

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC QUẢN LÝ VÀ CÔNG NGHỆ HẢI PHÒNG



ĐỒ ÁN TỐT NGHIỆP
NGÀNH CÔNG NGHỆ THÔNG TIN

Sinh viên: Vũ Thượng Khanh

Giảng viên hướng dẫn: ThS. Nguyễn Thị Thanh Toan

HẢI PHÒNG – 2025

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC QUẢN LÝ VÀ CÔNG NGHỆ HẢI PHÒNG

**XÂY DỰNG PHẦN MỀM QUẢN LÝ TRUNG TÂM TỔ
CHỨC DẠY HỌC**

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY
NGÀNH CÔNG NGHỆ THÔNG TIN

Sinh viên: Vũ Thượng Khanh

Giảng viên hướng dẫn: ThS. Nguyễn Thị Thanh Thoan

HẢI PHÒNG – 2025

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐH QUẢN LÝ VÀ CÔNG NGHỆ HẢI PHÒNG

NHIỆM VỤ ĐỀ TÀI TỐT NGHIỆP

Sinh viên: Vũ Thượng Khanh – **Mã SV:** 2212111014

Lớp : CT2601

Ngành : Công nghệ thông tin

Tên đề tài : Xây dựng phần mềm quản lý trung tâm tổ chức dạy học

NHIỆM VỤ ĐỀ TÀI

1. Nội dung và các yêu cầu cần giải quyết trong nhiệm vụ đề tài tốt nghiệp

a. Mô tả tóm tắt đề tài

- Tìm hiểu hệ thống phần mềm quản lý trung tâm tổ chức dạy học, đưa lên thành website.
- Tìm hiểu n8n tự động hóa gửi tin nhắn Zalo
- Tìm kiếm và xây dựng hệ thống xử lý.

b. Nội dung hướng dẫn

- Tìm hiểu tổng quan về phần mềm quản lý trung tâm tổ chức dạy học
- Tìm hiểu tổng quan về n8n
- Tìm hiểu các ứng dụng xung quanh đề tài

c. Kết quả cần đạt được

- Xây dựng thành công hệ thống quản lý trung tâm tổ chức dạy học, sử dụng được và mang tính chất chuyên nghiệp
- Tìm hiểu nghiệp vụ và xây dựng được các báo cáo theo các đơn vị phòng ban, các cấp bậc trong hệ thống quản lý.
- Cài đặt và xây dựng hệ thống báo cáo phân tích:
 - o Kết nối được với cơ sở dữ liệu
 - o Xây dựng được trang website hoàn chỉnh
- Báo cáo và trình bày kết quả

2. Các tài liệu, số liệu cần thiết

- [AI Workflow Automation Platform & Tools - n8n](#)

3. Địa điểm thực tập tốt nghiệp.

- Công ty cổ phần công nghệ tin học thương mại tân phong

CÁN BỘ HƯỚNG DẪN ĐỀ TÀI TỐT NGHIỆP

Họ và tên : Nguyễn Thị Thanh Thoan

Học hàm, học vị : Thạc sĩ

Cơ quan công tác : Trường TH – THCS – THPT Vinschool Hải Phòng

Nội dung hướng dẫn:

- Tìm hiểu hệ thống phần mềm quản lý trung tâm tổ chức dạy học, đưa lên thành website.
- Tìm hiểu n8n tự động hóa gửi tin nhắn Zalo
- Tìm kiếm và xây dựng hệ thống xử lý.

Kết quả cần đạt được

- Xây dựng thành công hệ thống quản lý trung tâm tổ chức dạy học, sử dụng được và mang tính chất chuyên nghiệp
- Tìm hiểu nghiệp vụ và xây dựng được các báo cáo theo các đơn vị phòng ban, các cấp bậc trong hệ thống quản lý.
- Cài đặt và xây dựng hệ thống báo cáo phân tích:
 - o Kết nối được với cơ sở dữ liệu
 - o Xây dựng được trang website hoàn chỉnh
- Báo cáo và trình bày kết quả

Đề tài tốt nghiệp được giao ngày 06 tháng 10 năm 2025

Yêu cầu phải hoàn thành xong trước ngày 27 tháng 12 năm 2025

Đã nhận nhiệm vụ ĐTTN

Sinh viên

Đã giao nhiệm vụ ĐTTN

Giảng viên hướng dẫn

Hải Phòng, ngày.....tháng 12 năm 2025

TRƯỞNG KHOA

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập - Tự do - Hạnh phúc

PHIẾU NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN TỐT NGHIỆP

Họ và tên giảng viên : ThS.Nguyễn Thị Thanh Thoan

Đơn vị công tác : Trường TH – THCS – THPT Vinschool Hải Phòng

Họ và tên sinh viên : Vũ Thượng Khanh

Chuyên Ngành : Công nghệ Thông tin

Nội dung hướng dẫn:

Hướng dẫn sinh viên khảo sát thực tế, phân tích yêu cầu bài toán; thiết kế và xây dựng phần mềm quản lý trung tâm tổ chức dạy học trên nền tảng web; triển khai các chức năng chính, xử lý dữ liệu, kiểm thử hệ thống; hoàn thiện báo cáo và chuẩn bị bảo vệ đồ án.

1. Tinh thần thái độ của sinh viên trong quá trình làm đề tài tốt nghiệp

Sinh viên Vũ Thượng Khanh có tinh thần học tập nghiêm túc, thái độ làm việc tích cực và trách nhiệm trong suốt quá trình thực hiện đề tài. Chủ động nghiên cứu tài liệu, khảo sát nghiệp vụ thực tế, triển khai nội dung đồ án đầy đủ theo đúng hướng dẫn; tiếp thu tốt các ý kiến góp ý của giảng viên và hoàn thành đồ án đúng tiến độ.

2. Đánh giá chất lượng của đồ án/khóa luận (so với nội dung yêu cầu đó đề ra trong nhiệm vụ Đ.T.T.N trên các mặt lý luận, thực tiễn, tính toán số liệu...)

Đồ án “*Xây dựng phần mềm quản lý trung tâm tổ chức dạy học*” đáp ứng tốt các nội dung và yêu cầu đã đề ra trong nhiệm vụ đề tài tốt nghiệp.

Về **mặt lý luận**, sinh viên trình bày đầy đủ và hệ thống cơ sở lý thuyết liên quan đến phát triển hệ thống web, mô hình Client–Server, công nghệ ReactJS, Node.js và cơ sở dữ liệu MySQL, phù hợp với hướng nghiên cứu của đề tài.

Về **mặt thực tiễn**, đề tài xuất phát từ nhu cầu quản lý thực tế tại các trung tâm dạy học, phân tích nghiệp vụ rõ ràng, đề xuất giải pháp hợp lý và có tính ứng dụng cao. Hệ thống được thiết kế đầy đủ các chức năng quản lý học viên, giáo viên, lớp học, điểm danh, chấm công và báo cáo thống kê, đáp ứng yêu cầu quản lý tập trung và tự động hóa quy trình.

Về **xử lý dữ liệu và kỹ thuật**, đồ án xây dựng được cơ sở dữ liệu chặt chẽ, thiết kế các bảng dữ liệu hợp lý, mô tả rõ luồng xử lý và các chức năng nghiệp

vụ; các chức năng chấm công, điểm danh và thống kê được triển khai logic, đáp ứng yêu cầu bài toán.

Nhìn chung, đồ án được thực hiện nghiêm túc, có tính khoa học, tính thực tiễn và khả năng triển khai, **đạt yêu cầu của đồ án tốt nghiệp ngành Công nghệ Thông tin.**

3. Ý kiến của giảng viên hướng dẫn tốt nghiệp

Được bảo vệ Không được bảo vệ Điểm hướng dẫn

Hải Phòng, ngày.....tháng năm 2025

Giảng viên hướng dẫn

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập - Tự do - Hạnh phúc

PHIẾU NHẬN XÉT CỦA GIÁNG VIÊN CHẤM PHẢN BIỆN

Họ và tên giảng viên :

Đơn vị công tác : Trường Đại học quản lý và công nghệ Hải Phòng

Họ và tên sinh viên : Vũ Thượng Khanh- **Chuyên ngành:** Công Nghệ Thông Tin

Đề tài tốt nghiệp : Xây dựng phần mềm quản lý trung tâm tổ chức dạy học

1. Phần nhận xét của giảng viên chấm phản biện

.....
.....
.....

2. Những mặt còn hạn chế

.....
.....
.....

3. Ý kiến của giảng viên chấm phản biện

.....

.....

.....

Được bảo vệ Không được bảo vệ Điểm phản biện

Hải Phòng, ngày.....tháng.....năm 2025

Giảng viên chấm phản biện

(ký và ghi rõ họ tên)

LỜI CẢM ƠN

Đồ án tốt nghiệp chuyên ngành Công nghệ thông tin với đề tài “Xây dựng phần mềm quản lý trung tâm tổ chức dạy học” là kết quả của quá trình nỗ lực không ngừng của bản thân em trong suốt thời gian học tập và nghiên cứu, cùng với sự hướng dẫn tận tình của quý thầy cô, sự giúp đỡ và động viên của gia đình, bạn bè và các anh chị.

Trước hết, em xin bày tỏ lòng kính trọng và biết ơn sâu sắc tới cô Nguyễn Thị Thanh Thoan , người đã trực tiếp hướng dẫn, tận tình chỉ bảo, định hướng và giúp em khắc phục những thiếu sót trong quá trình thực hiện đồ án. Những góp ý quý báu của thầy là nguồn động lực lớn giúp em hoàn thiện đề tài này.

Em cũng xin gửi lời cảm ơn chân thành tới Ban Giám hiệu Trường Đại học Quản lý và Công nghệ Hải Phòng, cùng Ban Lãnh đạo Khoa Công nghệ thông tin đã tạo mọi điều kiện thuận lợi về cơ sở vật chất và môi trường học tập để em có thể hoàn thành tốt đồ án tốt nghiệp của mình.

Cuối cùng, em xin chân thành cảm ơn gia đình, bạn bè và các anh chị đã luôn ở bên động viên, giúp đỡ em trong suốt quá trình học tập và thực hiện đề tài.

Em xin chân thành cảm ơn!

Hải Phòng, ngày.....tháng.....năm 2025

Sinh viên

Vũ Thượng Khanh

LỜI CAM ĐOAN

Tôi xin cam đoan rằng đề tài “Xây dựng phần mềm quản lý trung tâm tổ chức dạy học” được thực hiện một cách minh bạch, trung thực và công khai. Toàn bộ nội dung và kết quả trong đề án là kết quả của quá trình nghiên cứu, tìm hiểu và nỗ lực của bản thân, dưới sự hướng dẫn tận tình của thầy cô hướng dẫn cùng sự hỗ trợ từ bạn bè và gia đình.

Tôi xin cam đoan rằng các kết quả và nội dung trình bày trong đề án là trung thực, không sao chép hay sử dụng kết quả của bất kỳ công trình nghiên cứu nào khác.

Nếu phát hiện có sự sao chép hoặc gian lận học thuật, tôi xin chịu hoàn toàn trách nhiệm trước Nhà trường.

Hải Phòng, ngày.....tháng.....năm 2025

Sinh viên

Vũ Thượng Khanh

Mục Lục

LỜI CẢM ƠN	10
LỜI CAM ĐOAN.....	11
DANH MỤC CÁC TỪ VIẾT TẮT.....	15
LỜI MỞ ĐẦU	19
CHƯƠNG I: KHẢO SÁT HIỆN TRẠNG VÀ PHÁT BIỂU BÀI TOÁN	20
1.1. Tổng quan về bài toán và đối tượng khảo sát.....	20
1.2. Khảo sát quy trình nghiệp vụ hiện tại.....	20
1.2.1. Quy trình quản lý học viên và đăng ký khóa học	20
1.2.2. Quy trình xếp lớp và quản lý lịch học	20
1.2.3. Quy trình chăm sóc khách hàng và thông báo.....	21
1.3. Đánh giá ưu nhược điểm của hiện trạng.....	21
1.3.1. Ưu điểm	21
1.3.2. Nhược điểm và thách thức.....	21
1.4. Phát biểu bài toán và đề xuất giải pháp	22
1.4.1. Mục tiêu cốt lõi của hệ thống	22
1.4.2. Yêu cầu chức năng (Functional Requirements).....	22
1.4.3. Yêu cầu về công nghệ.....	23
1.5. Kết luận chương	23
CHƯƠNG II: PHÂN TÍCH THIẾT KẾ HỆ THỐNG	24
2.1. Kết luận chương	24
2.1.1. Yêu cầu chức năng.....	24
2.1.2. Yêu cầu phi chức năng	26
2.2. Tác nhân Hệ Thống	27
2.3. Biểu đồ Use Case.....	28
2.3.1. Usecase tổng quát	28
2.3.2. Biểu đồ Usecase Quản lý lớp học.....	29
2.3.3. Biểu đồ Usecase Quản lý điểm danh.....	29
2.3.4. Biểu đồ Usecase Quản lý user	30

2.3.5.	Biểu đồ Usecase Quản lý Giáo viên	30
2.3.6.	Biểu đồ Usecase Quản lý Học viên	31
2.3.7.	Biểu đồ Usecase Quản lý Chăm công	31
2.3.8.	Biểu đồ Usecase Quản lý Báo cáo & thống kê.....	32
2.4.	Sơ Đồ UseCase Tuần Tự	32
2.4.1.	UC Quản lý lớp học.....	32
2.4.2.	UC Giáo viên thực hiện điểm danh	39
2.4.3.	UC Quản lý người dùng.....	40
2.4.4.	UC Quản lý giáo viên	46
2.4.5.	UC Giáo viên thực hiện chăm công.....	53
2.5.	Thiết kế Cơ sở dữ liệu	54
2.5.1.	Sơ đồ thực thể liên kết (ERD - Mô tả sơ bộ).....	54
2.5.2.	Mô tả chi tiết các bảng dữ liệu (Data Dictionary)	55
CHƯƠNG III: CƠ SỞ LÝ THUYẾT		61
3.1.	Mô hình Client – Server trong phát triển web	61
3.1.1.	Khái niệm về mô hình Client – Server	62
3.1.2.	Vai trò và chức năng của Client	62
3.1.3.	Vai trò và chức năng của Server.....	63
3.1.4.	Nguyên lý hoạt động.....	64
3.1.5.	Ưu điểm của mô hình Client – Server	64
3.2.	Frontend.....	65
3.2.1.	React.js.....	65
3.2.2.	React Router	67
3.2.3.	Context API	68
3.3.	Backend	70
3.3.1.	Node.js.....	70
3.3.2.	Express.js	71
3.3.3.	Mô hình kiến trúc Backend	72
3.3.4.	Middleware	74

3.3.5.	Kết nối Frontend – Backend	76
3.4.	Cơ sở dữ liệu MySQL	77
3.4.1.	Giới thiệu MySQL	77
3.4.2.	Đặc điểm nổi bật	78
3.4.3.	Vai trò của MySQL trong dự án	78
3.5.	Kiểm thử	78
3.5.1.	Giới thiệu	78
3.5.2.	Kiểm thử hộp đen (Black Box Testing)	79
CHƯƠNG IV: Giao Diện Trang Web		81
4.1.	Giao diện website	81
4.1.1.	Trang login	81
4.1.2.	Trang Home	82
4.1.3.	Trang quản lý users	82
4.1.4.	Trang quản lý giảng viên	83
4.1.5.	Trang quản lý hồ sơ học sinh	83
4.1.6.	Trang quản lý lớp học	84
4.1.7.	Trang chấm công của giảng viên	84
4.1.8.	Trang điểm danh	85
4.1.9.	Trang thời khóa biểu	85
4.2.	Yêu cầu cài đặt	86
4.2.1.	Yêu cầu phần cứng	86
4.2.2.	Yêu cầu phần mềm:	86
4.2.3.	Yêu cầu mạng Internet	88
CHƯƠNG V: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN		89
5.1.	Kết quả đạt được	89
5.2.	Hạn chế	90
5.3.	Hướng phát triển	91
TÀI LIỆU THAM KHẢO		92

DANH MỤC CÁC TỪ VIẾT TẮT

STT	Từ viết tắt	Nguyên nghĩa (Tiếng Anh / Tiếng Việt)	Nghĩa Tiếng Việt / Giải thích
1	API	Application Programming Interface	Giao diện lập trình ứng dụng
2	CNTT	Công nghệ thông tin	Ngành Công nghệ thông tin
3	CORS	Cross-Origin Resource Sharing	Chia sẻ tài nguyên chéo nguồn
4	CRUD	Create, Read, Update, Delete	Tạo, Đọc, Cập nhật, Xóa (4 thao tác cơ bản của CSDL)
5	CSDL	Cơ sở dữ liệu	Hệ thống lưu trữ dữ liệu
6	DOM	Document Object Model	Mô hình đối tượng tài liệu
7	ERD	Entity Relationship Diagram	Sơ đồ thực thể liên kết

8	FK	Foreign Key	Khóa ngoại (trong CSDL)
9	FR	Functional Requirements	Yêu cầu chức năng
10	GVHD	Giảng viên hướng dẫn	Giảng viên hướng dẫn đồ án
11	HTTP	HyperText Transfer Protocol	Giao thức truyền tải siêu văn bản
12	JSON	JavaScript Object Notation	Định dạng đối tượng JavaScript (dùng để trao đổi dữ liệu)
13	JSX	JavaScript XML	Cú pháp mở rộng của JavaScript
14	JWT	JSON Web Token	Mã thông báo web JSON (dùng để xác thực)
15	MSV	Mã sinh viên	Mã số sinh viên
16	MVC	Model - View - Controller	Mô hình kiến trúc phần mềm 3 lớp

17	NFR	Non-Functional Requirements	Yêu cầu phi chức năng
18	ORM	Object-Relational Mapping	Ánh xạ quan hệ đối tượng
19	PK	Primary Key	Khóa chính (trong CSDL)
20	RDBMS	Relational Database Management System	Hệ quản trị cơ sở dữ liệu quan hệ
21	SPA	Single Page Application	Ứng dụng đơn trang
22	SQL	Structured Query Language	Ngôn ngữ truy vấn có cấu trúc
23	UC	Use Case	Trường hợp sử dụng
24	UI	User Interface	Giao diện người dùng
25	URL	Uniform Resource Locator	Định vị tài nguyên thống nhất (Đường dẫn web)

26	XSS	Cross-Site Scripting	Tấn công chèn mã độc vào trang web
----	-----	----------------------	------------------------------------

LỜI MỞ ĐẦU

Trong kỷ nguyên chuyên đổi số, việc ứng dụng công nghệ thông tin vào quản lý giáo dục không còn là lựa chọn mà là xu thế tất yếu. Tại các trung tâm tổ chức dạy học quy mô vừa và nhỏ, công tác quản lý hiện nay phần lớn vẫn thực hiện thủ công qua sổ sách hoặc các công cụ rời rạc như Excel. Phương thức này không chỉ tốn kém thời gian, nhân lực mà còn tiềm ẩn nhiều sai sót trong việc xếp lớp, theo dõi học phí và đặc biệt là thiếu sự kết nối tức thời giữa trung tâm và phụ huynh.

Xuất phát từ nhu cầu thực tiễn đó, em đã lựa chọn và thực hiện đề tài: “Xây dựng phần mềm quản lý trung tâm tổ chức dạy học”.

Mục tiêu cốt lõi của đề tài là xây dựng một hệ thống website quản lý tập trung, toàn diện với các chức năng: quản lý hồ sơ học viên - giáo viên, sắp xếp thời khóa biểu tự động, điểm danh và chấm công. Điểm nổi bật của hệ thống là khả năng ứng dụng công nghệ hiện đại (ReactJS, Node.js, MySQL) để tối ưu hóa hiệu năng, đồng thời tích hợp tính năng gửi thông báo tự động qua Zalo, giúp việc tương tác giữa nhà trường và gia đình trở nên chuyên nghiệp, nhanh chóng và hiệu quả hơn.

Đề án là kết quả của quá trình nghiên cứu lý thuyết và vận dụng vào thực tế, nhằm tạo ra một sản phẩm công nghệ có tính ứng dụng cao, giải quyết triệt để bài toán quản lý đang tồn tại ở các trung tâm hiện nay.

CHƯƠNG I: KHẢO SÁT HIỆN TRẠNG VÀ PHÁT BIỂU BÀI TOÁN

1.1. Tổng quan về bài toán và đối tượng khảo sát

Trong bối cảnh giáo dục ngày càng phát triển, các trung tâm tổ chức dạy học đóng vai trò quan trọng trong việc cung cấp tri thức và kỹ năng. Tuy nhiên, tại nhiều trung tâm quy mô vừa và nhỏ, việc quản lý vẫn đang thực hiện theo phương thức truyền thống hoặc sử dụng các công cụ rời rạc.

Đối tượng khảo sát của đề tài là quy trình vận hành tại một trung tâm tổ chức dạy học tiêu chuẩn, bao gồm các hoạt động chính: quản lý hồ sơ học viên, giảng viên, sắp xếp lịch học, quản lý học phí và báo cáo thống kê.

1.2. Khảo sát quy trình nghiệp vụ hiện tại

Hiện tại, công tác quản lý tại trung tâm chủ yếu dựa trên phương pháp thủ công hoặc sử dụng các công cụ văn phòng cơ bản (Microsoft Excel, sổ sách giấy). Quy trình hoạt động cụ thể như sau:

1.2.1. Quy trình quản lý học viên và đăng ký khóa học

- Tiếp nhận: Nhân viên tư vấn ghi chép thông tin học viên vào sổ hoặc nhập liệu vào file Excel rời rạc.
- Lưu trữ: Hồ sơ học viên được lưu trong các file folder trên máy tính cục bộ, khó tra cứu nhanh lịch sử học tập.
- Nhược điểm: Dữ liệu không đồng bộ, dễ thất lạc thông tin liên hệ, khó khăn trong việc theo dõi tiến trình học tập của học viên.

1.2.2. Quy trình xếp lớp và quản lý lịch học

- Sắp xếp: Cán bộ quản lý phải kiểm tra thủ công lịch rảnh của giáo viên và phòng học để xếp lịch.
- Thông báo: Việc thông báo lịch học, lịch nghỉ hoặc thay đổi phòng học được thực hiện qua gọi điện thoại hoặc nhắn tin thủ công cho từng người.

- Nhược điểm: Tốn nhiều thời gian, dễ xảy ra tình trạng trùng lịch (trùng phòng, trùng giáo viên) và sai sót tạo nên áp lực lớn trong quản lý.

1.2.3. Quy trình chăm sóc khách hàng và thông báo

- Hiện trạng: Việc gửi thông báo nhắc lịch học, chúc mừng sinh nhật, hoặc nhắc đóng học phí được thực hiện thủ công bởi nhân viên.
- Nhược điểm: Tốn nhân lực, không đảm bảo tính tức thời, dễ bỏ sót học viên. Chưa ứng dụng được các công cụ tự động hóa hiện đại.

1.3. Đánh giá ưu nhược điểm của hiện trạng

Qua quá trình khảo sát, có thể rút ra các đánh giá sau về phương thức quản lý hiện tại:

1.3.1. Ưu điểm

- Chi phí đầu tư ban đầu thấp, không cần hạ tầng phức tạp.
- Linh hoạt trong các tình huống phát sinh đơn giản, quy mô nhỏ.

1.3.2. Nhược điểm và thách thức

- Tốn thời gian và nhân lực: Các thao tác thủ công lặp đi lặp lại chiếm dụng thời gian làm việc của nhân viên.
- Dễ sai sót: Việc nhập liệu và tính toán thủ công thường xuyên xảy ra lỗi, đặc biệt là trong quản lý tài chính và xếp lịch.
- Thiếu tính chuyên nghiệp: Khó khăn trong việc tạo ra các báo cáo phân tích theo phòng ban, cấp bậc một cách nhanh chóng và chính xác.
- Khó mở rộng: Khi số lượng học viên tăng lên, phương pháp thủ công sẽ quá tải và không thể đáp ứng.

1.4. Phát biểu bài toán và đề xuất giải pháp

Từ những bất cập trên, yêu cầu cấp thiết là phải xây dựng hệ thống website quản lý trung tâm tổ chức dạy học. Hệ thống mới cần giải quyết được bài toán tự động hóa quy trình, quản lý tập trung và nâng cao trải nghiệm người dùng.

1.4.1. Mục tiêu cốt lõi của hệ thống

- Quản lý tập trung: Đưa toàn bộ dữ liệu (Học viên, Giáo viên, Lớp học, Khóa học) vào một cơ sở dữ liệu thống nhất, giúp dễ dàng tra cứu và quản lý.
- Tự động hóa quy trình nghiệp vụ: Giảm thiểu thao tác tay trong việc xếp lớp, đăng ký học và quản lý trạng thái lớp học.
- Hỗ trợ theo dõi: Giúp cả học viên lẫn giáo viên có thể chủ động theo dõi tiến trình học tập và lịch dạy.
- Báo cáo chuyên nghiệp: Xây dựng hệ thống báo cáo phân tích theo các đơn vị phòng ban và các cấp bậc, hỗ trợ người quản lý nắm bắt tình hình hoạt động nhanh chóng.

1.4.2. Yêu cầu chức năng (Functional Requirements)

Dựa trên nhiệm vụ đề tài đã được giao, hệ thống cần đáp ứng các chức năng sau:

- Nhóm chức năng Quản trị hệ thống (Admin):
 - Quản lý người dùng (Thêm, sửa, xóa, phân quyền: Admin, Giáo viên, Học viên).
 - Cấu hình hệ thống.
- Nhóm chức năng Quản lý Đào tạo:
 - Quản lý danh sách khóa học, môn học.
 - Quản lý lớp học và xếp thời khóa biểu.
 - Quản lý hồ sơ giáo viên và học viên.

- Nhóm chức năng Báo cáo - Thống kê:
 - Xây dựng hệ thống báo cáo phân tích kết nối với cơ sở dữ liệu.
 - Báo cáo doanh thu, số lượng học viên, tình trạng lớp học.

1.4.3. Yêu cầu về công nghệ

Hệ thống được xây dựng trên nền tảng Web với mô hình Client – Server:

- Frontend (Client): Sử dụng ReactJS để xây dựng giao diện người dùng thân thiện, tương tác mượt mà.
- Backend (Server): Sử dụng Node.js và Express.js để xây dựng API xử lý logic nghiệp vụ.
- Cơ sở dữ liệu: Sử dụng MySQL để lưu trữ và truy vấn dữ liệu có cấu trúc, đảm bảo tính toàn vẹn dữ liệu.

1.5. Kết luận chương

Việc chuyển đổi từ mô hình quản lý thủ công sang hệ thống phần mềm website là giải pháp tối ưu để nâng cao hiệu quả hoạt động, giảm thiểu sai sót và mang lại sự chuyên nghiệp cho trung tâm dạy học. Chương tiếp theo sẽ đi sâu vào phân tích và thiết kế chi tiết hệ thống này.

CHƯƠNG II: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

2.1. Kết luận chương

Phần này mô tả các yêu cầu cần có của Hệ thống Quản lý Trung tâm Giáo dục, bao gồm các yêu cầu về chức năng và các yêu cầu phi chức năng.

2.1.1. Yêu cầu chức năng

Đây là các chức năng cụ thể mà hệ thống phải cung cấp cho người dùng.

- FR1: Quản lý Xác thực và Phân quyền
 - Hệ thống phải cho phép người dùng đăng nhập bằng tên người dùng và mật khẩu.
 - Hệ thống phải xác thực thông tin đăng nhập và phân quyền truy cập dựa trên vai trò của người dùng (Quản trị viên, Giáo viên).
 - Hệ thống phải cho phép người dùng đăng xuất khỏi tài khoản.
- FR2: Quản lý Người dùng (Dành cho Quản trị viên)
 - Hệ thống phải cho phép Quản trị viên tạo, xem, cập nhật và xóa tài khoản người dùng trong hệ thống.
 - Hệ thống phải cho phép Quản trị viên gán vai trò (quyền hạn) cho từng tài khoản.
- FR3: Quản lý Giáo viên (Dành cho Quản trị viên)
 - Hệ thống phải cho phép Quản trị viên tạo, xem, cập nhật và xóa hồ sơ giáo viên (bao gồm thông tin cá nhân, liên lạc, chuyên môn).
- FR4: Quản lý Học viên (Dành cho Quản trị viên)
 - Hệ thống phải cho phép Quản trị viên tạo, xem, cập nhật và xóa hồ sơ học viên.

- Hệ thống phải cho phép lưu trữ thông tin chi tiết của học viên (họ tên, ngày sinh, thông tin liên lạc của phụ huynh...).
- FR5: Quản lý Lớp học (Dành cho Quản trị viên)
- Hệ thống phải cho phép Quản trị viên tạo, xem, cập nhật và xóa thông tin lớp học (tên lớp, lịch học, phòng học).
 - Hệ thống phải cho phép Quản trị viên chỉ định giáo viên phụ trách cho một lớp học.
 - Hệ thống phải cho phép Quản trị viên thêm hoặc xóa học viên ra khỏi một lớp học.
- FR6: Quản lý Điểm danh
- Hệ thống phải cho phép Giáo viên thực hiện điểm danh cho các học viên trong lớp mình phụ trách (có mặt, vắng mặt, đi trễ).
 - Hệ thống phải cho phép Quản trị viên xem và chỉnh sửa lại lịch sử điểm danh của bất kỳ lớp học nào.
- FR7: Quản lý Chấm công
- Hệ thống phải cho phép Giáo viên/Nhân viên tự ghi nhận thời gian làm việc (ví dụ: check-in, check-out).
 - Hệ thống phải cho phép Quản trị viên xem, quản lý và xác nhận bảng chấm công của tất cả nhân viên.
- FR8: Báo cáo và Thống kê (Dành cho Quản trị viên)
- Hệ thống phải có khả năng xuất các báo cáo về:
 - Tình hình điểm danh của một lớp hoặc một học viên cụ thể.
 - Danh sách các lớp học và sĩ số.
 - Bảng chấm công của giáo viên.

2.1.2. Yêu cầu phi chức năng

Đây là các tiêu chí về chất lượng, hiệu suất và các ràng buộc của hệ thống.

– NFR1: Hiệu suất (Performance)

- Thời gian phản hồi của các chức năng chính (tải trang, truy vấn dữ liệu) không quá 3 giây.
- Hệ thống có khả năng phục vụ đồng thời ít nhất 50 người dùng mà không bị suy giảm hiệu suất rõ rệt.

– NFR2: Bảo mật (Security)

- Mật khẩu của người dùng phải được mã hóa một chiều (hashing) trước khi lưu vào cơ sở dữ liệu.
- Hệ thống phải được bảo vệ khỏi các lỗ hổng bảo mật web phổ biến như SQL Injection, Cross-Site Scripting (XSS).
- Tất cả các API yêu cầu nghiệp vụ phải được bảo vệ và yêu cầu xác thực hợp lệ (ví dụ: JWT).

– NFR3: Tính khả dụng (Availability)

- Hệ thống phải hoạt động ổn định và sẵn sàng 24/7, với thời gian hoạt động (uptime) đạt 99.5%.
- Hệ thống cần có cơ chế sao lưu dữ liệu định kỳ để phòng trường hợp có sự cố.

– NFR4: Tính dễ sử dụng (Usability)

- Giao diện người dùng phải được thiết kế trực quan, thân thiện và dễ sử dụng.
- Hệ thống cần có sự nhất quán trong thiết kế giao diện giữa các trang.

- Hệ thống phải hiển thị tương thích trên các trình duyệt web phổ biến (Chrome, Firefox, Edge).
- NFR5: Khả năng bảo trì (Maintainability)
- Mã nguồn phải được tổ chức rõ ràng, tuân thủ các quy tắc lập trình (coding convention) và có tài liệu/chú thích ở những phần logic phức tạp.
 - Kiến trúc hệ thống (frontend, backend) cần được phân tách rõ ràng để dễ dàng nâng cấp, sửa lỗi.

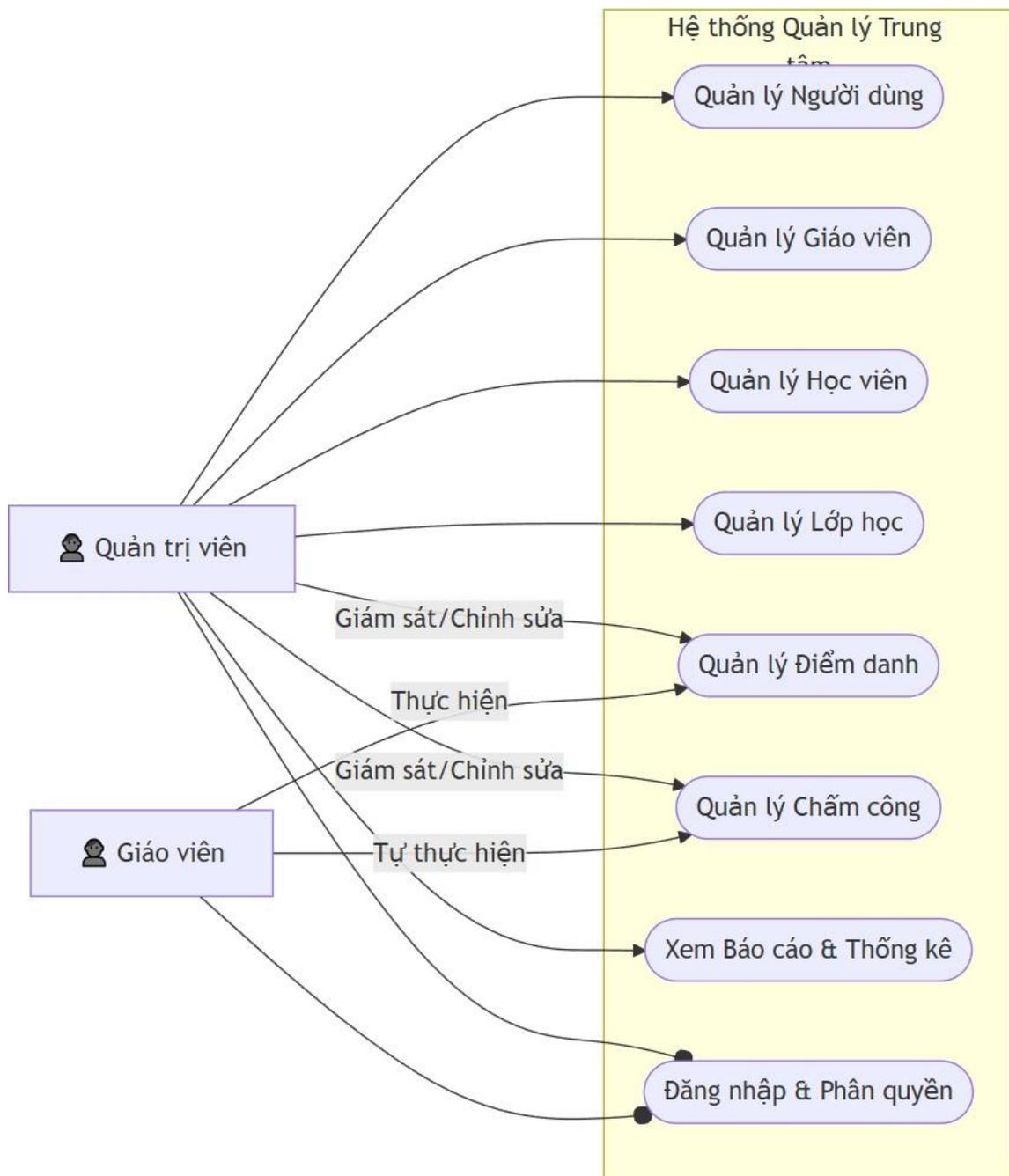
2.2. Tác nhân Hệ Thống

Quản trị viên (Administrator): Là người dùng có quyền cao nhất, chịu trách nhiệm quản lý toàn bộ dữ liệu cốt lõi của hệ thống như người dùng, giáo viên, học viên, lớp học và xem các báo cáo tổng thể.

Giáo viên (Teacher): Là người dùng chịu trách nhiệm chuyên môn giảng dạy, thực hiện điểm danh cho lớp mình phụ trách và chấm công.

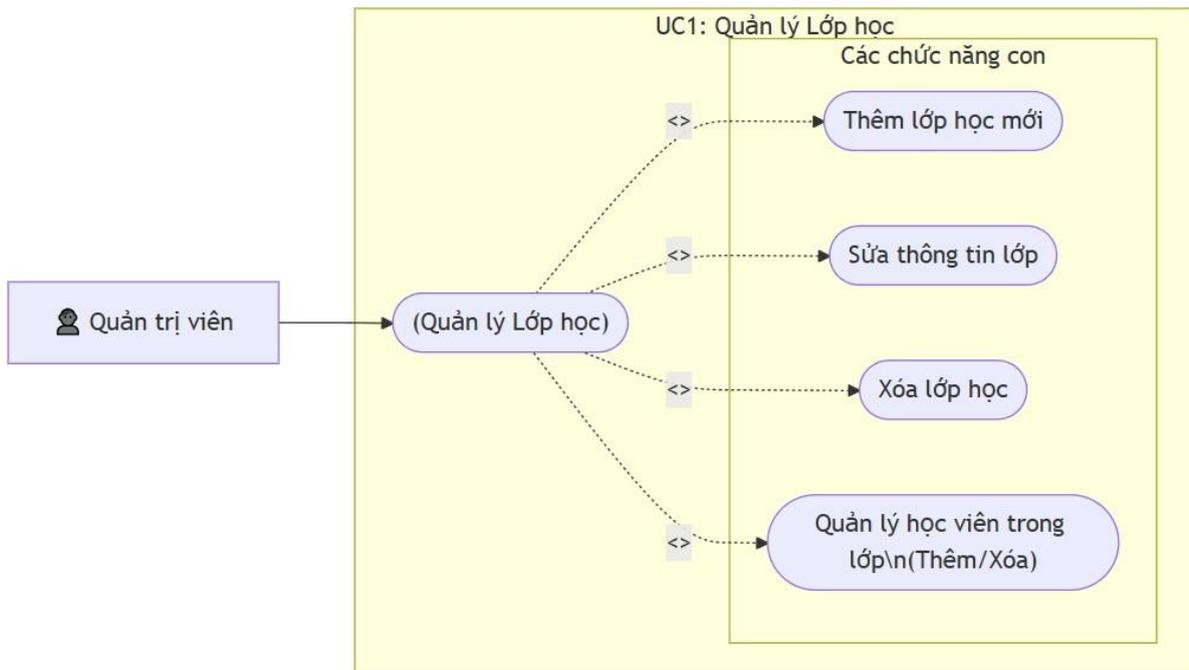
2.3. Biểu đồ Use Case

2.3.1. Usecase tổng quát



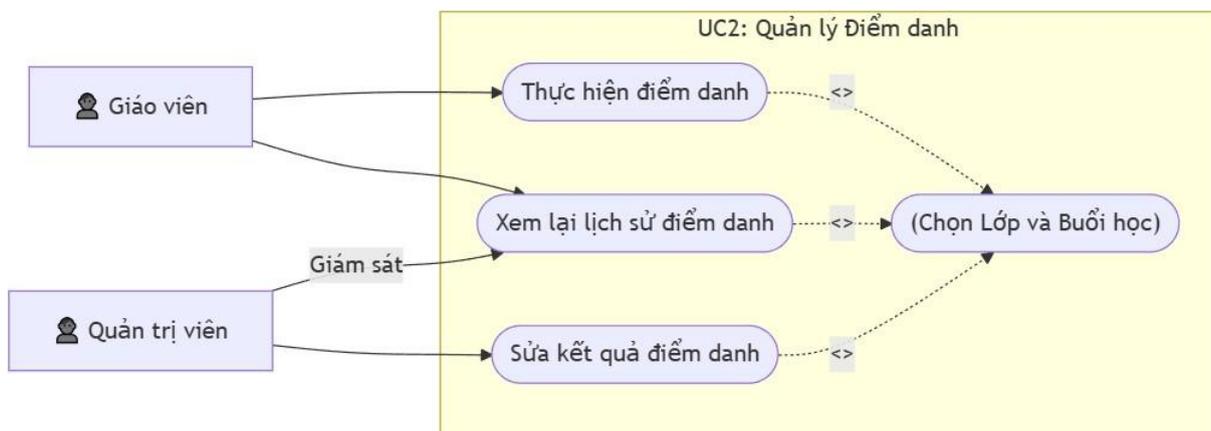
Hình 2.1. Biểu Đồ Usecase Tổng Quát

2.3.2. Biểu đồ Usecase Quản lý lớp học



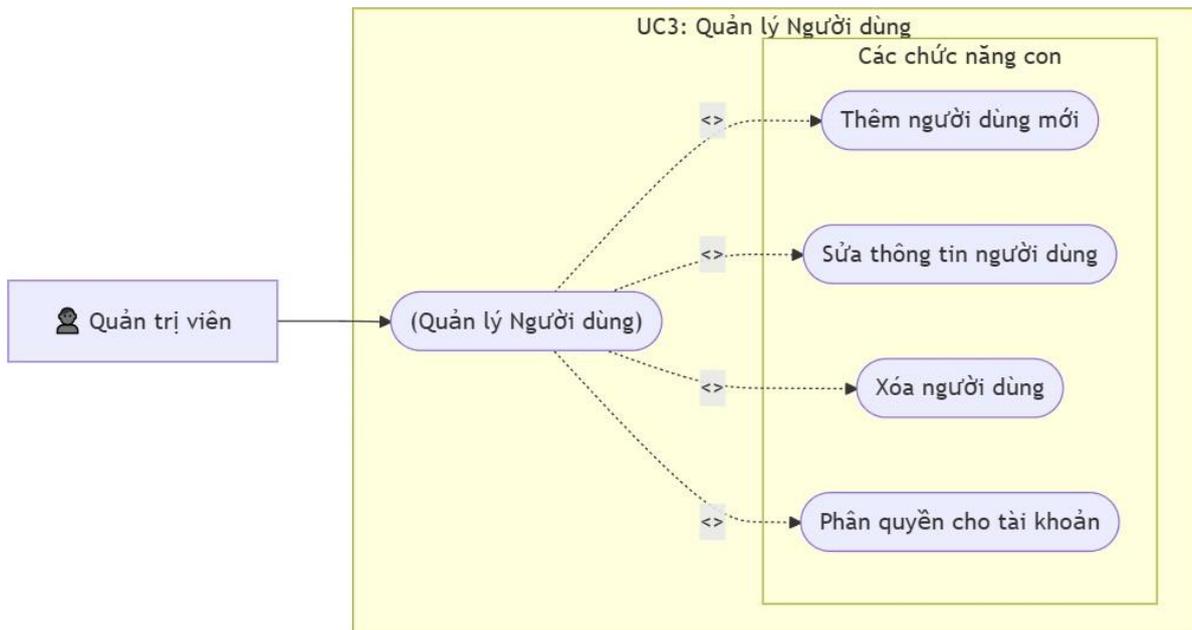
Hình 2.2. Biểu Đồ Usecase Quản lý lớp học

2.3.3. Biểu đồ Usecase Quản lý điểm danh



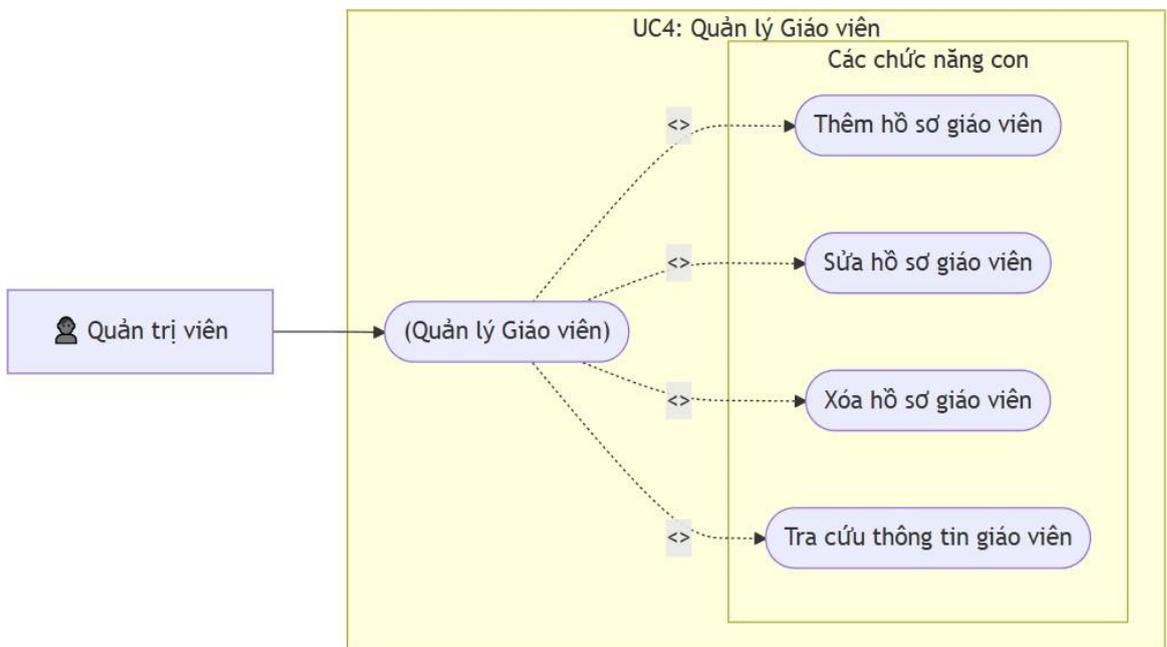
Hình 2.3. Biểu Đồ Usecase Quản lý điểm danh

2.3.4. Biểu đồ Usecase Quản lý user



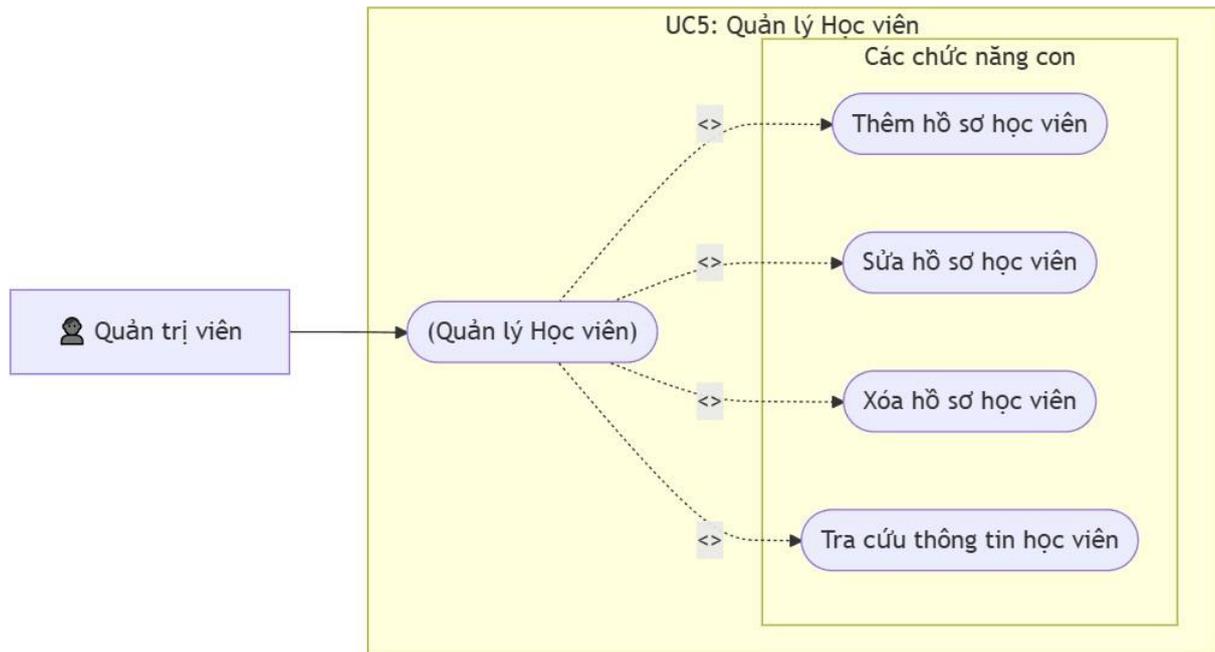
Hình 2.4. Biểu Đồ Usecase Quản lý user

2.3.5. Biểu đồ Usecase Quản lý Giáo viên



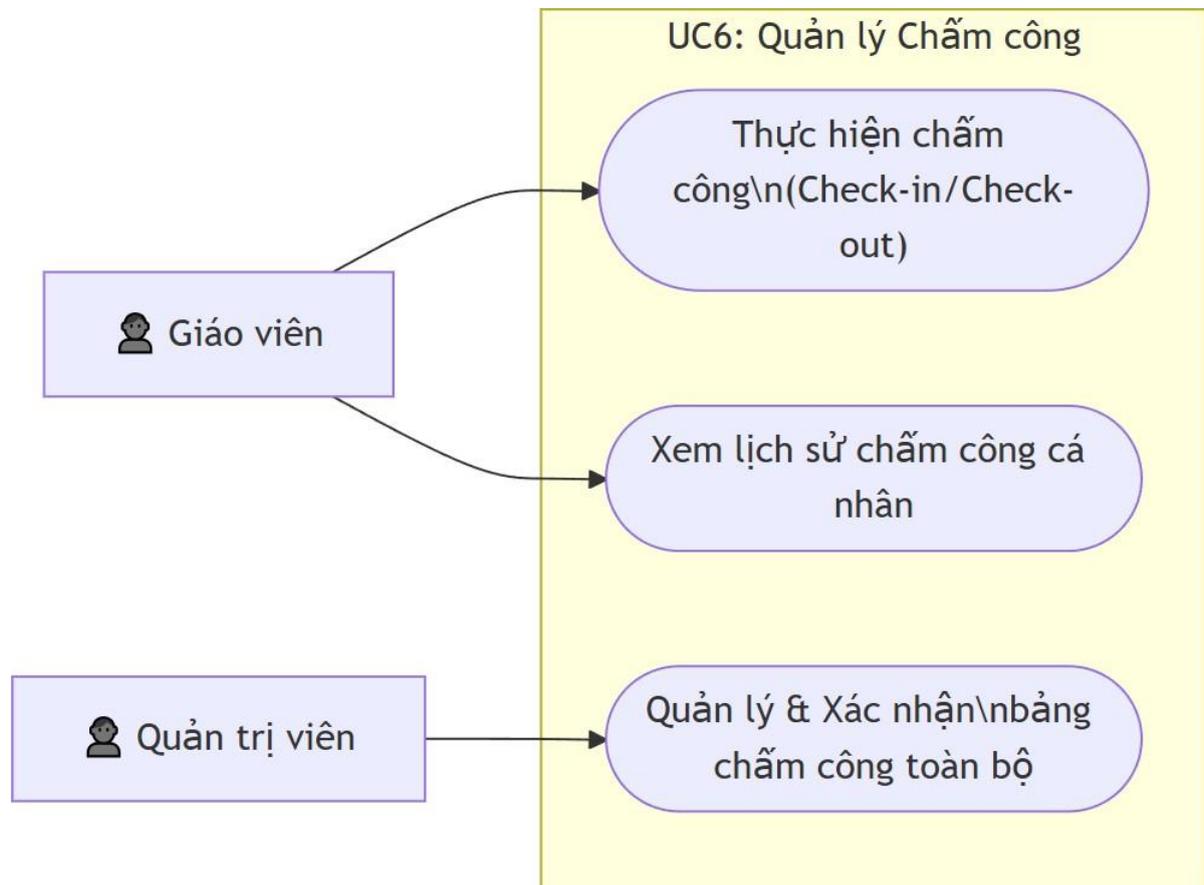
Hình 2.5. Biểu Đồ Usecase Quản lý giáo viên

2.3.6. Biểu đồ Usecase Quản lý Học viên



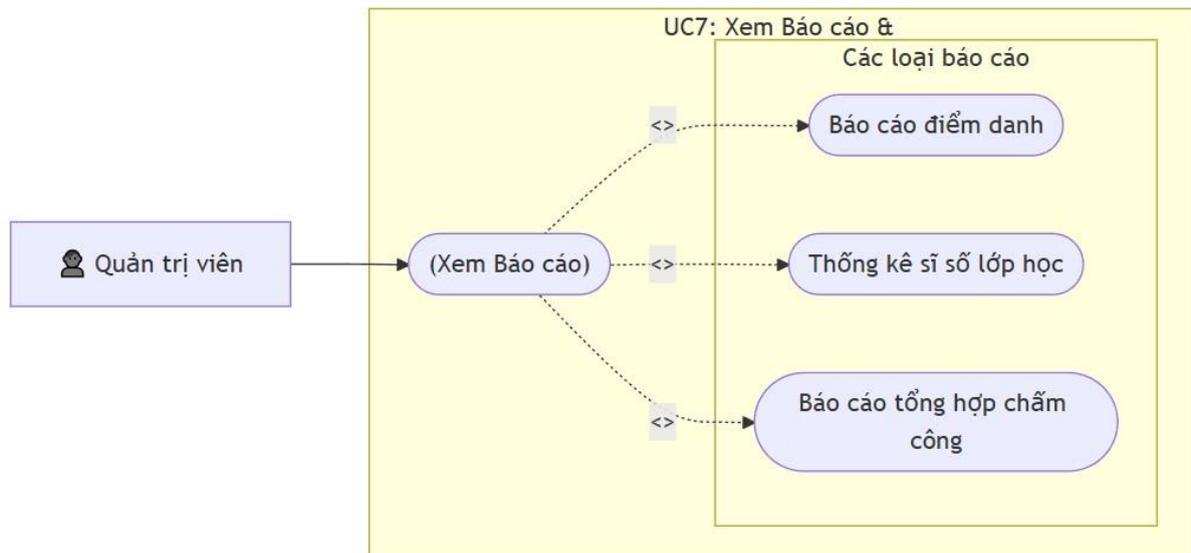
Hình 2.6. Biểu Đồ Usecase Quản lý học viên

2.3.7. Biểu đồ Usecase Quản lý Chấm công



Hình 2.7. Biểu Đồ Usecase Quản lý chấm công

2.3.8. Biểu đồ Usecase Quản lý Báo cáo & thống kê

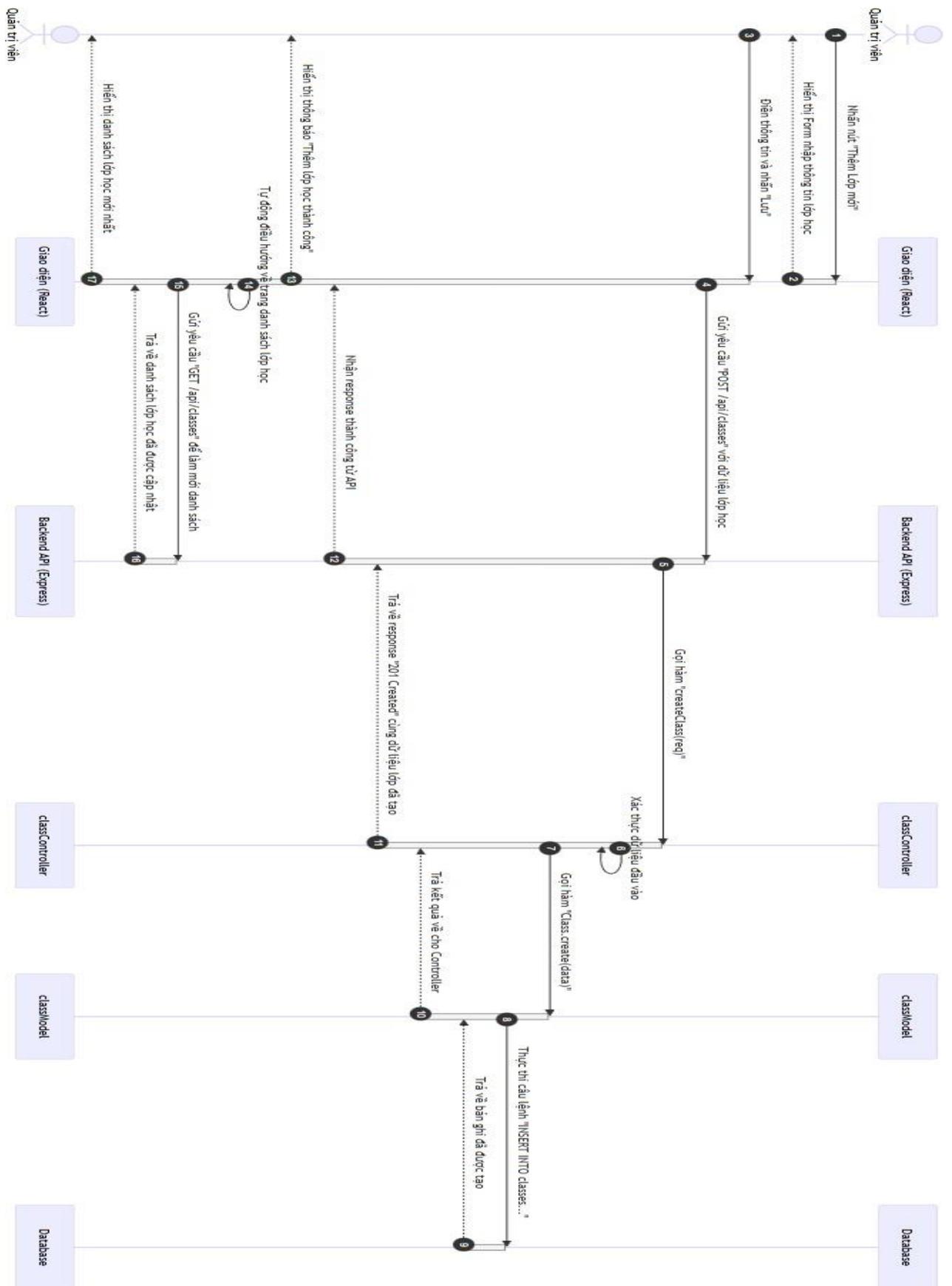


Hình 2.8. Biểu Đồ Usecase Quản lý báo cáo và thống kê

2.4. Sơ Đồ UseCase Tuần Tự

2.4.1. UC Quản lý lớp học

- ❖ Thêm lớp học

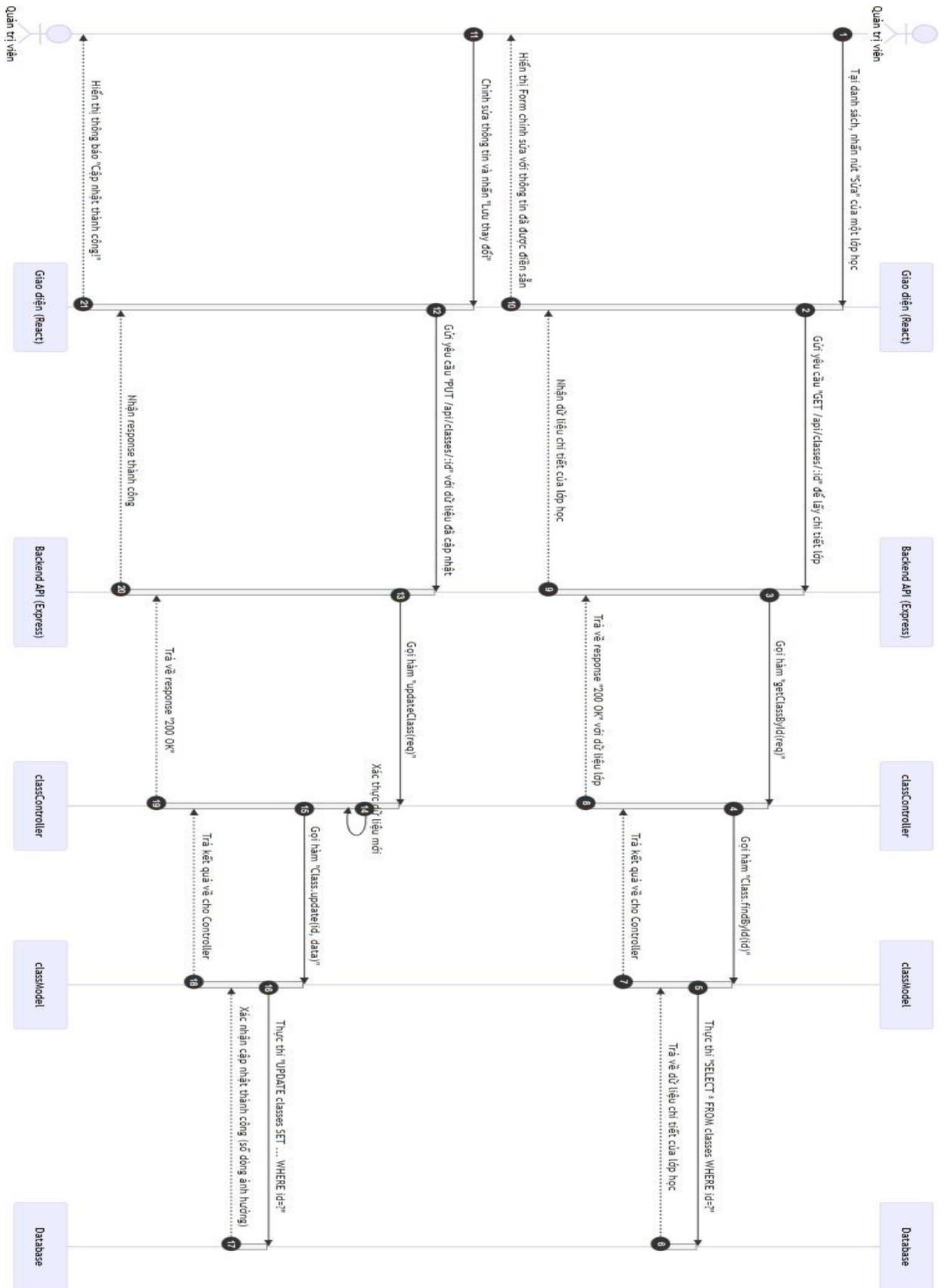


Hình 2.9. Biểu Đồ tuần tự chức năng thêm lớp học

➤ Luồng hoạt động

- 1-3: Quản trị viên tương tác với giao diện để yêu cầu chức năng và nhập dữ liệu.
- 4: Giao diện người dùng gửi một yêu cầu HTTP POST đến API của backend.
- 5-10: Backend xử lý yêu cầu:
 - API route gọi đến classController.
 - Controller xử lý logic, gọi classModel để thao tác với Database.
 - Model thực thi lệnh INSERT và trả kết quả về.
 - Controller tạo một response thành công và gửi lại.
- 11-15: Giao diện nhận được phản hồi, thông báo cho người dùng, và tự động làm mới dữ liệu để hiển thị thông tin mới nhất.

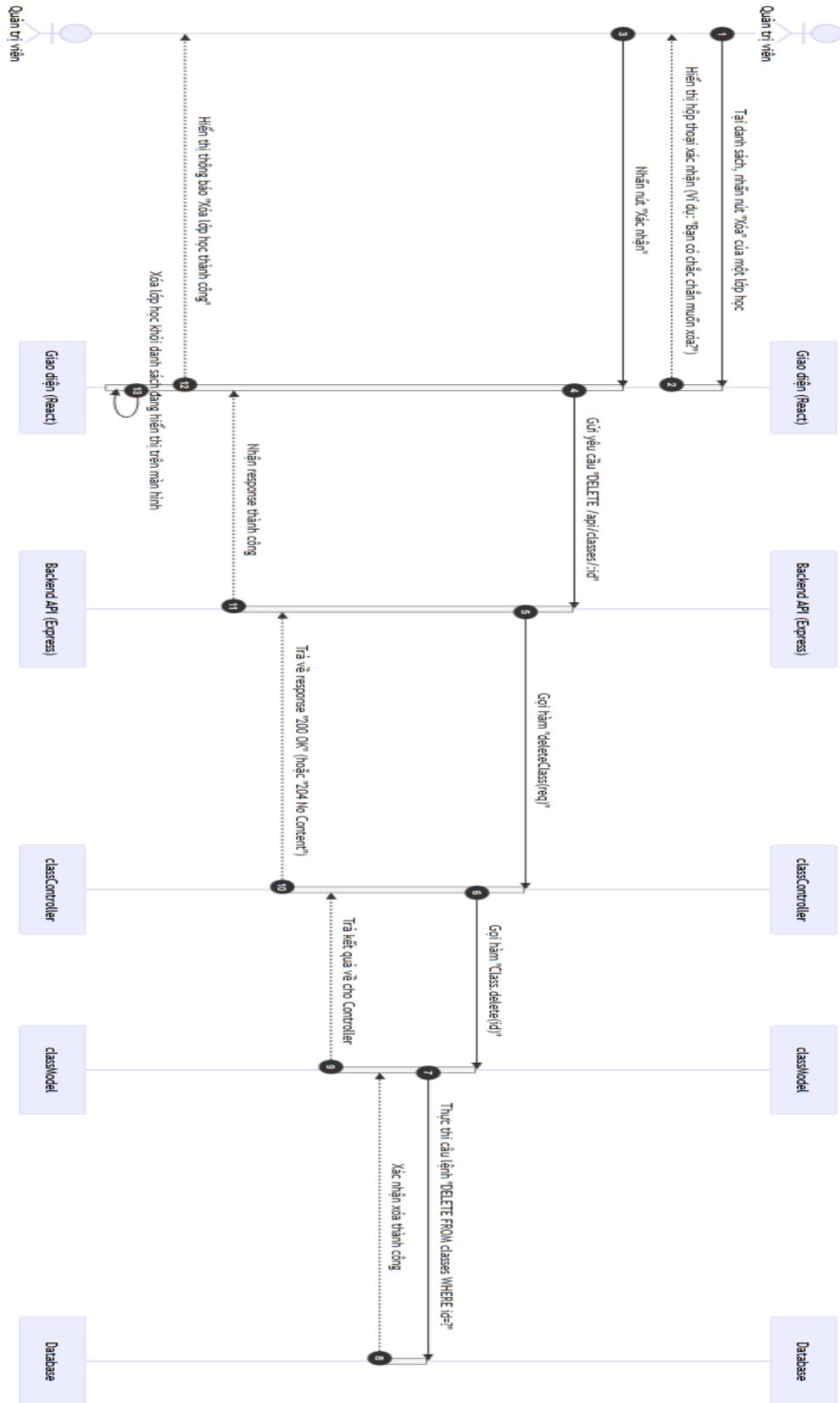
❖ Sử dụng thông tin lớp học



Hình 2.10. Biểu Đồ tuần tự chức năng sửa lớp học

- Luồng hoạt động
- Giai đoạn 1: Lấy và hiển thị dữ liệu cũ (Bước 1-10)
 - 1-2: Quản trị viên chọn một lớp học để sửa. Giao diện (UI) ngay lập tức gửi một yêu cầu GET đến Backend API để lấy thông tin đầy đủ của lớp học đó.
 - 3-8: Yêu cầu được chuyển qua các tầng của backend: API -> Controller -> Model. Model truy vấn Database để lấy dữ liệu.
 - 9-10: Dữ liệu được trả về qua các tầng và đến tay Giao diện (UI). UI dùng dữ liệu này để điền vào một biểu mẫu (form) chỉnh sửa và hiển thị cho Quản trị viên.
- Giai đoạn 2: Gửi và lưu dữ liệu mới (Bước 11-21)
 - 11-12: Sau khi Quản trị viên chỉnh sửa và nhấn lưu, UI gửi một yêu cầu PUT (cập nhật) đến cùng một địa chỉ API, mang theo dữ liệu mới.
 - 13-18: Backend nhận yêu cầu, xác thực dữ liệu, sau đó Controller yêu cầu Model thực thi lệnh UPDATE trong Database.
 - 19-21: Sau khi nhận xác nhận từ Database, Backend gửi một response thành công về cho UI. UI nhận được response này và hiển thị thông báo "Cập nhật thành công" cho người dùng.

❖ Xóa lớp học

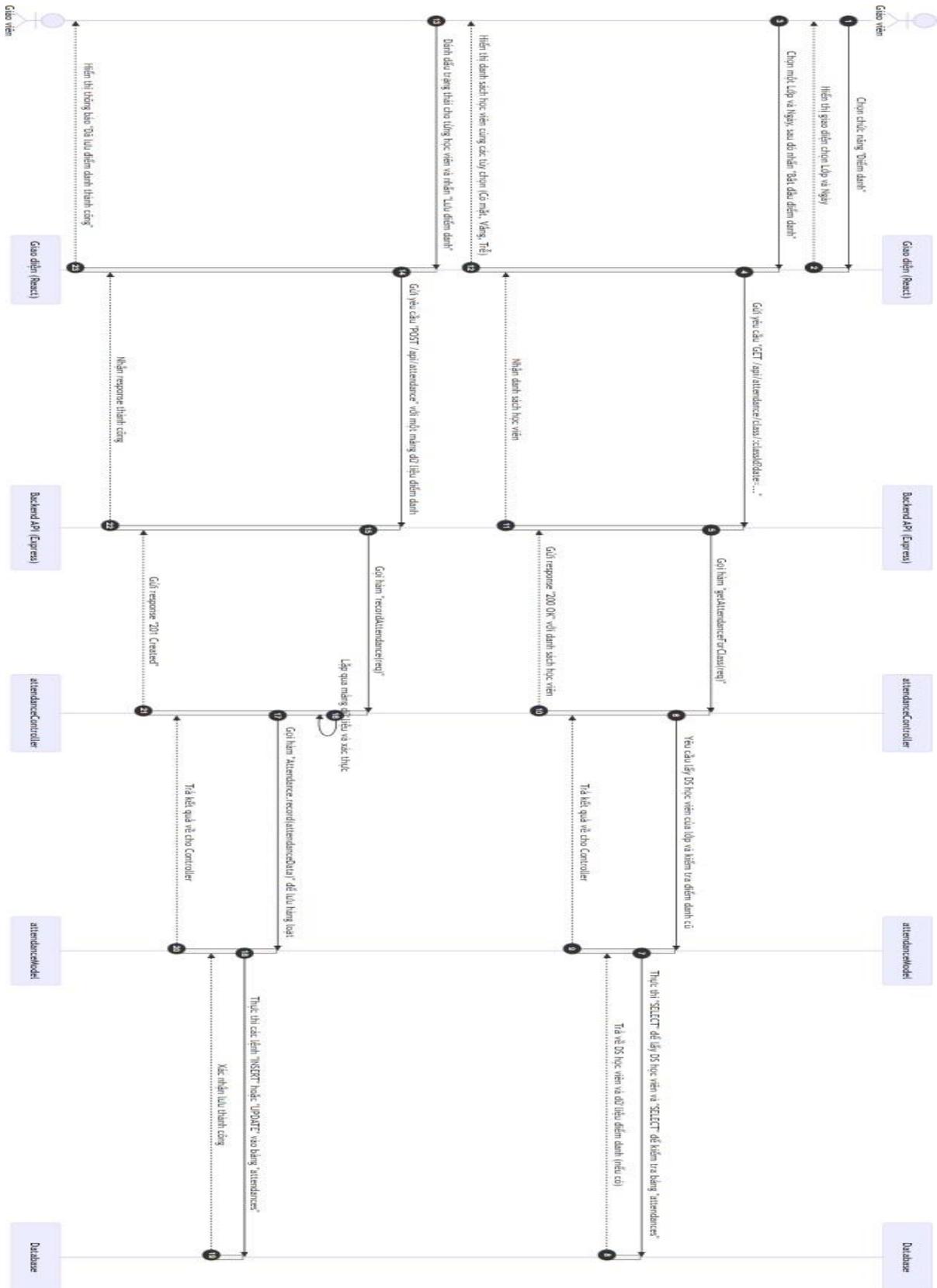


Hình 2.11. Biểu Đồ tuần tự chức năng xóa lớp học

➤ Luồng hoạt động

- 1-3: Quản trị viên bắt đầu hành động xóa. Để tránh thao tác nhầm, Giao diện (UI) sẽ hiển thị một hộp thoại yêu cầu xác nhận lại từ người dùng.
- 4: Sau khi người dùng xác nhận, UI gửi yêu cầu DELETE đến Backend API, chỉ định rõ ID của lớp học cần xóa.
- 5-10: Backend xử lý yêu cầu xóa. Controller gọi Model để thực thi lệnh DELETE trong Database. Sau khi Database xác nhận hoàn tất, backend trả về một response thành công cho UI.
- 11-13: UI nhận được response thành công, báo cho người dùng biết thao tác đã hoàn tất, và đồng thời cập nhật lại giao diện bằng cách loại bỏ dòng dữ liệu vừa bị xóa khỏi danh sách.

2.4.2. UC Giáo viên thực hiện điểm danh

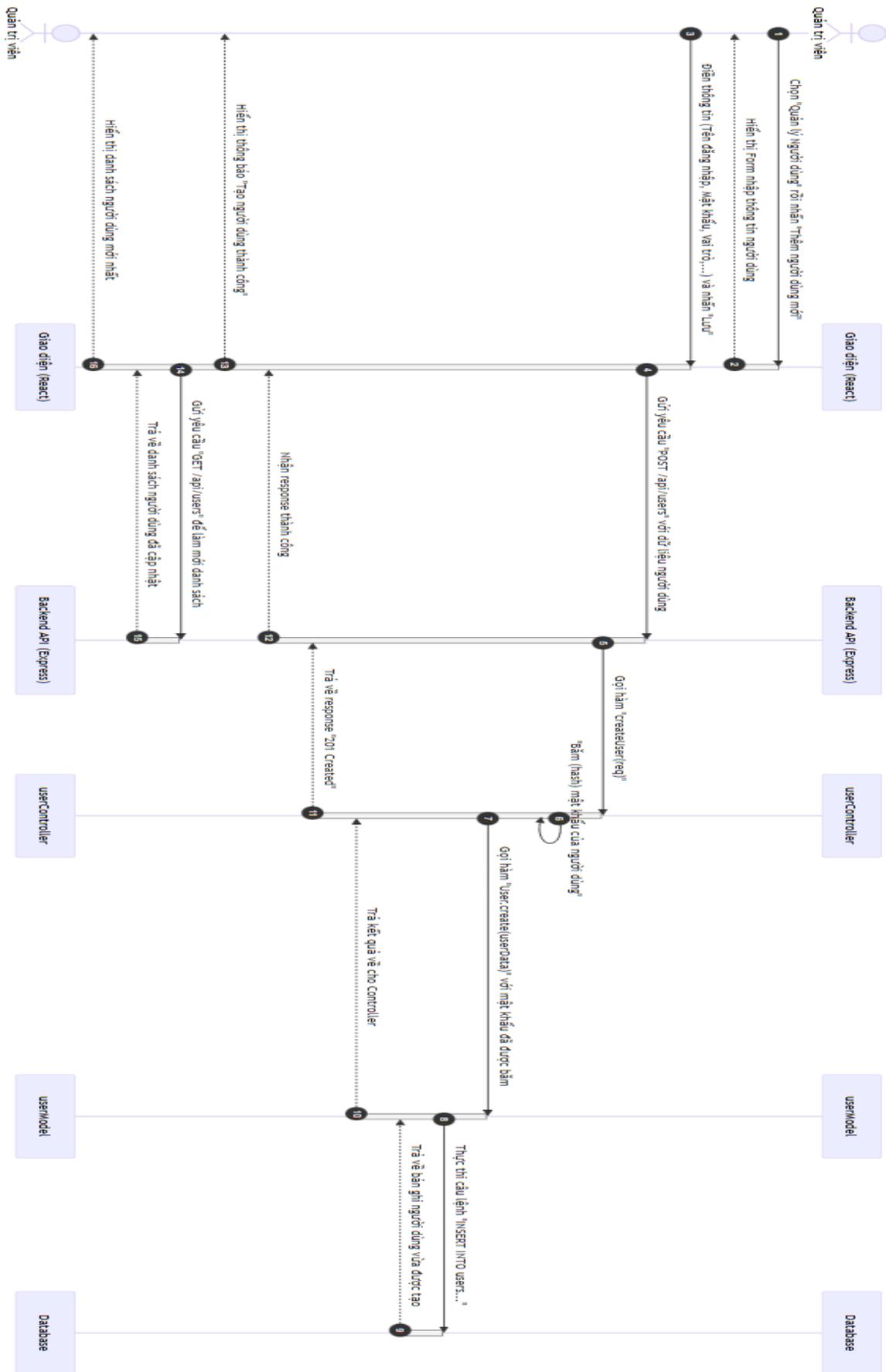


Hình 2.12. Biểu Đồ tuần tự giáo viên thực hiện điểm danh

- Luồng hoạt động
- Giai đoạn 1: Lấy danh sách lớp để điểm danh (Bước 1-11)
 - 1-4: Giáo viên chọn lớp và ngày muốn điểm danh. Giao diện (UI) gửi yêu cầu GET lên Backend API để lấy danh sách học viên của lớp đó.
 - 5-9: Backend xử lý yêu cầu, Controller thông qua Model để truy vấn Database lấy về danh sách học viên thuộc lớp học và kiểm tra xem ngày hôm đó đã điểm danh hay chưa.
 - 10-11: Dữ liệu được trả về cho UI để hiển thị một danh sách sinh viên kèm theo các nút để giáo viên thao tác điểm danh.
- Giai đoạn 2: Lưu kết quả điểm danh (Bước 12-22)
 - 12-13: Sau khi giáo viên hoàn tất việc đánh dấu, UI tổng hợp dữ liệu thành một mảng và gửi yêu cầu POST đến Backend API.
 - 14-19: Controller nhận dữ liệu, xử lý và gọi Model để ghi đồng loạt thông tin điểm danh của các học viên vào Database. Model có thể thực hiện nghiệp vụ "tạo mới nếu chưa có, cập nhật nếu đã có" (upsert).
 - 20-22: Sau khi Database xác nhận lưu thành công, một response Created được trả về cho UI, và UI hiển thị thông báo thành công cho giáo viên.

2.4.3. UC Quản lý người dùng

- ❖ Thêm người dùng mới

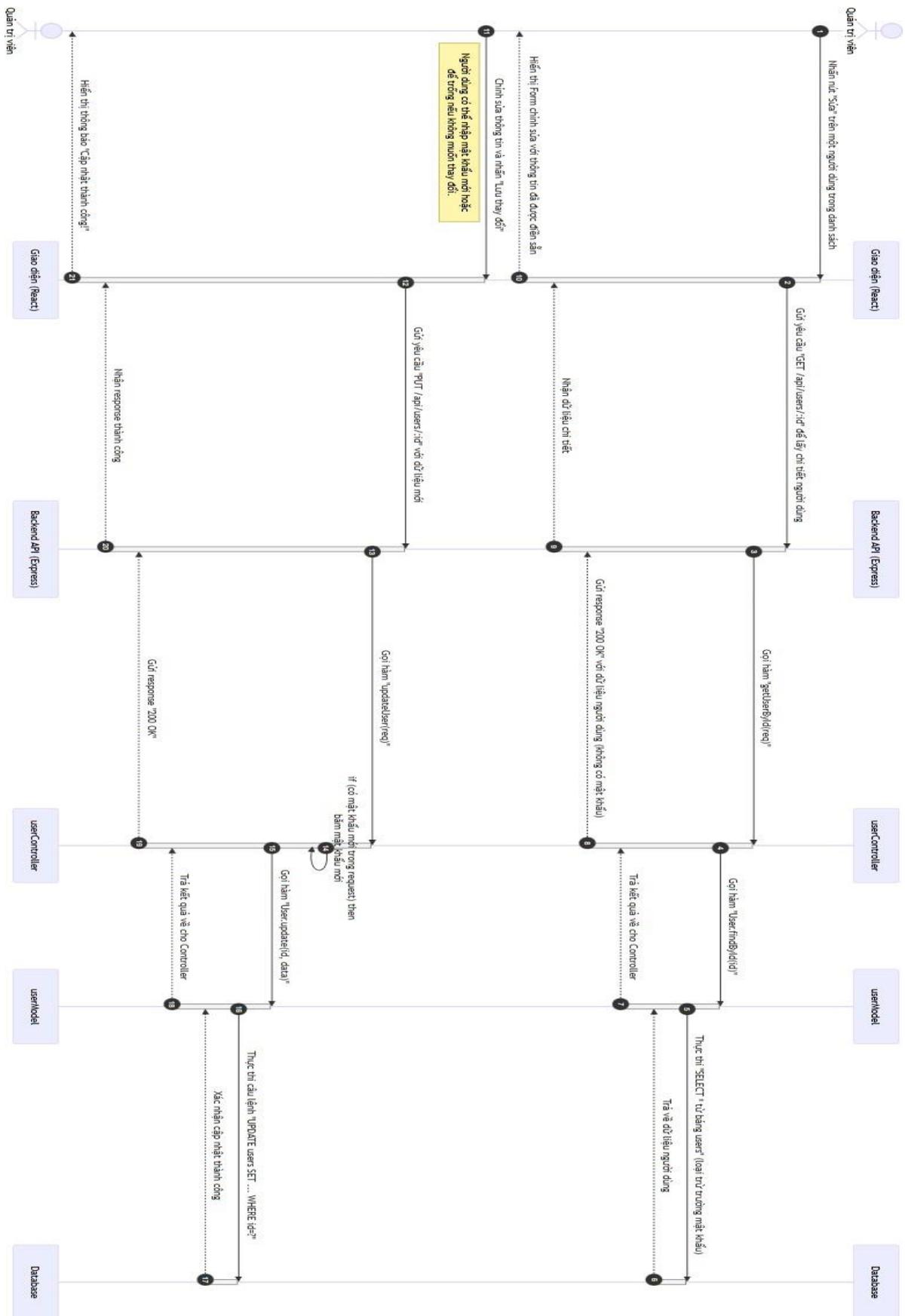


Hình 2.13. Biểu Đồ tuần tự chức năng thêm người dùng

➤ Luồng hoạt động

- 1-4: Quản trị viên nhập thông tin cho người dùng mới thông qua giao diện.
- 5: Giao diện (UI) gửi một yêu cầu POST chứa thông tin người dùng đến Backend API.
- 6-12: Backend xử lý yêu cầu:
 - 6-7: userController nhận yêu cầu. Bước quan trọng nhất ở đây là Controller không lưu mật khẩu dạng thô (plain text) mà sử dụng một thuật toán để băm mật khẩu (ví dụ: bcrypt), biến nó thành một chuỗi ký tự không thể dịch ngược.
 - 8: Controller gọi userModel để tạo người dùng mới, truyền vào dữ liệu người dùng với mật khẩu đã được băm an toàn.
 - 9-12: Model ghi dữ liệu vào Database và các tầng lần lượt trả về response thành công.
- 13-16: Giao diện nhận được phản hồi thành công, báo cho Quản trị viên và tự động tải lại danh sách người dùng để hiển thị tài khoản vừa được tạo.

❖ Sửa thông tin người dùng

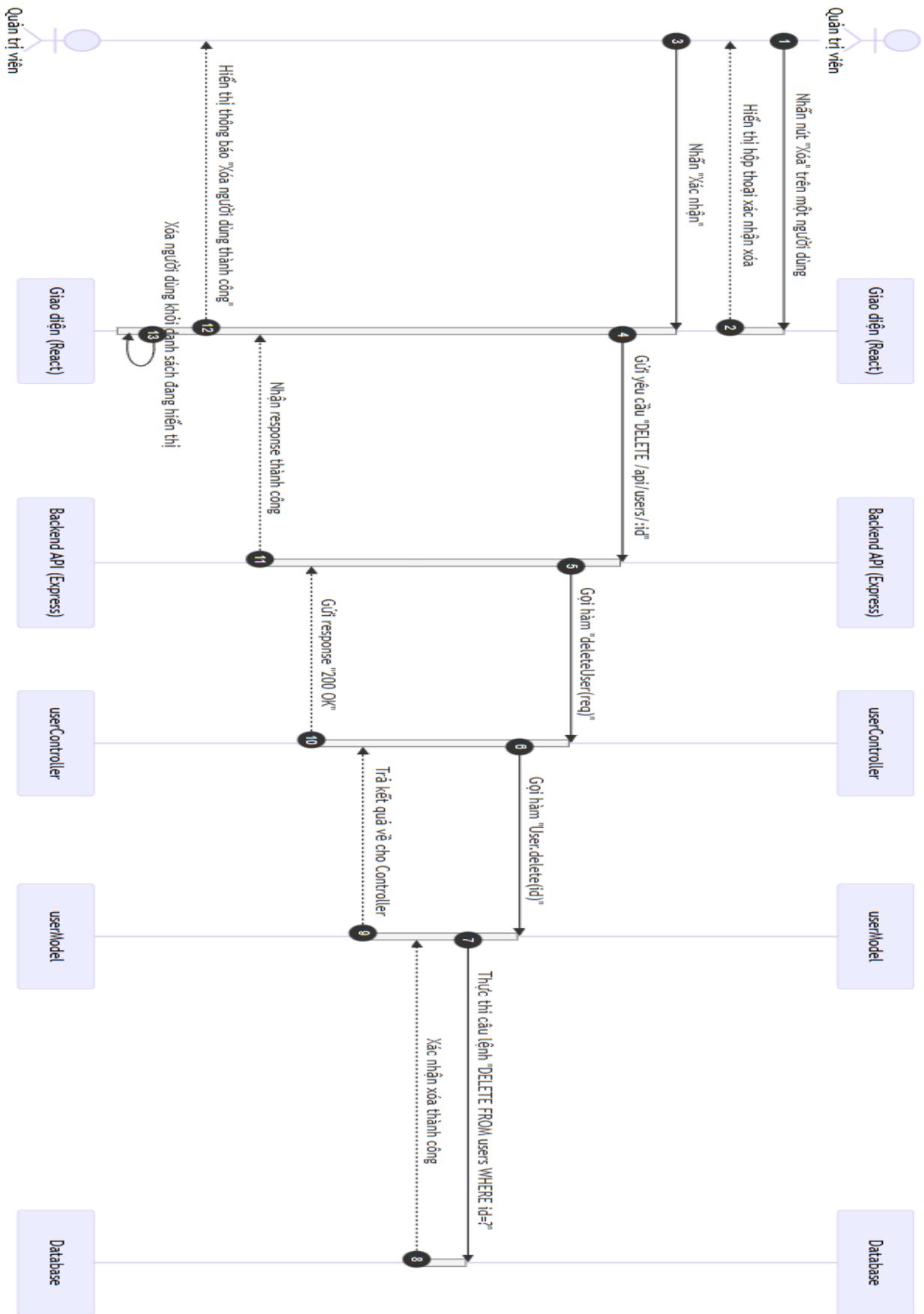


Hình 2.14. Biểu Đồ tuần tự chức năng sửa thông tin người dùng

➤ Luồng hoạt động

- 1-9: Tương tự các luồng "sửa" khác, UI lấy dữ liệu mới nhất của người dùng từ backend để hiển thị lên form. Lưu ý bảo mật: backend tuyệt đối không trả trường mật khẩu (dù đã băm) về cho client.
- 10-12: Người dùng thay đổi thông tin và nhấn lưu. Giao diện gửi yêu cầu PUT lên backend.
- 13-14: Đây là bước xử lý quan trọng. userController sẽ kiểm tra xem người dùng có gửi lên mật khẩu mới hay không. Nếu có, nó sẽ băm mật khẩu mới này. Nếu không, nó sẽ bỏ qua và chỉ cập nhật các thông tin khác.
- 15-21: Dữ liệu đã được xử lý sẽ được Model cập nhật vào Database. Hệ thống phản hồi thành công cho người dùng.

❖ Xóa người dùng



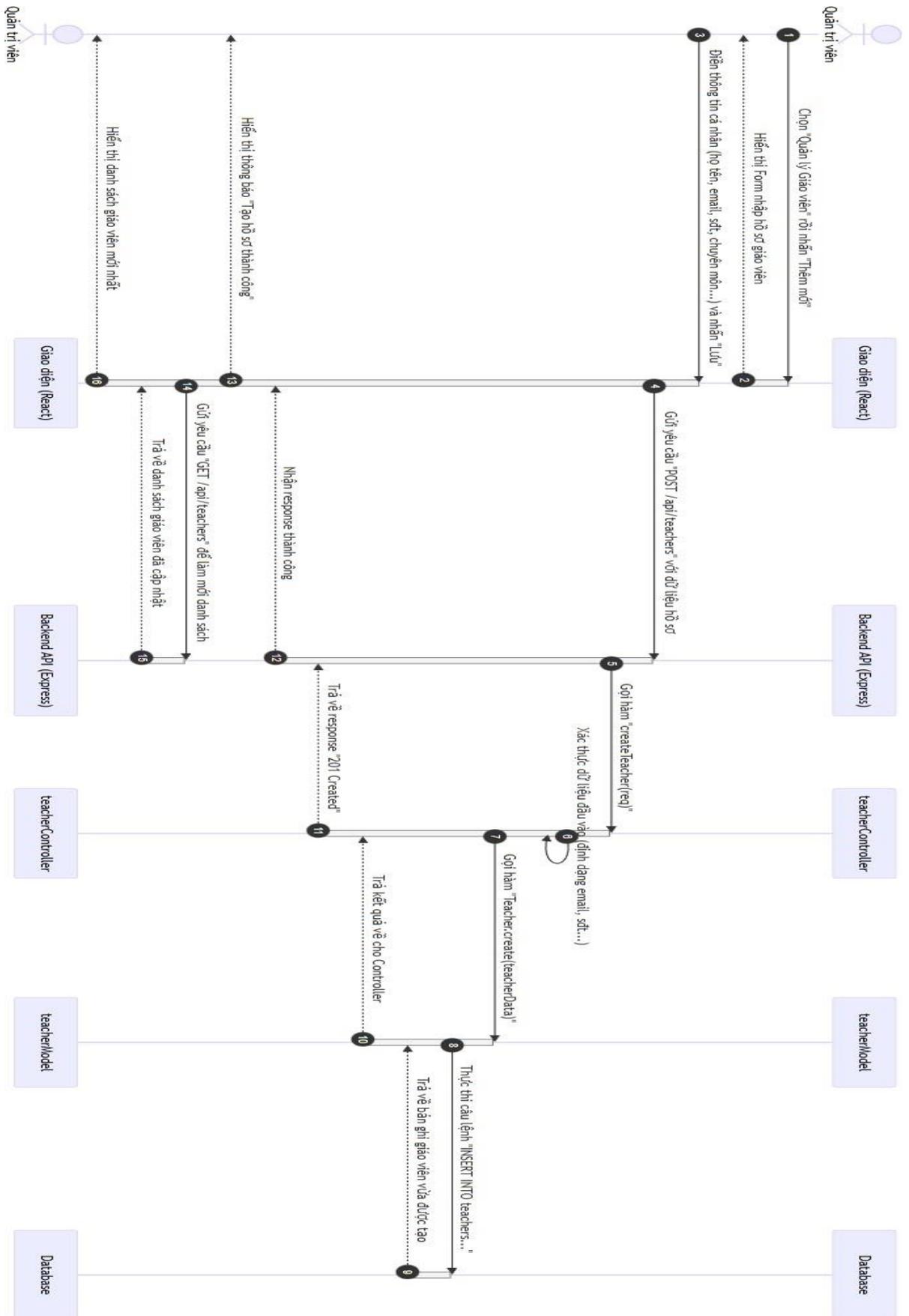
Hình 2.15. Biểu Đồ tuần tự chức năng xóa người dùng

➤ Luồng hoạt động

- 1-4: Người dùng yêu cầu xóa và xác nhận lại để đảm bảo an toàn.
- 5-10: Giao diện gửi yêu cầu DELETE và backend xử lý việc xóa bản ghi tương ứng trong cơ sở dữ liệu.
- 11-13: Giao diện nhận phản hồi, báo thành công và cập nhật lại màn hình.

2.4.4. UC Quản lý giáo viên

- ❖ Thêm hồ sơ giáo viên mới

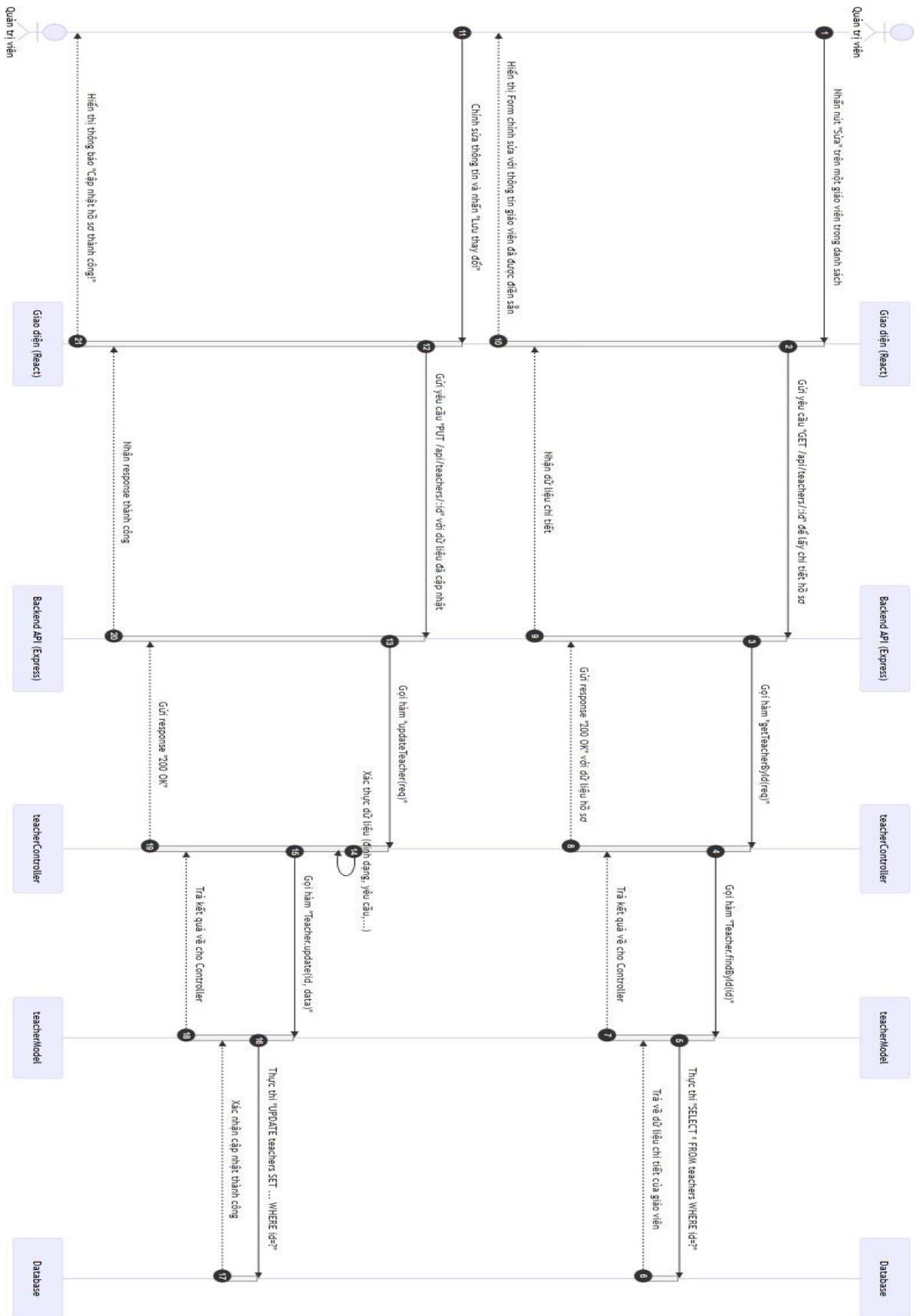


Hình 2.16. Biểu Đồ tuần tự chức năng thêm hồ sơ giáo viên mới

➤ Luồng hoạt động

- 1-4: Quản trị viên nhập thông tin của giáo viên mới vào một biểu mẫu trên giao diện.
- 5: Giao diện gửi yêu cầu POST chứa thông tin này đến Backend.
- 6-12: Backend nhận dữ liệu, xác thực tính hợp lệ, sau đó Controller yêu cầu Model tạo một bản ghi mới trong bảng teachers của Database. Sau khi hoàn tất, backend trả về một response thành công.
- 13-16: Giao diện thông báo cho người dùng và làm mới lại danh sách để hiển thị giáo viên vừa được thêm.

❖ Sửa hồ sơ giáo viên

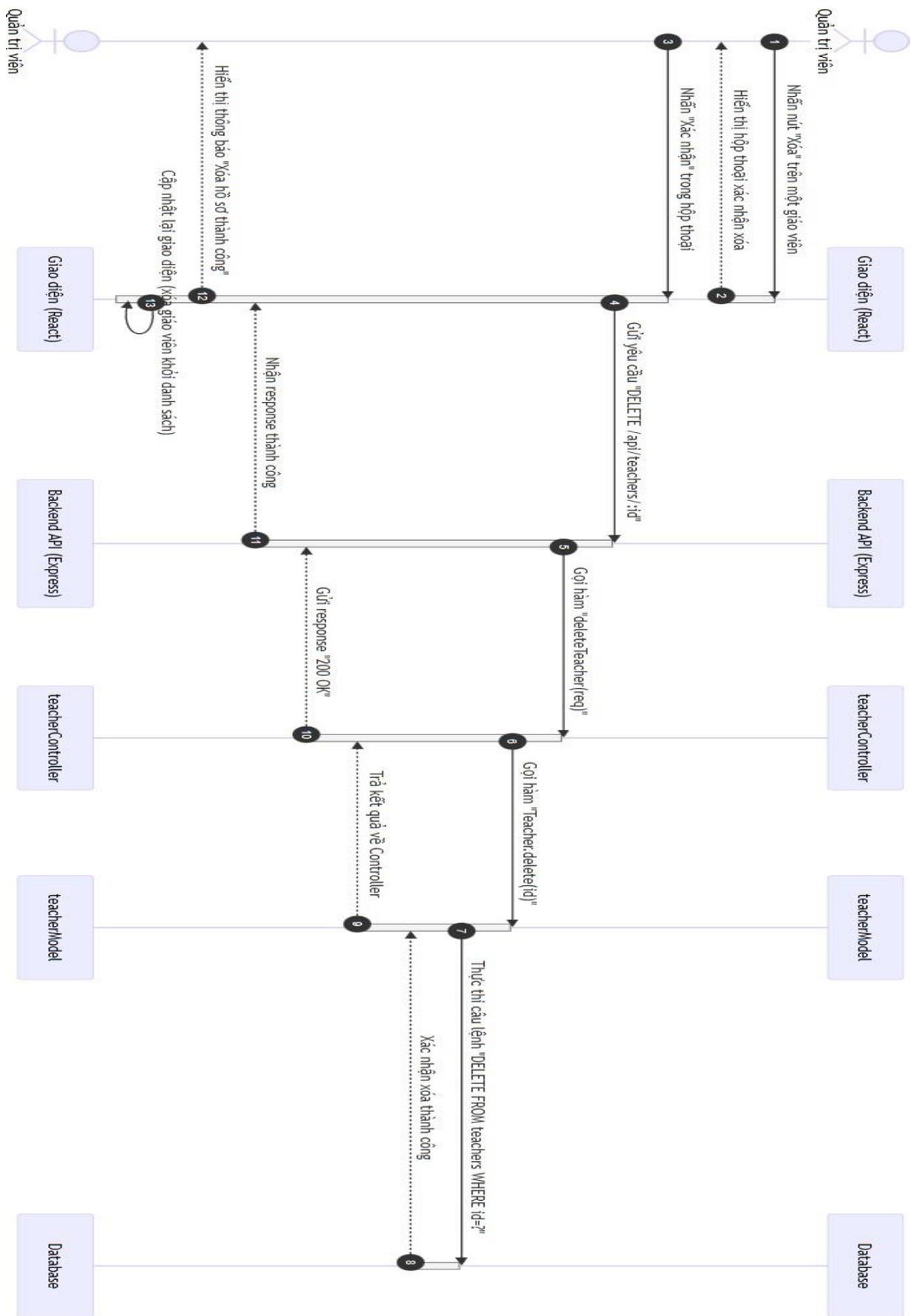


Hình 2.17. Biểu Đồ tuần tự chức năng sửa hồ sơ giáo viên

➤ Luồng hoạt động

- Giai đoạn 1 (Lấy dữ liệu): Quản trị viên chọn sửa, giao diện gửi yêu cầu GET để lấy thông tin hiện tại của giáo viên và hiển thị lên biểu mẫu.
- Giai đoạn 2 (Cập nhật dữ liệu): Sau khi Quản trị viên sửa và lưu, giao diện gửi yêu cầu PUT. Backend sẽ xác thực dữ liệu mới và yêu cầu Model thực hiện lệnh UPDATE trong cơ sở dữ liệu, sau đó phản hồi lại cho giao diện.

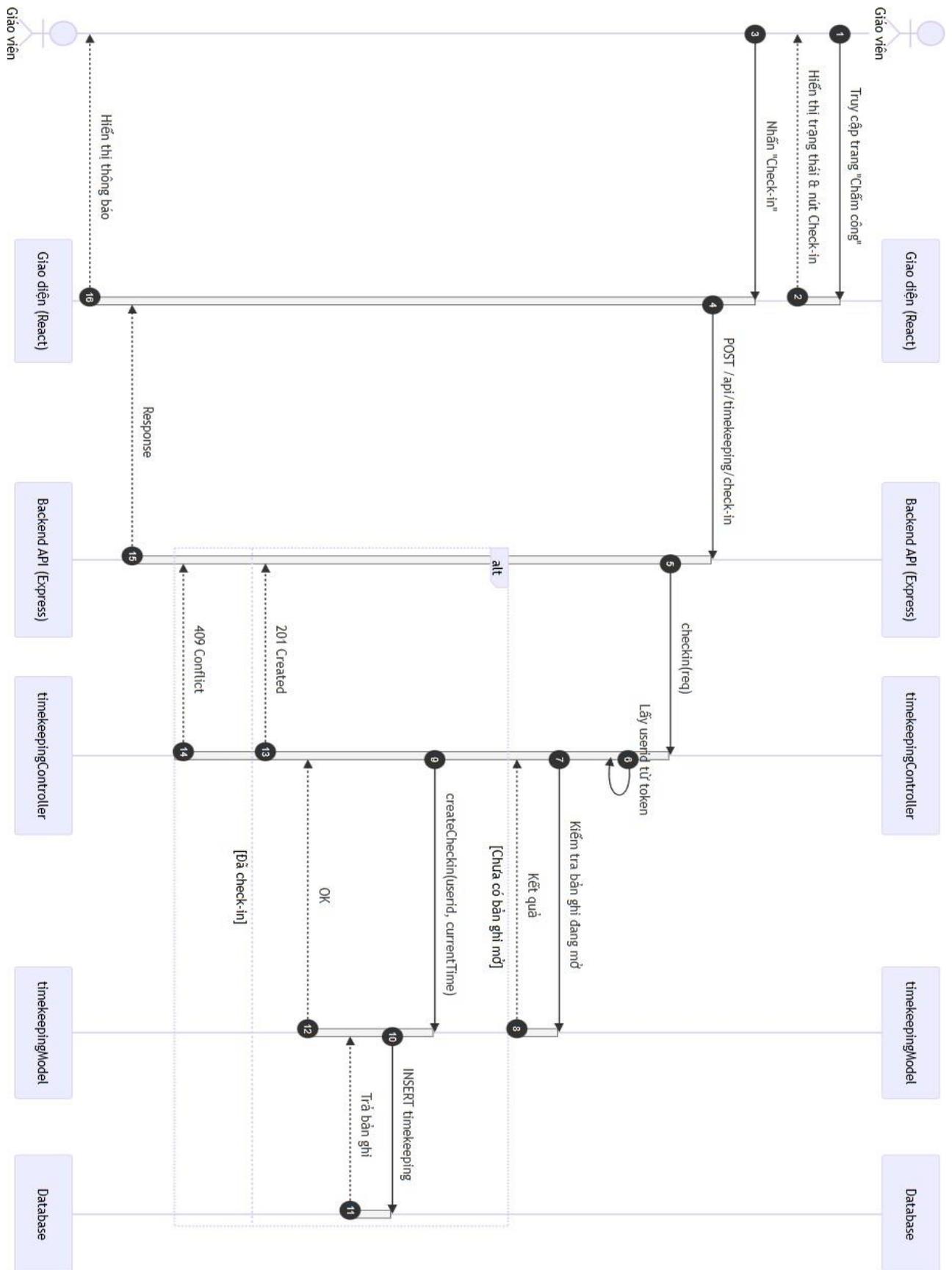
❖ Xóa hồ sơ giáo viên



Hình 2.18. Biểu Đồ tuần tự chức năng xóa hồ sơ giáo viên

- Bước xác nhận: Quản trị viên nhấn nút "Xóa", và giao diện sẽ yêu cầu xác nhận lại để đảm bảo an toàn dữ liệu.
- Bước thực thi: Khi được xác nhận, giao diện gửi yêu cầu DELETE. Backend xử lý yêu cầu bằng cách gọi đến Controller và Model để thực thi lệnh DELETE trong cơ sở dữ liệu.
- Bước phản hồi: Backend báo thành công, giao diện thông báo cho người dùng và cập nhật lại danh sách.

2.4.5. UC Giáo viên thực hiện chấm công

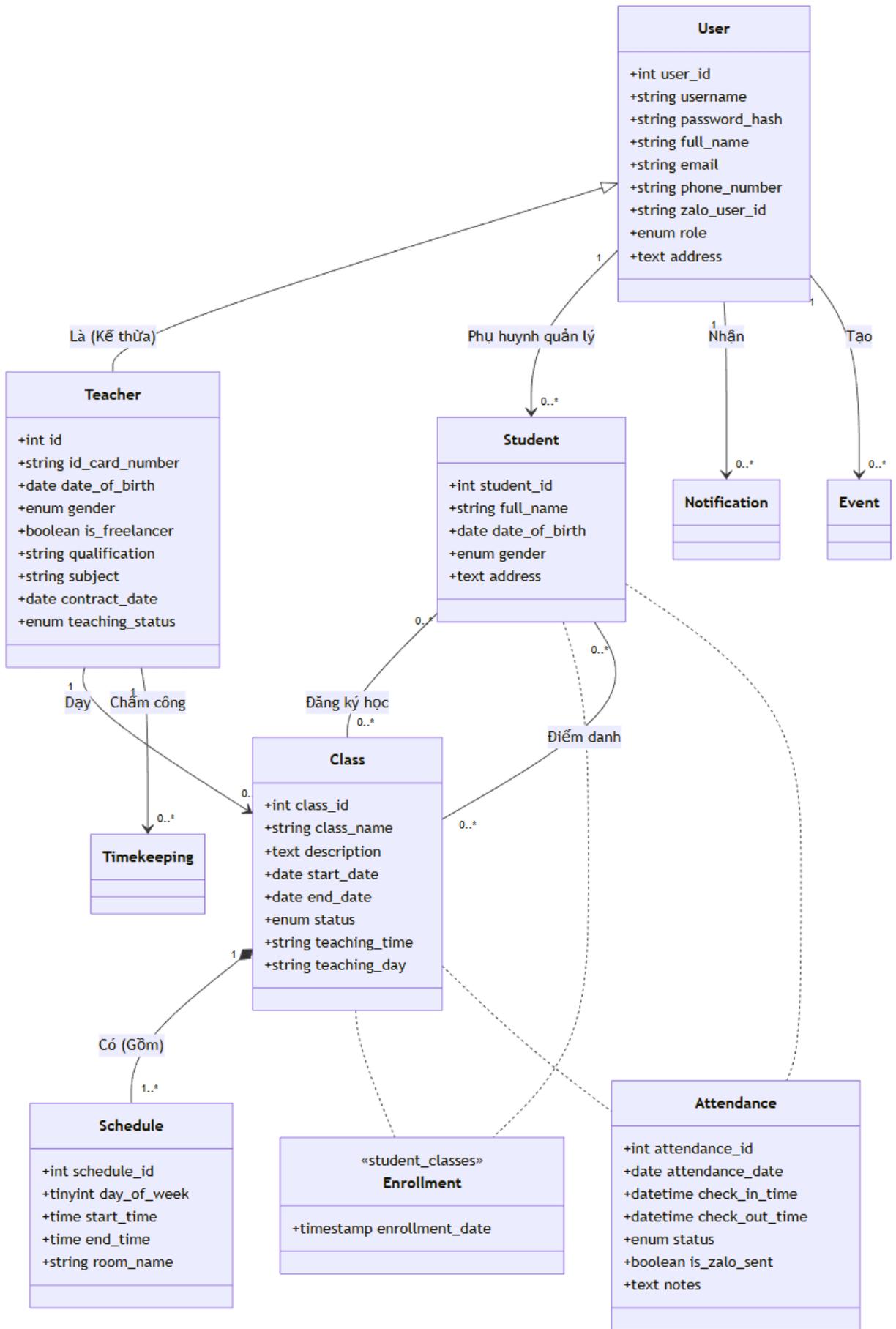


Hình 2.19. Biểu Đồ tuần tự chức năng giáo viên chấm công.

- 1-4: Giáo viên vào trang chấm công và nhấn nút "Check-in". Giao diện gửi yêu cầu POST đến backend. Một điểm quan trọng là giao diện không cần gửi ID của giáo viên, vì thông tin này đã có trong token xác thực (ví dụ: JWT) mà trình duyệt tự động đính kèm vào mỗi request.
- 5-8: Backend nhận yêu cầu. Controller lấy ID của giáo viên từ token và kiểm tra trong Database xem giáo viên này có đang trong một ca làm việc chưa check-out hay không, để tránh việc check-in nhiều lần.
- 9-15 (Luồng thành công): Nếu hợp lệ, Controller sẽ yêu cầu Model tạo một bản ghi chấm công mới với thời gian check-in là thời gian hiện tại và check_out_time là NULL. Sau đó, hệ thống báo thành công về cho giao diện để cập nhật lại trạng thái và thông báo cho giáo viên.
- 16-19 (Luồng lỗi): Nếu giáo viên đã check-in từ trước mà chưa check-out, hệ thống sẽ trả về lỗi và thông báo cho giáo viên biết.

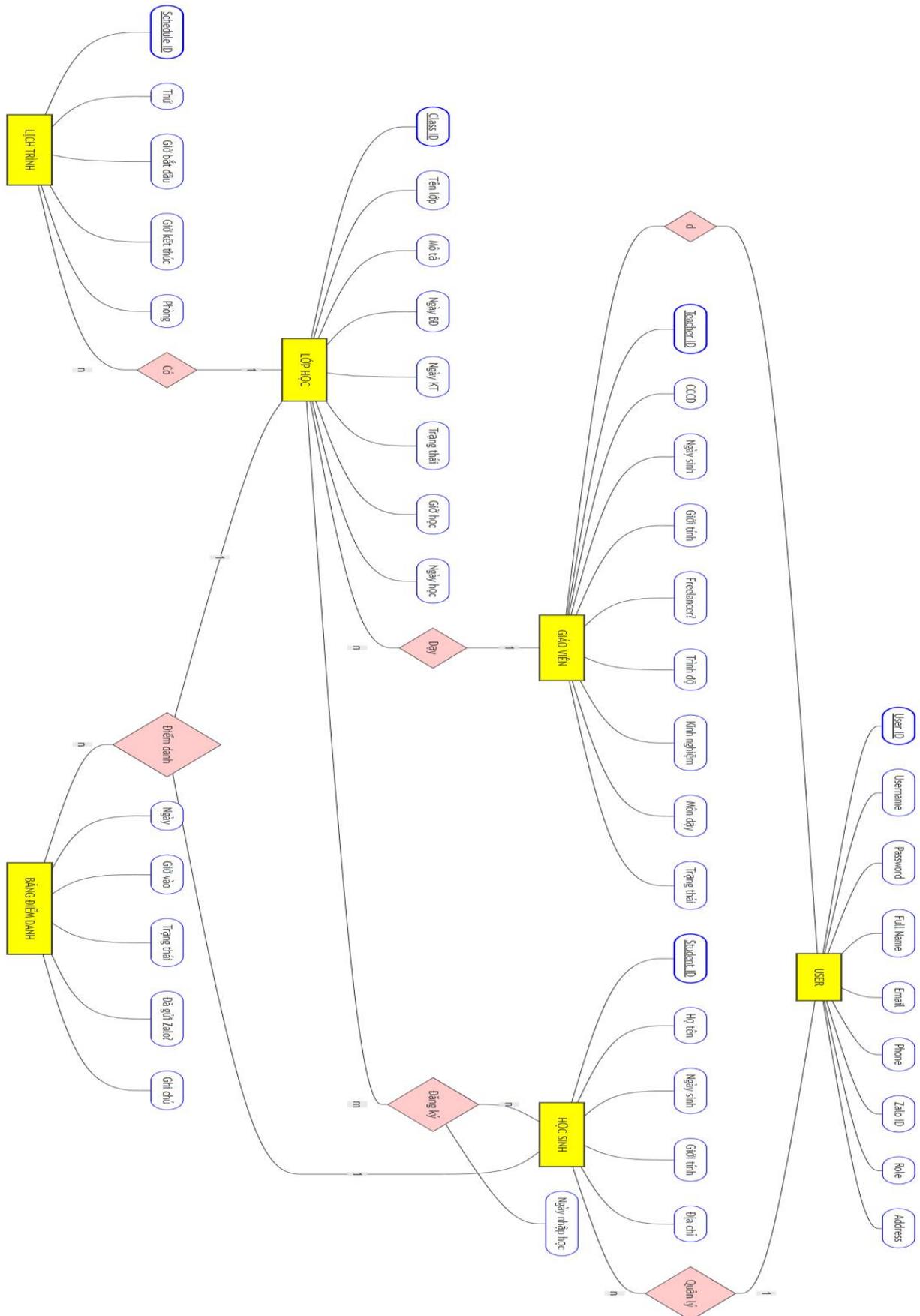
2.5. Thiết kế Cơ sở dữ liệu

2.5.1. Biểu đồ lớp UML



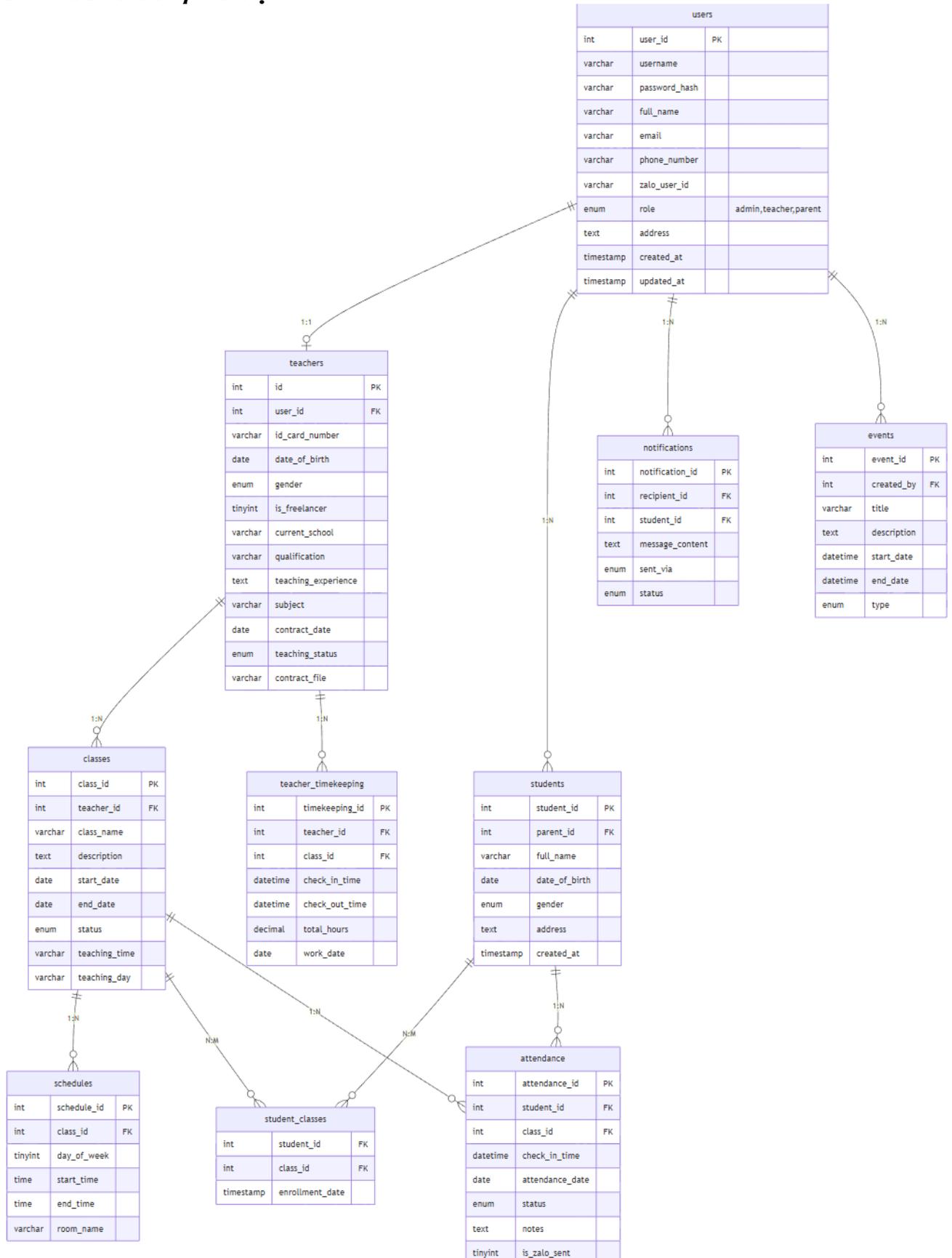
Hình 2.20. Biểu Đồ lớp UML.

2.5.2. Mô hình EER



Hình 2.21. Mô hình EER.

2.5.3. Mô hình quan hệ



Hình 2.22. Mô hình Quan hệ.

2.5.4. Mô tả chi tiết các bảng dữ liệu (Data Dictionary)

- Bảng users (Người dùng hệ thống) Bảng trung tâm quản lý tài khoản đăng nhập cho Quản trị viên, Giáo viên và Phụ huynh.
 - Mục đích: Xác thực người dùng và tích hợp định danh Zalo.

Tên trường	Kiểu dữ liệu	Mô tả
user_id	int(11)	Khóa chính (PK), mã định danh người dùng.
username	varchar(50)	Tên đăng nhập hệ thống.
password_hash	varchar(255)	Mật khẩu đã được mã hóa.
full_name	varchar(100)	Họ và tên đầy đủ.
email	varchar(100)	Địa chỉ Email.
phone_number	varchar(15)	Số điện thoại liên hệ.
zalo_user_id	varchar(100)	ID Zalo (Dùng để gửi thông báo qua Zalo OA).
role	enum	Vai trò: 'admin' (Quản trị), 'teacher' (Giáo viên), 'parent' (Phụ huynh).
address	text	Địa chỉ liên hệ.
created_at	timestamp	Thời gian tạo tài khoản.

- Bảng teachers (Hồ sơ Giáo viên) Lưu trữ thông tin chi tiết về chuyên môn và hợp đồng của giáo viên.

– Liên kết: user_id liên kết với bảng users.

Tên trường	Kiểu dữ liệu	Mô tả
id	int(11)	Khóa chính (PK).
user_id	int(11)	Khóa ngoại (FK) trỏ tới bảng users.
id_card_number	varchar(20)	Số CCCD/CMND.
date_of_birth	date	Ngày tháng năm sinh.
gender	enum	Giới tính ('Nam', 'Nữ', 'male', 'female', 'other').
is_freelancer	tinyint(1)	Trạng thái: 1 là giáo viên tự do, 0 là cơ hữu.
current_school	varchar(255)	Trường học hiện tại đang công tác (nếu có).
qualification	varchar(255)	Trình độ chuyên môn/Bằng cấp.
subject	varchar(100)	Môn giảng dạy chính.
teaching_status	enum	Trạng thái: 'active', 'inactive', 'on_leave'.

➤ Bảng students (Hồ sơ Học sinh) Quản lý thông tin học sinh.

– Liên kết: parent_id liên kết với bảng users (vai trò phụ huynh).

Tên trường	Kiểu dữ liệu	Mô tả
------------	--------------	-------

student_id	int(11)	Khóa chính (PK).
full_name	varchar(100)	Họ tên học sinh.
date_of_birth	date	Ngày sinh.
gender	enum	Giới tính.
parent_id	int(11)	Khóa ngoại (FK) trỏ tới users (Tài khoản phụ huynh).
address	text	Địa chỉ thường trú.

- Bảng classes (Lớp học) Quản lý thông tin các lớp học được mở tại trung tâm.

Tên trường	Kiểu dữ liệu	Mô tả
class_id	int(11)	Khóa chính (PK).
class_name	varchar(100)	Tên lớp học.
teacher_id	int(11)	Khóa ngoại (FK) trỏ tới bảng teachers.
start_date	date	Ngày bắt đầu khóa học.
end_date	date	Ngày kết thúc dự kiến.
status	enum	Trạng thái: 'active' (Đang học), 'completed', 'cancelled'.
teaching_time	varchar(100)	Thời gian học (Text hiển thị).
teaching_day	varchar(100)	Ngày học trong tuần (Text hiển thị).

- Bảng student_classes (Danh sách lớp học sinh tham gia) Bảng trung gian giải quyết mối quan hệ nhiều-nhiều giữa Học sinh và Lớp học.

Tên trường	Kiểu dữ liệu	Mô tả
student_id	int(11)	Khóa ngoại trở tới students.
class_id	int(11)	Khóa ngoại trở tới classes.
enrollment_date	timestamp	Ngày học sinh bắt đầu tham gia lớp.

- Bảng schedules (Thời khóa biểu chi tiết) Lưu trữ lịch học cụ thể từng buổi của lớp học để phục vụ việc xếp lịch và check trùng lịch.

Tên trường	Kiểu dữ liệu	Mô tả
schedule_id	int(11)	Khóa chính.
class_id	int(11)	Khóa ngoại trở tới classes.
day_of_week	tinyint(4)	Thứ trong tuần (2-8 hoặc 0-6).
start_time	time	Giờ bắt đầu ca học.
end_time	time	Giờ kết thúc ca học.
room_name	varchar(50)	Tên phòng học.

- Bảng attendance (Điểm danh học sinh) Lưu lịch sử điểm danh của học sinh, hỗ trợ tính năng gửi thông báo Zalo tự động.

Tên trường	Kiểu dữ liệu	Mô tả
attendance_id	int(11)	Khóa chính.

class_id	int(11)	Lớp học.
student_id	int(11)	Học sinh được điểm danh.
attendance_date	date	Ngày điểm danh.
check_in_time	datetime	Thời gian điểm danh thực tế.
status	enum	Trạng thái: 'present' (Có mặt), 'absent' (Vắng), 'late' (Muộn).
is_zalo_sent	tinyint(1)	Cờ đánh dấu: 1 = Đã gửi thông báo Zalo cho phụ huynh.

- Bảng teacher_timekeeping (Châm công giáo viên) Ghi nhận giờ dạy thực tế của giáo viên để tính lương.

Tên trường	Kiểu dữ liệu	Mô tả
timekeeping_id	int(11)	Khóa chính.
teacher_id	int(11)	Giáo viên.
class_id	int(11)	Dạy lớp nào.
check_in_time	datetime	Giờ vào lớp.
check_out_time	datetime	Giờ tan lớp.
total_hours	decimal(4,2)	Tổng số giờ dạy đã quy đổi.

- Bảng events (Sự kiện & Tin tức) Quản lý các sự kiện, thông báo chung của trung tâm.

Tên trường	Kiểu dữ liệu	Mô tả
------------	--------------	-------

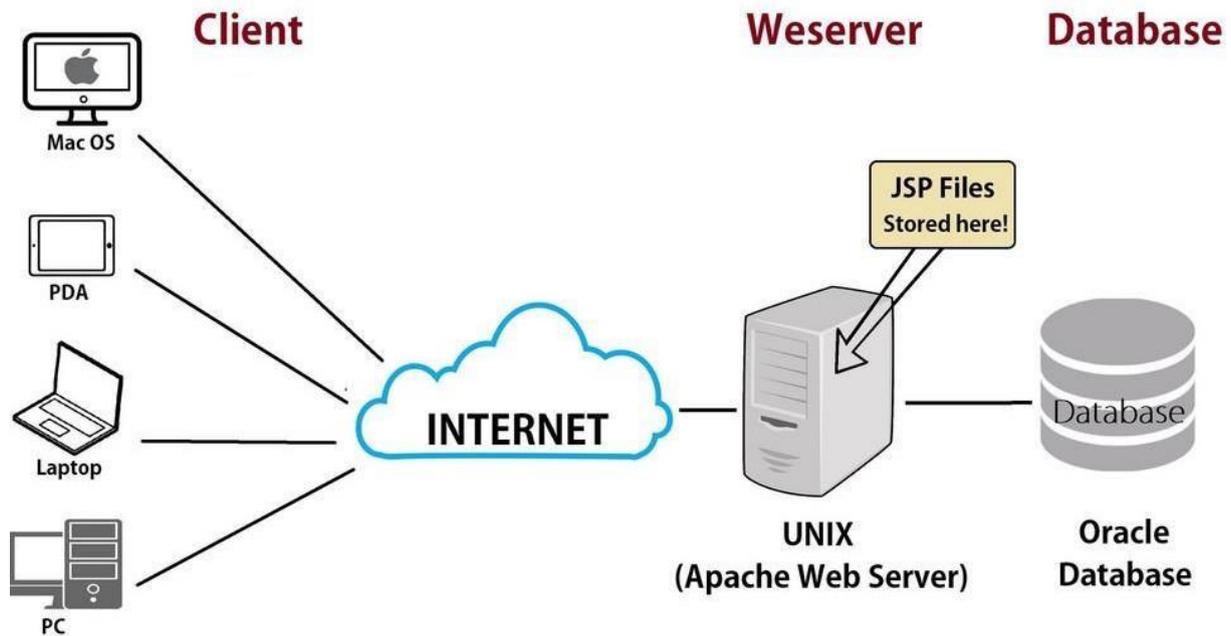
event_id	int(11)	Khóa chính.
title	varchar(255)	Tiêu đề sự kiện.
start_date	datetime	Thời gian bắt đầu sự kiện.
type	enum	Loại: 'event' (Sự kiện), 'notification' (Thông báo).
notification_sent	tinyint(1)	Trạng thái đã gửi thông báo hàng loạt hay chưa.

- Bảng notifications (Lịch sử gửi thông báo) Lưu log các tin nhắn đã gửi đi (qua Zalo, SMS, App) để đối soát.

Tên trường	Kiểu dữ liệu	Mô tả
notification_id	int(11)	Khóa chính.
recipient_id	int(11)	Người nhận (User ID).
sent_via	enum	Kênh gửi: 'zalo', 'sms', 'app'.
status	enum	Trạng thái gửi: 'success', 'failed'.

CHƯƠNG III: CƠ SỞ LÝ THUYẾT

3.1. Mô hình Client – Server trong phát triển web



Hình 3.1: Mô hình Client – Server-Data

Trong lĩnh vực phát triển phần mềm nói chung và lập trình web nói riêng, mô hình Client – Server là một kiến trúc cơ bản và phổ biến nhất. Kiến trúc này phân chia rõ ràng vai trò giữa hai thành phần chính: **Client** (máy khách) và **Server** (máy chủ), giúp hệ thống hoạt động hiệu quả, linh hoạt và dễ dàng mở rộng.

- **Client (Máy khách):** Là phía người dùng tương tác trực tiếp, trong dự án này chính là trình duyệt web mà người dùng sử dụng để truy cập website. Client có nhiệm vụ gửi yêu cầu (request) đến Server thông qua giao thức HTTP/HTTPS. Giao diện người dùng của dự án được phát triển bằng công nghệ **ReactJS**, kết hợp với HTML, CSS và JavaScript để tạo ra trải nghiệm tương tác và linh hoạt.

- **Server (Máy chủ):** Là nơi xử lý logic nghiệp vụ, tương tác với cơ sở dữ liệu và phản hồi lại các yêu cầu từ Client. Trong dự án này, Server được xây dựng bằng Node.js và framework Express.js. Server chịu trách nhiệm xử lý các yêu-cầu như xác thực người dùng, quản lý nhà hàng, xử lý đơn đặt hàng và truy vấn dữ liệu từ cơ sở dữ liệu.

3.1.1. Khái niệm về mô hình Client – Server

Mô hình Client – Server là một mô hình phân tán, trong đó Client là bên gửi yêu cầu dịch vụ và Server là bên cung cấp dịch vụ đó. Khi người dùng thực hiện một hành động trên website (ví dụ: **xem danh sách nhà hàng, đăng nhập, đặt món ăn**), các hành động đó sẽ được trình duyệt (Client) gửi dưới dạng yêu cầu HTTP tới Server.

Sau khi nhận và xử lý yêu cầu, Server sẽ gửi phản hồi (response) về cho Client, thường dưới dạng dữ liệu **JSON**. Client sẽ nhận dữ liệu này và sử dụng nó để cập nhật giao diện, hiển thị thông tin tương ứng cho người dùng.

Một đặc điểm quan trọng của mô hình này là Client không trực tiếp truy cập vào cơ sở dữ liệu. Toàn bộ tương tác với dữ liệu đều phải thông qua các API do Server cung cấp, giúp đảm bảo tính bảo mật, toàn vẹn và quản lý dữ liệu một cách tập trung.

3.1.2. Vai trò và chức năng của Client

Client là phần giao diện người dùng (UI – User Interface), nơi người dùng tương tác trực tiếp với hệ thống. Trong dự án này, phần Client được xây dựng bằng **React.js** – một thư viện JavaScript mạnh mẽ để xây dựng các giao diện người dùng có tính tương tác cao.

- Các chức năng chính của Client trong dự án bao gồm:
 - Hiển thị giao diện thân thiện, cho phép người dùng xem danh sách nhà hàng, thực đơn, và quản lý giỏ hàng.

- Gửi các yêu cầu HTTP đến Server khi người dùng thực hiện các hành động như đăng nhập, đặt hàng, hoặc tìm kiếm. Các phương thức phổ biến được dùng là GET, POST, PUT, DELETE.
 - Nhận dữ liệu (thường ở dạng JSON) từ Server và cập nhật lại giao diện một cách linh hoạt mà không cần tải lại trang.
 - Thực hiện điều hướng giữa các trang (ví dụ: từ trang chủ đến trang chi tiết nhà hàng), quản lý trạng thái của ứng dụng (state), và xử lý các sự kiện từ người dùng.
- Với việc sử dụng công nghệ Single Page Application (SPA), ứng dụng chỉ tải trang một lần duy nhất và sau đó cập nhật nội dung một cách linh hoạt, mang đến trải nghiệm mượt mà và nhanh chóng cho người dùng.

3.1.3. Vai trò và chức năng của Server

Server là nơi tiếp nhận, xử lý các yêu cầu từ Client và gửi phản hồi tương ứng. Trong dự án này, Server được xây dựng bằng **Node.js** kết hợp với framework **Express.js**.

- Các chức năng chính của Server bao gồm:
- Xử lý logic nghiệp vụ (business logic) như: xử lý đăng ký/đăng nhập, quản lý gi-ỏ hàng, tạo và quản lý đơn hàng, phê duyệt nhà hàng mới.
 - Quản lý và tương tác với cơ sở dữ liệu (thêm, sửa, xóa, truy vấn dữ liệu liên quan đến người dùng, nhà hàng, món ăn, đơn hàng).
 - Cung cấp các API theo tiêu chuẩn RESTful để Client có thể dễ dàng giao tiếp và sử dụng tài nguyên.
 - Bảo mật hệ thống bằng các phương pháp xác thực (authentication) để xác định danh tính người dùng và phân quyền (authorization) để

kiểm soát quyền truy cập tài nguyên (ví dụ: chỉ Admin mới có quyền duyệt nhà hàng).

- Xử lý việc tải lên và lưu trữ các tệp tin (uploads) như hình ảnh đại diện của nhà hàng hoặc món ăn.
- Server hoạt động trên một cổng (port) riêng, liên tục lắng nghe các yêu cầu HTTP đến và gửi phản hồi dữ liệu về cho Client.

3.1.4. Nguyên lý hoạt động

- Server hoạt động trên một cổng (port) riêng, liên tục lắng nghe các yêu cầu HTTP đến và gửi phản hồi dữ liệu về cho Client.
 - Người dùng thực hiện thao tác trên giao diện, ví dụ: nhấn nút "Thêm vào giỏ hàng" cho một món ăn.
 - Client (ReactJS) sẽ tạo một yêu cầu HTTP (cụ thể là phương thức POST) chứa thông tin về món ăn đó và gửi đến Server.
 - Server (Node.js/Express) nhận yêu cầu, xác thực người dùng, sau đó thực hiện xử lý logic: tìm thông tin người dùng và món ăn, rồi thêm món ăn đó vào giỏ hàng tương ứng trong cơ sở dữ liệu.
 - Sau khi xử lý xong, Server gửi lại một phản hồi HTTP (HTTP response), thường kèm theo dữ liệu giỏ hàng đã được cập nhật ở định dạng JSON.
 - Client nhận phản hồi này và cập nhật lại giao diện người dùng, ví dụ: cập nhật số lượng sản phẩm trên biểu tượng giỏ hàng.

3.1.5. Ưu điểm của mô hình Client – Server

- **Tính phân tách rõ ràng:** Tách biệt hoàn toàn giữa giao diện (Frontend) và phần xử lý logic (Backend), giúp cho việc phát triển, bảo trì và nâng cấp từng phần trở nên độc lập và dễ dàng hơn.

- **Tái sử dụng cao:** Backend API được xây dựng có thể phục vụ cho nhiều loại Client khác nhau trong tương lai (ví dụ: ứng dụng di động, ứng dụng desktop) mà không cần xây dựng lại logic.
- **Dễ mở rộng (Scalability):** Khi hệ thống có nhiều người dùng hơn, chúng ta có thể dễ dàng nâng cấp hiệu năng của Server (scale-up) hoặc triển khai thêm nhiều máy chủ (scale-out) mà không ảnh hưởng đến Client.
- **Bảo mật tập trung:** Toàn bộ dữ liệu và logic nghiệp vụ quan trọng được quản lý tập trung tại Server, giúp việc kiểm soát truy cập và bảo vệ dữ liệu trở nên hiệu quả hơn.

3.2. Frontend

3.2.1. React.js

Khái niệm

React.js (hay React) là một thư viện JavaScript mã nguồn mở do Facebook phát triển, ra mắt lần đầu vào năm 2013. React được sử dụng để xây dựng giao diện người dùng (UI) cho các ứng dụng web. Thay vì thao tác trực tiếp với DOM (Document Object Model) của trình duyệt, React sử dụng một cơ chế gọi là **Virtual DOM** để tối ưu hóa hiệu suất và tốc độ cập nhật giao diện.

React không phải là một framework toàn diện như Angular, mà chỉ tập trung vào lớp View (giao diện) trong mô hình MVC, cho phép lập trình viên dễ dàng tạo ra các thành phần giao diện có khả năng tái sử dụng cao.

Đặc điểm nổi bật của React.js

- **Virtual DOM:** React tạo ra một bản sao của DOM thật trong bộ nhớ. Khi trạng thái (state) của một component thay đổi, React sẽ tính toán sự khác biệt và chỉ cập nhật những phần cần thiết trên DOM thật, giúp ứng dụng hoạt động nhanh và mượt mà hơn.

- **Kiến trúc dựa trên Component:** Giao diện trong React được chia thành các thành phần (component) độc lập và có thể tái sử dụng. Mỗi component quản lý trạng thái riêng, giúp mã nguồn trở nên rõ ràng, dễ quản lý và bảo trì.
- **JSX (JavaScript XML):** Là một phần mở rộng của JavaScript, cho phép viết mã trông giống HTML ngay trong các tệp JavaScript. JSX giúp việc mô tả giao diện trở nên trực quan và dễ hiểu hơn.
- **Ràng buộc dữ liệu một chiều (One-Way Data Binding):** Dữ liệu trong React chảy theo một chiều, từ component cha xuống các component con. Luồng dữ liệu này giúp việc theo dõi và gỡ lỗi trở nên dễ dàng hơn.
- **Hooks:** Kể từ phiên bản 16.8, Hooks (ví dụ: useState, useEffect, useContext) cho phép sử dụng state và các tính năng khác của React trong các function component mà không cần viết class component.

Ứng dụng của React trong chuyên đề

Trong dự án website nhà hàng này, React.js được sử dụng để:

- Xây dựng toàn bộ giao diện người dùng, bao gồm các trang như: Trang chủ, Trang chi tiết nhà hàng, Giỏ hàng, Đăng nhập, Đăng ký, Trang thông tin người dùng, và các trang quản lý dành cho Chủ nhà hàng và Admin.
- Tạo ra các component có khả năng tái sử dụng cao như: Header, Footer, RestaurantCard (thẻ nhà hàng), MenuItemCard (thẻ món ăn), Sidebar (thanh điều hướng bên), Modal (hộp thoại)...
- Sử dụng thư viện **Axios** để gửi yêu cầu HTTP và nhận dữ liệu từ Backend API.
- Sử dụng **React Router DOM** (react-router-dom) để xử lý việc điều hướng, chuyển trang trong ứng dụng.

- Quản lý trạng thái toàn cục của ứng dụng, chẳng hạn như thông tin người dùng đang đăng nhập (AuthContext) và dữ liệu giỏ hàng (CartContext) bằng **Context API** và hook useContext.

3.2.2. React Router

Khái niệm

React Router là một thư viện định tuyến (routing) tiêu chuẩn cho các ứng dụng React. Nó cho phép xây dựng các ứng dụng đơn trang (SPA) với trải nghiệm điều hướng mượt mà như một trang web đa trang truyền thống. Bằng cách đồng bộ giao diện người dùng với URL của trình duyệt, React Router giúp quản lý việc di chuyển giữa các component khác nhau mà không cần tải lại trang.

Cách hoạt động

React Router theo dõi sự thay đổi của URL trong trình duyệt, sau đó ánh xạ các đường dẫn (path) này với các component React tương ứng đã được cấu hình để hiển thị. Nó sử dụng History API của trình duyệt để quản lý lịch sử điều hướng, cho phép người dùng sử dụng các nút back/forward một cách tự nhiên

Các thành phần chính được sử dụng trong dự án:

- `<BrowserRouter>`: Component bao bọc toàn bộ ứng dụng để kích hoạt chức năng định tuyến.
- `<Routes>`: Component dùng để nhóm các định nghĩa `<Route>`.
- `<Route>`: Định nghĩa một tuyến đường, ánh xạ một path (đường dẫn URL) tới một element (component) cụ thể.
- `<Link>`: Component thay thế cho thẻ `<a>` truyền thống, cho phép chuyển hướng giữa các trang mà không gây tải lại trang.
- Các Hooks: `useNavigate()` để điều hướng theo lập trình, `useParams()` để lấy các tham số động từ URL (ví dụ: ID của nhà hàng), và `useLocation()` để truy cập thông tin về URL hiện tại.

Ứng dụng trong đồ án

Trong dự án website nhà hàng, React Router được sử dụng để:

- Tạo hệ thống định tuyến cho toàn bộ ứng dụng với các trang chính như:
 - /: Trang chủ hiển thị danh sách nhà hàng.
 - /restaurants/:id: Trang chi tiết của một nhà hàng cụ thể.
 - /cart: Trang giỏ hàng.
 - /login, /register: Các trang đăng nhập, đăng ký.
 - /profile, /orders: Các trang quản lý thông tin cá nhân và lịch sử đơn hàng của người dùng.
 - Các tuyến đường được bảo vệ cho Admin và Chủ nhà hàng (ví dụ: /admin/dashboard, /owner/menu-manager).
- Sử dụng `useNavigate()` để tự động chuyển hướng người dùng sau khi thực hiện các hành động như **đăng nhập, đăng ký thành công, hoặc sau khi đặt hàng**.
- Sử dụng `useParams()` để lấy id của nhà hàng từ URL, từ đó gọi API để tải và hiển thị đúng thông tin chi tiết của nhà hàng đó.

3.2.3. Context API

Khái niệm

Context API là một cơ chế được tích hợp sẵn trong React, cho phép truyền dữ liệu giữa các component mà không cần phải sử dụng phương pháp truyền props qua từng cấp (một vấn đề thường được gọi là "prop drilling"). Thay vì truyền dữ liệu qua nhiều lớp component trung gian, Context API cho phép chia sẻ "trạng thái toàn cục" (global state) từ một nơi trung tâm đến bất kỳ component nào trong cây component có nhu cầu sử dụng.

Context API đặc biệt hữu ích để quản lý những dữ liệu cần được truy cập ở nhiều nơi trong ứng dụng, ví dụ như: thông tin người dùng đang đăng nhập, trạng thái giỏ hàng, chủ đề giao diện (theme), hoặc ngôn ngữ hiển thị.

Ưu điểm của Context API

- Giúp mã nguồn sạch hơn bằng cách loại bỏ việc truyền props không cần thiết.
- Cung cấp một giải pháp quản lý trạng thái đơn giản và hiệu quả cho các ứng dụng vừa và nhỏ.
- Là một tính năng có sẵn trong React, không cần cài đặt thêm thư viện bên ngoài như Redux.
- Dễ dàng kết hợp với React Hooks (cụ thể là hook useContext) để giúp việc truy cập dữ liệu từ Context trở nên ngắn gọn và trực quan.

Ứng dụng trong dự án

Trong dự án website nhà hàng, Context API đóng vai trò trung tâm trong việc quản lý trạng thái của ứng dụng, cụ thể qua hai context chính:

- **AuthContext ([AuthContext.js](#)):**
 - Quản lý trạng thái xác thực của người dùng trên toàn bộ ứng dụng (đã đăng nhập hay chưa, thông tin người dùng, vai trò - user/owner/admin).
 - Cung cấp thông tin này cho các component khác như Header (để hiển thị nút đăng nhập/đăng xuất) và ProtectedRoute (để kiểm tra quyền truy cập vào các trang quản trị).
- **CartContext ([CartContext.js](#)):**
 - Quản lý toàn bộ dữ liệu của giỏ hàng, bao gồm danh sách các món ăn, số lượng, và tổng tiền.

- Cho phép các component khác nhau như MenuItemCard (nút "Thêm vào giỏ"), CartPage (trang giỏ hàng), và Header (hiển thị số lượng món trong giỏ) có thể cùng lúc truy cập và thay đổi dữ liệu giỏ hàng một cách nhất quán.

Việc áp dụng Context API giúp giảm độ phức tạp trong việc quản lý và đồng bộ hóa dữ liệu, làm cho mã nguồn trở nên dễ bảo trì và mở rộng hơn.

3.3. Backend

3.3.1. Node.js

Khái niệm

Node.js là một môi trường thực thi (runtime environment) cho phép chạy mã JavaScript ở phía máy chủ (server-side), được xây dựng trên V8 JavaScript Engine của Google. Khác với JavaScript truyền thống chỉ hoạt động trên trình duyệt, Node.js cho phép lập trình viên sử dụng JavaScript để xây dựng các ứng dụng backend mạnh mẽ, ví dụ như RESTful API, ứng dụng thời gian thực, và microservices.

Node.js sử dụng mô hình I/O không chặn (non-blocking I/O) và kiến trúc hướng sự kiện (event-driven), giúp nó có khả năng xử lý hàng nghìn kết nối đồng thời một cách hiệu quả, rất phù hợp cho các ứng dụng web đòi hỏi hiệu suất cao.

Ứng dụng trong dự án

Trong dự án website nhà hàng này, Node.js đóng vai trò là nền tảng chính cho toàn bộ hệ thống backend, với các nhiệm vụ:

- Xây dựng một hệ thống RESTful API hoàn chỉnh để phía frontend (React) có thể giao tiếp, gửi và nhận dữ liệu.
- Xử lý toàn bộ logic nghiệp vụ của ứng dụng, bao gồm: **đăng ký, đăng nhập, quản lý người dùng, quản lý nhà hàng, món ăn, giỏ hàng và xử lý đơn hàng.**

- Tương tác với cơ sở dữ liệu **MySQL** thông qua một ORM (Object-Relational Mapping) là Sequelize để thực hiện các thao tác truy vấn và cập nhật dữ liệu.
- Quản lý các tuyến đường (routes) của API một cách rõ ràng và có tổ chức.

3.3.2. Express.js

Khái niệm

Express.js là một framework web tối giản và linh hoạt được xây dựng trên nền tảng Node.js. Nó cung cấp một bộ công cụ mạnh mẽ để phát triển các ứng dụng web và API một cách nhanh chóng. Express là thành phần chính trong ngăn xếp công nghệ **MERN/MEVN** (MongoDB/MySQL, Express, React/Vue, Node.js) và là một trong những framework phổ biến nhất trong hệ sinh thái Node.js

Ưu điểm của Express.js

- Đơn giản và linh hoạt: Cung cấp các tính năng cốt lõi mà không áp đặt một cấu trúc cứng nhắc, cho phép lập trình viên tự do tổ chức dự án.
- Quản lý Route mạnh mẽ: Dễ dàng định nghĩa các điểm cuối (endpoint) cho API, hỗ trợ tất cả các phương thức HTTP và các mẫu URL động.
- Hệ thống Middleware: Đây là tính năng nổi bật nhất của Express. Middleware là các hàm được thực thi tuần tự trong chu trình xử lý request, cho phép thực hiện các tác vụ như xác thực, ghi log, kiểm tra dữ liệu đầu vào, và xử lý lỗi.
- Cộng đồng lớn: Có một hệ sinh thái rộng lớn các thư viện và công cụ hỗ trợ.

Ứng dụng trong chuyên đề

Trong dự án này, Express.js được sử dụng để:

- Định nghĩa và tổ chức toàn bộ các API endpoint để quản lý người dùng, nhà hàng, thực đơn, và các đơn đặt hàng.
- Sử dụng các middleware để xử lý yêu cầu. Ví dụ:
 - Middleware isAuth để kiểm tra xem người dùng đã đăng nhập hay chưa bằng cách xác thực JSON Web Token (JWT).
 - Middleware isAdmin và isOwner để thực hiện phân quyền, đảm bảo chỉ những người dùng có vai trò phù hợp mới có thể truy cập các tài nguyên quản trị.
- Kết nối với cơ sở dữ liệu MySQL để truy xuất và cập nhật dữ liệu theo yêu cầu từ client.
- Xử lý việc tải lên các tệp tin như hình ảnh món ăn, hình ảnh nhà hàng.

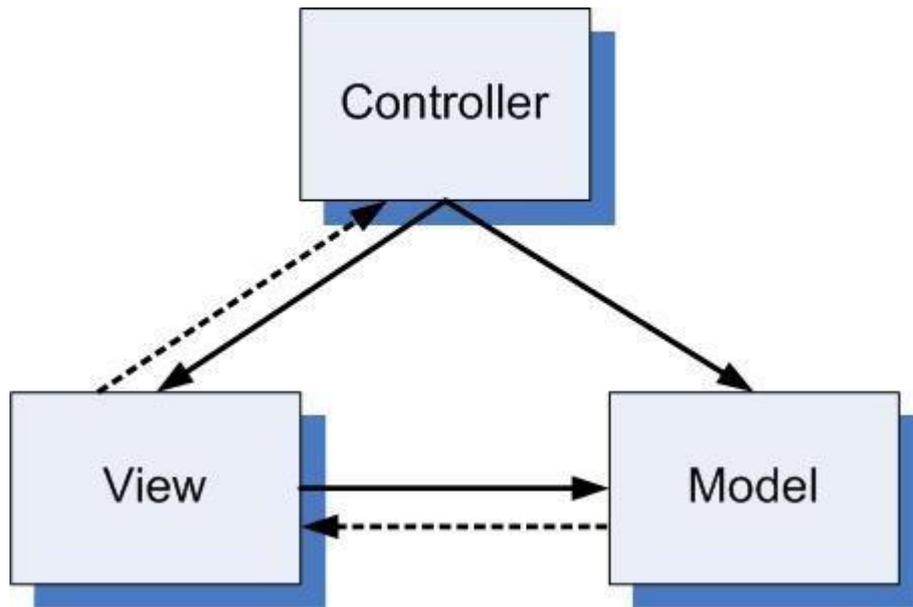
3.3.3. Mô hình kiến trúc Backend

Dự án áp dụng một kiến trúc phần mềm phía backend dựa trên mô hình MVC (Model - View - Controller) được biến thể để phù hợp với việc xây dựng RESTful API. Cấu trúc này giúp phân tách rõ ràng các thành phần của ứng dụng, làm cho mã nguồn trở nên có tổ chức, dễ đọc và dễ bảo trì.

Các thành phần được tổ chức như sau:

- **Model:**
 - Vị trí: Thư mục backend/models.
 - Nhiệm vụ: Định nghĩa cấu trúc dữ liệu của các đối tượng trong ứng dụng (ví dụ: User, Restaurant, Order). Model chịu trách nhiệm cho tất cả các tương tác với cơ sở dữ liệu như tạo, đọc, cập nhật, và xóa (CRUD). Trong dự án này, các model của Sequelize định nghĩa bảng và các mối quan hệ trong DB.
- **Controller:**

- **Vị trí:** Thư mục backend/controllers.
 - **Nhiệm vụ:** Chứa đựng logic nghiệp vụ chính của ứng dụng. Mỗi controller nhận yêu cầu được chuyển tiếp từ router, sử dụng các hàm trong **Model** để xử lý dữ liệu, sau đó chuẩn bị và gửi phản hồi về cho client. Ví dụ, auth.controller.js xử lý logic đăng ký và đăng nhập.
- **View:**
- **Khái niệm:** Trong kiến trúc REST API, không có "View" truyền thống tạo ra giao diện HTML. Thay vào đó, View chính là dữ liệu được gửi về cho client, thường ở định dạng JSON. Controller sau khi xử lý xong sẽ gửi một phản hồi JSON, và phía client (React) sẽ chịu trách nhiệm "vẽ" ra giao diện người dùng dựa trên dữ liệu đó.
- **Routes:**
- **Vị trí:** Thư mục backend/routes.
 - **Nhiệm vụ:** Đóng vai trò điều hướng. Nó định nghĩa các URL endpoint và các phương thức HTTP tương ứng (GET, POST, PUT, DELETE), sau đó ánh xạ mỗi yêu cầu đến một hàm xử lý (handler) cụ thể trong Controller.



Hình 3.2: Mô hình MVC

3.3.4. Middleware

Khái niệm

Middleware là một khái niệm phổ biến trong lập trình backend, đặc biệt trong các framework như Express.js. Middleware có thể hiểu là một hàm trung gian được gọi trong quá trình xử lý HTTP request và response. Nó có nhiệm vụ can thiệp, kiểm tra, xử lý dữ liệu trước khi request được chuyển tiếp đến route chính hoặc khi response được trả về client.

Trong Express.js, middleware có thể thực hiện nhiều chức năng như:

- Kiểm tra xác thực người dùng (authentication).
- Ghi log thông tin yêu cầu.
- Phân tích body của request (body parsing).
- Xử lý lỗi.
- Hạn chế số lượng truy cập (rate limiting), kiểm tra quyền (authorization), hoặc xác minh token (JWT).

Cơ chế hoạt động

Middleware hoạt động theo chuỗi: Mỗi middleware sau khi xử lý xong có thể gọi next() để chuyển sang middleware tiếp theo. Nếu không gọi next(), quá trình sẽ dừng lại tại middleware đó. Điều này giúp kiểm soát linh hoạt luồng xử lý của ứng dụng.

Ưu điểm của Middleware

Tái sử dụng cao: Một middleware có thể áp dụng cho nhiều route.

Tăng bảo mật và kiểm soát: Nhờ có thể kiểm tra token, quyền truy cập, và trạng thái request trước khi xử lý chính.

Giúp mã nguồn rõ ràng, dễ tổ chức: Tách biệt các bước xử lý request thành từng phần riêng biệt.

Dễ mở rộng và bảo trì: Có thể thêm mới chức năng mà không ảnh hưởng logic cũ.

Ứng dụng trong chuyên đề

Trong chuyên đề xây dựng website bán hàng, Middleware đóng vai trò quan trọng trong:

- Xác thực người dùng thông qua JWT để đảm bảo chỉ người đã đăng nhập mới được thực hiện các thao tác như đặt thuê, xem thông tin tài khoản.
- Phân quyền người dùng, ví dụ: chỉ admin mới được thêm/sửa sản phẩm.
- Kiểm tra dữ liệu đầu vào (validate) như kiểm tra email, mật khẩu, tên sản phẩm...
- Xử lý lỗi chung, trả về thông báo lỗi rõ ràng và có định dạng cho frontend.
- Bảo vệ API khỏi các cuộc tấn công như spam hoặc gửi nhiều request bất hợp pháp.

Middleware giúp tăng độ bảo mật, hiệu quả và khả năng mở rộng của hệ thống backend.

3.3.5. *Kết nối Frontend – Backend*

Khái niệm

Trong kiến trúc ứng dụng web hiện đại, Frontend và Backend là hai hệ thống độc lập nhưng giao tiếp chặt chẽ với nhau.

- **Frontend (Client-side):** Là giao diện người dùng mà người dùng tương tác, được xây dựng bằng React.
- **Backend (Server-side):** Là máy chủ xử lý logic nghiệp vụ, quản lý dữ liệu và cung cấp API.

Quá trình kết nối giữa chúng được thực hiện thông qua giao thức HTTP, nơi Frontend gửi yêu cầu đến các API do Backend cung cấp để trao đổi dữ liệu.

Cơ chế kết nối phổ biến

- **Giao tiếp qua RESTful API:** Backend xây dựng một hệ thống API theo tiêu chuẩn REST. Frontend sử dụng thư viện Axios để gửi các yêu cầu HTTP (GET, POST, PUT, DELETE) đến các API này.
- **Trao đổi dữ liệu bằng JSON:** Dữ liệu được trao đổi giữa hai bên chủ yếu dưới định dạng JSON, một định dạng nhẹ và dễ dàng cho JavaScript xử lý.
- **Xử lý CORS (Cross-Origin Resource Sharing):** Vì Frontend (chạy trên một port, ví dụ localhost:3000) và Backend (chạy trên port khác, ví dụ localhost:5000) là hai "nguồn" khác nhau, Backend phải cấu hình middleware CORS để cho phép trình duyệt chấp nhận các yêu cầu từ Frontend.
- **Xác thực bằng Token (JWT):** Sau khi người dùng đăng nhập thành công, Backend sẽ cấp một JWT (JSON Web Token). Frontend lưu token

này lại (thường trong Local Storage) và gửi kèm nó trong header của mỗi yêu cầu cần xác thực sau đó.

Vai trò trong dự án chuyên đề

Việc kết nối này là xương sống của toàn bộ ứng dụng:

- **Tải dữ liệu động:** Khi người dùng truy cập trang chủ, Frontend gọi API để lấy và hiển thị **danh sách các nhà hàng** từ cơ sở dữ liệu.
- **Thực hiện hành động:** Khi người dùng **thêm một món ăn vào giỏ hàng** hoặc **tạo đơn hàng**, Frontend gửi thông tin đó đến API của Backend để xử lý và lưu trữ.
- **Đồng bộ trạng thái:** Thông tin người dùng, nội dung giỏ hàng, và trạng thái đơn hàng luôn được đồng bộ giữa giao diện và cơ sở dữ liệu thông qua các lượt gọi API.
- **Hiển thị thông báo và lỗi:** Các phản hồi từ Backend (thành công hay thất bại) giúp Frontend hiển thị các thông báo phù hợp cho người dùng.

Công cụ và công nghệ sử dụng

Frontend: React.js kết hợp với thư viện Axios để gửi HTTP requests.

Backend: Express.js xây dựng hệ thống API RESTful.

Token (JWT): Dùng để xác thực và phân quyền giữa frontend và backend.

CORS Middleware: Được sử dụng tại backend để cho phép frontend truy cập từ domain khác.

3.4. Cơ sở dữ liệu MySQL

3.4.1. Giới thiệu MySQL

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mã nguồn mở phổ biến, sử dụng Ngôn ngữ Truy vấn có Cấu trúc (SQL). Nó được sử dụng rộng

rãi trong các ứng dụng web nhờ tính ổn định, hiệu suất cao và khả năng tương thích tốt với các ngôn ngữ phía server như Node.js.

3.4.2. Đặc điểm nổi bật

- Dữ liệu được lưu trữ dưới dạng bảng (table) có cấu trúc.
- Hỗ trợ chuẩn SQL với đầy đủ các thao tác CRUD (Create, Read, Update, Delete).
- Cho phép liên kết dữ liệu giữa các bảng thông qua khóa ngoại (foreign key).
- Tính bảo mật cao, hỗ trợ phân quyền người dùng.
- Tương thích đa nền tảng (Windows, macOS, Linux).

3.4.3. Vai trò của MySQL trong dự án

Trong dự án xây dựng website nhà hàng, MySQL được dùng làm cơ sở dữ liệu chính, có nhiệm vụ:

- **Lưu trữ thông tin người dùng:** Tên, email, mật khẩu (đã mã hóa), vai trò (quyền truy cập)...
- **Quản lý nhà hàng và thực đơn:** Bao gồm thông tin về nhà hàng (tên, địa chỉ, ảnh) và các món ăn (tên, mô tả, giá, ảnh)...
- **Xử lý đơn hàng:** Lưu thông tin giỏ hàng, chi tiết đơn hàng, trạng thái đơn hàng và lịch sử đặt món.
- Các bảng chính thường có trong hệ thống gồm: users, restaurants, menuItems, orders, orderItems và cartItems.

3.5. Kiểm thử

3.5.1. Giới thiệu

Định nghĩa: Kiểm thử phần mềm (Software Testing) là hoạt động nhằm tìm kiếm và phát hiện lỗi của phần mềm. Đây là một tiến trình hoặc một tập hợp các

tiến trình được thiết kế để đảm bảo chương trình máy tính thực hiện đúng theo yêu cầu của khách hàng. Theo "The Art of Software Testing", kiểm thử phần mềm là quá trình thực thi một chương trình với mục đích tìm ra lỗi.

Mục tiêu: Phát hiện sớm các lỗi và tìm được càng nhiều lỗi càng tốt để tránh sai sót, sửa đi sửa lại nhiều lần. Ngăn chặn các lỗi phát sinh, hạn chế rủi ro cho hệ thống và chứng minh rằng sản phẩm phù hợp, hiệu quả và đúng với đặc tả của nó. Từ đó, chúng ta được rằng các trường hợp kiểm thử (test cases) và các báo cáo được tạo ra là đúng và hữu dụng.

3.5.2. Kiểm thử hộp đen (Black Box Testing)

Định nghĩa: Black box testing (Kiểm thử hộp đen) là một phương pháp kiểm thử phần mềm tập trung vào việc kiểm tra hành vi bên ngoài của một ứng dụng mà không quan tâm đến cấu trúc nội bộ hay chi tiết cài đặt của nó. Trong phương pháp này, người kiểm thử chỉ xem ứng dụng như một "hộp đen" và tập trung vào việc kiểm tra dữ liệu đầu vào, kết quả đầu ra, cũng như các tình huống và điều kiện biên của hệ thống.

Ưu điểm	Nhược điểm
<ul style="list-style-type: none"> – Phát hiện các lỗi và hành vi không mong muốn từ góc độ người dùng. – Giúp đảm bảo tính tương tác mượt mà và đáp ứng nhanh chóng, mang lại trải nghiệm người dùng tốt. – Người kiểm thử không cần biết về ngôn ngữ lập trình hay cấu trúc mã nguồn. – Việc kiểm thử có thể được thực hiện bởi một nhóm độc lập, đảm bảo tính khách quan. – Có thể tiết kiệm thời gian và nguồn lực vì không cần phân tích mã nguồn phức tạp. 	<ul style="list-style-type: none"> – Không thể đảm bảo độ phủ kiểm thử cao như White box testing. – Có thể bỏ qua các lỗi nằm sâu trong hệ thống như lỗi logic hoặc lỗi thiết kế. – Khó kiểm tra được các tình huống hoặc các đường đi (path) phức tạp bên trong mã nguồn. – Chất lượng kiểm thử phụ thuộc nhiều vào khả năng và kinh nghiệm của người kiểm thử; nếu thiếu kỹ năng có thể dẫn đến bỏ sót lỗi.

Bảng 0.1: Ưu nhược điểm của kiểm thử hộp đen

Các phương pháp được sử dụng:

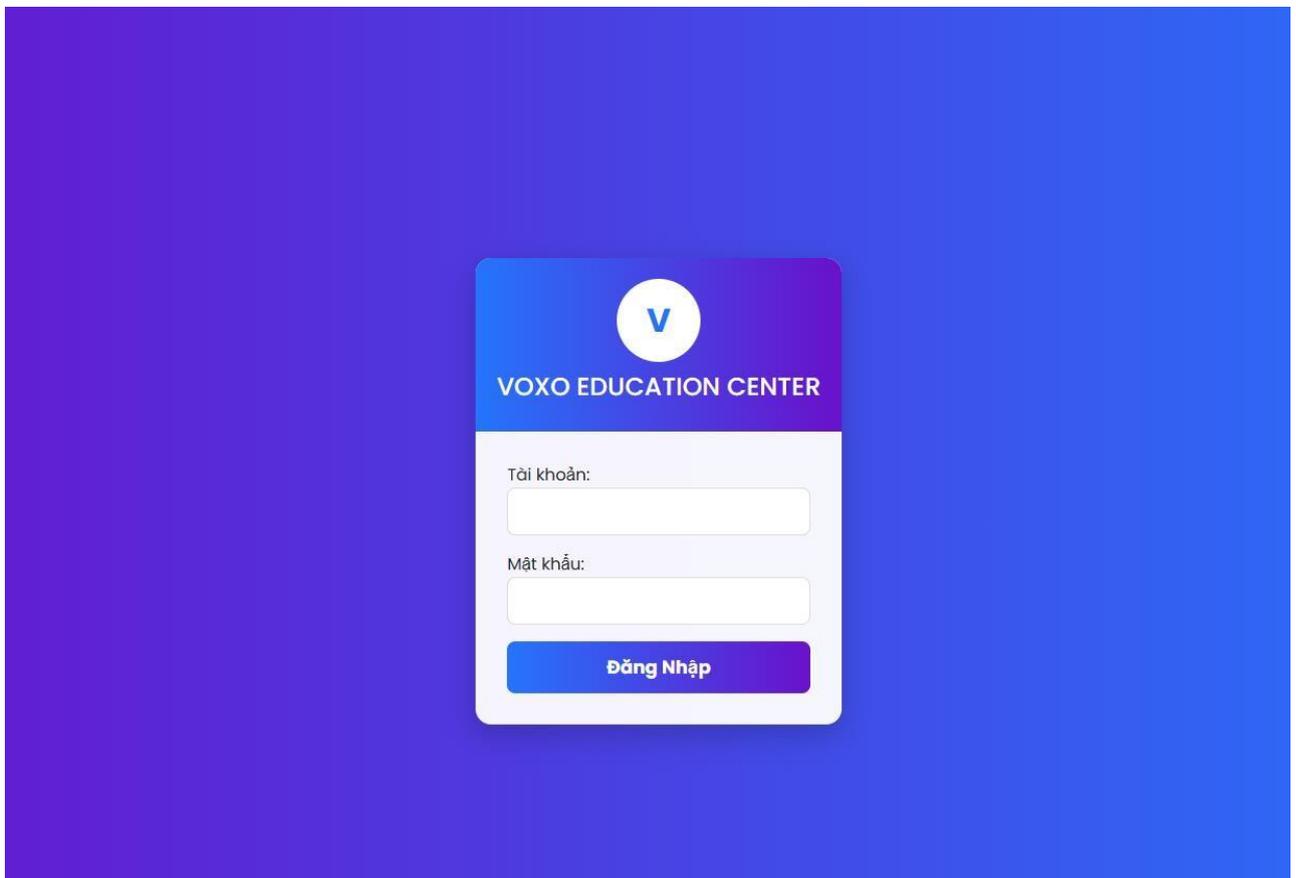
- **Equivalence Partitioning (Phân vùng tương đương):** Chia các giá trị đầu vào thành các nhóm tương đương và chỉ kiểm tra một giá trị đại diện trong mỗi nhóm. Điều này giúp giảm số lượng ca kiểm thử cần thực hiện mà vẫn đảm bảo bao quát các trường hợp quan trọng.
- **Boundary Value Analysis (Phân tích giá trị biên):** Tập trung vào việc kiểm tra các giá trị tại và gần các điểm biên của các vùng dữ liệu đầu vào, nơi thường xảy ra lỗi.
- **Decision Table Testing (Kiểm thử bảng quyết định):** Sử dụng các bảng quyết định để xác định các điều kiện và hành động tương ứng của ứng dụng. Các ca kiểm thử được tạo dựa trên các kết hợp của các điều kiện trong bảng.
- **State Transition Testing (Kiểm thử chuyển trạng thái):** Tập trung vào việc kiểm thử các quá trình chuyển đổi trạng thái của ứng dụng. Các ca kiểm thử được tạo dựa trên các trạng thái và các sự kiện gây ra sự chuyển đổi.
- **Error Guessing (Đoán lỗi):** Dựa trên kinh nghiệm và trực giác của người kiểm thử để dự đoán các lỗi có thể xảy ra trong ứng dụng.
- **Exploratory Testing (Kiểm thử khám phá):** Người kiểm thử đồng thời thiết kế và thực thi các bài kiểm thử để khám phá ứng dụng một cách tự do, không theo kịch bản cứng nhắc.
- **Regression Testing (Kiểm thử hồi quy):** Đảm bảo rằng các thay đổi hoặc sửa lỗi trong ứng dụng không gây ảnh hưởng tiêu cực đến các chức năng đã hoạt động tốt trước đó.

CHƯƠNG IV: Giao Diện Trang Web

Tiếp nối phần cơ sở lý thuyết và thiết kế, nội dung dưới đây tập trung mô tả trực quan các giao diện chính của website 'Quản lý trung tâm tổ chức dạy học'. Các hình ảnh minh họa sẽ làm rõ cách thức người dùng tương tác với phần mềm để giải quyết các bài toán quản lý đã đặt ra. Bên cạnh đó, phần này cũng đưa ra các thông số kỹ thuật cần thiết về phần cứng và phần mềm để cài đặt và sử dụng ứng dụng một cách ổn định.

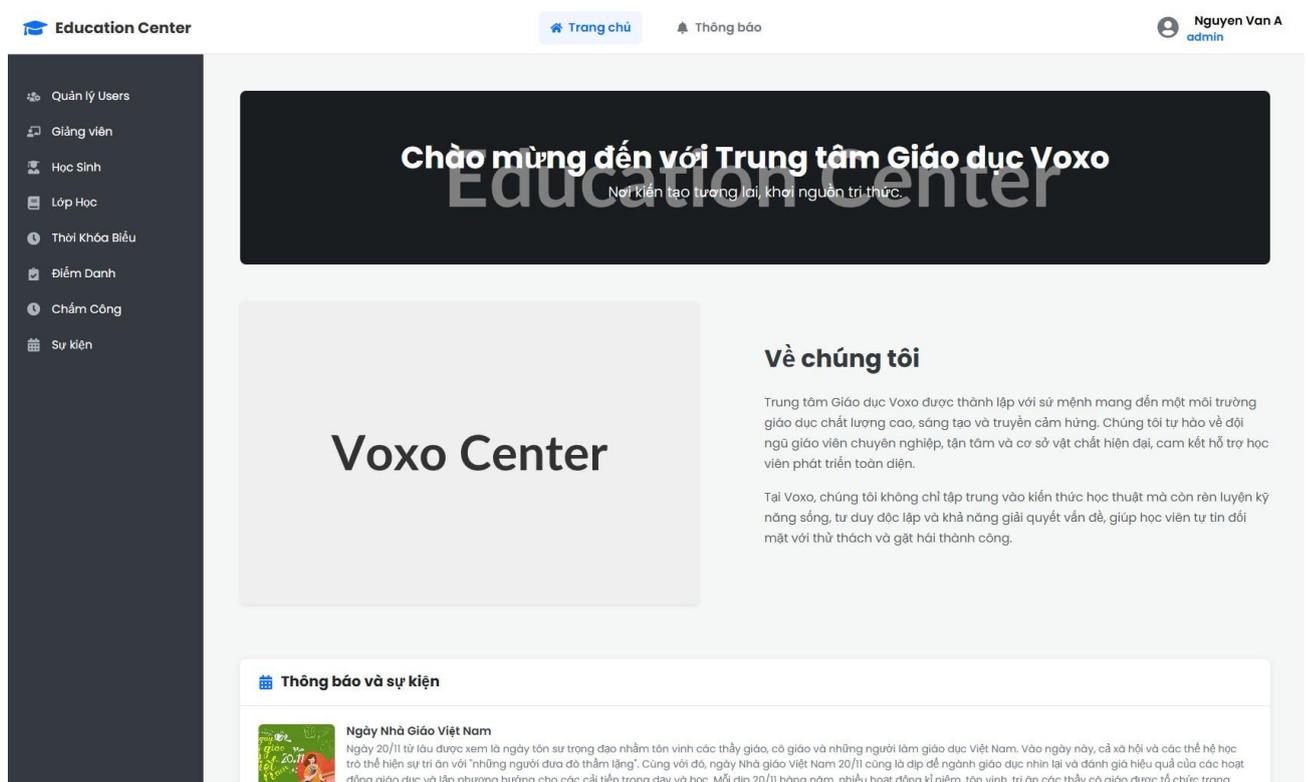
4.1. Giao diện website

4.1.1. Trang login



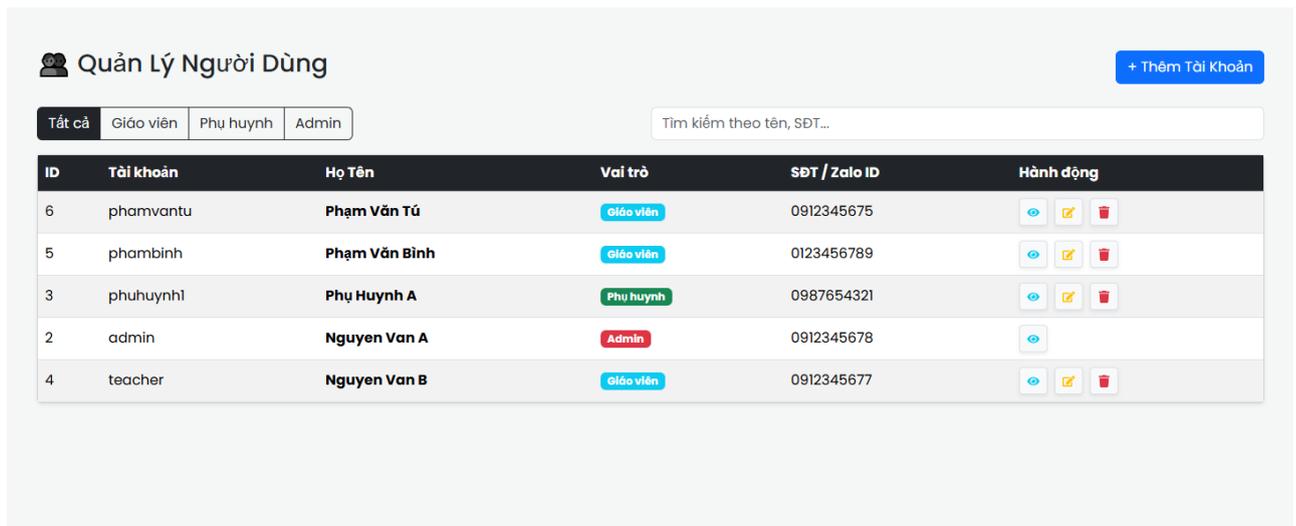
Hình 4.1: Giao diện trang login

4.1.2. Trang Home



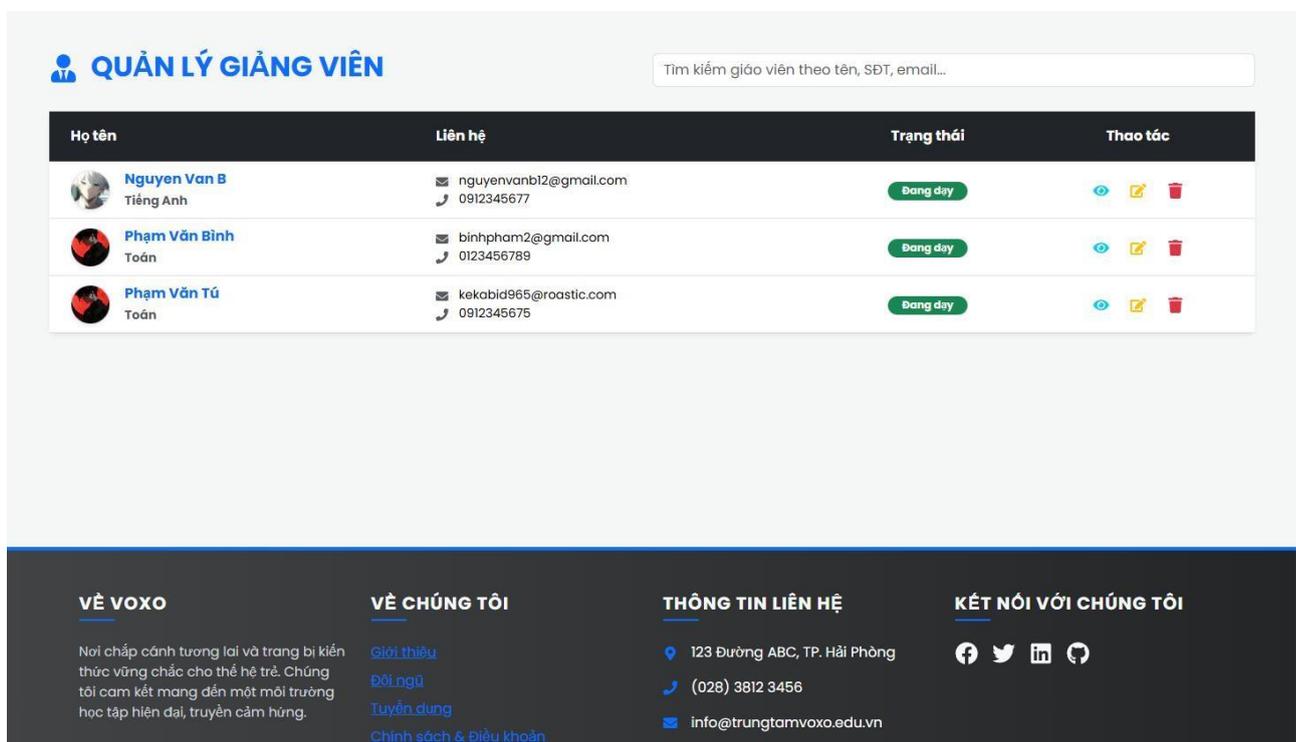
Hình 4.2: Giao diện trang Dashboard

4.1.3. Trang quản lý users



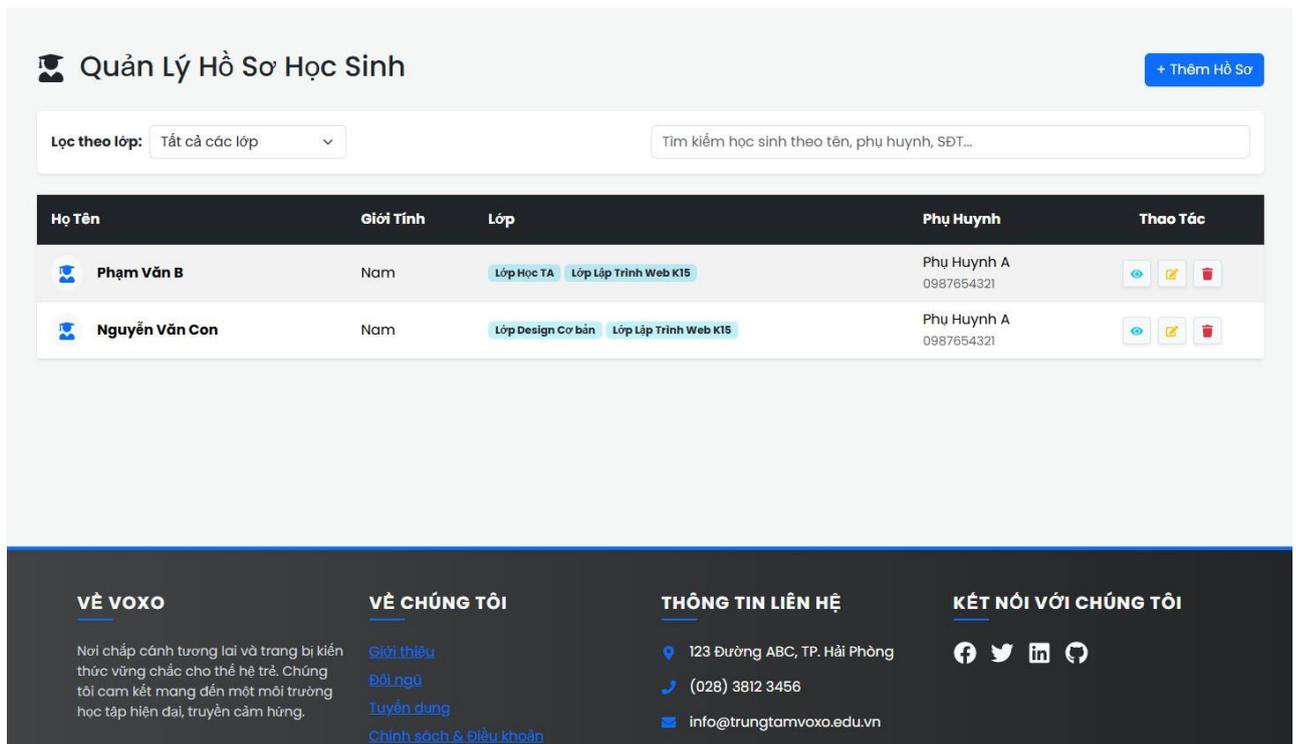
Hình 4.3: Giao diện trang quản lý users

4.1.4. Trang quản lý giảng viên



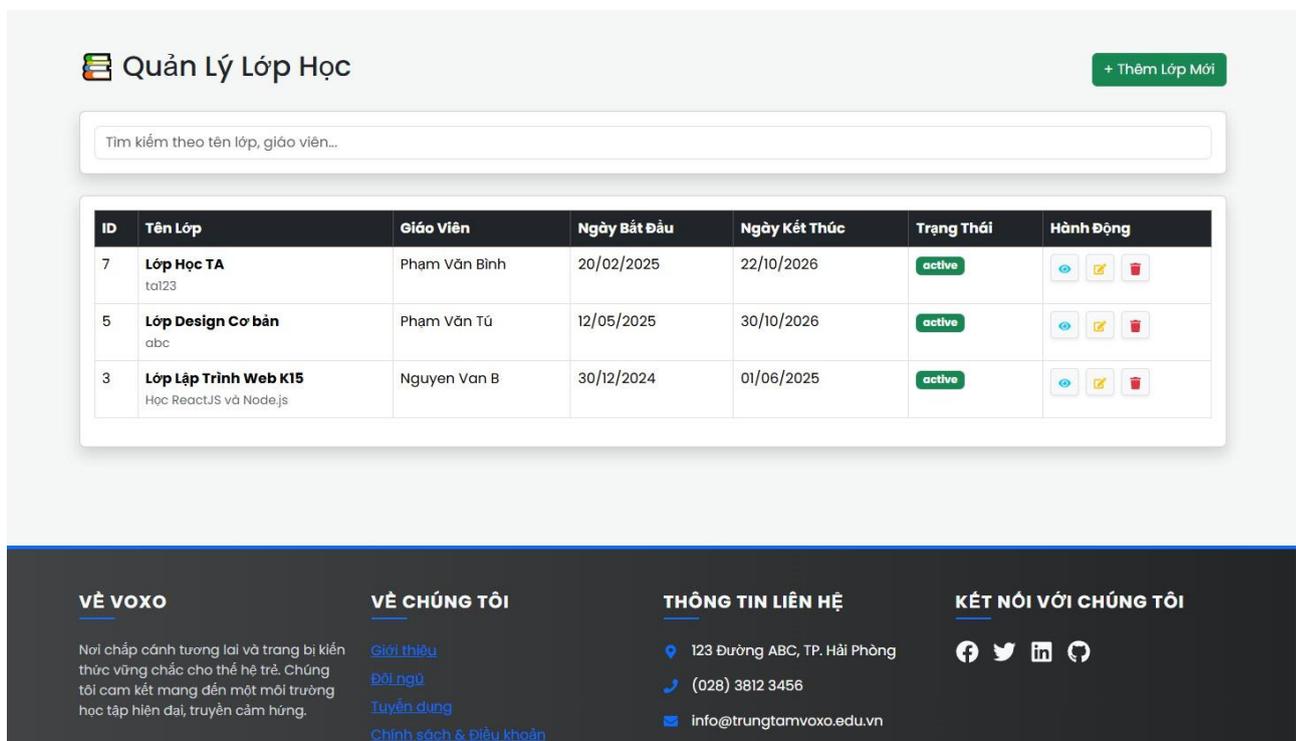
Hình 4.4: Giao diện trang quản lý giảng viên

4.1.5. Trang quản lý hồ sơ học sinh



Hình 4.5: Giao diện trang quản lý hồ sơ học sinh

4.1.6. Trang quản lý lớp học



Hình 4.6: Giao diện trang quản lý lớp học

4.1.7. Trang chấm công của giảng viên

Chấm Công Tiết Dạy

Thao tác

Chọn lớp dạy:
-- Chọn lớp --

BẮT ĐẦU DẠY

Lịch sử Excel

Ngày	Lớp	Vào	Ra	Tổng Giờ
26/12/2025	Lớp Lập Trình Web K15	20:30	20:36	0.08
26/12/2025	Lớp Lập Trình Web K15	20:26	20:26	0.00
26/12/2025	Lớp Lập Trình Web K15	20:03	20:06	0.03
20/12/2025	Lớp Lập Trình Web K15	07:18	---:--	...
20/12/2025	Lớp Lập Trình Web K15	07:04	07:17	0.22

Hình 4.7: Giao diện trang chấm công của giảng viên

4.1.8. Trang điểm danh

Quản lý Điểm danh

LỚP HỌC
Lớp Lập Trình Web K15

Tháng 12/2025

Th 2 29	Th 3 30 Hôm nay	Th 4 31	Th 5 1	Th 6 2	Th 7 3	CN 4
------------	-----------------------	------------	-----------	-----------	-----------	---------

Danh sách học sinh
Ngày: 30/12/2025 Tất cả Có mặt Tất cả Vắng

Học viên	Trạng thái	Ghi chú
Phạm Văn B ID: 2	Có mặt	Ghi chú...
Nguyễn Văn Con ID: 1	Có mặt	Ghi chú...

Báo Tan Học **LƯU LẠI**

Hình 4.8: Giao diện trang điểm danh

4.1.9. Trang thời khóa biểu

 **THỜI KHÓA BIỂU**
+ Thêm Thời Khóa Biểu

Thứ	Thời gian	Lớp học	Giáo viên	Phòng học	Hành động
Thứ Hai	09:00 - 11:00	Lớp Học TA	Phạm Văn Bình	abc	 
Thứ Ba	08:30 - 10:30	Lớp Design Cơ bản	Phạm Văn Tú	abc	 

VỀ VOXO

Nơi chấp cánh tương lai và trang bị kiến thức vững chắc cho thế hệ trẻ. Chúng tôi cam kết mang đến một môi trường học tập hiện đại, truyền cảm hứng.

VỀ CHÚNG TÔI

[Giới thiệu](#)
[Đội ngũ](#)
[Tuyển dụng](#)
[Chính sách & Điều khoản](#)

THÔNG TIN LIÊN HỆ

 123 Đường ABC, TP. Hải Phòng
 (028) 3812 3456
 info@trungtamvoxoxo.edu.vn

KẾT NỐI VỚI CHÚNG TÔI

Hình 4.9: Giao diện trang thời khóa biểu

4.2. Yêu cầu cài đặt

4.2.1. Yêu cầu phần cứng:

- Để đảm bảo hệ thống hoạt động ổn định, máy tính phát triển và triển khai cần đáp ứng các yêu cầu tối thiểu sau:
 - + CPU: Intel Core i3 hoặc tương đương.
 - + RAM: Tối thiểu 8GB.
 - + Ổ cứng: Tối thiểu 20GB dung lượng trống.

4.2.2. Yêu cầu phần mềm:

- Để cài đặt và vận hành dự án, cần có các phần mềm sau:
 - + 1. Hệ điều hành

- Hệ điều hành Windows 10 trở lên, hoặc các hệ điều hành khác như macOS, Linux.

+ 2. Node.js và NPM

- Node.js: Là môi trường thực thi (runtime) cho cả backend (Express.js) và frontend (React.js). Phiên bản đề nghị là 16.x trở lên.
- NPM (Node Package Manager): Đi kèm với Node.js, được sử dụng để quản lý các thư viện và gói phụ thuộc của dự án.

+ 3. Môi trường Cơ sở dữ liệu

- XAMPP: Là một gói phần mềm tích hợp, cung cấp môi trường máy chủ web cần thiết. Trong dự án này, XAMPP được sử dụng chủ yếu để cung cấp:
- Cơ sở dữ liệu MariaDB (hoặc MySQL): Hệ quản trị cơ sở dữ liệu để lưu trữ toàn bộ dữ liệu của ứng dụng.
- phpMyAdmin: Giao diện web để quản lý cơ sở dữ liệu một cách trực quan.

+ 4. Trình soạn thảo mã nguồn (Code Editor)

- Visual Studio Code (VSCode): Là một trình soạn thảo mã nguồn đa nền tảng, được phát triển bởi Microsoft. Nhờ vào hệ sinh thái tiện ích mở rộng (extensions) phong phú, VSCode có thể hoạt động như một IDE nhẹ hỗ trợ nhiều ngôn ngữ lập trình như JavaScript, Python, C++, Java, PHP, Go và nhiều ngôn ngữ khác.
- VSCode cung cấp các tính năng nổi bật như:
- Tự động hoàn thiện mã (IntelliSense)
- Gỡ lỗi trực tiếp trong trình soạn thảo (debugging)
- Tích hợp Git để quản lý mã nguồn

- Terminal tích hợp
 - Hỗ trợ lập trình từ xa (Remote Development)
 - Hàng nghìn tiện ích mở rộng để hỗ trợ framework, ngôn ngữ và công cụ phát triển.
- + 5. Git
- Hệ thống quản lý phiên bản phân tán, được sử dụng để theo dõi các thay đổi trong mã nguồn trong quá trình phát triển.
- + 6. Trình duyệt Web
- Một trình duyệt web hiện đại như Google Chrome, Mozilla Firefox, hoặc Microsoft Edge để chạy và kiểm thử ứng dụng web phía người dùng.

4.2.3. Yêu cầu mạng Internet:

- Thiết bị cần kết nối mạng để:
 - + Tải và cập nhật tiện ích mở rộng (extensions)
 - + Cập nhật phần mềm và tính năng mới
 - + Đồng bộ hóa cài đặt nếu dùng tài khoản Microsoft
 - + Làm việc với các dịch vụ như GitHub, Live Share, Remote SSH, v.v.

CHƯƠNG V: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Kết quả đạt được

Trong quá trình thực hiện đồ án "Xây dựng phần mềm quản lý trung tâm tổ chức dạy học", em đã nỗ lực nghiên cứu, tìm hiểu công nghệ và đạt được những kết quả quan trọng sau:

- Về mặt tài liệu và phân tích hệ thống:
 - + Đã khảo sát được quy trình nghiệp vụ thực tế tại các trung tâm dạy học, từ đó phát hiện các nhược điểm của phương thức quản lý thủ công .
 - + Xây dựng hoàn chỉnh bản phân tích đặc tả yêu cầu, biểu đồ Use Case (tổng quát và chi tiết), biểu đồ tuần tự và thiết kế cơ sở dữ liệu sát với thực tế.
- Về mặt kỹ thuật và công nghệ:
 - + Cơ sở dữ liệu: Thiết kế và triển khai thành công cơ sở dữ liệu MySQL (sử dụng XAMPP/MariaDB), đảm bảo lưu trữ có cấu trúc cho các thực thể quan trọng như: Users, Teachers, Students, Classes, Attendance và Schedules.
 - + Backend: Xây dựng hệ thống API RESTful hoàn chỉnh sử dụng Node.js và Express.js, xử lý tốt các logic nghiệp vụ về xác thực (JWT), phân quyền và quản lý dữ liệu.
 - + Frontend: Phát triển giao diện người dùng thân thiện, hiện đại bằng React.js, áp dụng mô hình Component và Context API để quản lý trạng thái, giúp trải nghiệm người dùng mượt mà (SPA).
 - + Tự động hóa: Bước đầu tìm hiểu và ứng dụng công cụ n8n để tích hợp tính năng gửi thông báo tự động qua Zalo, giúp tăng cường kết nối giữa trung tâm và phụ huynh.

- Về mặt chức năng hệ thống:
 - + Quản trị viên (Admin): Quản lý toàn diện hệ thống bao gồm: Quản lý người dùng, hồ sơ giáo viên/học viên, thiết lập lớp học, xếp thời khóa biểu và xem các báo cáo thống kê.
 - + Giáo viên (Teacher): Thực hiện các nghiệp vụ hàng ngày như xem lịch dạy, điểm danh học viên (có mặt/vắng/muộn) và thực hiện chấm công cá nhân (Check-in/Check-out).
 - + Hệ thống: Đảm bảo quy trình điểm danh và chấm công hoạt động chính xác, dữ liệu được đồng bộ tức thời giữa Client và Server.

5.2. Hạn chế

Mặc dù đã hoàn thành các mục tiêu cơ bản đề ra, đề án vẫn còn tồn tại một số hạn chế cần khắc phục:

- Chức năng học phí chưa chuyên sâu: Hệ thống hiện tại tập trung tốt vào quản lý đào tạo (lớp học, điểm danh) nhưng phân hệ quản lý tài chính (thu học phí, công nợ, tính lương giáo viên) mới chỉ dừng lại ở mức cơ bản, chưa hỗ trợ thanh toán trực tuyến.
- Thông báo còn phụ thuộc: Tính năng gửi thông báo qua Zalo tuy đã được đề xuất và nghiên cứu (n8n) nhưng vẫn phụ thuộc vào chính sách của bên thứ ba (Zalo OA), có thể phát sinh chi phí hoặc giới hạn số lượng tin nhắn trong thực tế.
- Chưa có ứng dụng di động (Mobile App): Hiện tại hệ thống chỉ hoạt động trên nền tảng Web. Việc thiếu vắng ứng dụng di động riêng biệt (Android/iOS) là một hạn chế đối với trải nghiệm của phụ huynh và giáo viên khi cần thao tác nhanh trên điện thoại.

- Báo cáo thống kê chưa đa dạng: Các biểu đồ thống kê hiện tại chủ yếu dạng bảng biểu cơ bản, chưa có các biểu đồ trực quan (chart) phân tích sâu về hiệu quả đào tạo hay biến động sĩ số theo thời gian.

5.3. Hướng phát triển

Dựa trên nền tảng đã xây dựng, em đề xuất các hướng phát triển để hoàn thiện sản phẩm trong tương lai:

- Tích hợp thanh toán điện tử: Kết nối với các cổng thanh toán (VNPAY, MoMo) để phụ huynh có thể đóng học phí trực tuyến ngay trên hệ thống, tự động hóa quy trình ghi nhận doanh thu.
- Nâng cấp điểm danh bằng AI: Tích hợp công nghệ nhận diện khuôn mặt (Face ID) để hỗ trợ quy trình chấm công giáo viên và điểm danh học sinh nhanh chóng, chính xác hơn, giảm thiểu gian lận.
- Phát triển Mobile App: Xây dựng phiên bản ứng dụng di động (sử dụng React Native hoặc Flutter) để đồng bộ với hệ thống website, giúp giáo viên điểm danh và phụ huynh nhận thông báo thuận tiện hơn.
- Mở rộng tính năng tương tác (LMS): Bổ sung các chức năng hỗ trợ học tập như: giao bài tập về nhà, chấm điểm trực tuyến, và diễn đàn trao đổi giữa giáo viên - phụ huynh - học sinh.
- Tối ưu hóa Deployment: Triển khai hệ thống lên các nền tảng Cloud (như AWS, VPS) với tên miền chính thức và chứng chỉ bảo mật SSL để đảm bảo tính sẵn sàng và an toàn dữ liệu khi đưa vào sử dụng thực tế.

TÀI LIỆU THAM KHẢO

- [1] Mozilla Developer Network (MDN). *Client–Server overview*. https://developer.mozilla.org/en-US/docs/Learn/Server-side/First_steps/Client-Server_overview
- [2] React Official Documentation – <https://react.dev>
- [3] Vite Official Documentation – <https://vitejs.dev>
- [4] React Router – Official Documentation: <https://reactrouter.com>
- [5] Nguyễn Thanh Hưng (2023). *Ứng dụng SCSS và CSS Modules trong phát triển giao diện React*, Trường Đại học Công nghệ Thông tin TP.HCM.
- [6] MDN WebDocs–*ReactContextAPI Overview*:<https://developer.mozilla.org/>
- [7] Node.js Official Website – <https://nodejs.org/en>
- [8] Express.js Official Website – <https://expressjs.com/>
- [9] Express.js Official Website – <https://expressjs.com/>
- [10] Phạm Quang Huy (2022). *Ứng dụng kiến trúc MVC trong phát triển web với Node.js và Express*, Trường Đại học FPT TP.HCM.
- [11] Trần Đức Tài (2023). *Sử dụng Middleware trong Express.js để bảo vệ Website API*, Đại học Công nghệ Thông tin - ĐHQG TP.HCM.
- [12] Nguyễn Văn Minh (2022). *Xây dựng và kết nối frontend-backend trong ứng dụng thương mại điện tử*, Trường Đại học Bách Khoa Hà Nội.
- [13] W3Schools – MySQL Tutorial:
<https://www.w3schools.com/mysql/>