NGHIÊN CỨU XÂY DỰNG API KẾT NỐI PHẦN MỀM UNETI-ONLINE TRÊN THIẾT BỊ DI ĐỘNG

RESEARCH TO BUILD UNETI-ONLINE SOFTWARE CONNECTION API ON MOBILE DEVICES

Đỗ Tuấn Hạnh[1], Nguyễn Văn Hòa[1], Trần Bảo Trung[2], Hà Đăng Huy[2]

Lê Công Minh[2], Phan Thanh Hải[2], Tống Bá Quang Anh[2], Chu Đăng Tuấn[2]

[1]Phòng Đào tạo, Trường Đại học Kinh tế - Kỹ thuật Công nghiệp

[2]Sinh viên Khoa Công nghệ thông tin, Trường Đại học Kinh tế - Kỹ thuật Công nghiệp

Đến Tòa soạn ngày 15/03/2023, chấp nhận đăng ngày 01/04/2023

Tóm tắt: Nghiên cứu API (giao diện lập trình ứng dụng) đã trở thành cốt lõi của giao tiếp liền mạch giữa các ứng dụng phần mềm. Với sự phụ thuộc ngày càng nhiều vào các API, điều quan trọng là phải tối ưu hóa hiệu suất của chúng và tránh thời gian phản hồi chậm có thể cản trở đáng kể trải nghiệm người dùng. Khi API ngày càng trở nên không thể thiếu đối với các hệ thống phần mềm hiện đại, phục vụ cho nhiều người dùng, việc duy trì hiệu suất tối ưu ngày càng trở nên cấp thiết. Sử dụng Nodejs kết hợp với ReactJS mô hình MVC và JWT Token để tiến hành xây dựng API giúp tăng hiệu suất sử dụng. Chúng tôi tiến hành thực nghiệm cho hơn 50000 sinh viên có thể gửi nhận các dữ liệu, đảm bảo hiệu suất tốt trên thiết bị di động với các chức năng nghiệp vụ của khảo thí một cửa. Đảm bảo cho việc sinh viên có thể giải quyết các thủ tục hành chính mà không phải trực tiếp đến phòng một cửa.

Từ khóa: API, MVC, JWT, ReactJS.

Abstract: Researching APIs (Application Programming Interfaces) has become the core of seamless communication between software applications. With a growing reliance on APIs, it's important to optimize their performance and avoid slow response times that can significantly hinder the user experience. As APIs become increasingly integral to modern software systems that cater to multiple users, maintaining optimal performance becomes increasingly imperative. Using Nodejs in combination with ReactJS MVC model and JWT Token to advance in building APIs to increase performance. We conduct experiments for more than 50,000 students who can send and receive data, ensuring good performance on mobile devices with the business functions of one-stop testing. Ensure that students can handle administrative procedures without having to go directly to the one-stop room.

Keywords: API, MVC, JWT, ReactJS.

1. GIỚI THIỆU

Giao diện lập trình ứng dụng (API) là một trong các cơ chế phức tạp tạo điều kiện cho sự tương tác và giao tiếp liền mạch giữa các hệ thống phần mềm riêng biệt bằng cách thiết lập một ngôn ngữ chung và một bộ giao thức. API trao quyền cho các nhà phát triển sử dụng các thành phần và dịch vụ dựng sẵn, từ đó đẩy nhanh quá trình phát triển phần mềm và thúc đẩy mô đun hóa. Chẳng hạn, hãy nghĩ đến việc kết hợp dịch vụ truyền phát nhạc của Spotify vào một ứng dụng dành cho thiết bị di động: API cho phép nhà phát triển sử dụng thư viện nhạc phong phú của Spotify mà không cần phải tạo nền tảng âm nhạc của riêng họ từ đầu. Bằng cách tận dụng API, các nhà phát triển có

thể khai thác hiệu quả các điểm mạnh của các hệ thống khác nhau như xây dựng MSTIL hỗ trợ trong học tập [2]. Sử dụng API viết giao tiếp với hệ thống điều khiển các thiết bị phần cứng giúp cho điều khiển hệ thống, việc thăm dò dầu khí thông qua hệ thống điều khiển các mũi khoan để nâng cao hiệu suất thăm dò [3].

Tình hình ứng dụng API không còn bó hẹp trong công nghệ thông tin. Với nhu cầu chuyển đổi số phát triển mạnh ở Việt Nam, việc xây dựng nghiên cứu ứng dụng API ở Việt Nam là vấn đề cấp thiết vì có thể sử dụng cho nhiều lĩnh vực, tuy còn rất nhiều hạn chế trong việc nghiên cứu ứng dụng, để tận dụng hiệu quả các dữ liệu dùng chung. Như trong khai thác dữ liệu bản đồ đã xây dựng API để có thể khai thác công nghệ ArcGIS API [1] có thể mang đến công dụng trong công tác quản lý tài nguyên.

API được phân loại theo cả kiến trúc và phạm vi sử dụng. Có 4 loại API [4]:

- Open APIs hoặc Public APIs (API mở): Còn có tên gọi khác là API công khai, có sẵn nên có thể được sử dụng bởi bất kỳ nhà phát triển nào. Đổi lại, các Open APIs thông thường sẽ yêu cầu các biện pháp xác thực hoặc ủy quyền thấp và bị hạn chế chức năng khi chia sẻ công khai. Một số Open APIs sẽ được chia sẻ miễn phí, một số khác sẽ yêu cầu tính phí khi sử dụng. Chi phí này thường được tính dựa trên số lượng "lệnh gọi" (calls) đến API được sử dụng.

- Partner APIs (API đối tác): API này cần có quyền hoặc giấy phép cụ thể mới truy cập được. Thường dành cho các nhà phát triển bên ngoài ủy quyền để hỗ trợ đầu mối hợp tác giữa doanh nghiệp với doanh nghiệp. Một số doanh nghiệp lựa chọn Partner APIs vì muốn kiểm soát tốt hơn người dùng có thể truy cập vào tài nguyên của họ và chỉ rõ cách thức sử dụng các tài nguyên đó.

- Internal APIs (API nội bộ): Không giống như API mở hay API đối tác, API nội bộ không dành cho các bên thứ ba sử dụng, thường dùng trong phạm vi công ty. Công ty sử dụng API này để kết nối các hệ thống cũng như dữ liệu nội bộ của công ty/tổ chức.

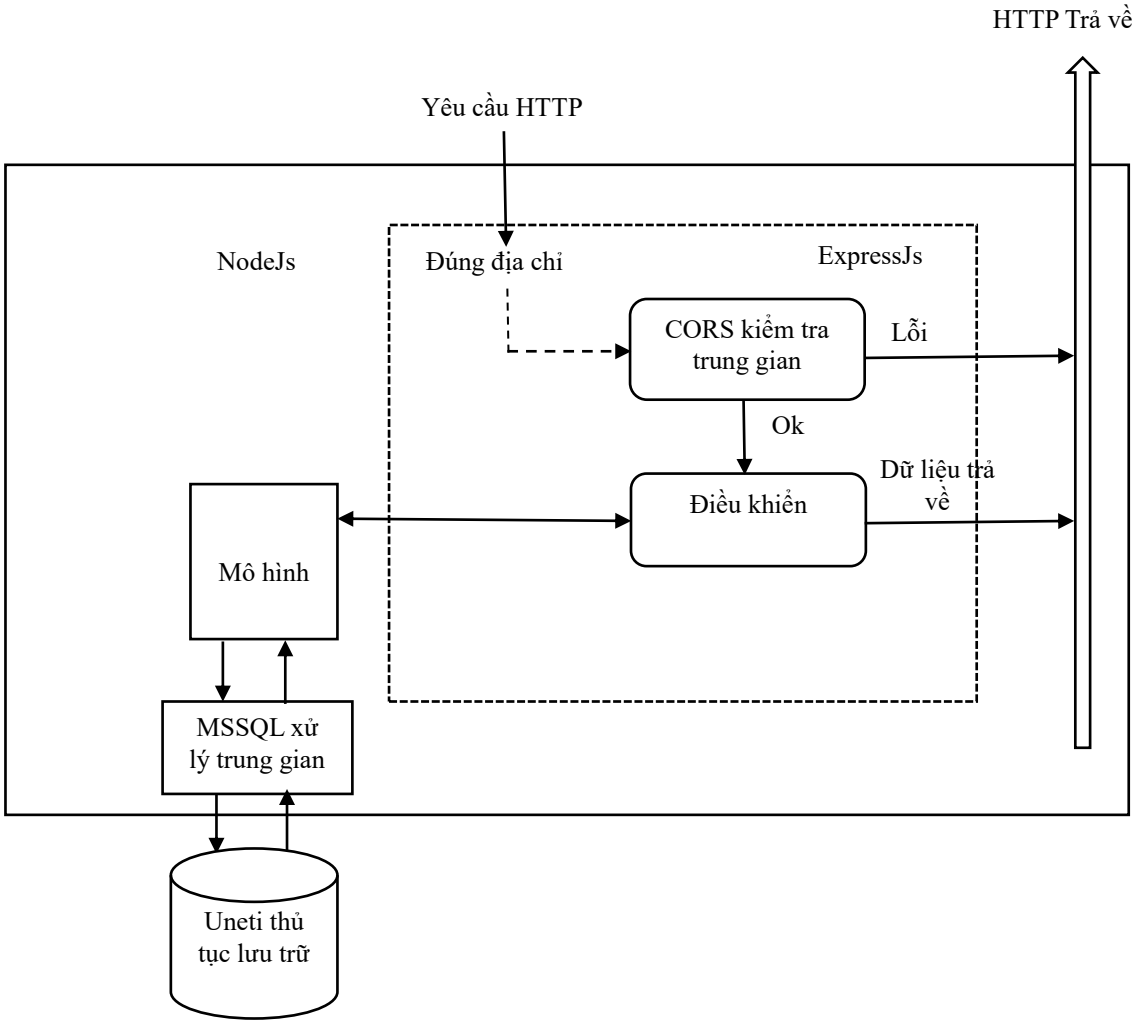
- Composite APIs (API tổng hợp): Kết hợp hai hay nhiều API khác nhau để giải quyết các yêu cầu phức tạp của hệ thống. Nếu cần dữ liệu từ các ứng dụng hoặc từ nhiều nguồn dữ liệu khác nhau, bạn nên sử dụng API tổng hợp. Ngoài ra, có thể sử dụng API tổng hợp để thiết lập một chuỗi các "lệnh gọi" (calls) và phản hồi tự động mà không cần chủ động can thiệp vào.

Để có thể dùng chung cơ sở dữ liệu phần mềm Uneti-Online trên PC chúng tôi sử dụng API nội bộ cho hệ thống của nhà trường.

2. DỮ LIỆU VÀ PHƯƠNG PHÁP NGHIÊN CỨU

2.1. Mô hình luồng dữ liệu API

Phần mềm Uneti-Online là hệ thống phần mềm quản lý hỗ trợ giải quyết các thủ tục một cửa cho tất cả sinh viên trong trường. Đây là hệ phần mềm nội bộ trên máy tính PC do các cán bộ một cửa của nhà trường quản lý. Phần mềm Uneti-Online sử dụng cơ sở dữ liệu là Uneti Store procedure. Để có thể sử dụng dữ liệu dùng chung khi thiết kế phần mềm để chạy trên thiết bị di động và không ảnh hưởng đến dữ liệu dùng chung. Yêu cầu đặt ra cần xây dựng API cho phép dùng chung cơ sở dữ liệu là Uneti Store procedure. Xây dựng mô hình luồng cho API hoạt động như sau:



Hình 1. Mô hình luồng dữ liệu API

Sau đây là các bước thực hiện quy trình sơ đồ luồng dữ liệu:

Bước 1: Người dùng gửi vào một HTTP Request bằng cách sử dụng đúng địa chỉ API có được từ server.

Bước 2: Server kiểm tra HTTP Request thông qua CORS Middleware.

Bước 3: Nếu CORS Middleware báo lỗi sẽ trả về Error trong HTTP Response cho người dùng. Nếu thỏa mãn yêu cầu của CORS HTTP Request sẽ được chuyển tiếp cho controller xử lý.

Bước 4: Controller điều hướng HTTP Request đến cho các Model chịu trách nhiệm cho đường dẫn API xử lý.

Bước 5: Model xử lý các thông tin nhận vào và chuyển cho MSSQL Middleware.

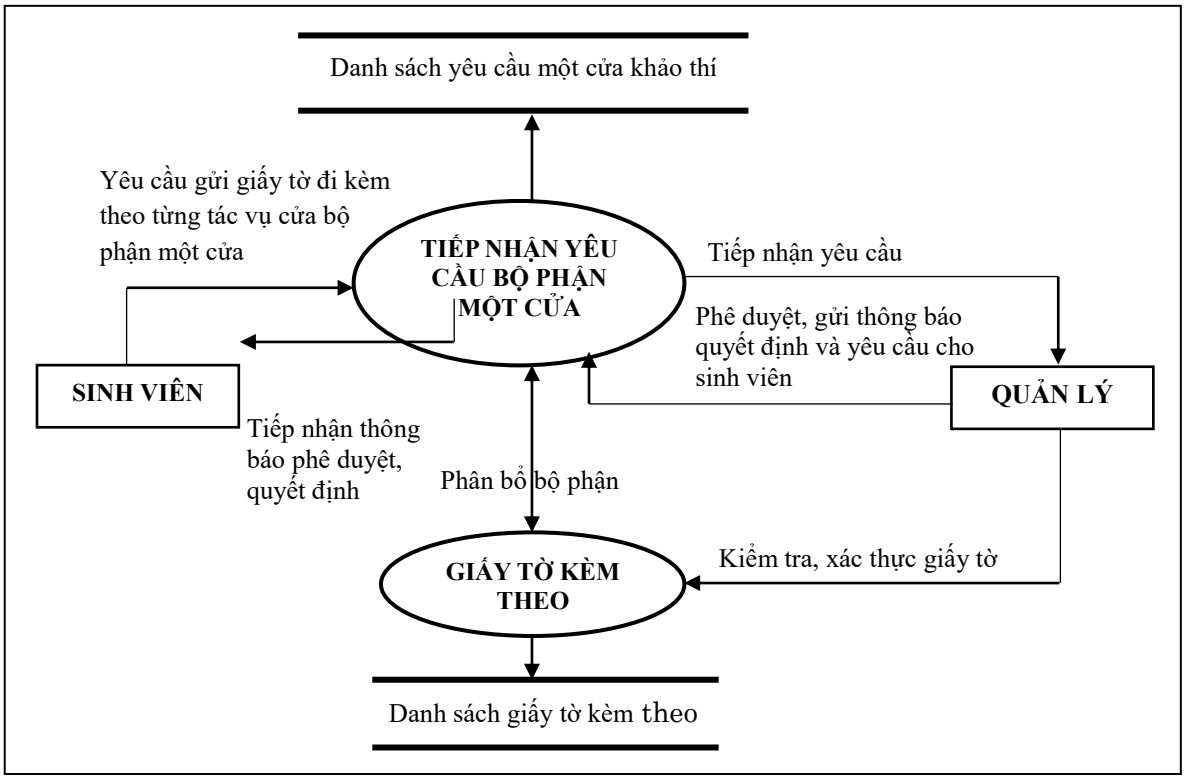
Bước 6: MSSQL Middleware sẽ làm việc trực tiếp với các Store Procedure với các thông tin nhận được từ Model và trả về kết quả lấy được từ Store Procedure.

Bước 7: Các dữ liệu nhận được từ Store Procedure sẽ được chuyển về Controller thông qua MSSQL Middleware và Model để trở thành Response Data.

Bước 8: Response Data sẽ được gửi trả cho người dùng dưới dạng HTTP Response. Khi đã có luồng dữ liệu hoạt động của API, cần xây dựng mô hình hệ thống phần mềm trên thiết bị di động.

2.2. Mô hình hệ thống phần mềm

Để tạo điều kiện giúp sinh viên giải quyết thủ tục hành chính một cửa một cách nhanh chóng kịp thời, chúng tôi tiến hành xây dựng hệ thống phần mềm Uneti-Online trên thiết bị di động. Hệ thống cung cấp công cụ cho sinh viên, nhà trường sử dụng công cụ trực tuyến qua môi trường internet thông qua hệ thống hồ sơ điện tử thay thế cho các hình thức hồ sơ giấy thông thường được thực hiện trực tiếp tại Bộ phận một cửa. Ngoài ra, nhằm nâng cao tính thuận tiện giúp tiết kiệm thời gian và chi phí cho sinh viên, nhà trường. Hệ thống di động giải quyết thủ tục hành chính còn cung cấp công cụ theo dõi về tiến trình, nhật ký hồ sơ, ghi nhận tất cả nội dung yêu cầu giúp cho sinh viên trực tiếp theo dõi, giám sát hồ sơ của mình, đồng thời qua đó tạo thành công cụ để theo dõi quá trình xử lý các thủ tục của Bộ phận một cửa.



Hình 2. Mô hình luồng dữ liệu Uneti-Online

Tiến trình: Tiếp nhận yêu cầu bộ phận một cửa, giấy tờ kèm theo.

Tác nhân: Quản lý, sinh viên.

Kho dữ liệu: Danh sách yêu cầu một cửa khảo thí.

Để có thể đánh giá hiệu năng của API cần sử dụng bộ công cụ kiểm thử.

2.3. Công cụ phân tích hiệu năng API

Postman là một công cụ nổi tiếng để thử nghiệm và đánh giá hiệu năng API. Nó có giao diện đồ họa dễ sử dụng và cho phép người dùng gửi các yêu cầu API, xem phản hồi của API. Postman có khả năng hỗ trợ tất cả các phương thức HTTP bao gồm POST, PUT, PATCH, DELETE hay GET. Không dừng lại ở đó, Postman còn thể hiện sự tiện lợi của mình khi chúng cho phép các lập trình viên thực hiện lưu lại lịch sử các lần request, là trợ thủ đắc lực khi có nhu cầu sử dụng lại [5].

Các tính năng mà postman mang lại khi sử dụng:

- Postman sử dụng Collections, vì vậy người dùng hoàn toàn có thể tạo ra bộ dữ liệu mẫu lưu trữ cho những lệnh gọi API. Ngoài ra, ở mỗi bộ dữ liệu mẫu lưu trữ, người dùng cũng có thể tạo thêm các thư mục con với nhiều request. Đây được đánh giá là điểm mạnh cực lớn giúp quá trình tổ chức các bộ thử nghiệm trở nên dễ dàng.
- Postman Collections và Environment cung cấp tính năng import hoặc export, từ đó người dùng sẽ có thể chia sẻ các tệp dữ liệu dễ dàng hơn. Không chỉ vậy, các liên kết trực tiếp cũng có thể được dùng với mục đích chia sẻ bộ dữ liệu mẫu lưu trữ.
- Postman mang khả năng vượt trội khi chúng có thể test trạng thái phản hồi của HTTP.
- Postman có thể hỗ trợ gỡ lỗi. Theo đó, nhờ bộ phận bảng điều khiển của Postman, người dùng sẽ dễ dàng kiểm tra các dữ liệu đã xuất. Nhờ vậy, quá trình gỡ lỗi sẽ trở nên linh hoạt hơn bao giờ hết.
- Postman hỗ trợ tạo thử nghiệm. Khi những điểm kiểm tra thử nghiệm cũng như xác định trạng thái phản hồi HTTP thành công, vai trò xác nhận sẽ có xu hướng thêm vào mỗi lệnh gọi API để đảm bảo tối đa phạm vi kiểm tra.
- Quá trình sử dụng newman và bộ dữ liệu mẫu lưu trữ sẽ đảm bảo rằng các kiểm thử có thể chạy được trong những lần lặp lại. Nhờ đó, người dùng sẽ tiết kiệm nhiều thời gian hơn cho các thử nghiệm có tính lặp cao.
- Postman có khả năng hỗ trợ tích hợp một cách liên tục cho các hoạt động phát triển.

2.4. Phương pháp xử lý số liệu

Kết quả thực nghiệm được phân tích qua số liệu công cụ Postman, để xác định độ ổn định dữ liệu và đo thời gian đáp ứng cho người dùng.

3. KẾT QUẢ VÀ THẢO LUẬN

3.1. Xây dựng Server API

Hệ thống server rest API là một máy chủ có hệ thống nhằm tiếp nhận, thông báo và xử lý các yêu cầu của sinh viên đến nhà trường thông qua các thiết bị di động hoặc website. Các thiết bị này được kết nối với server thông qua các API sinh ra từ server. Hệ thống server rest API phục vụ cho công tác sinh viên, yêu cầu ban đầu phải có các chức năng sau:

- Lấy thông tin cá nhân: Sinh viên có thể lấy được các thông tin cá nhân của mình.
- Yêu cầu miễn học, miễn thi tiếng Anh: Sinh viên có thể gửi yêu cầu xin miễn học, miễn thi học phần tiếng anh có kèm giấy tờ xác nhận thông qua các ứng dụng kết nối với API.
- Phúc khảo điểm: Sinh viên có thể gửi yêu cầu phúc khảo điểm của mình thông qua các ứng dụng kết nối với API tương ứng với các đợt đăng ký được cho phép.
- Xem lịch thi: Sinh viên có thể xem lịch thi.
- Đăng ký thi: Sinh viên có thể đăng ký thi lại các môn hiện đang có lịch thi tương ứng với học phần của bản thân thông qua ứng dụng.
- Hoãn thi: Sinh viên có thể gửi yêu cầu hoãn thi các môn.
- Hủy đăng ký thi lại: Sinh viên có thể gửi các yêu cầu hủy đăng ký lịch thi lại của các môn có trong học phần của mình.
- Yêu cầu liên quan đến kết quả học tập: Sinh viên có thể lấy thông tin về kết quả học tập của bản thân.
- Khóa các chức năng: Các chức năng có

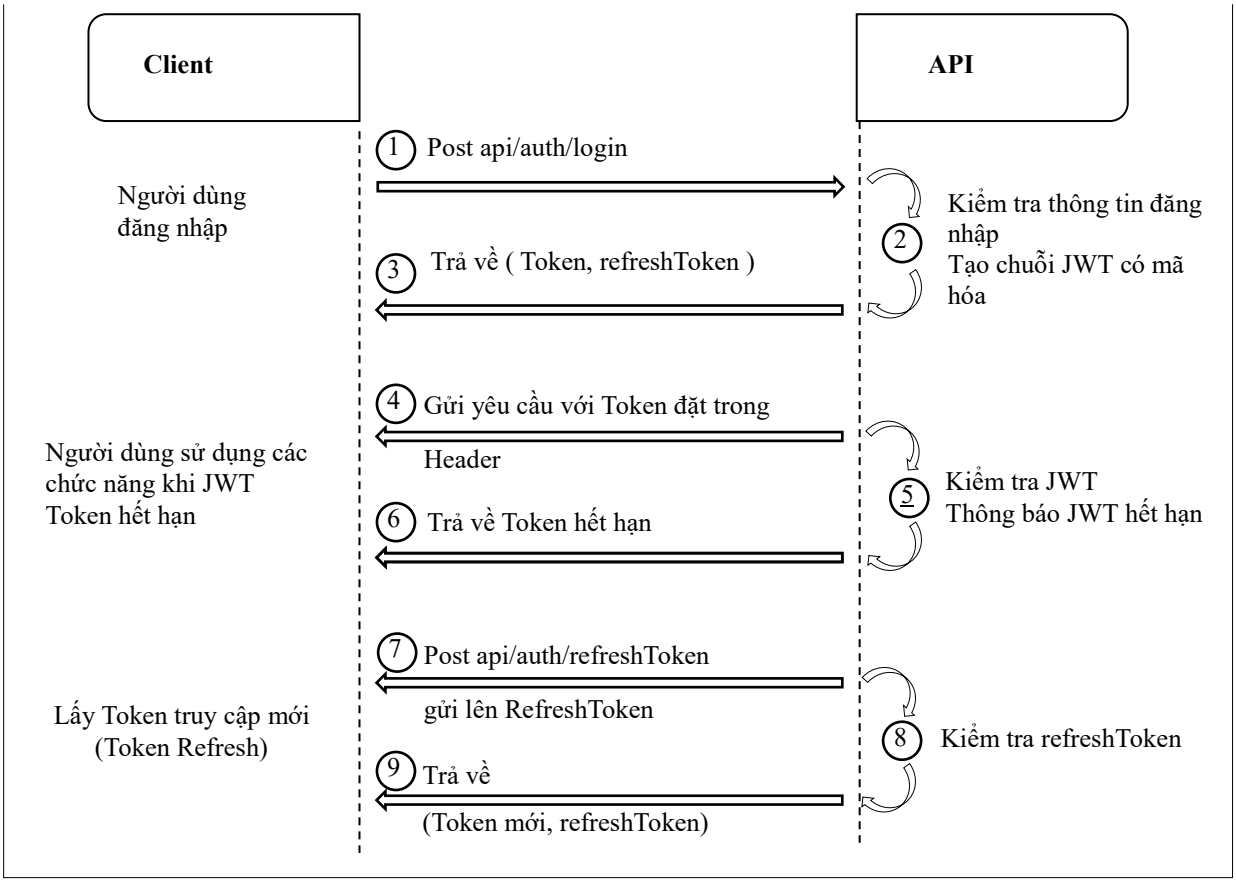
nhu cầu nâng cấp cần được tạm dừng hoạt động để tiến hành bảo trì.

- Đăng nhập, cấp quyền người dùng: Thiết lập hệ thống đăng nhập riêng sử dụng mã sinh viên và mật khẩu có sẵn để thực hiện. Sử dụng JWT Token để tăng tính bảo mật.
- Gửi thông báo: Thông báo được gửi đến cho sinh viên khi sinh viên thực hiện các yêu cầu hoặc các yêu cầu được xử lý hoàn thành.

Yếu tố quan trọng trong xây dựng API là phải đảm bảo tính bảo mật dữ liệu để tránh bị tấn công.

3.2. Xây dựng hệ thống bảo mật cho API.

Sử dụng JWT Token được thực hiện như mô hình sau:



Hình 3. Mô hình bảo mật API

Xây dựng cấu hình cơ sở dữ liệu MySQL và Sequelize sau đó định cấu hình khóa xác thực. Có được luồng dữ liệu hoạt động thông qua tệp tin "Auth.routes.js": để giúp POST đăng ký và đăng nhập, "user.routes.js" để nhận tài nguyên công cộng và được bảo mật. Tiếp theo xây dựng bộ đệm trung gian thông qua chức năng tệp tin sau "verifySignUp.js" để kiểm tra tên người dùng hoặc email trùng lặp, tệp tin "authJwt.js" dùng để xác minh mã thông báo, kiểm tra vai trò người dùng trong cơ sở dữ liệu. Xây dựng bộ điều khiển gồm tập tin "auth.controller.js" để xử lý các hành động đăng ký và đăng nhập. Tệp tin "user.controller.js" giúp việc trả lại nội dung công khai và được bảo vệ. Cần tạo mô hình Sequelize bằng cách xây dựng 2 tệp tin là "ngườidùng.model.js" và tệp tin "vai trò.model.js". Tiếp theo ta xây dựng tệp tin "server.js" dùng để nhập, khởi tạo các môđun

và tuyến đường cần thiết, cho tiếp nhận các kết nối. Để đảm bảo tính bảo mật, sẽ sử dụng bảo mật chuỗi kết nối. Trước hết sẽ sử dụng thuật toán mã hóa AES-256-CBC. Việc tiếp theo cần mã hóa chuỗi kết nối, bộ mã hóa này sẽ tiến hành mã hóa chuỗi kết nối với server. Sau khi mã hóa cần bộ giải mã chuỗi kết nối, bộ giải mã này sẽ tiến hành giải mã chuỗi kết nối đó với server thông qua một key mã hóa (chìa khóa mã hóa). Cuối cùng cần bảo mật code API và key mã hóa, giải mã chuỗi kết nối. Tiến hành dùng một bên thứ ba để bảo vệ code Api và key (mã hóa và giải mã), lúc này sẽ được bảo vệ trên GitLab. Vậy cần phải xây dựng một hệ thống GitLab, sau đó xây dựng một tài khoản git trên hệ thống GitLab đó. Khi đó toàn bộ code API được đẩy và bảo vệ bởi git. Vậy lúc này, key mã hóa sẽ được chuyển thành biến dưới dạng môi trường và được lưu, bảo vệ bởi git.

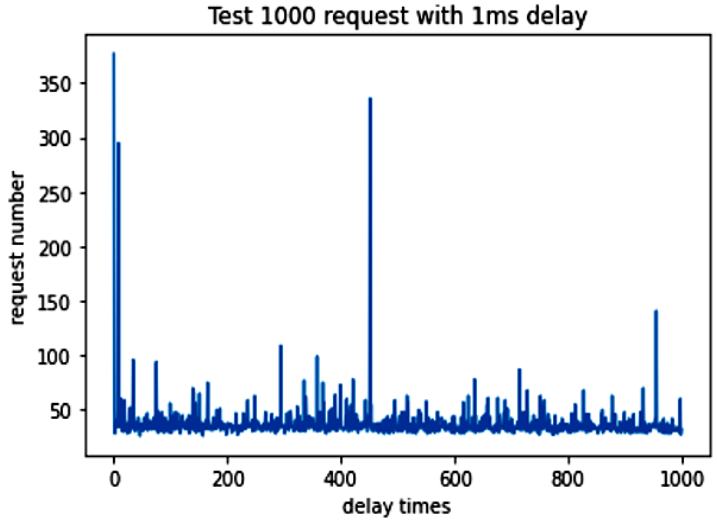
3.3. Đánh giá hiệu năng sử dụng API

Tiến hành chạy kiểm thử trên máy Lenovo Workstation ThinkStation P520 cấu hình sau đây:

- CPU Intel Xeon W-2255 Processor (3.70 GHz up to 4.50 GHz).
- Ram: 32 GB DDR4-2933 MHz.
- Ổ cứng: SSD 256 GB SSD Pcle và HDD 2TB SATA.
- - Card màn hình: NVIDIA GeForce RTX 3080 Ti 12GB GĐR6X, K+M USD, DOS, BT+Wifi, 3Y Onsite & KYD – 30BFSJT100.

Thử nghiệm đầu tiên tiến hành cho 1000 lượt truy cập với độ trễ của mỗi lượt truy cập 1 ms. Thời gian duy trì giá trị và hiệu lực của Token: 10 h.

Source	Environment	Iterations	Duration	All tests	Avg. Resp. Time
Runner	none	1000	2m 18s	0	37 ms

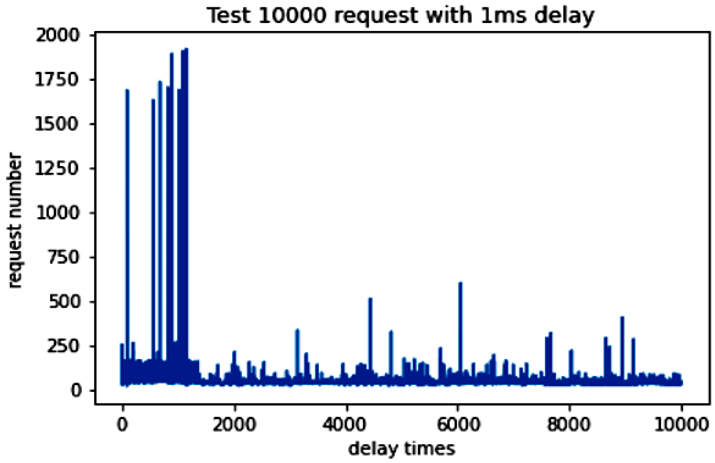


Hình 4. Kết quả chạy 1000 lượt truy cập chụp từ postman

Kết quả chạy 1000 lượt truy cập mất tầm hơn 2m cho tất cả dữ liệu đến, thời gian trung bình để hoàn thành một lượt truy cập để gửi và nhận mất 37 ms.

Thử nghiệm tiếp theo tiến hành cho 10000 lượt truy cập với độ trễ của mỗi lượt truy cập 1 ms. Thời gian duy trì giá trị và hiệu lực của Token: 10 h.

Source	Environment	Iterations	Duration	All tests	Avg. Resp. Time
Runner	none	10000	23m 17s	0	35 ms

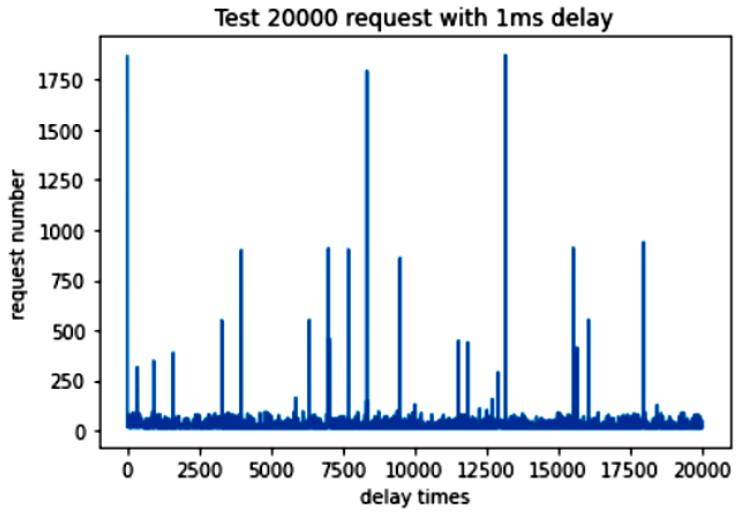


Hình 5. Kết quả chạy 10000 lượt truy cập chụp từ postman

Kết quả chạy 10000 lượt truy cập mất tầm hơn 23 m cho tất cả dữ liệu đến, thời gian trung bình để hoàn thành một lượt truy cập để gửi và nhận mất 35 ms.

Thử nghiệm tiến hành cho 20000 lượt truy cập với độ trễ của mỗi lượt truy cập 1 ms. Thời gian duy trì giá trị và hiệu lực của Token: 10 h.

Source	Environment	Iterations	Duration	All tests	Avg. Resp. Time
Runner	none	20000	41m 20s	0	22 ms

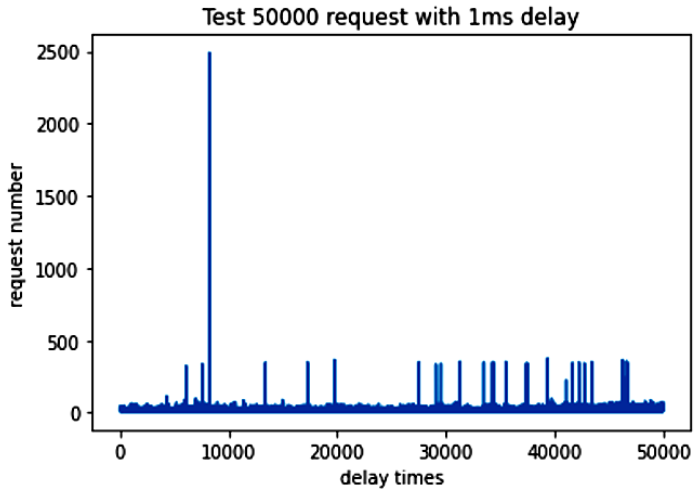


Hình 6. Kết quả chạy 20000 lượt truy cập chụp từ postman

Kết quả chạy 20000 lượt truy cập mất tầm hơn 41 m cho tất cả dữ liệu đến, thời gian trung bình để hoàn thành một lượt truy cập để gửi và nhận mất 22 ms. Thử nghiệm tiến hành cho 50000 lượt truy cập với độ trễ của mỗi lượt truy cập 1 ms. Thời gian duy trì giá trị và hiệu lực của Token: 10 h.

Kết quả chạy 50000 lượt truy cập mất tầm hơn gần 2 h cho tất cả dữ liệu đến, thời gian trung bình để hoàn thành một lượt truy cập để gửi và nhận mất 17 ms.

Source	Environment	Iterations	Duration	All tests	Avg. Resp. Time
Runner	none	50000	1h 98m	0	17 ms



Hình 7. Kết quả chạy 50000 lượt truy cập chụp từ postman

4. KẾT LUẬN VÀ KIẾN NGHỊ

Từ những kết quả thu được trong quá trình nghiên cứu xây dựng API kết nối phần mềm Uneti-Online trên thiết bị di động, hệ thống đã đáp ứng cho 50000 sinh viên cùng lúc truy cập và trải nghiệm. Để có thể hoạt động tốt hơn, cần nâng cấp phần cứng tăng bộ nhớ đệm cấp ứng dụng, có thể giảm đáng kể thời gian phản hồi bằng cách giảm thiểu việc truy xuất và xử lý dữ liệu dư thừa. Hiện tại có tầm 20000 sinh viên, nhà trường đáp ứng tốt nhu cầu hoạt động. Ngoài ra để tăng hiệu suất tốt hơn cần tiếp tục nghiên cứu công nghệ mới và có thể cải tiến các giải thuật lưu trữ hệ thống cơ sở dữ liệu.

TÀI LIỆU THAM KHẢO

[1] Trần Xuân Miên, Nguyễn Thế Công, Phạm Thị Kim Hoa, Nguyễn Thị Dung, Đặng Thị Hoàng Nga, Ứng dụng Arcgis API xây dựng cơ sở dữ liệu thống kê, kiểm kê đất đai tại Huyện Phú Lương, Tỉnh Thái Nguyên, Tạp chí Khoa học đo đạc và Bản đồ, số 51-3/2022.

[2] Rong Qin, Zeyu Wang, Sheng Huang, Luwen Huangfu, *MSTIL: Multi-cue Shape-aware Transferable Imbalance Learning for effective graphic API recommendation*, Journal of Systems and Software, Volume 200, June 2023.

[3] Gianmario Sorrentino, Giovanna Biscontin, *Improved filtration parameters from modified API filter press,* Geoenergy Science and Engineering, Volume 224, May 2023.

[4] https://aws.amazon.com/vi/what-is/api/

[5] https://www.postman.com/product/api-repository/

Thông tin liên hệ: **Đỗ Tuấn Hạnh**

Điện thoại: 0913009379 - Email: dthanh@uneti.edu.vn

Phòng Đào tạo, Trường Đại học Kinh tế - Kỹ thuật Công nghiệp.

-
-