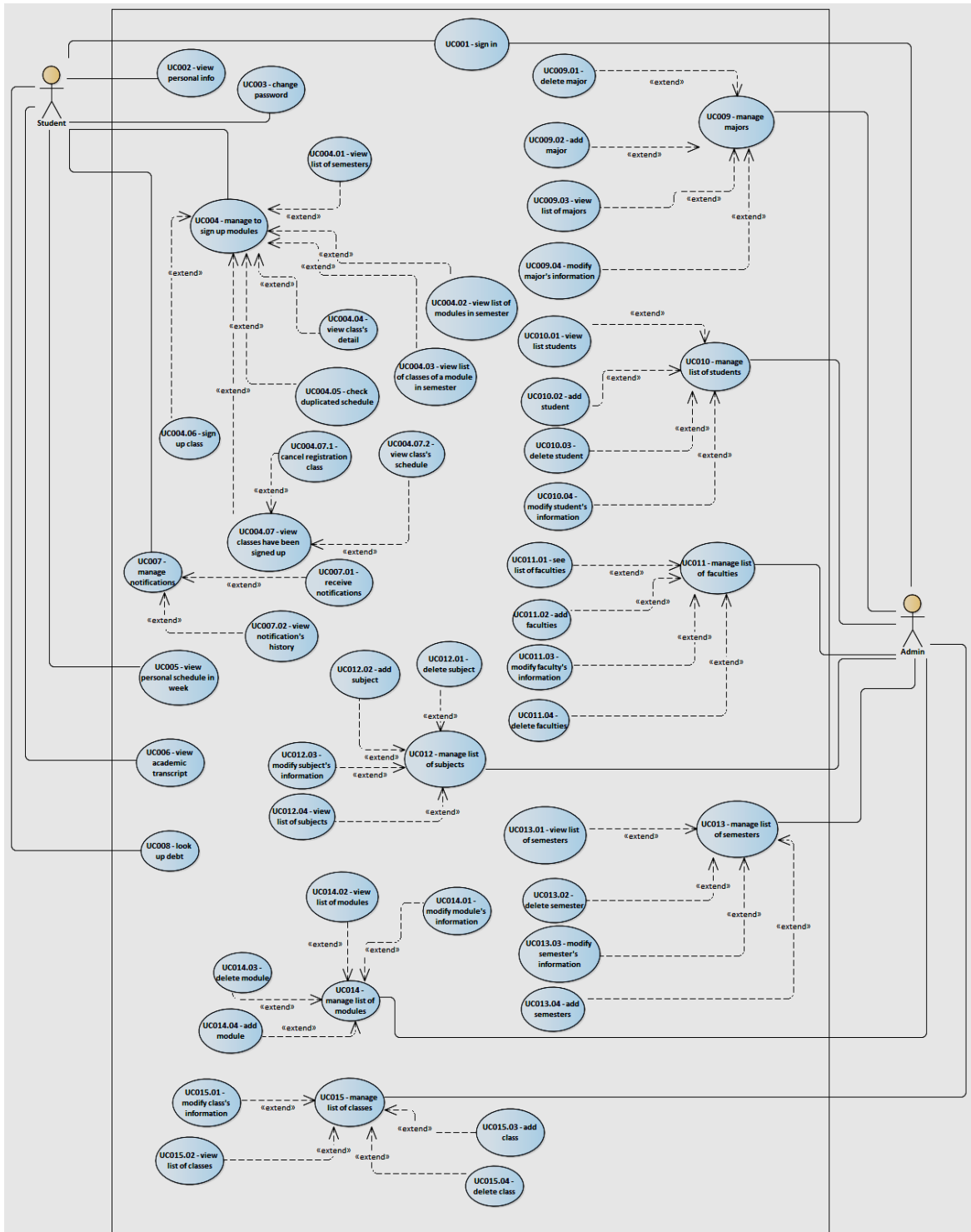
Đối tượng người dùng cho hệ thống AcadIUH gồm có sinh viên và quản trị viên của trường. Các chức năng của từng đối tượng được mô tả trong *Bảng 3.3.1*. Trong đó, chúng tôi tập trung vào các chức năng chính của hệ thống mà hiện nay nhà trường đang gặp phải vấn đề khó khăn như đăng ký học phần. Ngoài ra, các chức năng của quản trị viên cũng được đưa vào nhằm hỗ trợ cho hệ thống đồng bộ các nghiệp vụ đào tạo và các thao tác của sinh viên.

Bảng 3.3.1 Danh các chức năng trong công cụ AcadIUH

Đối tượng	Tên chức năng	Mô tả
Sinh viên	Xem thông tin cá nhân	Xem thông tin cá nhân của sinh viên đã đăng nhập vào ứng dụng
	Đăng nhập tài khoản	Đăng nhập vào hệ thống với tài khoản được tạo
	Đổi mật khẩu	Đổi mật khẩu của tài khoản đã đăng nhập vào ứng dụng
	Quản lý đăng ký học phần	Gồm các chức năng như xem danh sách học phần đã đăng ký, đăng ký lớp học phần, hủy đăng ký lớp học phần, xem chi tiết lịch học lớp học phần
	Quản lý thông báo	Cho phép nhận thông báo cảnh báo lịch thi, cảnh báo học vụ, xem lịch sử thông báo
	Tra cứu công nợ	Xem công nợ của sinh viên (công nợ đã thanh toán, chưa thanh toán, tổng công nợ)
Quản trị viên	Quản lý khoa	cho phép thêm, xóa, sửa thông tin của khoa, xem danh sách khoa
	Quản lý chuyên ngành	Cho phép thêm, xóa, sửa thông tin của chuyên ngành, xem danh sách chuyên ngành
	Quản lý sinh viên	Cho phép thêm, xóa, sửa thông tin của sinh viên, xem danh sách sinh viên
	Quản lý môn học	Cho phép thêm, xóa, sửa thông tin của môn học, xem danh sách môn học
	Quản lý học phần	Cho phép thêm, xóa, sửa thông tin của học phần, xem danh sách học phần
	Quản lý lớp học phần	Cho phép thêm, xóa, sửa thông tin của lớp học phần, xem danh sách lớp học phần

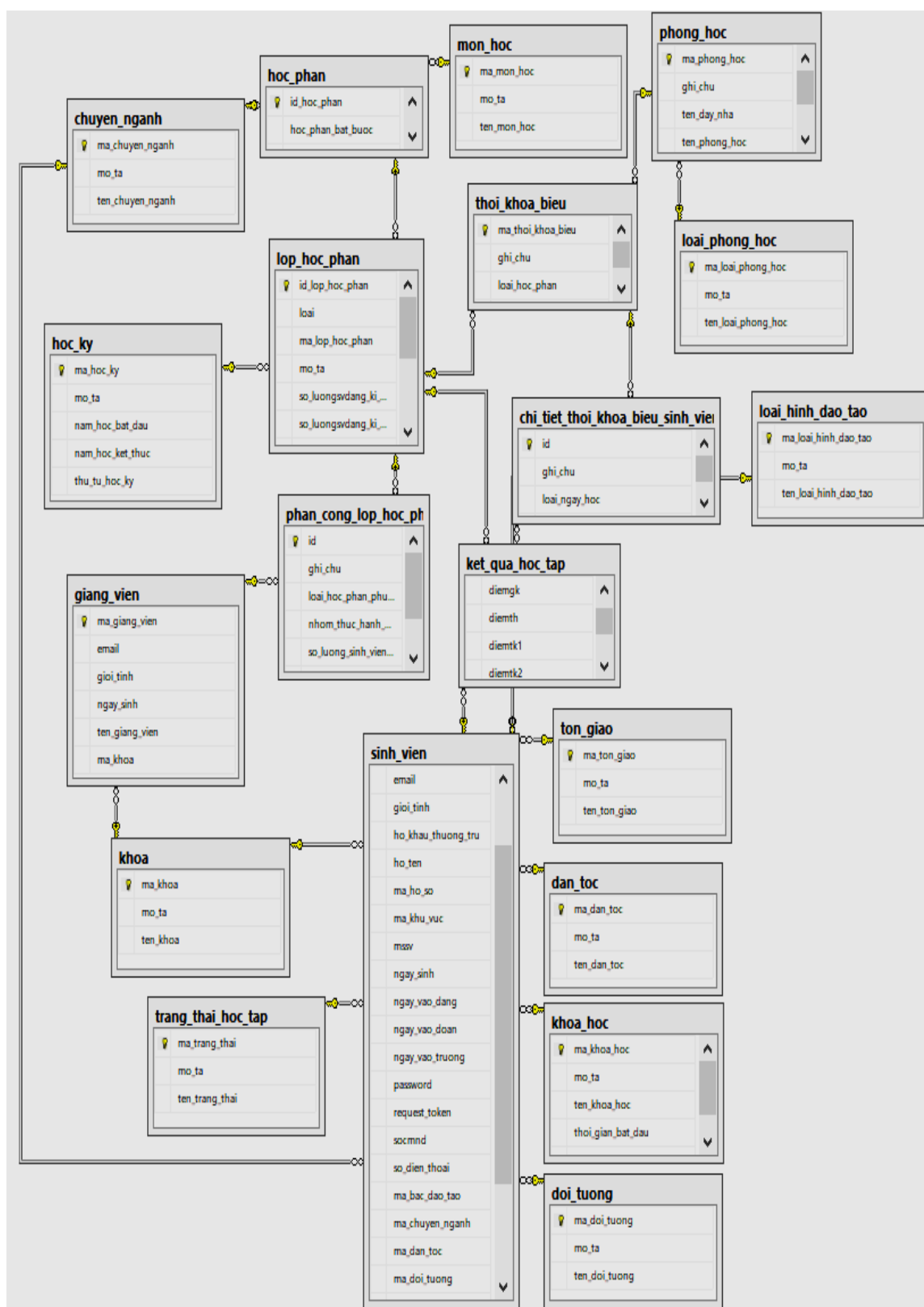
Từ những chức năng trên, chúng tôi tập trung vào các chức năng chính như: Đăng ký học phần, quản lý lớp học phần. Sau đó chúng tôi mở rộng các chức năng tương ứng như trong *Hình 3.3.1*.

Trong mô hình use-case chúng tôi trình bày, gồm có hai thành phần actor chính mô tả cho hai đối tượng người dùng mà chúng tôi hướng đến (sinh viên, nhân viên quản trị trường học). Trong đó, những use-case chính và phức tạp về nghiệp vụ đòi hỏi nhiều xử lý cho hệ thống đó là: use-case số *UC004-manage to sign up modules* và use-case *UC014-manage list of modules*. Ngoài ra, chúng tôi cũng mở rộng các use-case cho các chức năng của hệ thống nhằm tạo điều kiện thuận lợi cho người dùng là sinh viên và nhân viên quản trị. Tuy nhiên, những chức năng của các dịch vụ trên hệ thống điện toán đám mây như dịch vụ cân bằng (Load Balancer service), dịch vụ chứng thực (Authentication service) và các dịch vụ khác chúng tôi chưa đề cập ở đây. Chúng tôi sẽ trình bày các dịch vụ này trong phần *3.2 Kiến trúc của phần mềm*.



Hình 3.3.1 Danh sách các Use-case trong công cụ AcadIUH

Với danh sách các chức năng trên, chúng tôi xây dựng mô hình dữ liệu thể hiện toàn bộ cấu trúc cơ sở dữ liệu của AcadIUH (Hình 3.3.2). Trong đó, chúng tôi đặc biệt quan tâm đến quan hệ *sinh_vien*, khi số lượng thuộc tính khá nhiều và phức tạp. Đồng thời, quan hệ này cũng có số lượng mối kết hợp nhiều nhất. Việc xác định *sinh_vien* càng chi tiết như thế, giúp cho việc đặc tả sinh viên trong thế giới thực một cách rõ ràng và tường minh. Bên cạnh đó, quan hệ *lop_hoc_phan* cũng là quan hệ cần phải bận tâm. Đây là quan hệ mà các thao tác xử lý cho cả hai đối tượng sinh viên và quản trị viên đều thường xuyên truy xuất. Việc xác định các thuộc tính chính xác cũng góp phần cho nghiệp vụ xử lý dễ dàng hơn.

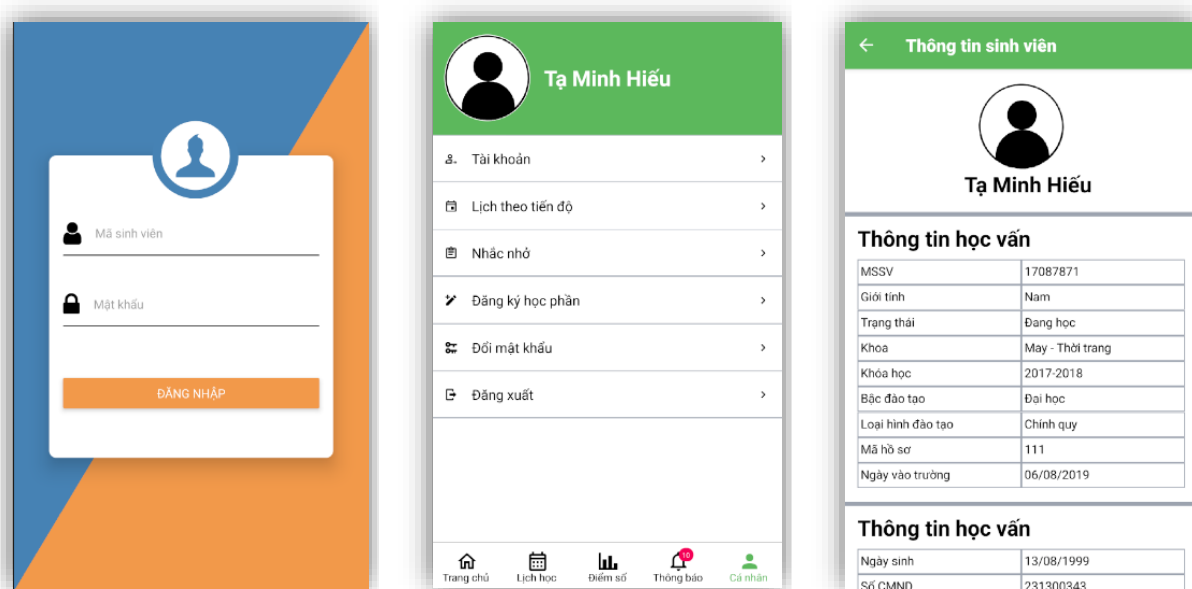


Hình 3.3.2 Lược đồ cơ sở dữ liệu trong công cụ AcadIUH

2.4 Giao diện đồ họa

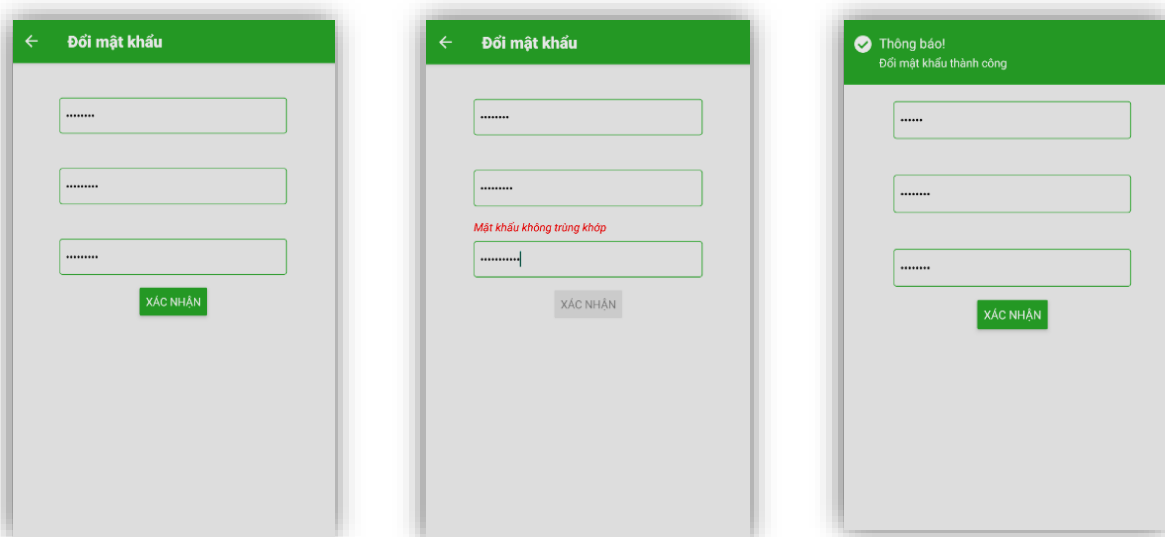
Công cụ AcadIUH triển khai trên hai nền tảng là web và app. Trong đó hệ thống app cho phép người dùng download và cài đặt trên các thiết bị chạy trên nền Android/iOS. Công cụ này sẽ có giao diện người dùng

thân thiết và đảm bảo tính ổn định trên các dòng điện thoại khác nhau. Chúng tôi cũng xin giới thiệu vài giao diện cho các chức năng trên ứng dụng trên. Đầu tiên, người dùng đăng nhập vào hệ thống, tài khoản sinh viên sử dụng là tài khoản do nhà trường cung cấp với tên đăng nhập ứng là mã số sinh viên và mật khẩu (Hình 3.4.1)



Hình 3.4.1 Giao diện đăng nhập của AcadIUH

Ở chức năng đổi mật khẩu, người dùng có thể đổi mật khẩu theo ràng buộc nhất định, thay vì sử dụng mật khẩu mặc định mà nhà trường đã cung cấp. Giao diện đổi mật khẩu bao gồm mật khẩu hiện tại, mật khẩu mới và xác nhận mật khẩu, người dùng phải điền đầy đủ thông tin ba trường này để đổi mật khẩu. Sau khi người dùng bấm đổi mật khẩu, ứng dụng sẽ hiển thị thông báo rõ ràng đổi mật khẩu thành công hay thất bại, và ghi rõ lý do nếu thất bại (Hình 3.4.2)



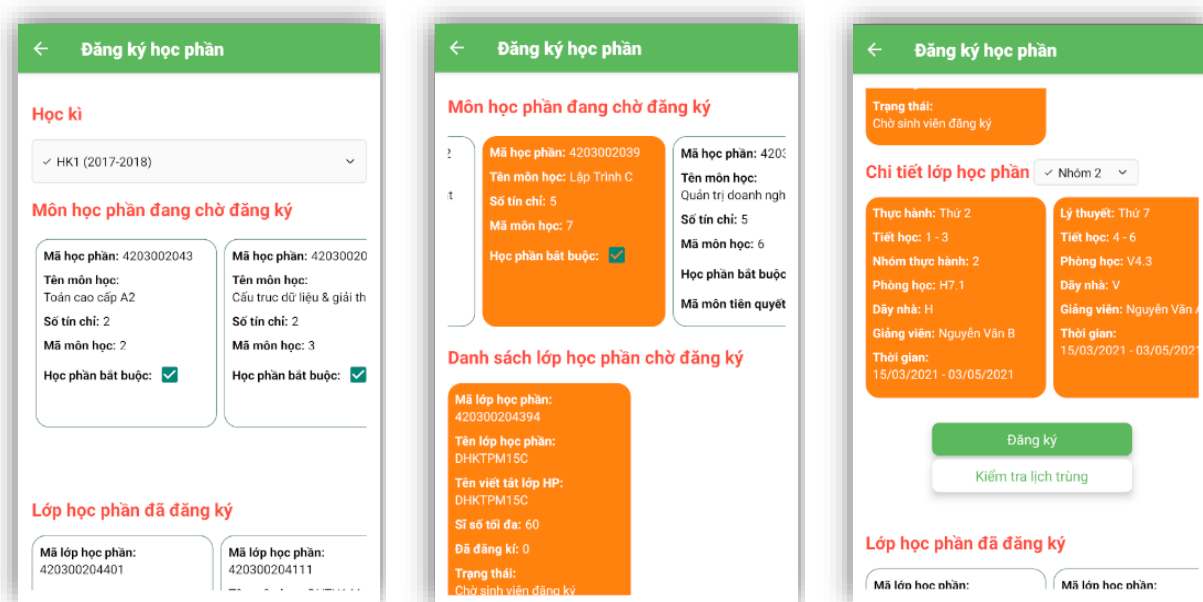
Hình 3.4.2 Giao diện thay đổi mật khẩu người dùng

Ở mục *Cá nhân*, người dùng có thể xem được các chức năng cơ bản một cách nhanh chóng như xem lịch, đăng ký học phần. Thêm vào đó, tính năng đăng ký học phần sẽ là chức năng chính cho hệ thống. Ta có thể

biết được chi tiết lớp, kiểm tra lịch trùng, kiểm tra môn đã đăng ký hay những lớp học phần được mở trong học kỳ đó. Để thực hiện chức năng này, sinh viên phải thực hiện trong 4 bước.

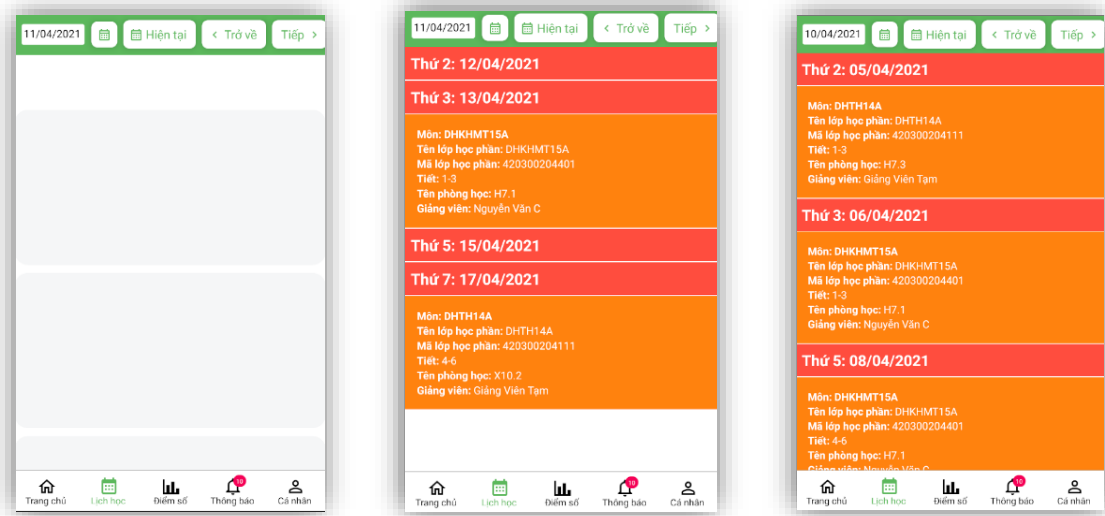
- **Bước một:** sinh viên phải chọn học kỳ cần đăng ký học phần, danh sách học kỳ hiển thị dưới dạng danh sách (dropdown). Ứng dụng sẽ tìm nạp và hiển thị những học phần sẽ mở lớp trong học kỳ đó.
- **Bước hai:** sinh viên phải chọn học phần muốn đăng ký. Sau khi chọn, ứng dụng sẽ tìm nạp và hiển thị danh sách những lớp học phần tương ứng với học phần đã chọn.
- **Bước ba:** sinh viên chọn lớp muốn đăng ký học. Ứng dụng sẽ tìm nạp và hiển thị chi tiết (thời khóa biểu, nhóm thực hành) của lớp học đó. Nếu đó là lớp có học phần thực hành, sinh viên phải chọn nhóm thực hành cho mình rồi mới có thể chọn nút đăng ký.
- **Bước cuối:** sinh viên nhấn nút đăng ký, ứng dụng sẽ thông báo rõ ràng đăng ký thành công hay thất bại, nếu thất bại thì sẽ nêu rõ lý do.

Ngoài ra, người dùng có thể kiểm tra xem lớp học phần và nhóm thực hành đã chọn có trùng với những môn học đã đăng ký trước đó hay không. Ứng dụng sẽ hiển thị danh sách các học phần bị trùng lịch học (Hình 3.4.3). Ở cuối giao diện là danh sách các lớp học phần đã đăng ký trong học kỳ này. Người dùng có thể chọn hủy lớp học phần nếu muốn. Khi hủy, ứng dụng sẽ thông báo thành công hay không và lý do đi kèm nếu thất bại (lớp học phần đã chấp nhận mở lớp, lớp học phần đã khóa...). Người dùng có thể chọn xem lịch học chi tiết của lớp học phần đã đăng ký đó. Hệ thống sẽ hiển thị chi tiết thời khóa biểu.



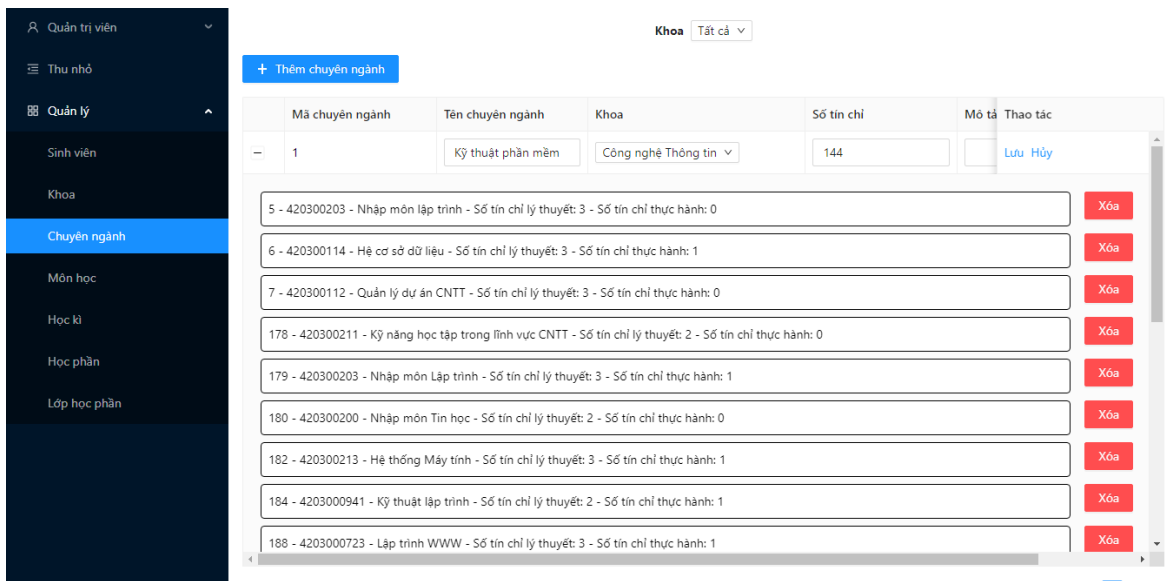
Hình 3.4.3 Quá trình đăng ký học phần trên công cụ AcadIUH

Tính năng xem lịch học theo tuần, có thể xem lịch học ở các tuần quá khứ, hiện tại, tương lai. Song với đó tính năng này sẽ thông báo người dùng mỗi khi có tiết, có môn sắp học trong ngày sau (hoặc tuần sau...). Cũng như việc thông báo lịch thi giữa kỳ, cuối kỳ.... Người dùng có thể click vào các link học online ở mỗi môn để được dẫn tới các app tương ứng trong điện thoại như Zoom, Google Meet, LMS của IUH. (Hình 3.4.4).



Hình 3.4.4 Màn hình xem thời khóa biểu và các thông báo cho các lịch học tiếp theo

Ngoài ra, các tính năng xử lý trên Web cho nhân viên đào tạo (Quản trị viên) chúng tôi cũng đã triển khai. Nhân viên đào tạo có thể đăng nhập vào hệ thống và thực hiện các chức năng như: Thêm các chương trình đào tạo, mở các học phần trong các học kỳ,... Bên cạnh đó, hệ thống cũng sẽ cho phép thực hiện các chức năng với ràng buộc về thời gian cũng như các ràng buộc nghiệp vụ khác trong công tác đào tạo tại trường (Hình 3.4.5).

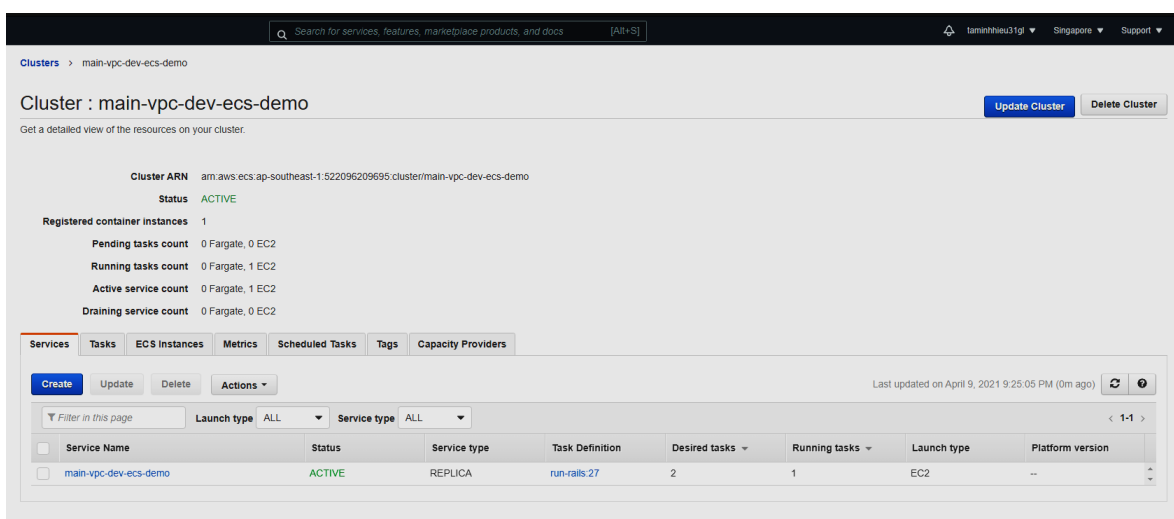


Hình 3.4.5 Màn hình hệ thống Web dành cho nhân viên đào tạo với các chức năng tương ứng

4. KIỂM NGHIỆM THỰC TẾ

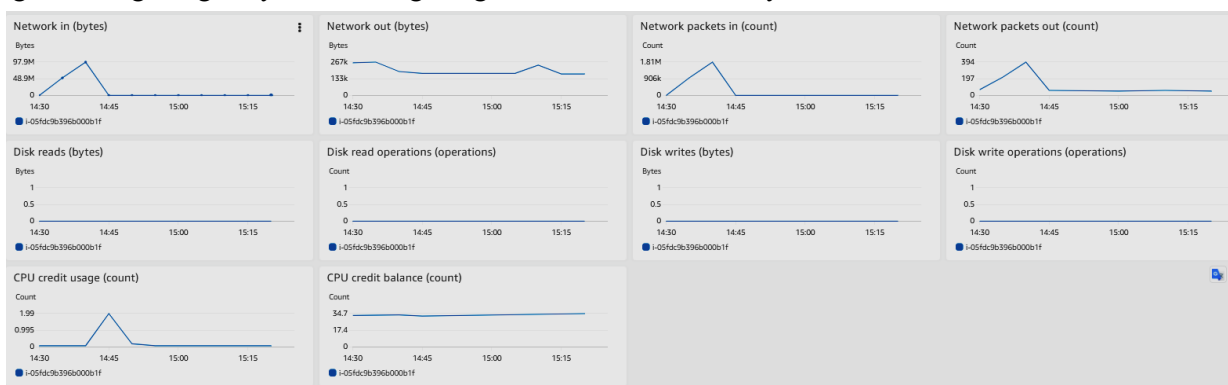
Chúng tôi đã tiến hành triển khai hệ công cụ AcadIUH trên hệ thống điện toán đám mây của AWS². Trong giai đoạn này chúng tôi sử dụng Infrastructure của AWS. Dịch vụ này gồm một ECS Cluster dùng để quản lý các instance (máy chủ). Mỗi service sẽ được xác định chạy các container như hình service type là EC2 (chúng tôi chọn hệ điều hành Linux, phiên bản t2.micro) như Hình 4.1

² <https://www.youtube.com/watch?v=PJLdOdJR69g&t=1s>



Hình 4.1 Cấu hình chi tiết của ECS

Các service được tạo ra hoạt động ổn định cho công cụ AcadiUH (Hình 4.2). Những thông số về tài nguyên sử dụng CPU (CPU credit balance) cũng như dung lượng đường truyền (Network packets) được đảm bảo trong các khoảng thời gian khi hệ thống tăng tốc xử lý (Bảng 4.1 mô tả ý nghĩa của các thông số). Tuy nhiên đây là phiên bản thử nghiệm nên sẽ dùng các service có hiệu suất thấp hơn phiên bản triển khai trong thực tế. Ứng dụng web đã triển khai trên AWS, đường dẫn truy cập https sẽ được xác định bởi nhà trường. Ngoài ra ứng dụng chạy trên di động cũng sẽ được đưa lên CHPlay với tên là AcadiUH.



Hình 4.2 Các thông số Mornitoring hệ thống khi triển khai AcadiUH

Bảng 4.1 Ý nghĩa các chỉ số trong monitoring

Network in (bytes)	Số byte mà ứng dụng sẽ nhận được trên tất cả các giao diện mạng. Số liệu này xác định khối lượng lưu lượng mạng đến cho một ứng dụng duy nhất. Đơn vị: Byte
Network out (bytes)	Số byte được gửi bởi ứng dụng trên tất cả các giao diện mạng. Số liệu này xác định khối lượng lưu lượng mạng đi từ một ứng dụng duy nhất. Đơn vị: Byte
Network packets in (count)	Số lượng gói tin mà ứng dụng nhận được trên tất cả các giao diện mạng. Số liệu này xác định khối lượng lưu lượng đến về số lượng gói trên một ứng dụng. Đơn vị: count
Disk reads (bytes)	Các byte được đọc từ tất cả các khối lượng lưu trữ ứng dụng có sẵn cho ứng dụng. Số liệu này được sử dụng để xác định khối lượng dữ liệu mà ứng dụng đọc từ đĩa cứng. Điều này có thể được sử dụng để xác định tốc độ của ứng dụng.

	dụng. Nếu không có khối lượng lưu trữ nào, thì giá trị là 0 hoặc số liệu không được báo cáo. Đơn vị: Byte
Disk, read operations (operations)	Các thao tác đọc đã hoàn thành từ tất cả các khối lượng lưu trữ ứng dụng có sẵn cho ứng dụng trong một khoảng thời gian cụ thể. Nếu không có khối lượng lưu trữ nào, thì giá trị là 0 hoặc số liệu không được báo cáo. Đơn vị: count
Disk writes (bytes)	Các byte được ghi vào tất cả các khối lượng lưu trữ có sẵn cho ứng dụng đó. Số liệu này được sử dụng để xác định khối lượng dữ liệu mà ứng dụng ghi vào đĩa cứng. Điều này có thể được sử dụng để xác định tốc độ của ứng dụng. Nếu không có khối lượng lưu trữ nào, thì giá trị là 0 hoặc số liệu không được báo cáo. Đơn vị: Byte
Disk, write operations (operations)	Các thao tác ghi đã hoàn thành vào tất cả khối lượng lưu trữ có sẵn cho ứng dụng trong một khoảng thời gian cụ thể. Nếu không có khối lượng lưu trữ nào, thì giá trị là 0 hoặc số liệu không được báo cáo. Đơn vị: count
CPU credit usage (count)	CPU Credit quy định số lượng CPU bùng nổ của một phiên bản. Có thể sử dụng CPU Credit này để tăng hiệu suất của CPU trong suốt thời gian Burst. Giả sử đang vận hành ứng dụng ở 100% hiệu suất CPU trong 5 phút, sẽ tiêu tốn 5 (tức là $5 * 1,0$) CPU Credit. Tương tự, nếu chạy một ứng dụng ở 50% hiệu suất CPU trong 5 phút, sẽ tiêu tốn 2,5 (tức là $5 * 0,5$) Tín dụng CPU.
CPU credit balance (count)	CPU Credit Balance là số lượng CPU Credit có sẵn trong tài khoản của bất kỳ lúc nào.

5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Bài báo này chúng tôi đã đưa ra hai giải pháp cho bài toán xây dựng công cụ AcadIUH nhằm hỗ trợ cho công việc học tập của sinh viên trường đại học Công nghiệp Tp.HCM. Trong đó, giải pháp đầu tiên chúng tôi đã triển khai ứng dụng chạy trên thiết bị di động có tên là AcadIUH. Ứng dụng này chạy trên cả hai nền tảng thông dụng là Android và iOS. Giải pháp thứ hai, chúng tôi cũng triển khai cho công cụ trên và hệ thống website trên nền tảng điện toán đám mây. Đồng thời cũng kiểm chứng được tính ổn định của hệ thống khi triển khai trên nền tảng này. Trong cả hai giải pháp trên chúng tôi đều áp dụng công nghệ phù hợp nhằm tối ưu tài nguyên cũng như đảm bảo được tính hiệu quả của hệ thống. Cụ thể là phần ứng dụng trên di động chúng tôi triển khai trên công nghệ như React Native tạo ứng dụng, ReactJS tạo website cho quản trị, Spring Boot tạo các hàm API trong kiến trúc microservice. Phần triển khai trên điện toán đám mây chúng tôi dùng Elastic Container Service (ECS); RDS; AutoScalling; Load Balancer; Elastic Container Registry (ECR). Tuy nhiên, công cụ vẫn còn một số hạn chế là quá trình chúng tôi thử nghiệm chỉ với số lượng truy xuất vừa và nhỏ. Hệ thống cần thêm việc thử nghiệm DOS server, tạo ra lượng truy cập ảo (có thể lên tới 100.000 request) để tiến hành monitoring, maintain nhằm đảm bảo tính mở rộng của hệ thống. Ngoài ra, các chức năng trên hệ thống web dành cho nhà quản trị vẫn còn hạn chế, chưa đáp ứng hết các chức năng hỗ trợ. Phần ứng dụng trên thiết bị di động cũng chưa được triển khai trên các store để hỗ trợ cho sinh viên sử dụng. Trong tương lai, chúng tôi sẽ dành tài nguyên và nhân lực để giải quyết các hạn chế trên.

REFERENCES

- [1] Liu, J., Ahmed, E., Shiraz, M., Gani, A., Buyya, R., & Qureshi, A. (2015). Application partitioning algorithms in mobile cloud computing: Taxonomy, review and future directions. *Journal of Network and Computer Applications*, 48, 99-117.
- [2] Kishore, P., & Mahendra, B. M. (2020). Evolution of Client-Side Rendering over Server-Side Rendering. *Recent Trends in Information Technology and its Application*, 3(2).
- [3] Balalaie, A., Heydarnoori, A., & Jamshidi, P. (2016). Microservices architecture enables devops: Migration to a cloud-native architecture. *IEEE Software*, 33(3), 42-52.
- [4] Philips, J. (2019). Workflow automation using Docker Swarm and GitLab CI at Flanders Institute for Biomechanical Experimentation. In proceeding of NL-RSE19 conference on Research Software Engineering, Amsterdam, Netherlands.
- [5] https://java.com/en/download/help/whatis_java.html, accessed 03/4/2021
- [6] <https://www.javatpoint.com/what-are-the-uses-of-javascript>, accessed 03/4/2021
- [7] Walls, C. (2015). *Spring Boot in action*. Simon and Schuster.
- [8] Danielsson, W. (2016). React Native application development. *Linköpings universitet*, Swedia, 10(4).
- [9] Rawat, P., & Mahajan, A. N. (2018) ReactJS: A Modern Web Development Framework.
- [10] <https://www.simform.com/best-front-end-frameworks/>, accessed 10/4/2021
- [11] <https://scout24.com/blog/nodejs-architecture-and-12-best-practices-for-nodejs-development> accessed 03/4/2021
- [12] Correia, R. C. M. (2020). Analysis of Node.js Application Performance Using MongoDB Drivers. *Information Technology and Systems: In Proceedings of ICITS 2020*, 1137, 213.
- [13] <https://aws.amazon.com/vi/getting-started/hands-on> accessed 05/4/2021
- [14] <https://docs.aws.amazon.com/whitepapers/latest/introduction-devops-aws/infrastructure-as-code.html> accessed 05/4/2021
- [15] <https://www.terraform.io/docs/cloud/> accessed 05/4/2021

Ngày nhận bài: 28/04/2021

Ngày chấp nhận đăng: 15/07/2021