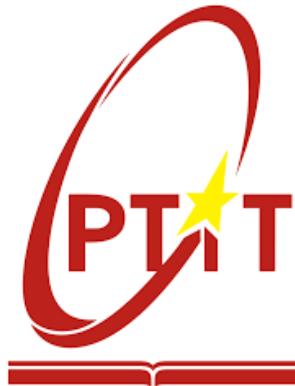


HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

KHOA CÔNG NGHỆ THÔNG TIN I



**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC
XÂY DỰNG HỆ THỐNG QUẢN LÝ
RẠP CHIẾU PHIM**

Sinh viên thực hiện : Nguyễn Ngọc Quang - B21DCCN630

Trần Thu Hà - B21DCCN315

Giảng viên hướng dẫn : PGS. TS. Nguyễn Trọng Khánh

Khóa : 2021 – 2026

Hệ : Đại học Chính quy

Hà Nội – 2025

LỜI CẢM ƠN

Trước hết, nhóm xin bày tỏ lòng biết ơn sâu sắc nhất tới PGS.TS. Nguyễn Trọng Khánh, người đã trực tiếp hướng dẫn, dành nhiều thời gian quý báu để chỉ bảo và định hướng cho nhóm trong suốt quá trình nghiên cứu và thực hiện đồ án.

Em xin chân thành cảm ơn quý thầy, cô trong Khoa Công nghệ Thông tin I đã truyền đạt cho em những kiến thức nền tảng và chuyên sâu trong suốt những năm học tập tại trường. Những kiến thức và kinh nghiệm mà thầy cô chia sẻ không chỉ giúp em có đủ năng lực và tư duy để hoàn thành đồ án tốt nghiệp, mà còn là hành trang quý báu, tạo nền tảng vững chắc cho em trong quá trình làm việc và phát triển sự nghiệp sau khi ra trường.

Bên cạnh đó, em xin gửi lời tri ân đến gia đình và bạn bè đã luôn ở bên cạnh động viên, khích lệ và tạo mọi điều kiện tốt nhất để em có thể tập trung hoàn thành tốt đồ án tốt nghiệp này.

Trân trọng cảm ơn!

Hà Nội, ... tháng ... năm 2025

Đại diện nhóm sinh viên thực hiện

Nguyễn Ngọc Quang

NHẬN XÉT, ĐÁNH GIÁ, CHO ĐIỂM

(Của giảng viên hướng dẫn)

Điểm:(Bằng chữ.....)

Đồng ý/không đồng ý cho sinh viên bảo vệ trước hội đồng chấm tốt nghiệp.

Hà Nội, ...tháng ... năm 2025

CÁN BỘ - GIẢNG VIÊN HƯỚNG DẪN

NHẬN XÉT, ĐÁNH GIÁ, CHO ĐIỂM

(Của giảng viên phản biện)

Điểm:(Bằng chữ.....)

Đồng ý/không đồng ý cho sinh viên bảo vệ trước hội đồng chấm tốt nghiệp.

Hà Nội, ...tháng ... năm 2025

CÁN BỘ - GIẢNG VIÊN PHẢN BIỆN

BẢNG PHÂN CHIA CÔNG VIỆC

Thành viên	Nhiệm vụ
Nguyễn Ngọc Quang	<ul style="list-style-type: none">- Thiết kế hệ thống Microservices- Nghiên cứu công nghệ và xem xét tích hợp vào hệ thống- Xây dựng giao diện người dùng- Triển khai logic nghiệp vụ- Xây dựng và tối ưu thời gian phản hồi API- Triển khai Deploy- Rà soát lỗi hệ thống và thực hiện vá lỗi
Trần Thu Hà	<ul style="list-style-type: none">- Phân tích nghiệp vụ- Thiết kế kiến trúc Cơ sở dữ liệu- Xây dựng giao diện người dùng- Triển khai phân quyền RBAC- Triển khai RAG với Gemini cho Chatbot

LINK SẢN PHẨM

Link đến hệ thống: <https://cinema-project-mu.vercel.app/>

Link source code github của hệ thống: <https://github.com/GnauqTheBeast/cinema-project>

MỤC LỤC

DANH MỤC BẢNG BIỂU	
DANH MỤC HÌNH ẢNH	
KÝ HIỆU VÀ THUẬT NGỮ	
LỜI MỞ ĐẦU	1
1. Đặt vấn đề	1
2. Giải pháp	2
3. Cấu trúc đồ án	3
CHƯƠNG 1: CƠ SỞ LÝ THUYẾT	4
1.1. Công nghệ Frontend	4
1.2. Công nghệ Backend	5
1.3. Hệ quản trị cơ sở dữ liệu	9
1.4. Thanh toán và bảo mật	11
1.5. Tích hợp AI	12
1.6. DevOps và hạ tầng triển khai	14
1.7. Kết chương	15
CHƯƠNG 2: PHÂN TÍCH YÊU CẦU VÀ THIẾT KẾ HỆ THỐNG	17
2.1. Phân tích yêu cầu chức năng	17
2.2. Phân tích yêu cầu phi chức năng	38
2.3. Thiết kế hệ thống	39
2.4. Kết chương	54
CHƯƠNG 3: CÀI ĐẶT VÀ KIỂM THỬ HỆ THỐNG	55
3.1. Cài đặt	55
3.2. Môi trường triển khai	57
3.3. Giao diện và chức năng	58
3.4. Kết chương	70
CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	71
4.1. Kết luận	71
4.2. Các hạn chế	71
4.3. Hướng phát triển	71
4.4. Kết chương	72
TÀI LIỆU THAM KHẢO	73

DANH MỤC BẢNG BIÊU

Bảng 1: Đặc tả usecase Đăng nhập.....	19
Bảng 2: Đặc tả usecase Đăng ký	20
Bảng 3: Đặc tả usecase đặt vé xem phim	21
Bảng 4: Đặc tả usecase quản lý phim	22
Bảng 5: Đặc tả usecase quản lý phòng chiếu.....	23
Bảng 6: Đặc tả usecase quản lý ghế	25
Bảng 7: Đặc tả usecase quản lý suất chiếu	27
Bảng 8: Đặc tả usecase quản lý nhân viên.....	28
Bảng 9: Đặc tả usecase quản lý tin tức về phim	29
Bảng 10: Đặc tả usecase quản lý tài liệu chatbot	30
Bảng 11: Đặc tả usecase chat với hệ thống	31
Bảng 12: Đặc tả usecase xem báo cáo thống kê doanh thu	32
Bảng 13: Đặc tả usecase xem tin tức	32
Bảng 14: Đặc tả usecase xem thông báo	33
Bảng 15: Đặc tả usecase bán vé tại quầy	34
Bảng 16: Đặc tả usecase tra cứu vé của khách hàng	35
Bảng 17: Đặc tả usecase tìm kiếm phim.....	35
Bảng 18: Đặc tả usecase xem lịch sử đặt vé	36
Bảng 19: Đặc tả usecase chỉnh sửa thông tin cá nhân	37
Bảng 20: Đặc tả usecase xuất vé	37
Bảng 21: Phân tích chi tiết các lớp	42

DANH MỤC HÌNH ẢNH

Hình 1: Trang web của Trung tâm Chiếu phim Quốc gia bị sập [1]	1
Hình 2: Thủ tướng lắng nghe trình bày giải pháp Blockchain cho dịch vụ công – tư [2]	2
Hình 3: ReactJS	4
Hình 4: Ngôn ngữ lập trình Go.....	5
Hình 5: Framework ExpressJS	6
Hình 6: Kiến trúc Microservices	7
Hình 7: Cơ chế Redis Publish/Subscribe.....	9
Hình 8: Hệ quản trị cơ sở dữ liệu PostgreSQL.....	9
Hình 9: Hệ quản trị dữ liệu Redis.....	10
Hình 10: Công nghệ Blockchain	11
Hình 11: Mô tả cơ bản cách LLM hoạt động	12
Hình 12: Gemini Model.....	13
Hình 13: Mô tả cơ chế RAG hoạt động.....	14
Hình 14: Nền tảng container Docker	15
Hình 15: Máy chủ web và reverse proxy Nginx.....	15
Hình 16: Biểu đồ usecase tổng quan của hệ thống	18
Hình 17: Biểu đồ lớp thiết kế hệ thống.....	39
Hình 18: Cơ sở dữ liệu hệ thống.....	42
Hình 19: Sơ đồ tuần tự chức năng đăng nhập.....	43
Hình 20: Sơ đồ tuần tự chức năng đăng ký	44
Hình 21: Sơ đồ tuần tự chức năng đặt vé (1).....	45
Hình 22: Sơ đồ tuần tự chức năng đặt vé (2).....	46
Hình 23: Sơ đồ tuần tự chức năng xem suất chiếu	47
Hình 24: Sơ đồ tuần tự chức năng thêm suất chiếu	47
Hình 25: Sơ đồ tuần tự chức năng sửa suất chiếu	48
Hình 26: Sơ đồ tuần tự chức năng xóa suất chiếu	48
Hình 27: Sơ đồ tuần tự chức năng tạo tài khoản mới cho nhân viên.....	49
Hình 28: Sơ đồ tuần tự chức năng xem chi tiết thông tin nhân viên	49
Hình 29: Sơ đồ tuần tự chức năng sửa thông tin nhân viên.....	50
Hình 30: Sơ đồ tuần tự chức năng xóa thông tin nhân viên	50
Hình 31: Sơ đồ tuần tự chức năng thống kê doanh thu theo phim	51
Hình 32: Sơ đồ tuần tự chức năng chat với hệ thống	51
Hình 33: Sơ đồ tuần tự chức năng quản lý tài liệu RAG chatbot	52
Hình 34: Sơ đồ tuần tự chức năng xem tin tức	53

Hình 35: Sơ đồ tuần tự chức năng xem lịch sử đặt vé	53
Hình 36: Sơ đồ tuần tự chức năng xuất vé	54
Hình 37: Sơ đồ mô tả kiến trúc hệ thống.....	55
Hình 38: Sơ đồ triển khai Transactional Outbox.....	56
Hình 39: Quy trình CI/CD	56
Hình 40: Giao diện module đăng nhập	58
Hình 41: Giao diện module đăng ký.....	58
Hình 42: Giao diện module đặt vé xem phim (1)	59
Hình 43: Giao diện module đặt vé xem phim (2)	59
Hình 44: Giao diện module đặt vé xem phim (3)	60
Hình 45: Giao diện module đặt vé xem phim (4)	60
Hình 46: Giao diện module đặt vé xem phim (5)	61
Hình 47: Giao diện module đặt vé xem phim (6)	61
Hình 48: Giao diện module quản lý suất chiếu (1).....	62
Hình 49: Giao diện module quản lý suất chiếu (2).....	62
Hình 50: Giao diện module quản lý nhân viên (1)	63
Hình 51: Giao diện module quản lý nhân viên (2)	63
Hình 52: Giao diện module quản lý nhân viên (3)	64
Hình 53: Giao diện module xem tin tức (1).....	64
Hình 54: Giao diện module xem tin tức (2).....	65
Hình 55: Giao diện module Chatbot.....	65
Hình 56: Giao diện module xem lịch sử đặt vé	66
Hình 57: Giao diện module quản lý tài liệu RAG (1)	66
Hình 58: Giao diện module quản lý tài liệu RAG (2)	67
Hình 59: Giao diện module bán vé tại quầy (1)	67
Hình 60: Giao diện module bán vé tại quầy (2)	68
Hình 61: Giao diện module bán vé tại quầy (3)	69
Hình 62: Giao diện module xuất vé	69

KÝ HIỆU VÀ THUẬT NGỮ

ST T	Thuật ngữ	Ý nghĩa	Chú thích
1	AI	Artificial Intelligence	Trí tuệ nhân tạo
2	HTTP	Hypertext Transfer Protocol	Giao thức truyền tải siêu văn bản giữa client và server
3	API	Application Programming Interface	Giao diện lập trình ứng dụng cho phép các thành phần phần mềm giao tiếp với nhau
4	RBAC	Role-Based Access Control	Mô hình quản lý quyền truy cập, cấp phát quyền dựa trên vai trò của người dùng
5	CI/CD	Continuous Integration / Continuous Deployment	Tập hợp các phương pháp và quy trình tự động hóa trong phát triển phần mềm
6	ACID	Atomicity, Consistency, Isolation, Durability	Tập hợp các tính chất đảm bảo tính toàn vẹn của giao dịch trong cơ sở dữ liệu
7	ORDBMS	Object-relational database management system	Hệ quản trị cơ sở dữ liệu quan hệ - đối tượng
8	MVCC	Multi-Version Concurrency Control	Kiểm soát truy cập đồng thời đa phiên bản
9	LLM	Large Language Model	Mô hình ngôn ngữ lớn, được huấn luyện trên tập dữ liệu quy mô lớn nhằm hiểu và sinh ngôn ngữ tự nhiên
10	RAG	Retrieval-Augmented Generation	Kiến trúc kết hợp giữa truy xuất dữ liệu và sinh nội dung

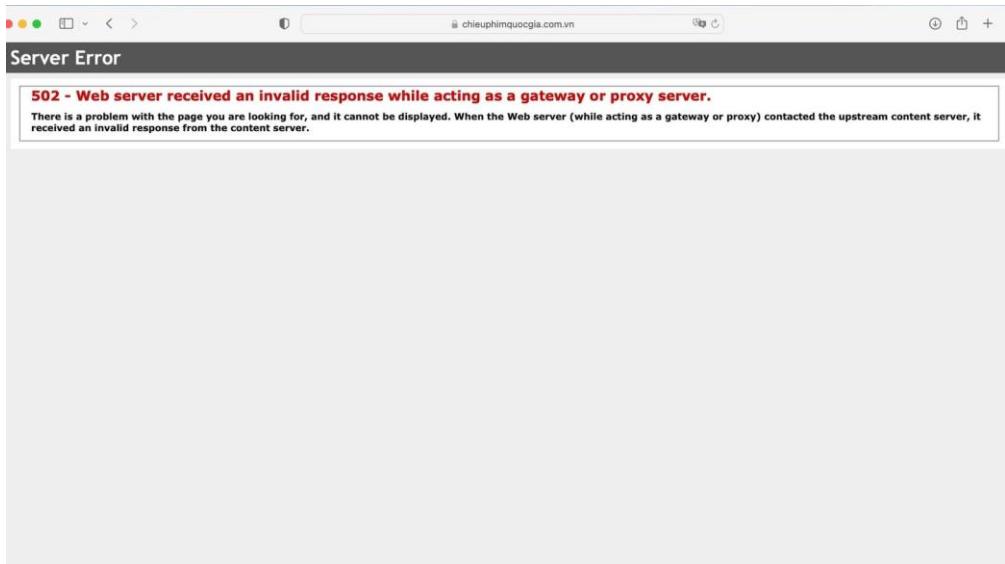
11	RPC	Remote Procedure Call	Cơ chế gọi hàm từ xa giữa các tiến trình hoặc service
12	EVM	Ethereum Virtual Machine	Môi trường máy tính thực thi hợp đồng thông minh
13	CORS	Cross-Origin Resource Sharing	Cơ chế kiểm soát truy cập tài nguyên giữa các nguồn
14	JWT	JSON Web Token	Xác thực người dùng
15	IDE	Integrated Development Environment	Môi trường phát triển tích hợp

LỜI MỞ ĐẦU

1. Đặt vấn đề

Trong những năm gần đây, nhu cầu giải trí, đặc biệt là xem phim tại rạp, ngày càng gia tăng, kéo theo sự phát triển mạnh mẽ của các nền tảng đặt vé trực tuyến. Tuy nhiên, phần lớn các trang đặt vé phim hiện nay tại Việt Nam vẫn còn tồn tại nhiều hạn chế, đặc biệt là khả năng đáp ứng khi lưu lượng truy cập lớn. Hệ thống thường hoạt động thiếu ổn định, dễ xảy ra tình trạng quá tải mỗi khi có bộ phim hấp dẫn được công chiếu, dẫn đến việc quá trình đặt vé bị gián đoạn và làm suy giảm đáng kể trải nghiệm người dùng.

Trên thực tế đã ghi nhận trường hợp hệ thống đặt vé bị sập do lượng người truy cập tăng đột biến, điển hình như trang web của Trung tâm Chiếu phim Quốc gia trong các đợt cao điểm phát hành phim “Đào, phở và piano”.



Hình 1: Trang web của Trung tâm Chiếu phim Quốc gia bị sập [1]

Bên cạnh đó, các nền tảng đặt vé hiện nay vẫn chưa tận dụng các công nghệ mới như blockchain và trí tuệ nhân tạo. Việc sử dụng các cơ chế thanh toán truyền thống khiến tính minh bạch và độ an toàn của giao dịch chưa được đảm bảo một cách tối ưu. Trong khi đó, việc ứng dụng công nghệ AI vào các mô hình kinh doanh truyền thống có thể giúp tối ưu hóa nguồn nhân lực, tiết kiệm thời gian và chi phí vận hành.

Bối cảnh xu hướng công nghệ toàn cầu ngày càng hướng tới tính minh bạch và an toàn dữ liệu, Việt Nam cũng đang từng bước đẩy mạnh nghiên cứu, phát triển và ứng dụng công nghệ blockchain trong nhiều lĩnh vực, đặc biệt là dịch vụ công. Minh chứng cho điều này là việc công

nghệ blockchain đã được ứng dụng trong hệ thống truy xuất nguồn gốc tiền chát, hóa chất của Bộ Công an trên nền tảng VNNeID, đồng thời được tích cực triển khai trong hoạt động của nhiều tổ chức và doanh nghiệp dịch vụ công – tư, góp phần nâng cao hiệu quả quản lý nhà nước và thúc đẩy phát triển kinh tế tư nhân. Điều này cho thấy việc nghiên cứu và triển khai blockchain vào các hệ thống dịch vụ trực tuyến là một xu hướng tất yếu và có ý nghĩa thực tiễn cao.



Hình 2: Thủ tướng lắng nghe trình bày giải pháp Blockchain cho dịch vụ công – tư [2]

2. Giải pháp

Xuất phát từ thực tế, đồ án thực hiện với mục tiêu xây dựng một nền tảng đặt vé xem phim hiện đại, thân thiện và ổn định hơn. Với yêu cầu hệ thống có khả năng mở rộng, nâng cao hiệu suất và đảm bảo tính sẵn sàng ngay cả khi có lượng lớn người dùng truy cập đồng thời. Không chỉ cải thiện trải nghiệm truyền thống, nền tảng còn tích hợp thêm các công nghệ mới như thanh toán bằng crypto nhằm mang đến sự tiện lợi và linh hoạt cho người dùng trong thời đại số.

Bên cạnh đó, hệ thống còn tích hợp chatbot thông minh, giúp tự động hóa việc giải đáp thắc mắc và hỗ trợ người dùng trong suốt quá trình đặt vé. Chatbot AI giúp giảm tải cho bộ phận chăm sóc khách hàng, đồng thời đảm bảo người dùng luôn nhận được thông tin nhanh chóng, chính xác và nhất quán.

Với sự kết hợp giữa kiến trúc hệ thống hiệu năng cao, phương thức thanh toán tiên tiến và công nghệ AI hỗ trợ người dùng, đồ án hướng tới xây dựng một mô hình đặt vé xem phim toàn

diện, đáp ứng tốt nhu cầu của thị trường hiện tại và mở ra nhiều tiềm năng phát triển trong tương lai.

3. Cấu trúc đồ án

Nội dung đồ án bao gồm có 4 chương:

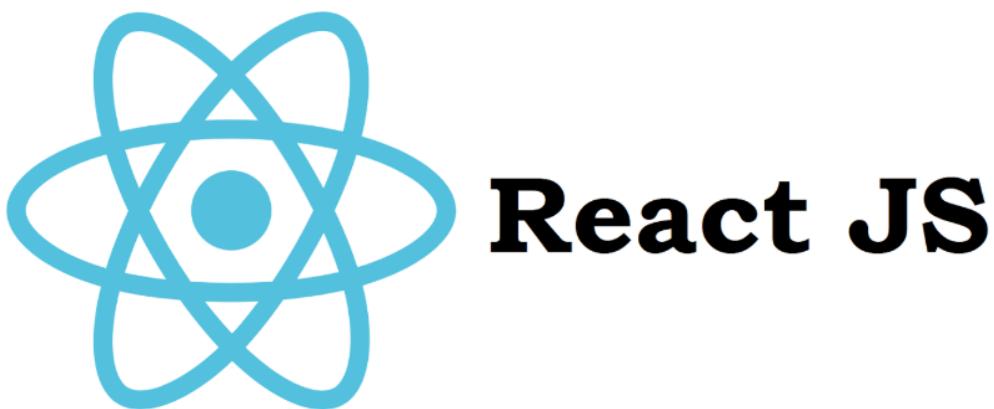
- Chương 1: Cơ sở lý thuyết
- Chương 2: Phân tích yêu cầu và thiết kế hệ thống
- Chương 3: Cài đặt và kiểm thử hệ thống
- Chương 4: Kết luận và hướng phát triển

CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

1.1. Công nghệ Frontend

1.1.1. ReactJS

ReactJS [3] là một thư viện JavaScript mã nguồn mở được sử dụng để xây dựng giao diện người dùng. ReactJS tập trung vào việc xây dựng các thành phần giao diện có khả năng tái sử dụng cao, giúp đơn giản hóa quá trình phát triển và bảo trì ứng dụng. Thư viện này được thiết kế theo mô hình hướng thành phần, cho phép tách biệt giao diện thành các khôi độc lập và dễ quản lý.



Hình 3: ReactJS

ReactJS được phát triển và duy trì bởi Meta cùng với cộng đồng mã nguồn mở, và được phát hành lần đầu vào năm 2013. Thư viện có khả năng chạy trên hầu hết các trình duyệt hiện đại và dễ dàng tích hợp với nhiều công nghệ khác trong hệ sinh thái JavaScript. Với tính chất mã nguồn mở và miễn phí, ReactJS cho phép người dùng tự do sử dụng, chỉnh sửa và phân phối cho nhiều mục đích khác nhau, từ các dự án cá nhân đến các hệ thống quy mô lớn trong doanh nghiệp.

1.1.2. Ethers.js

Ethers.js [4] là một thư viện JavaScript mã nguồn mở được sử dụng để tương tác với Blockchain Ethereum và các mạng tương thích EVM. Thư viện cung cấp các API cấp cao cho phép ứng dụng kết nối với blockchain, thực hiện giao dịch, quản lý ví người dùng một cách an toàn. Ethers.js được thiết kế gọn nhẹ, dễ sử dụng và tập trung mạnh vào tính bảo mật, phù hợp cho các ứng dụng Web3 hiện đại.

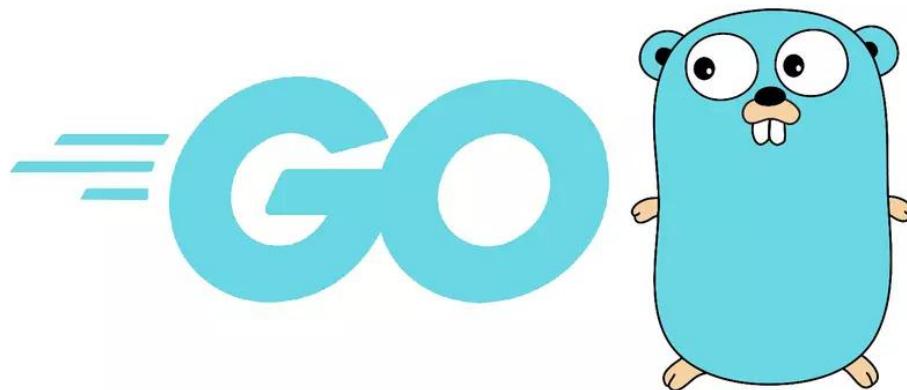
Ethers.js hỗ trợ kết nối với nhiều provider phổ biến như MetaMask, Infura, Alchemy thông qua giao thức RPC, cho phép ứng dụng truy vấn dữ liệu blockchain hoặc gửi giao dịch trực tiếp lên mạng lưới.

1.2. Công nghệ Backend

1.2.1. Golang

Go [5] là một ngôn ngữ lập trình mã nguồn mở được thiết kế nhằm xây dựng các hệ thống phần mềm hiệu năng cao, có khả năng mở rộng và dễ bảo trì. Golang đặc biệt phù hợp cho việc phát triển các ứng dụng phía máy chủ, hệ thống phân tán, dịch vụ web, microservices và các công cụ hạ tầng. Ngôn ngữ này theo đuổi triết lý thiết kế đơn giản, cú pháp rõ ràng và thời gian biên dịch nhanh, giúp tăng năng suất phát triển và giảm độ phức tạp của hệ thống.

Golang được phát triển bởi Google, với sự tham gia của các kỹ sư nổi tiếng như Robert Griesemer, Rob Pike và Ken Thompson, và được phát hành lần đầu vào năm 2009. Ngôn ngữ có khả năng hoạt động trên nhiều nền tảng khác nhau như Linux, macOS và Windows, đồng thời hỗ trợ biên dịch chéo để tạo ra các file thực thi độc lập. Là một ngôn ngữ mã nguồn mở và miễn phí, Golang cho phép người dùng tự do sử dụng, chỉnh sửa và phân phối cho nhiều mục đích khác nhau.



Hình 4: Ngôn ngữ lập trình Go

Golang cung cấp nhiều đặc điểm và tính năng hiện đại phục vụ cho việc xây dựng các hệ thống phần mềm quy mô lớn và có độ tin cậy cao:

- Cú pháp đơn giản, dễ đọc và dễ học, giảm thiểu lỗi trong quá trình phát triển.

- Hiệu năng cao nhờ biên dịch trực tiếp sang mã máy.
- Cơ chế xử lý đồng thời mạnh mẽ thông qua goroutine và channel.
- Quản lý bộ nhớ tự động với cơ chế garbage collection hiệu quả.

1.2.2. ExpressJS

ExpressJS [6] là một framework web nhẹ được xây dựng trên nền tảng NodeJS, nhằm đơn giản hóa quá trình phát triển các ứng dụng web và API phía máy chủ. ExpressJS cung cấp các cơ chế routing, middleware và xử lý request/response linh hoạt, giúp lập trình dễ dàng xây dựng các dịch vụ RESTful web app.



Hình 5: Framework ExpressJS

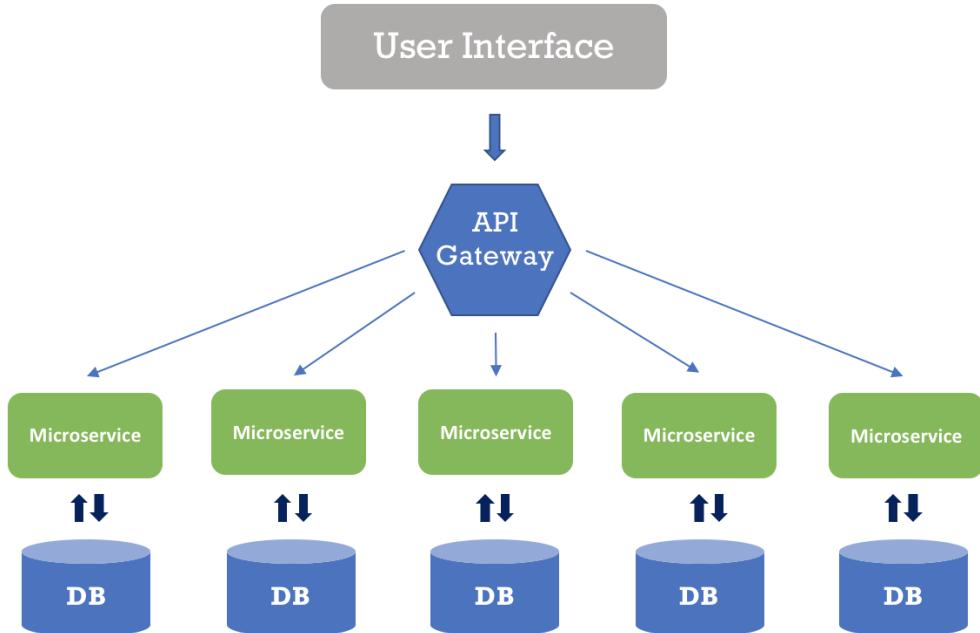
ExpressJS được phát triển bởi TJ Holowaychuk và được phát hành lần đầu vào năm 2010. Framework này có khả năng chạy trên mọi môi trường hỗ trợ NodeJS và thường được sử dụng như một nền tảng cốt lõi cho nhiều framework và công cụ khác. Là phần mềm mã nguồn mở và miễn phí, ExpressJS cho phép người dùng tự do sử dụng và mở rộng theo nhu cầu của hệ thống.

1.2.3. Microservices

Microservices [7] là một phong cách kiến trúc phần mềm trong đó hệ thống được chia thành nhiều dịch vụ nhỏ, độc lập, mỗi dịch vụ đảm nhiệm một chức năng nghiệp vụ cụ thể. Các dịch vụ này giao tiếp với nhau thông qua các giao thức nhẹ như HTTP/REST, gRPC hoặc message queue. Kiến trúc Microservices giúp giảm sự phụ thuộc giữa các thành phần trong hệ thống, từ đó nâng cao khả năng mở rộng, bảo trì và phát triển song song.

Kiến trúc Microservices được hình thành và phổ biến rộng rãi từ khoảng năm 2014, dựa trên các thực tiễn phát triển phần mềm của các công ty công nghệ lớn như Netflix, Amazon và

Google. Microservices có thể được triển khai trên nhiều nền tảng khác nhau, thường kết hợp với các công nghệ hiện đại như container (Docker), hệ thống điều phối (Kubernetes) và hạ tầng điện toán đám mây. Đây không phải là một sản phẩm cụ thể mà là một mô hình kiến trúc, được áp dụng linh hoạt tùy theo yêu cầu của từng hệ thống.



Hình 6: Kiến trúc Microservices

Kiến trúc Microservices mang lại nhiều ưu điểm trong việc xây dựng các hệ thống phần mềm quy mô lớn và phức tạp:

- Mỗi dịch vụ được phát triển, triển khai và mở rộng độc lập.
- Giảm ảnh hưởng lan truyền khi một dịch vụ gặp sự cố.
- Hỗ trợ phát triển song song bởi nhiều nhóm khác nhau.
- Dễ dàng mở rộng theo từng chức năng nghiệp vụ cụ thể.

Bên cạnh các ưu điểm, Microservices cũng đặt ra một số thách thức như quản lý phức tạp, giám sát hệ thống phân tán và đảm bảo nhất quán dữ liệu. Việc chia hệ thống thành nhiều dịch vụ độc lập làm gia tăng số lượng thành phần cần quản lý. Vì vậy, Microservices đòi hỏi các cơ chế điều phối, tự động hóa và giám sát hiệu quả.

1.2.4. gRPC

gRPC [8] là một framework RPC hiệu năng cao được thiết kế để hỗ trợ xây dựng các hệ thống phân tán và kiến trúc Microservices. gRPC cho phép các ứng dụng ở những môi trường khác nhau giao tiếp với nhau thông qua việc gọi hàm từ xa như các hàm cục bộ, giúp đơn giản hóa quá

trình trao đổi dữ liệu giữa các dịch vụ. Framework này sử dụng giao thức HTTP/2 làm nền tảng truyền thông và định dạng dữ liệu nhị phân Protocol Buffers, từ đó mang lại hiệu năng cao và độ trễ thấp.

gRPC được phát triển bởi Google và được công bố chính thức vào năm 2015. Framework này hỗ trợ đa ngôn ngữ lập trình như Go, Java, C++, Python, Node.js và nhiều ngôn ngữ khác, cho phép các hệ thống dễ dàng tích hợp với nhau. gRPC có thể hoạt động trên nhiều nền tảng và hệ điều hành khác nhau, đồng thời là một công nghệ mã nguồn mở và miễn phí, được sử dụng rộng rãi trong các hệ thống phân tán quy mô lớn.

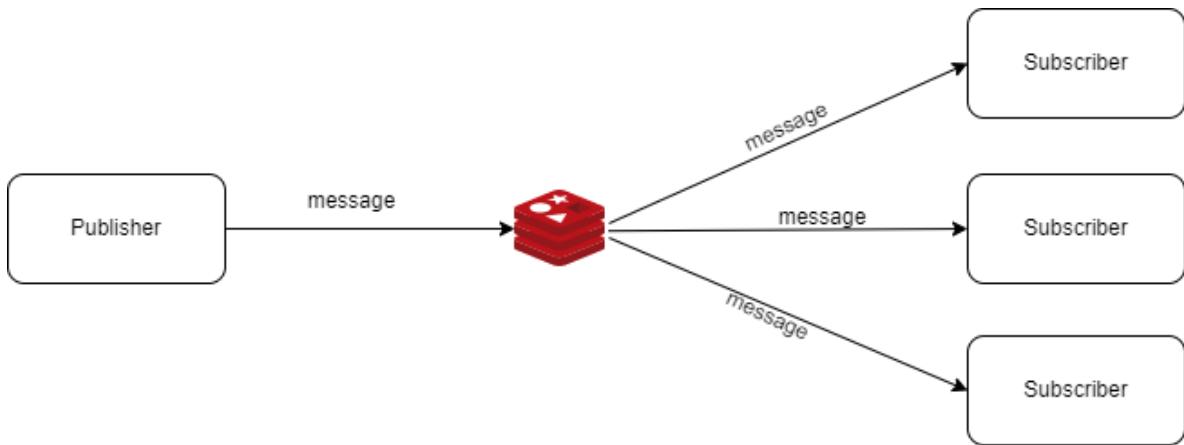
gRPC cung cấp nhiều tính năng hiện đại phục vụ cho việc xây dựng các hệ thống giao tiếp hiệu quả và có độ tin cậy cao. Đây là một số ưu điểm của gRPC:

- Sử dụng HTTP/2 giúp hỗ trợ multiplexing, nén header và truyền dữ liệu hai chiều.
- Định dạng Protocol Buffers nhỏ gọn, giúp giảm kích thước dữ liệu và tăng tốc độ truyền.
- Tự động sinh mã cho nhiều ngôn ngữ từ file định nghĩa .proto.

1.2.5. Redis Pub/Sub

Redis Pub/Sub [9] là một cơ chế truyền thông theo mô hình Publish/Subscribe được tích hợp sẵn trong hệ quản trị dữ liệu Redis. Cơ chế này cho phép các thành phần trong hệ thống giao tiếp với nhau theo thời gian thực thông qua các channel, trong đó các publisher gửi thông điệp và các subscriber đăng ký lắng nghe mà không cần biết trực tiếp về nhau. Redis Pub/Sub thường được sử dụng trong các hệ thống phân tán, ứng dụng thời gian thực và kiến trúc microservices nhằm truyền tải sự kiện hoặc thông báo với độ trễ thấp.

Redis Pub/Sub là một phần trong hệ sinh thái Redis, được phát triển bởi Redis Labs và cộng đồng mã nguồn mở. Redis Pub/Sub là một tính năng của hệ thống mã nguồn mở và miễn phí, cho phép người dùng tự do triển khai và sử dụng trong các ứng dụng có quy mô từ nhỏ đến lớn.



Hình 7: Cơ chế Redis Publish/Subscribe

1.3. Hệ quản trị cơ sở dữ liệu

1.3.1. PostgreSQL

PostgreSQL [10] là một hệ quản trị cơ sở dữ liệu ORDBMS mã nguồn mở và miễn phí, được đánh giá cao về tính ổn định, khả năng mở rộng và mức độ tuân thủ các tiêu chuẩn kỹ thuật. Hệ thống được thiết kế để xử lý nhiều loại khối lượng công việc khác nhau, từ các ứng dụng quy mô nhỏ trên máy tính cá nhân đến các hệ thống doanh nghiệp, kho dữ liệu lớn và dịch vụ web có số lượng người dùng đồng thời cao.

PostgreSQL được phát triển và duy trì bởi PostgreSQL Global Development Group, và được phát hành lần đầu vào ngày 08/07/1996. Hệ quản trị này có khả năng hoạt động trên nhiều nền tảng khác nhau như Linux, macOS và Windows.



Hình 8: Hệ quản trị cơ sở dữ liệu PostgreSQL

Bên cạnh các tính năng cốt lõi, PostgreSQL còn nổi bật nhờ khả năng mở rộng mạnh mẽ thông qua hệ thống extension phong phú. Các extension cho phép bổ sung thêm chức năng mới mà không cần thay đổi mã nguồn lõi của hệ quản trị, giúp PostgreSQL thích ứng linh hoạt với nhiều bài toán khác nhau, từ xử lý dữ liệu truyền thống đến các ứng dụng hiện đại như tìm kiếm ngữ nghĩa và trí tuệ nhân tạo.

Một số tính năng hiện đại của PostgreSQL:

- Tuân thủ chặt chẽ chuẩn SQL và hỗ trợ nhiều mở rộng nâng cao.
- Hỗ trợ các ràng buộc toàn vẹn dữ liệu.
- Đảm bảo toàn vẹn giao dịch theo mô hình ACID.
- Áp dụng cơ chế MVCC.
- Hệ sinh thái extension hỗ trợ linh hoạt.

1.3.2. Redis

Redis [11] là một hệ quản trị cơ sở dữ liệu mã nguồn mở lưu trữ trong bộ nhớ, thường được sử dụng như một hệ thống cache, message broker và kho dữ liệu tạm thời cho các ứng dụng hiệu năng cao. Redis hoạt động theo mô hình key–value, cho phép truy xuất dữ liệu với độ trễ rất thấp, phù hợp cho các hệ thống yêu cầu tốc độ xử lý nhanh và khả năng mở rộng tốt.

Redis được phát triển bởi Salvatore Sanfilippo và được phát hành lần đầu vào năm 2009. Redis là phần mềm mã nguồn mở và miễn phí, Redis cho phép người dùng tự do sử dụng, chỉnh sửa và phân phối cho nhiều mục đích khác nhau, từ các ứng dụng nhỏ đến các hệ thống phân tán quy mô lớn.



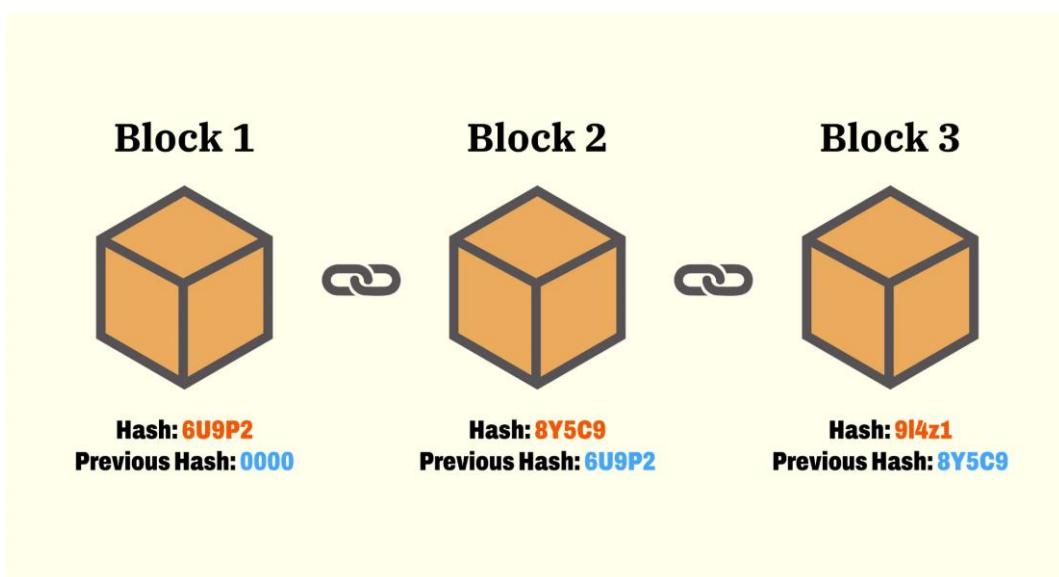
Hình 9: Hệ quản trị dữ liệu Redis

1.4. Thanh toán và bảo mật

1.4.1. Blockchain

Blockchain [12] là một công nghệ số cáy phân tán, cho phép lưu trữ và xác thực dữ liệu theo dạng các khối được liên kết với nhau bằng cơ chế mật mã. Mỗi khối chứa thông tin giao dịch và được gắn với khối trước đó, tạo thành một chuỗi dữ liệu liên tục và khó bị thay đổi. Nhờ đặc tính phi tập trung và bất biến, Blockchain giúp tăng tính minh bạch, an toàn và tin cậy trong việc lưu trữ và trao đổi dữ liệu.

Blockchain được giới thiệu lần đầu vào năm 2008 thông qua công trình nghiên cứu của Satoshi Nakamoto, làm nền tảng cho hệ thống tiền điện tử Bitcoin. Từ đó, công nghệ này đã được phát triển và mở rộng bởi nhiều cộng đồng và tổ chức khác nhau, ứng dụng trong nhiều lĩnh vực ngoài tiền điện tử.



Hình 10: Công nghệ Blockchain

1.4.2. SePay

SePay [13] là một nền tảng hỗ trợ thanh toán điện tử, được sử dụng để kết nối và tự động hóa quá trình xử lý giao dịch chuyển khoản ngân hàng trong các hệ thống phần mềm. SePay cho phép hệ thống nhận diện, xác thực và đối soát giao dịch thanh toán theo thời gian gần thực, từ đó hỗ trợ các nghiệp vụ như nạp tiền, thanh toán dịch vụ hoặc xác nhận đơn hàng một cách tự động mà không cần xử lý thủ công.

SePay được phát triển nhằm giải quyết bài toán tích hợp thanh toán ngân hàng nội địa tại Việt Nam, đặc biệt trong các hệ thống web và ứng dụng trực tuyến. Nền tảng này thường được sử

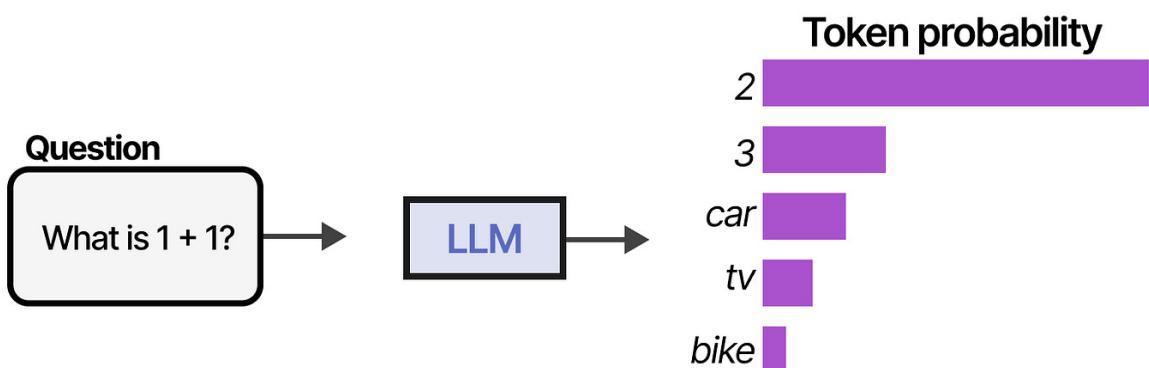
dụng thông qua API và webhook, cho phép các hệ thống backend tự động nhận thông tin giao dịch và cập nhật trạng thái thanh toán. SePay có thể tích hợp với nhiều ngôn ngữ lập trình và kiến trúc hệ thống khác nhau, phù hợp cho các ứng dụng thương mại điện tử và dịch vụ trực tuyến.

1.5. Tích hợp AI

1.5.1. Large Language Model

LLM [14] là một loại mô hình trí tuệ nhân tạo được xây dựng dựa trên Deep Learning, có khả năng hiểu và sinh ngôn ngữ tự nhiên ở quy mô lớn. LLM được huấn luyện trên tập dữ liệu văn bản khổng lồ nhằm học các quy luật ngôn ngữ, ngữ cảnh và tri thức tổng quát, từ đó có thể thực hiện nhiều tác vụ khác nhau như trả lời câu hỏi, tóm tắt văn bản, sinh nội dung, dịch ngôn ngữ và hỗ trợ lập trình. LLM hiện là nền tảng cốt lõi của nhiều hệ thống AI hiện đại và các ứng dụng NLP.

LLM được phát triển bởi nhiều tổ chức nghiên cứu và công ty công nghệ lớn như OpenAI, Google, Meta, Microsoft và các cộng đồng mã nguồn mở. Các mô hình ngôn ngữ lớn bắt đầu thu hút sự chú ý mạnh mẽ từ khoảng năm 2018 với sự ra đời của kiến trúc Transformer. LLM có thể được triển khai trên nhiều nền tảng khác nhau, từ hệ thống máy chủ hiệu năng cao, điện toán đám mây đến các môi trường nội bộ. Tùy theo mô hình cụ thể, LLM có thể là mã nguồn mở hoặc thương mại, phục vụ nhiều mục đích sử dụng khác nhau.



Hình 11: Mô tả cơ bản cách LLM hoạt động

1.5.2. Gemini

Gemini [15] là một dòng mô hình trí tuệ nhân tạo do Google phát triển, thuộc nhóm LLM. Gemini được thiết kế để xử lý và hiểu nhiều dạng dữ liệu khác nhau như văn bản, hình ảnh, mã nguồn và dữ liệu có cấu trúc, phục vụ cho các bài toán như sinh nội dung, hỏi đáp, phân tích ngữ cảnh và hỗ trợ lập trình. Bên cạnh các mô hình sinh ngôn ngữ, Gemini còn cung cấp các mô hình

embedding nhằm chuyển đổi dữ liệu văn bản thành vector số, phục vụ cho các bài toán tìm kiếm ngữ nghĩa và truy xuất thông tin.



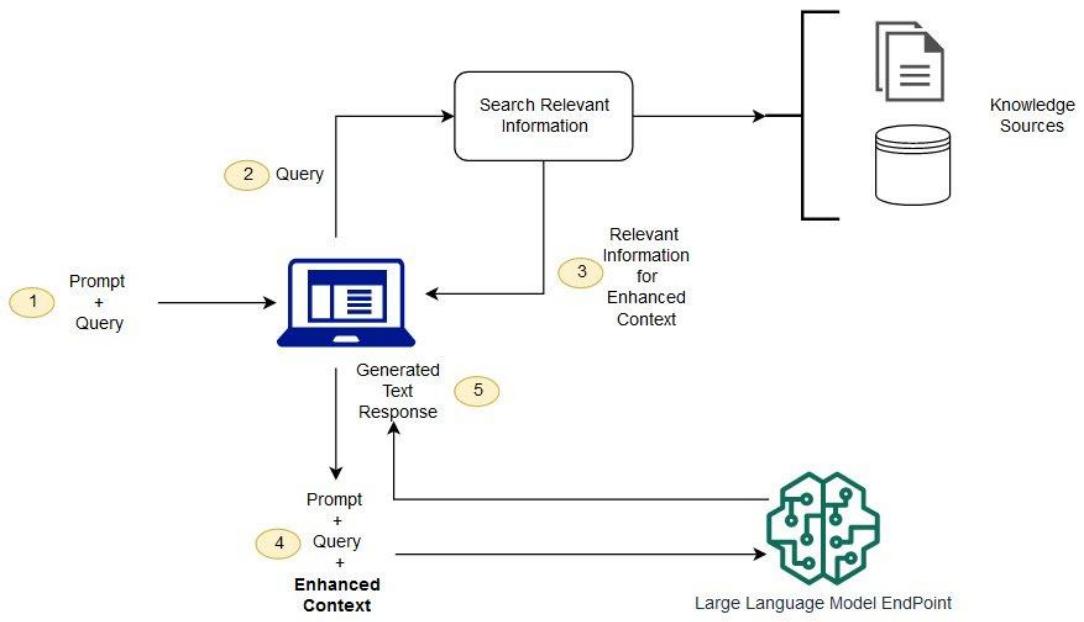
Hình 12: Gemini Model

Gemini được phát triển bởi Google DeepMind và được công bố lần đầu vào năm 2023. Các mô hình Gemini được triển khai chủ yếu thông qua hạ tầng điện toán đám mây của Google và tích hợp trong nhiều sản phẩm, dịch vụ AI. Tùy theo mục đích sử dụng, Gemini cung cấp các biến thể mô hình khác nhau, bao gồm mô hình sinh (generative model) và mô hình embedding, phục vụ cho các hệ thống AI quy mô từ nhỏ đến lớn.

1.5.3. Retrieval-Augmented Generation

RAG [16] là một kiến trúc kết hợp giữa cơ chế truy xuất thông tin và mô hình sinh ngôn ngữ, được sử dụng phổ biến trong các hệ thống trí tuệ nhân tạo hiện đại nhằm nâng cao độ chính xác và độ tin cậy của câu trả lời. Thay vì chỉ dựa vào tri thức đã được học trong quá trình huấn luyện, RAG cho phép hệ thống truy xuất dữ liệu liên quan từ các nguồn tri thức bên ngoài, sau đó sử dụng LLM để sinh nội dung dựa trên thông tin đã truy xuất.

RAG được giới thiệu và nghiên cứu đầu tiên bởi Facebook AI Research vào năm 2020. Kiến trúc này có thể được triển khai trên nhiều nền tảng khác nhau và tích hợp với các hệ thống lưu trữ tri thức như cơ sở dữ liệu truyền thống, cơ sở dữ liệu vector hoặc kho tài liệu nội bộ. RAG không phải là một sản phẩm đơn lẻ mà là một mô hình kiến trúc, thường được áp dụng trong các hệ thống AI doanh nghiệp, chatbot thông minh và hệ thống hỏi đáp dựa trên tri thức.



Hình 13: Mô tả cơ chế RAG hoạt động

Kiến trúc RAG mang lại nhiều ưu điểm quan trọng trong việc xây dựng các hệ thống sinh nội dung thông minh:

- Cho phép truy xuất thông tin từ nguồn dữ liệu bên ngoài trước khi sinh câu trả lời.
- Giảm hiện tượng sinh thông tin sai lệch của mô hình ngôn ngữ.
- Dễ dàng cập nhật tri thức mà không cần huấn luyện lại toàn bộ mô hình.
- Kết hợp hiệu quả giữa mô hình ngôn ngữ lớn và cơ sở dữ liệu tri thức.

1.6. DevOps và hạ tầng triển khai

1.6.1. Docker

Docker [17] là một nền tảng mã nguồn mở dùng để phát triển, đóng gói và triển khai ứng dụng theo mô hình container hóa. Docker cho phép đóng gói ứng dụng cùng với toàn bộ các thư viện, phụ thuộc và cấu hình cần thiết vào một container độc lập, giúp ứng dụng có thể chạy nhất quán trên nhiều môi trường khác nhau như máy phát triển, máy chủ nội bộ và hạ tầng điện toán đám mây. Công nghệ này giúp giải quyết hiệu quả vấn đề khác biệt môi trường giữa quá trình phát triển và triển khai hệ thống.

Docker được phát triển bởi Docker Inc. và được phát hành lần đầu vào năm 2013. Nền tảng này hoạt động trên nhiều hệ điều hành như Linux, macOS và Windows, đồng thời tận dụng các tính năng của hệ điều hành như Linux namespaces và cgroups để cô lập tài nguyên. Là một nền

tảng mã nguồn mở và miễn phí, Docker cho phép người dùng tự do sử dụng và tích hợp vào nhiều loại hệ thống khác nhau, từ các ứng dụng nhỏ đến các hệ thống phân tán quy mô lớn.



Hình 14: Nền tảng container Docker

1.6.2. Nginx

Nginx [18] là một web server mã nguồn mở, hỗ trợ reverse proxy, được sử dụng rộng rãi trong các hệ thống web hiện đại. Nginx có khả năng tiếp nhận các yêu cầu từ phía người dùng, phục vụ nội dung web và chuyển tiếp các request đến các máy chủ ứng dụng phía sau. Nhờ đó, Nginx thường được triển khai như một lớp trung gian giúp che giấu cấu trúc nội bộ của hệ thống và tối ưu việc phân phối lưu lượng truy cập.



Hình 15: Máy chủ web và reverse proxy Nginx

Với vai trò reverse proxy, Nginx giúp cải thiện hiệu năng, tăng khả năng mở rộng và hỗ trợ cân bằng tải giữa nhiều máy chủ backend. Công nghệ này thường được sử dụng trong các kiến trúc Microservices và các hệ thống web có lượng truy cập lớn, đóng vai trò quan trọng trong việc đảm bảo hệ thống hoạt động ổn định và hiệu quả.

1.7. Kết chương

Chương 1 đã trình bày các cơ sở lý thuyết và công nghệ làm nền tảng cho việc xây dựng hệ thống ứng dụng. Nội dung chương bao gồm kiến trúc phần mềm Microservices, công nghệ AI, Backend, Frontend cùng với các thành phần hạ tầng và triển khai. Trên cơ sở những nền tảng lý

thuyết và công nghệ đã được trình bày, chương tiếp theo sẽ tập trung vào việc phân tích yêu cầu và thiết kế hệ thống, nhằm làm rõ các yêu cầu chức năng, yêu cầu phi chức năng cũng như đề xuất kiến trúc tổng thể phù hợp cho hệ thống được xây dựng.

CHƯƠNG 2: PHÂN TÍCH YÊU CẦU VÀ THIẾT KẾ HỆ THỐNG

2.1. Phân tích yêu cầu chức năng

2.1.1. Xác định các tác nhân

Hệ thống có 4 tác nhân chính là quản trị viên, nhân viên quản lý, nhân viên bán hàng, và khách hàng, mỗi tác nhân đảm nhận các chức năng riêng biệt nhằm hỗ trợ quản lý các nghiệp vụ trong rạp chiếu phim:

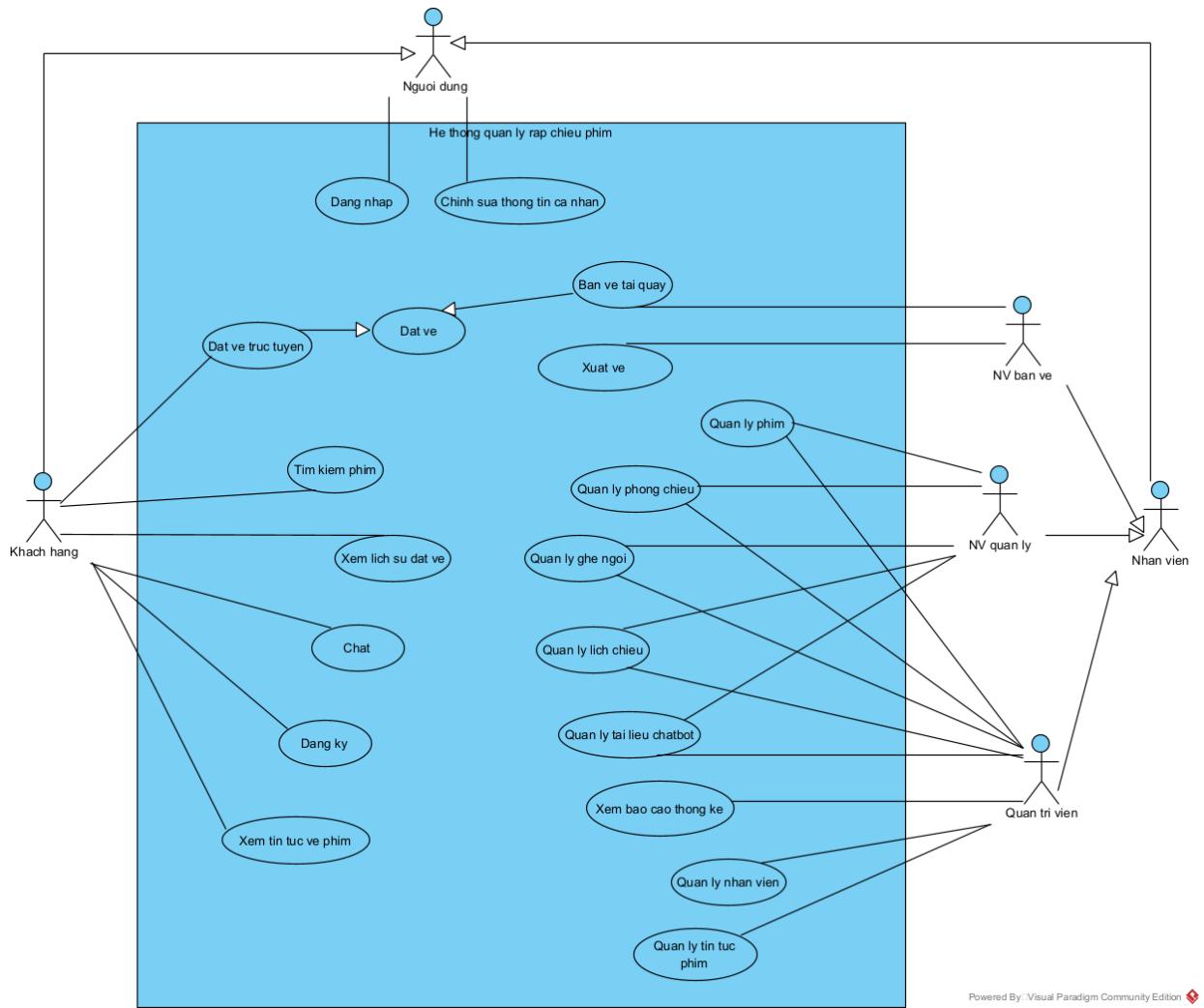
Tác nhân thứ 1 (Quản trị hệ thống): Là người quản lý toàn bộ hệ thống rạp chiếu phim, chịu trách nhiệm đảm bảo hoạt động vận hành ổn định và giám sát mọi nghiệp vụ diễn ra trong rạp. Quản trị hệ thống có quyền truy cập cao nhất và có thể sử dụng tất cả các chức năng trong hệ thống. Người quản trị có thể thực hiện việc quản lý thông tin phim, phòng chiếu, ghế của từng phòng, lịch chiếu và giá vé, bao gồm thêm mới, chỉnh sửa, xóa hoặc tìm kiếm dữ liệu. Người quản trị cũng có nhiệm vụ quản lý thông tin nhân viên, cũng như quản lý tài liệu chatbot phục vụ việc hỗ trợ khách hàng. Ngoài ra, quản trị hệ thống có thể xem các thống kê tổng hợp về doanh thu, số vé bán ra và hiệu suất hoạt động của rạp, từ đó đưa ra các quyết định điều hành phù hợp.

Tác nhân thứ 2 (Nhân viên quản lý): Là người phụ trách các nghiệp vụ quản lý nội bộ của rạp, chịu trách nhiệm vận hành các chức năng cần thiết nhưng không có đầy đủ quyền hạn như quản trị hệ thống. Nhân viên quản lý có thể thực hiện các tác vụ liên quan đến quản lý thông tin phim, phòng chiếu và ghế ngồi, lịch chiếu và giá vé, bao gồm thêm, sửa xóa, tìm kiếm dữ liệu.

Tác nhân thứ 3 (Nhân viên bán vé): Là người trực tiếp phục vụ khách hàng tại quầy, chịu trách nhiệm xử lý các yêu cầu liên quan đến mua vé và hỗ trợ khách hàng. Nhân viên bán vé có thể thực hiện việc bán vé trực tiếp tại quầy bằng cách lựa chọn phim, suất chiếu, ghế ngồi và in vé cho khách hàng. Đồng thời cũng có thể kiểm tra các đơn đặt vé trực tuyến để xác nhận thông tin, hỗ trợ khách hàng hoặc giải đáp thắc mắc về suất chiếu và thông tin phim.

Tác nhân thứ 4 (Khách hàng): Là người sử dụng hệ thống đặt vé trực tuyến thông qua website, có thể truy cập và sử dụng các chức năng phục vụ nhu cầu xem phim. Khách hàng có thể tìm kiếm phim, xem thông tin chi tiết về suất chiếu, phòng chiếu và giá vé để đưa ra lựa chọn phù hợp. Khách hàng thực hiện việc đặt vé trực tuyến bằng cách chọn phim, suất chiếu, ghế ngồi và thanh toán qua các phương thức hỗ trợ trong hệ thống. Ngoài ra, khách hàng có thể quản lý tài khoản cá nhân, chỉnh sửa thông tin và xem lịch sử đặt vé.

2.1.2. Biểu đồ Usecase tổng quan hệ thống



Hình 16: Biểu đồ usecase tổng quan của hệ thống

2.1.3. ĐẶC TẢ Usecase

Usecase name	Dăng nhập
Description	Cho phép người dùng đăng nhập vào hệ thống
Actor	Khách hàng, nhân viên bán vé, nhân viên quản lý, quản trị viên
Pre-condition	Người dùng đã có tài khoản trên hệ thống
Post condition	Đăng nhập thành công

Flow	<ol style="list-style-type: none"> 1. Người dùng chọn chức năng đăng nhập 2. Hệ thống hiển thị form đăng nhập 3. Người dùng nhập email và password để đăng nhập 4. Người dùng nhấn nút Đăng nhập 5. Hệ thống kiểm tra tính hợp lệ, đầy đủ và chính xác của các trường đã nhập 6. Hệ thống chuyển đến trang Home
Exception	<p>5a. Hệ thống thông báo lỗi: “Vui lòng điền đầy đủ thông tin” nếu nhập thiếu email hoặc mật khẩu</p> <p>5b. Hệ thống thông báo lỗi: “Tài khoản/mật khẩu không chính xác” nếu nhập sai email hoặc mật khẩu</p>

Bảng 1: Đặc tả usecase Đăng nhập

Usecase name	Đăng ký
Description	Cho phép khách hàng đăng ký
Actor	Khách hàng
Pre-condition	Người dùng chưa có tài khoản trên hệ thống
Post condition	Đăng ký thành công
Flow	<ol style="list-style-type: none"> 1. Người dùng chọn chức năng đăng ký 2. Hệ thống hiển thị form đăng ký 3. Người dùng nhập thông tin đăng ký 4. Người dùng nhấn nút Đăng ký 5. Hệ thống kiểm tra tính hợp lệ, đầy đủ và chính xác của các trường đã nhập 6. Hệ thống gửi mã xác nhận về cho người dùng qua email đăng ký và hiển thị giao diện cho phép nhập mã xác nhận 7. Người dùng nhập mã xác nhận 8. Hệ thống kiểm tra mã xác nhận có chính xác hay không 9. Hệ thống chuyển đến trang đăng nhập cho phép người dùng đăng nhập tài khoản vừa đăng ký

Exception	<p>5a. Hệ thống thông báo lỗi: “Vui lòng điền đầy đủ thông tin” nếu nhập thiếu email hoặc mật khẩu</p> <p>5b. Hệ thống thông báo lỗi: “Tài khoản/mật khẩu không chính xác” nếu nhập sai email hoặc mật khẩu</p> <p>8. Hệ thống thông báo: ”Mã xác nhận không chính xác” nếu nhập sai mã xác nhận</p>
-----------	--

Bảng 2: Đặc tả usecase Đăng ký

Usecase name	Đặt vé xem phim
Description	Usecase cho phép người dùng lựa chọn phim, suất chiếu, ghế, và tiến hành đặt vé xem phim.
Actor	Người dùng
Pre-condition	<p>Người dùng đã đăng nhập thành công.</p> <p>Phim, suất chiếu và phòng chiếu đang hoạt động và còn ghế trống.</p>
Post Condition	<p>Vé được đặt thành công và lưu vào hệ thống.</p> <p>Số lượng ghế trống được cập nhật lại.</p> <p>Hệ thống gửi email mã thông tin vé.</p>
Flow	<ol style="list-style-type: none"> Người dùng chọn chức năng Đặt vé hoặc chọn một phim từ danh sách. Hệ thống hiển thị danh sách suất chiếu của phim. Người dùng chọn suất chiếu mong muốn. Hệ thống hiển thị sơ đồ ghế ngồi và thông tin giá vé. Người dùng chọn ghế cần đặt. Người dùng xác nhận thông tin đặt vé (phim, suất chiếu, ghế, giá). Người dùng chọn phương thức thanh toán và tiến hành thanh toán. Hệ thống xử lý thanh toán. Hệ thống thông báo đặt vé thành công.

	10. Hệ thống gửi email cho người dùng thông tin vé xem phim.
Exception	<p>4a. Hệ thống thông báo: “Ghé bạn chọn đã được đặt, vui lòng chọn ghé khác.”</p> <p>7a. Hệ thống thông báo: “Phương thức thanh toán không hợp lệ hoặc thiếu thông tin.”</p> <p>8a. Hệ thống thông báo: “Thanh toán không thành công, vui lòng thử lại.”</p>

Bảng 3: Đặc tả usecase đặt vé xem phim

Usecase name	Quản lý phim
Description	Usecase cho phép quản trị viên/nhân viên xem, thêm, sửa, xoá các phim của rạp chiếu
Actor	Quản trị viên, nhân viên quản lý
Pre-condition	Quản trị viên, nhân viên đăng nhập vào trang admin và ở mục quản lý phim.
Post Condition	Hệ thống cập nhật danh sách phim.
Flow	<ol style="list-style-type: none"> Quản trị viên/nhân viên quản lý chọn chức năng thêm phim. Hệ thống hiển thị form thêm phim mới. Quản trị viên/nhân viên quản lý nhập đầy đủ thông tin phim và bấm nút thêm mới. Hệ thống kiểm tra tính hợp lệ và đầy đủ của các trường quản trị viên/nhân viên quản lý nhập. Hệ thống chuyển về trang quản lý phim, hiển thị thông báo thêm phim thành công và cập nhật lại danh sách phim. Quản trị viên/nhân viên quản lý chọn vào phim cần quản lý. Hệ thống hiển thị nút sửa, xóa phim và thông tin chi tiết của phim được chọn. Quản trị viên/nhân viên quản lý chọn chức năng sửa phim.

	<p>9. Hệ thống hiển thị form sửa phim.</p> <p>10. Quản trị viên/nhân viên quản lý nhập đầy đủ thông tin phim cần sửa và ấn nút sửa.</p> <p>11. Hệ thống kiểm tra tính hợp lệ và đầy đủ của các trường quản trị viên/nhân viên quản lý nhập.</p> <p>12. Hệ thống chuyển về trang quản lý phim, hiển thị thông báo sửa phim thành công và cập nhật lại phim.</p> <p>13. Quản trị viên/nhân viên quản lý chọn vào phim cần quản lý và chọn nút xoá.</p> <p>14. Hệ thống xóa phim mà quản trị viên/nhân viên quản lý vừa chọn.</p>
Exception	<p>4. Hệ thống thông báo: “Vui lòng điền đầy đủ thông tin.” nếu nhập thiếu dữ liệu phim.</p> <p>11. Hệ thống thông báo: “Vui lòng điền đầy đủ thông tin.” nếu nhập thiếu dữ liệu phim.</p>

Bảng 4: Đặc tả usecase quản lý phim

Usecase name	Quản lý phòng chiếu
Description	Usecase cho phép quản trị viên xem, thêm, sửa, xoá các phòng của rạp chiếu
Actor	Quản trị viên, nhân viên quản lý
Pre-condition	Quản trị viên, nhân viên đăng nhập vào trang admin và ở mục quản lý phòng chiếu.
Post Condition	Hệ thống cập nhật danh sách phòng chiếu.
Flow	<ol style="list-style-type: none"> Quản trị viên hoặc nhân viên quản lý chọn chức năng thêm phòng chiếu. Hệ thống hiển thị form thêm phòng chiếu mới. Quản trị viên hoặc nhân viên quản lý nhập đầy đủ thông tin phòng chiếu và ấn nút tạo mới. Hệ thống kiểm tra tính hợp lệ và đầy đủ của các thông tin đã

	<p>nhập.</p> <p>5. Hệ thống lưu phòng chiếu mới, chuyển về trang quản lý phòng chiếu, đồng thời hiển thị thông báo thêm phòng chiếu thành công và cập nhật danh sách phòng chiếu.</p> <p>6. Quản trị viên hoặc nhân viên quản lý nhập tên phòng chiếu cần tìm vào ô tìm kiếm.</p> <p>7. Hệ thống hiển thị danh sách phòng chiếu phù hợp với từ khóa tìm kiếm</p> <p>8. Quản trị viên hoặc nhân viên quản lý chọn phòng chiếu cần quản lý.</p> <p>9. Hệ thống hiển thị thông tin chi tiết của phòng chiếu đó.</p> <p>10. Quản trị viên hoặc nhân viên quản lý chọn chức năng sửa phòng chiếu.</p> <p>11. Hệ thống hiển thị form chỉnh sửa thông tin phòng chiếu.</p> <p>12. Quản trị viên hoặc nhân viên quản lý cập nhật thông tin phòng chiếu và ấn nút lưu.</p> <p>13. Hệ thống kiểm tra tính hợp lệ của dữ liệu</p> <p>14. Hệ thống cập nhật phòng chiếu và hiển thị thông báo sửa phòng chiếu thành công.</p> <p>15. Quản trị viên hoặc nhân viên quản lý chọn chức năng xóa phòng chiếu.</p> <p>16. Hệ thống hiển thị hộp thoại xác nhận với nội dung: “Bạn có chắc chắn muốn xóa phòng chiếu này không?”</p> <p>17. Quản trị viên hoặc nhân viên quản lý xác nhận Đồng ý.</p> <p>18. Hệ thống thực hiện xóa phòng chiếu và cập nhật lại danh sách phòng chiếu.</p>
Exception	<p>4. Hệ thống thông báo: “Vui lòng điền đầy đủ và đúng định dạng thông tin phòng chiếu.”</p> <p>5. Hệ thống thông báo: “Thêm phòng chiếu không thành công.”</p> <p>7. Hệ thống thông báo: “Không tìm thấy phòng chiếu phù hợp.”</p> <p>13. Hệ thống thông báo: “Vui lòng điền đầy đủ và đúng định dạng thông tin phòng chiếu.”</p> <p>18. Hệ thống thông báo: “Xóa phòng chiếu không thành công.”</p>

Bảng 5: Đặc tả usecase quản lý phòng chiếu

Usecase name	Quản lý ghế ngồi
Description	Usecase cho phép quản trị viên xem, thêm, sửa, xóa các ghế ngồi trong phòng của rạp chiếu.
Actor	Quản trị viên, nhân viên quản lý
Pre-condition	Quản trị viên/nhân viên quản lý đã đăng nhập vào hệ thống và có quyền quản lý ghế ngồi.
Post Condition	Hệ thống cập nhật danh sách ghế ngồi.
Flow	<ol style="list-style-type: none"> 1. Quản trị viên/nhân viên quản lý truy cập chức năng quản lý ghế ngồi. 2. Hệ thống hiển thị danh sách ghế ngồi theo chế độ bảng hoặc lưới. 3. Quản trị viên/nhân viên quản lý chọn chức năng thêm ghế mới. 4. Hệ thống hiển thị form thêm ghế mới. 5. Quản trị viên/nhân viên quản lý nhập đầy đủ thông tin ghế và ấn nút tạo mới. 6. Hệ thống kiểm tra tính hợp lệ và đầy đủ của các trường quản trị viên/nhân viên quản lý nhập. 7. Hệ thống chuyển về trang quản lý ghế ngồi, hiển thị thông báo thêm ghế thành công và cập nhật lại danh sách ghế. 8. Quản trị viên/nhân viên quản lý sử dụng bộ lọc và tìm kiếm để tìm kiếm ghế. 9. Hệ thống hiển thị ghế cần tìm. 10. Quản trị viên/nhân viên quản lý chọn chức năng chỉnh sửa ghế. 11. Hệ thống hiển thị form chỉnh sửa ghế với thông tin hiện tại. 12. Quản trị viên/nhân viên quản lý chỉnh sửa thông tin ghế và ấn nút cập nhật. 13. Hệ thống kiểm tra tính hợp lệ và đầy đủ của các trường quản trị viên/nhân viên quản lý nhập. 14. Hệ thống chuyển về trang quản lý ghế ngồi, hiển thị thông

	<p>báo cập nhật ghế thành công và cập nhật lại danh sách ghế.</p> <p>15. Quản trị viên/nhân viên quản lý thay đổi trạng thái ghế trực tiếp trên danh sách.</p> <p>16. Hệ thống cập nhật trạng thái ghế và làm mới dữ liệu hiển thị.</p> <p>17. Quản trị viên/nhân viên quản lý chọn chức năng xóa ghế.</p> <p>18. Hệ thống hiển thị hộp thoại xác nhận xóa.</p> <p>19. Quản trị viên/nhân viên quản lý xác nhận xóa ghế.</p> <p>20. Hệ thống xóa ghế đã chọn và cập nhật lại danh sách ghế ngồi.</p>
Exception	<p>6. Hệ thống thông báo: “Vui lòng điền đầy đủ và đúng định dạng thông tin ghế ngồi”.</p> <p>7. Hệ thống thông báo: “Ghế ngồi đã tồn tại”.</p> <p>9. Hệ thống thông báo: “Không thấy ghế ngồi cần tìm”.</p> <p>13. Hệ thống thông báo: “Vui lòng điền đầy đủ và đúng định dạng thông tin ghế ngồi.”</p>

Bảng 6: Đặc tả usecase quản lý ghế

Usecase name	Quản lý suất chiếu
Description	Usecase cho phép quản trị viên xem, thêm, sửa, xóa các suất chiếu
Actor	Quản trị viên, nhân viên quản lý
Pre-condition	Quản trị viên, nhân viên đăng nhập vào trang admin và có quyền quản lý suất chiếu.
Post Condition	Hệ thống cập nhật danh sách suất chiếu.
Flow	<ol style="list-style-type: none"> Quản trị viên/nhân viên quản lý truy cập chức năng quản lý suất chiếu. Hệ thống hiển thị danh sách suất chiếu. Quản trị viên/nhân viên quản lý chọn chức năng thêm suất chiếu mới. Hệ thống hiển thị form thêm suất chiếu mới.

	<p>5. Quản trị viên/nhân viên quản lý nhập đầy đủ thông tin suất chiếu và ấn nút tạo mới.</p> <p>6. Hệ thống kiểm tra tính hợp lệ và đầy đủ của các trường quản trị viên/nhân viên quản lý nhập.</p> <p>7. Hệ thống chuyển về trang quản lý suất chiếu, hiển thị thông báo thêm suất chiếu thành công và cập nhật lại danh sách.</p> <p>8. Quản trị viên/nhân viên quản lý xem danh sách suất chiếu và sử dụng bộ lọc, tìm kiếm suất chiếu theo các tiêu chí.</p> <p>9. Hệ thống hiển thị suất chiếu cần tìm.</p> <p>10. Quản trị viên/nhân viên quản lý chọn chức năng chỉnh sửa thông tin suất chiếu tại dòng suất chiếu tương ứng trong danh sách.</p> <p>11. Hệ thống hiển thị form chỉnh sửa suất chiếu với thông tin hiện tại.</p> <p>12. Quản trị viên/nhân viên quản lý chỉnh sửa thông tin suất chiếu và ấn nút cập nhật.</p> <p>13. Hệ thống kiểm tra tính hợp lệ và đầy đủ của các trường quản trị viên/nhân viên quản lý nhập.</p> <p>14. Hệ thống chuyển về trang quản lý suất chiếu, hiển thị thông báo cập nhật thành công và cập nhật lại thông tin suất chiếu.</p> <p>15. Quản trị viên/nhân viên quản lý thay đổi trạng thái suất chiếu trực tiếp tại danh sách.</p> <p>16. Hệ thống cập nhật trạng thái suất chiếu và làm mới dữ liệu hiển thị.</p> <p>17. Quản trị viên/nhân viên quản lý chọn chức năng xóa suất chiếu tại dòng ghé tương ứng trong danh sách.</p> <p>18. Hệ thống hiển thị hộp thoại xác nhận xóa.</p> <p>19. Quản trị viên/nhân viên quản lý xác nhận xóa.</p> <p>20. Hệ thống xóa suất chiếu đã chọn và cập nhật lại danh sách suất chiếu.</p>
Exception	<p>6a. Hệ thống thông báo: “Vui lòng điền đầy đủ và đúng định dạng thông tin suất chiếu.”</p> <p>6b. Hệ thống thông báo: “Không thể tạo suất chiếu trong quá khứ.”</p> <p>6c. Hệ thống thông báo: “Thời lượng suất chiếu ngắn hơn thời</p>

	<p>lượng phim.”</p> <p>7a. Hệ thống thông báo: “Suất chiếu đã tồn tại.”</p> <p>7b. Hệ thống thông báo: “Lịch chiếu bị trùng với lịch chiếu khác trong phòng.”</p> <p>9. Hệ thống thông báo: “Không tìm thấy suất chiếu phù hợp.”</p> <p>14. Hệ thống thông báo: “Vui lòng điền đầy đủ và đúng định dạng thông tin suất chiếu.”</p> <p>20. Hệ thống thông báo: “Không thể xoá suất chiếu đã hoặc đang chiếu.”</p>
--	--

Bảng 7: Đặc tả usecase quản lý suất chiếu

Usecase name	Quản lý nhân viên
Description	Usecase cho phép quản trị viên xem, thêm, sửa, xóa nhân viên trong hệ thống rạp chiếu.
Actor	Quản trị viên, nhân viên quản lý
Pre-condition	Quản trị viên/nhân viên quản lý đã đăng nhập vào hệ thống và có quyền quản lý nhân viên.
Post Condition	Hệ thống cập nhật danh sách nhân viên.
Flow	<ol style="list-style-type: none"> Quản trị viên/nhân viên quản lý truy cập chức năng quản lý nhân viên. Hệ thống hiển thị danh sách nhân viên. Quản trị viên/nhân viên quản lý chọn chức năng thêm nhân viên mới. Hệ thống hiển thị form thêm nhân viên mới Quản trị viên/nhân viên quản lý nhập đầy đủ thông tin nhân viên và ấn nút tạo mới. Hệ thống kiểm tra tính hợp lệ và đầy đủ của các trường quản trị viên/nhân viên quản lý nhập. Hệ thống chuyển về trang quản lý nhân viên, hiển thị thông báo thêm nhân viên mới thành công và cập nhật lại danh

	<p>sách.</p> <ol style="list-style-type: none"> 8. Quản trị viên/nhân viên quản lý xem danh sách nhân viên và sử dụng bộ lọc, tìm kiếm nhân viên theo các tiêu chí. 9. Hệ thống hiển thị nhân viên cần tìm. 10. Quản trị viên/nhân viên quản lý chọn chức năng chỉnh sửa nhân viên tại dòng nhân viên tương ứng trong danh sách. 11. Hệ thống hiển thị form chỉnh sửa nhân viên với thông tin hiện tại. 12. Quản trị viên/nhân viên quản lý chỉnh sửa thông tin nhân viên và ấn nút cập nhật. 13. Hệ thống kiểm tra tính hợp lệ và đầy đủ của các trường quản trị viên/nhân viên quản lý nhập. 14. Hệ thống chuyển về trang quản lý nhân viên, hiển thị thông báo cập nhật thành công và cập nhật lại thông tin nhân viên. 15. Quản trị viên/nhân viên quản lý chọn chức năng xóa nhân viên tại dòng tương ứng trong danh sách. 16. Hệ thống hiển thị hộp thoại xác nhận xóa. 17. Quản trị viên/nhân viên quản lý xác nhận xóa. 18. Hệ thống xóa suất chiếu đã chọn và cập nhật lại danh sách nhân viên.
Exception	<ol style="list-style-type: none"> 6. Hệ thống thông báo: “Vui lòng điền đầy đủ và đúng định dạng thông tin nhân viên.” 9. Hệ thống thông báo: “Không tìm thấy nhân viên.” 13. Hệ thống thông báo: “Vui lòng điền đầy đủ và đúng định dạng thông tin suất chiếu.”

Bảng 8: *Đặc tả usecase quản lý nhân viên*

Usecase name	Quản lý tin tức về phim
Description	Usecase cho phép quản trị viên sửa hoặc xoá tin tức.
Actor	Quản trị viên, nhân viên quản lý
Pre-condition	Quản trị viên/nhân viên quản lý đã đăng nhập vào hệ thống và có

	quyền quản lý tin tức.
Post Condition	Hệ thống cập nhật danh sách tin tức.
Flow	<ol style="list-style-type: none"> 1. Quản trị viên/nhân viên quản lý truy cập chức năng quản lý tin tức. 2. Hệ thống hiển thị danh sách tin tức. 3. Quản trị viên/nhân viên quản lý xem danh sách tin tức và sử dụng bộ lọc, tìm kiếm tin tức theo các tiêu chí. 4. Hệ thống hiển thị tin tức cần tìm. 5. Quản trị viên/nhân viên quản lý chọn chức năng chỉnh sửa nội dung tin tức tại dòng tương ứng trong danh sách. 6. Hệ thống hiển thị form chỉnh sửa nội dung với thông tin hiện tại. 7. Quản trị viên/nhân viên quản lý chỉnh sửa nội dung và ấn nút cập nhật. 8. Hệ thống kiểm tra tính hợp lệ và đầy đủ của các trường quản trị viên/nhân viên quản lý nhập. 9. Hệ thống chuyển về trang quản lý tin tức, hiển thị thông báo cập nhật thành công và cập nhật lại tin tức. 10. Quản trị viên/nhân viên quản lý chọn chức năng xóa tin tức tại dòng tương ứng trong danh sách. 11. Hệ thống hiển thị hộp thoại xác nhận xóa. 12. Quản trị viên/nhân viên quản lý xác nhận xóa. 13. Hệ thống xóa tin tức đã chọn và cập nhật lại danh sách tin tức.
Exception	<ol style="list-style-type: none"> 2. Hệ thống thông báo: “Có lỗi xảy ra khi tải dữ liệu”. 4. Hệ thống thông báo: “Không tìm thấy tin tức.”

Bảng 9: ĐẶC TẢ USECASE QUẢN LÝ TIN TỨC VỀ PHIM

Usecase name	Quản lý tài liệu RAG chatbot
Description	Usecase cho phép quản trị viên cung cấp hoặc xoá tài liệu RAG

	cho chatbot.
Actor	Quản trị viên, nhân viên quản lý
Pre-condition	Quản trị viên/nhân viên quản lý đã đăng nhập vào hệ thống và có quyền cung cấp RAG cho chatbot.
Post Condition	Hệ thống cập nhật danh sách tài liệu.
Flow	<ol style="list-style-type: none"> 1. Quản trị viên/nhân viên quản lý truy cập chức năng quản lý RAG chat bot. 2. Hệ thống hiển thị danh sách tài liệu đã được tải lên. 3. Quản trị viên chọn file tài liệu rồi ấn nút đăng tải. 4. Hệ thống cập nhật danh sách tài liệu. 5. Quản trị viên/nhân viên quản lý chọn chức năng xóa tin tức tại dòng tương ứng trong danh sách. 6. Hệ thống hiển thị hộp thoại xác nhận xóa. 7. Quản trị viên/nhân viên quản lý xác nhận xóa. 8. Hệ thống xóa tin tức đã chọn và cập nhật lại danh sách tin tức.
Exception	<ol style="list-style-type: none"> 3. Hệ thống thông báo: “Lỗi đăng tải file tài liệu.” 6. Hệ thống thông báo: “Xóa không thành công.”

Bảng 10: Đặc tả usecase quản lý tài liệu chatbot

Usecase name	Chat với hệ thống
Description	Usecase cho phép người dùng chat với hệ thống
Actor	Người dùng
Pre-condition	Người dùng truy cập vào hệ thống rạp chiếu phim.
Post Condition	Người dùng nhận được câu trả lời.
Flow	<ol style="list-style-type: none"> 1. Người dùng chọn chức năng chat bot 2. Hệ thống hiển thị box chat 3. Người dùng đặt câu hỏi, trò chuyện với chat bot

	4. Hệ thống phản hồi câu trả lời của chat bot cho người dùng
Exception	4. Hệ thống thông báo: "Chat bot hiện đang lỗi, thử lại sau."

Bảng 11: Đặc tả usecase chat với hệ thống

Usecase name	Xem báo cáo thống kê doanh thu
Description	Usecase cho quản trị viên xem báo cáo doanh thu
Actor	Quản trị viên, nhân viên quản lý
Pre-condition	Quản trị viên/nhân viên quản lý đã đăng nhập vào hệ thống và có quyền xem thống kê doanh thu.
Post Condition	Thông tin doanh thu được hiển thị đầy đủ, chính xác theo các tiêu chí mà quản trị viên/nhân viên quản lý lựa chọn.
Flow	<ol style="list-style-type: none"> 1. Quản trị viên/nhân viên quản lý truy cập chức năng quản lý doanh thu. 2. Hệ thống hiển thị trang quản lý doanh thu với các bộ lọc thời gian và thống kê tổng quan. 3. Quản trị viên/nhân viên quản lý chọn khoảng thời gian cần xem doanh thu theo tháng hoặc theo năm. 4. Hệ thống hiển thị doanh thu tương ứng theo khoảng thời gian đã chọn. 5. Quản trị viên/nhân viên quản lý chọn chức năng xem doanh thu theo phim. 6. Hệ thống hiển thị danh sách phim kèm theo doanh thu tương ứng của từng phim. 7. Quản trị viên/nhân viên quản lý chọn chức năng xem doanh thu theo thể loại phim. 8. Hệ thống hiển thị doanh thu được tổng hợp theo từng thể loại phim. 9. Quản trị viên/nhân viên quản lý thay đổi các bộ lọc hoặc tiêu chí thống kê khi cần. 10. Hệ thống cập nhật và hiển thị lại dữ liệu doanh thu theo

	tiêu chí mới được chọn.
Exception	<p>2. Hệ thống thông báo: “Không thể hiển thị dữ liệu doanh thu.”</p> <p>4. Hệ thống thông báo: “Không có dữ liệu doanh thu trong khoảng thời gian đã chọn.”</p> <p>6. Hệ thống thông báo: “Không thể hiển thị doanh thu theo phim.”</p> <p>8. Hệ thống thông báo: “Không thể hiển thị doanh thu theo thể loại phim.”</p>

Bảng 12: *Đặc tả usecase xem báo cáo thống kê doanh thu*

Usecase name	Xem tin tức
Description	Use case cho phép người dùng xem các tin tức liên quan đến phim ảnh được hệ thống cung cấp.
Actor	Người dùng
Pre-condition	Người dùng truy cập vào hệ thống rạp chiếu phim.
Post Condition	Người dùng nhận được câu trả lời.
Flow	<p>1. Người dùng truy cập vào trang tin tức.</p> <p>2. Hệ thống hiển thị danh sách các tin tức về phim ảnh.</p> <p>3. Người dùng chọn một tin tức bất kỳ trong danh sách.</p> <p>4. Hệ thống hiển thị nội dung chi tiết của tin tức được chọn.</p>
Exception	<p>2. Hệ thống thông báo: “Không thể hiển thị danh sách tin tức.”</p> <p>4. Hệ thống thông báo: “Không thể hiển thị nội dung tin tức.”</p>

Bảng 13: *Đặc tả usecase xem tin tức*

Usecase name	Xem thông báo
Description	Use case cho phép người dùng xem danh sách các thông báo cá

	nhân của mình và cập nhật trạng thái thông báo sau khi đã xem.
Actor	Người dùng
Pre-condition	Người dùng đã đăng nhập vào hệ thống rạp chiếu phim.
Post Condition	Thông báo được hiển thị cho người dùng và trạng thái thông báo được cập nhật thành đã xem.
Flow	<ol style="list-style-type: none"> 1. Người dùng nhấn vào biểu tượng chuông thông báo trên giao diện hệ thống. 2. Hệ thống hiển thị danh sách các thông báo của người dùng. 3. Người dùng chọn một thông báo trong danh sách để xem nội dung chi tiết. 4. Hệ thống hiển thị nội dung thông báo được chọn. 5. Hệ thống cập nhật trạng thái thông báo thành đã xem.
Exception	<ol style="list-style-type: none"> 2. Hệ thống thông báo: “Không thể hiển thị danh sách thông báo.” 4. Hệ thống thông báo: “Không thể hiển thị nội dung thông báo.” 5. Hệ thống thông báo: “Cập nhật trạng thái thông báo không thành công.”

Bảng 14: Đặc tả usecase xem thông báo

Usecase name	Bán vé tại quầy
Description	Use case cho phép nhân viên bán vé trực tiếp cho khách hàng tại quầy rạp chiếu phim thông qua hệ thống.
Actor	Nhân viên bán vé
Pre-condition	Nhân viên bán vé đã đăng nhập vào hệ thống và có quyền bán vé tại quầy.
Post Condition	Vé được bán thành công và thông tin vé được lưu vào hệ

	thống.
Flow	<ol style="list-style-type: none"> 1. Nhân viên bán vé truy cập chức năng bán vé tại quầy. 2. Hệ thống hiển thị danh sách phim đang chiếu và suất chiếu tương ứng. 3. Nhân viên bán vé chọn phim và suất chiếu theo yêu cầu của khách hàng. 4. Hệ thống hiển thị sơ đồ ghế của phòng chiếu. 5. Nhân viên bán vé chọn ghế cho khách hàng. 6. Hệ thống hiển thị thông tin vé và tổng số tiền cần thanh toán. 7. Nhân viên bán vé xác nhận thông tin và thực hiện thanh toán. 8. Hệ thống ghi nhận giao dịch, tạo vé và hiển thị thông báo bán vé thành công.
Exception	<ol style="list-style-type: none"> 2. Hệ thống thông báo: “Không thể hiển thị danh sách phim và suất chiếu.” 4. Hệ thống thông báo: “Không thể hiển thị sơ đồ ghế.” 6. Hệ thống thông báo: “Không thể tạo thông tin vé.” 8. Hệ thống thông báo: “Bán vé không thành công.”

Bảng 15: Đặc tả usecase bán vé tại quầy

Usecase name	Tra cứu vé của khách hàng
Description	Use case cho phép nhân viên tra cứu thông tin vé của khách hàng trong hệ thống.
Actor	Nhân viên bán vé
Pre-condition	Nhân viên đã đăng nhập vào hệ thống và có quyền tra cứu vé.
Post Condition	Thông tin vé của khách hàng được hiển thị hoặc xác nhận không tồn tại.
Flow	<ol style="list-style-type: none"> 1. Nhân viên truy cập chức năng tra cứu vé của khách hàng.

	<ol style="list-style-type: none"> 2. Hệ thống hiển thị giao diện tra cứu vé. 3. Nhân viên nhập thông tin tra cứu vé và thực hiện tìm kiếm. 4. Hệ thống hiển thị thông tin vé tương ứng với dữ liệu đã nhập.
Exception	<ol style="list-style-type: none"> 3. Hệ thống thông báo: “Vui lòng nhập thông tin tra cứu vé.” 4. Hệ thống thông báo: “Không tìm thấy vé của khách hàng.”

Bảng 16: Đặc tả usecase tra cứu vé của khách hàng

Usecase name	Tìm kiếm phim
Description	Use case cho phép người dùng tìm kiếm phim trong hệ thống rạp chiếu phim theo các tiêu chí khác nhau.
Actor	Người dùng
Pre-condition	Người dùng truy cập vào hệ thống rạp chiếu phim.
Post Condition	Danh sách phim phù hợp với tiêu chí tìm kiếm được hiển thị.
Flow	<ol style="list-style-type: none"> 1. Người dùng truy cập chức năng tìm kiếm phim. 2. Hệ thống hiển thị ô tìm kiếm và các bộ lọc phim. 3. Người dùng nhập từ khóa tìm kiếm hoặc chọn tiêu chí lọc và thực hiện tìm kiếm. 4. Hệ thống hiển thị danh sách phim phù hợp với yêu cầu tìm kiếm.
Exception	<ol style="list-style-type: none"> 3. Hệ thống thông báo: “Vui lòng nhập thông tin tìm kiếm.” 4. Hệ thống thông báo: “Không tìm thấy phim phù hợp.”

Bảng 17: Đặc tả usecase tìm kiếm phim

Usecase name	Xem lịch sử đặt vé
Description	Use case cho phép người dùng xem lại danh sách các vé đã đặt

	trước đó trong hệ thống.
Actor	Người dùng
Pre-condition	Người dùng truy cập vào hệ thống rạp chiếu phim.
Post Condition	Danh sách lịch sử đặt vé của người dùng được hiển thị.
Flow	<ol style="list-style-type: none"> 1. Người dùng truy cập chức năng xem lịch sử đặt vé. 2. Hệ thống hiển thị danh sách các vé mà người dùng đã đặt. 3. Người dùng xem thông tin chi tiết của từng vé trong lịch sử.
Exception	<ol style="list-style-type: none"> 2. Hệ thống thông báo: “Không thể hiển thị lịch sử đặt vé.” 3. Hệ thống thông báo: “Không có dữ liệu lịch sử đặt vé.”

Bảng 18: Đặc tả usecase xem lịch sử đặt vé

Usecase name	Chỉnh sửa thông tin cá nhân
Description	Use case cho phép người dùng cập nhật thông tin cá nhân của mình trong hệ thống.
Actor	Người dùng
Pre-condition	Người dùng truy cập vào hệ thống rạp chiếu phim.
Post Condition	Thông tin cá nhân của người dùng được cập nhật thành công.
Flow	<ol style="list-style-type: none"> 1. Người dùng truy cập chức năng chỉnh sửa thông tin cá nhân. 2. Hệ thống hiển thị thông tin cá nhân hiện tại. 3. Người dùng chọn nút chỉnh sửa thông tin cá nhân và điền form. 4. Hệ thống kiểm tra tính hợp lệ của dữ liệu người dùng nhập. 5. Hệ thống cập nhật thông tin và hiển thị thông báo cập

	nhật thành công.
Exception	<p>4. Hệ thống thông báo: “Vui lòng nhập đầy đủ thông tin cá nhân.”</p> <p>5. Hệ thống thông báo: “Cập nhật thông tin cá nhân không thành công.”</p>

Bảng 19: Đặc tả usecase chính sửa thông tin cá nhân

Usecase name	Xuất vé
Description	Use case cho phép nhân viên bán vé xuất vé.
Actor	Nhân viên bán vé
Pre-condition	Nhân viên bán vé truy cập vào hệ thống rạp chiếu phim.
Post Condition	Xuất vé cho khách hàng thành công.
Flow	<ol style="list-style-type: none"> 1. Nhân viên bán vé truy cập chức năng xuất vé. 2. Hệ thống hiển thị ô nhập mã đơn. 3. Nhân viên chọn nút tìm kiếm. 4. Hệ thống hiển thị các vé của đơn đặt. 5. Nhân viên chọn vé và ấn nút xuất vé. 6. Hệ thống hiển thị hộp thoại thông báo xác nhận. 7. Nhân viên ấn đồng ý. 8. Hệ thống cập nhật vé xuất và thông báo thành công.
Exception	<p>4. Hệ thống thông báo: “Đơn đặt vé không tồn tại.”</p> <p>8. Hệ thống thông báo: “Xuất vé không thành công.”</p>

Bảng 20: Đặc tả usecase xuất vé

2.2. Phân tích yêu cầu phi chức năng

2.2.1. Yêu cầu bảo mật

Hệ thống được đề ra mục tiêu đảm bảo các yêu cầu bảo mật theo nhiều lớp, tập trung vào nội dung chính: bảo mật phân quyền, bảo mật xác thực thông qua cơ chế mã hóa token và cấu hình CORS.

Về bảo mật phân quyền, hệ thống áp dụng mô hình RBAC với bốn vai trò chính gồm: quản trị viên, nhân viên quản lý, nhân viên bán vé và khách hàng. Mỗi vai trò được gán các quyền truy cập cụ thể. Người dùng chỉ có thể thực hiện các chức năng phù hợp với quyền hạn được cấp, trong khi các chức năng ngoài quyền hạn dữ liệu chỉ được phép thực hiện bởi các vai trò có thẩm quyền.

Về bảo mật thông tin, hệ thống sử dụng cơ chế xác thực dựa trên JWT với thuật toán ký HMAC-SHA256. Sau khi đăng nhập thành công, người dùng được cấp access token có thời hạn sử dụng nhất định và hỗ trợ refresh token nhằm giảm thiểu rủi ro lộ thông tin xác thực. Token được quản lý tạm thời trên Redis, cho phép kiểm soát vòng đời và hỗ trợ thu hồi khi cần thiết.

Cơ chế CORS được cấu hình nhằm chỉ cho phép truy cập từ các nguồn đáng tin cậy, đồng thời các endpoint quan trọng của hệ thống đều được bảo vệ thông qua middleware xác thực và phân quyền chi tiết, đảm bảo an toàn trong quá trình truy cập và xử lý dữ liệu.

2.2.2. Yêu cầu hiệu năng

Hệ thống quản lý được xây dựng theo mô hình Client – Server, trong đó phía Frontend sử dụng ReactJS để xây dựng giao diện người dùng, còn Backend được triển khai theo kiến trúc Microservices. Frontend áp dụng cơ chế lazy loading nhằm giảm băng thông truyền tải và nâng cao trải nghiệm người dùng.

Các service Backend được triển khai độc lập, mỗi service đảm nhiệm một chức năng nghiệp vụ cụ thể của hệ thống. Việc giao tiếp giữa các service được thực hiện thông qua gRPC đối với các tác vụ đồng bộ yêu cầu hiệu năng cao, và thông qua Redis Pub/Sub đối với các kịch bản giao tiếp bất đồng bộ. Dữ liệu của hệ thống được lưu trữ trên PostgreSQL, đồng thời kết hợp Redis làm bộ nhớ đệm nhằm giảm tải cho cơ sở dữ liệu và cải thiện tốc độ truy xuất.

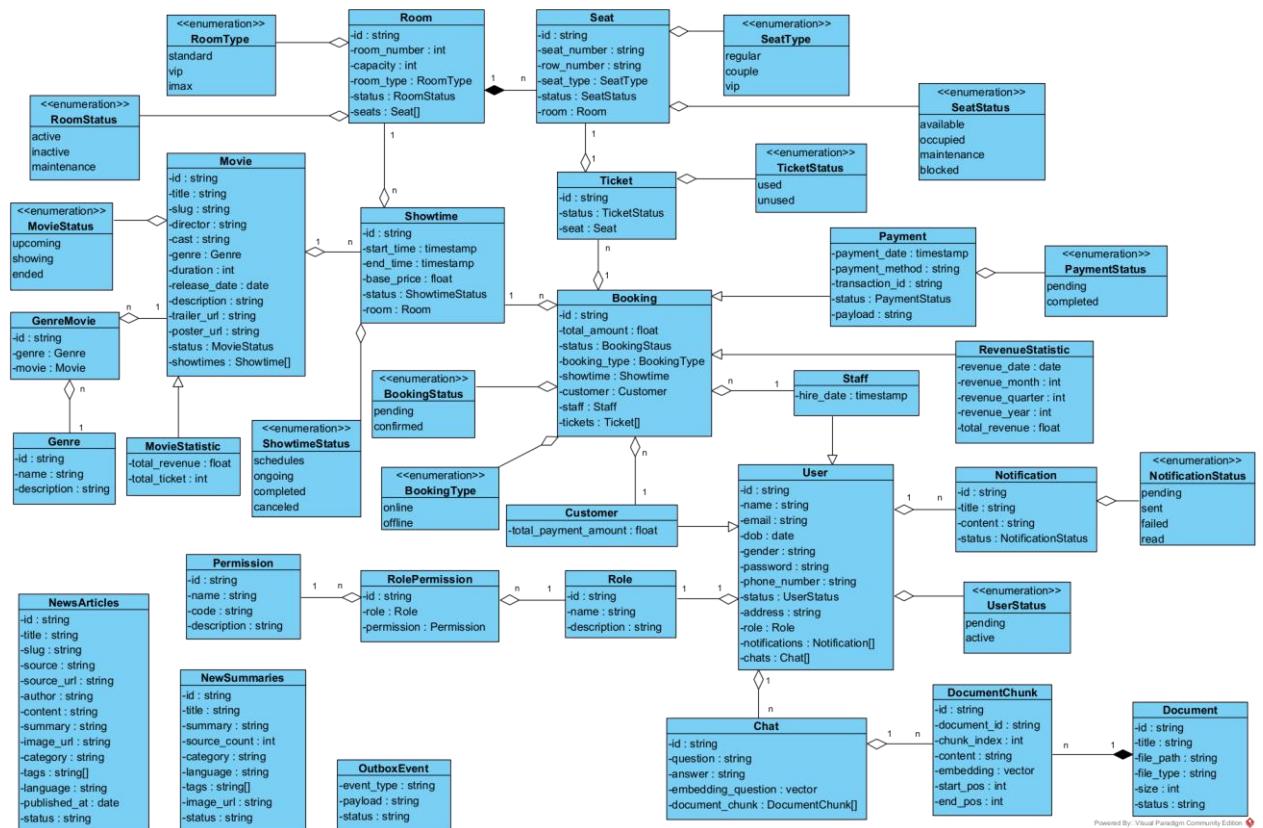
Bên cạnh đó, hệ thống áp dụng Transactional Outbox Pattern đảm bảo các sự kiện nghiệp vụ được xử lý một cách tin cậy và duy trì tính nhất quán trong môi trường phân tán. Đối với các kịch bản xử lý đồng thời, đặc biệt là những nghiệp vụ nhạy cảm như đặt vé và thanh toán, hệ thống

triển khai cơ chế khóa ở mức nghiệp vụ nhằm kiểm soát truy cập tài nguyên dùng chung, qua đó ngăn chặn hiệu quả tình trạng nhiều yêu cầu được xử lý cùng thời điểm.

Toàn bộ Backend được truy cập thông qua API Gateway, chịu trách nhiệm định tuyến yêu cầu và kiểm soát lưu lượng truy cập. Với cách tổ chức này, hệ thống đáp ứng được các yêu cầu về hiệu năng, khả năng mở rộng và độ ổn định trong môi trường vận hành thực tế.

2.3. Thiết kế hệ thống

2.3.1. Thiết kế biểu đồ lớp hệ thống



Hình 17: Biểu đồ lớp thiết kế hệ thống

Sau đây là bảng phân tích chi tiết các lớp:

STT	Tên lớp	Mô tả
1	User	Lớp User dùng để lưu thông tin người dùng trong hệ thống, bao gồm họ tên, email, ngày sinh, giới tính, số điện thoại, mật khẩu, địa chỉ, trạng thái tài khoản
2	Customer	Lớp Customer dùng để lưu thông tin khách hàng đặt

		vé, bao gồm tổng số tiền đã thanh toán trong hệ thống.
3	Staff	Lớp Staff dùng để lưu thông tin nhân viên như ngày được tuyển dụng.
4	Role	Lớp Role dùng để định nghĩa các vai trò trong hệ thống (quản trị viên, nhân viên bán vé, nhân viên quản lý, khách hàng), bao gồm tên vai trò, mô tả.
5	Permission	Lớp Permission dùng để quản lý các quyền truy cập chức năng trong hệ thống, bao gồm mã quyền, tên quyền và mô tả.
6	RolePermission	Lớp trung gian RolePermission dùng để liên kết từng quyền với vai trò, cho phép xác định mỗi vai trò được phép thực hiện những chức năng nào.
7	Movie	Lớp Movie dùng để lưu thông tin phim, bao gồm tiêu đề, đạo diễn, diễn viên, thể loại, thời lượng, ngày phát hành, mô tả, trailer, poster, trạng thái phim.
8	Genre	Lớp Genre dùng để lưu thông tin thể loại phim, bao gồm tên thể loại, mô tả.
9	GenreMovie	Lớp trung gian GenreMovie dùng để liên kết phim với thể loại, cho phép một phim có thể thuộc nhiều thể loại khác nhau.
10	Room	Lớp Room dùng để lưu thông tin phòng chiếu phim, bao gồm số phòng, sức chứa, loại phòng và trạng thái phòng.
11	Seat	Lớp Seat dùng để lưu thông tin ghế trong phòng chiếu, bao gồm số ghế, hàng ghế, loại ghế và trạng thái ghế.
12	Showtime	Lớp Showtime dùng để lưu thông tin suất chiếu phim, bao gồm thời gian bắt đầu, thời gian kết thúc, giá vé cơ bản, trạng thái suất chiếu, phòng chiếu và phim

		được trình chiếu.
13	Booking	Lớp Booking dùng để lưu thông tin đặt vé của khách hàng, bao gồm tổng tiền, trạng thái đặt vé, hình thức đặt vé, suất chiếu, khách hàng, nhân viên xử lý và danh sách vé.
14	Ticket	Lớp Ticket dùng để lưu thông tin vé xem phim, bao gồm trạng thái vé và ghế tương ứng.
15	Payment	Lớp Payment dùng để lưu thông tin thanh toán cho đơn đặt vé, bao gồm thời gian thanh toán, phương thức thanh toán, mã giao dịch, trạng thái thanh toán và dữ liệu thanh toán.
16	Notification	Lớp Notification dùng để lưu thông tin thông báo gửi tới người dùng, bao gồm người nhận thông báo, tiêu đề, nội dung và trạng thái thông báo.
17	NewsArticles	Lớp NewsArticles dùng để lưu thông tin bài viết tin tức, bao gồm tiêu đề, nguồn, tác giả, nội dung, tóm tắt, hình ảnh, danh mục, ngôn ngữ, ngày đăng và trạng thái bài viết.
18	NewsSummaries	Lớp NewsSummaries dùng để lưu thông tin tóm tắt tin tức, bao gồm tiêu đề, nội dung tóm tắt, nguồn, danh mục, hình ảnh và trạng thái.
19	Document	Lớp Document dùng để lưu thông tin tài liệu, bao gồm tiêu đề, đường dẫn tệp, loại tệp, kích thước và trạng thái tài liệu.
20	DocumentChunk	Lớp DocumentChunk dùng để lưu các đoạn nhỏ của tài liệu, bao gồm nội dung đoạn, vị trí bắt đầu - kết thúc và vector embedding phục vụ tìm kiếm.
21	Chat	Lớp Chat dùng để lưu thông tin hội thoại hỏi – đáp, bao gồm câu hỏi, câu trả lời, embedding câu hỏi và

		các đoạn tài liệu liên quan.
22	OutboxEvent	Lớp OutboxEvent dùng để lưu các sự kiện booking, payment trong hệ thống, bao gồm loại sự kiện, dữ liệu sự kiện và trạng thái xử lý.

Bảng 21: Phân tích chi tiết các lớp

2.3.2. Thiết kế cơ sở dữ liệu hệ thống



Hình 18: Cơ sở dữ liệu hệ thống

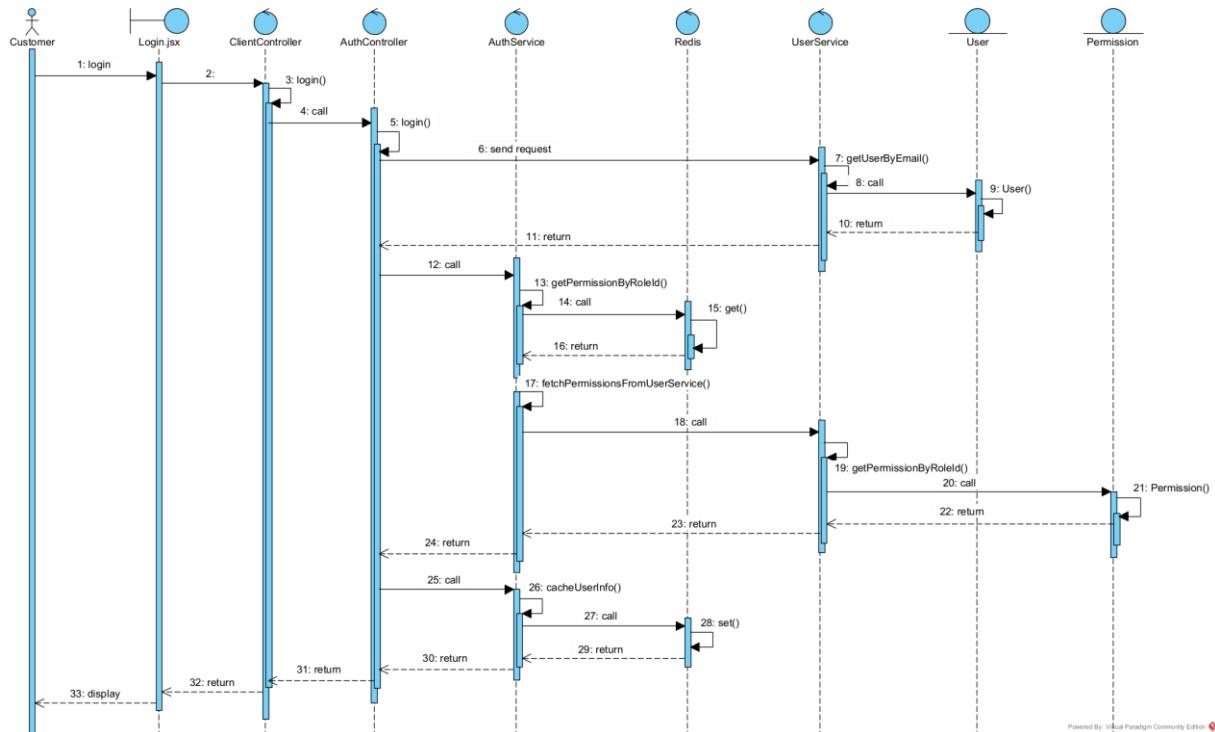
Cơ sở dữ liệu của hệ thống được phân tách theo từng service, mỗi service đảm nhiệm một nhóm chức năng nghiệp vụ riêng và quản lý độc lập các bảng dữ liệu thuộc miền nghiệp vụ của mình:

- User service: bao gồm các bảng liên quan tới người dùng và phân quyền hệ thống như tblUser, tblRole, tblPermission, tblRolePermission, để quản lý thông tin tài khoản, vai trò và quyền truy cập của người dùng.
- Movie service: bao gồm các bảng liên quan tới nghiệp vụ chính của rạp chiếu phim như tblMovie, tblGenre, tblMovieGenre, tblShowtime, tblRoom, tblSeat, để quản lý thông tin phim, thể loại, suất chiếu, phòng chiếu và ghế ngồi trong rạp.

- Booking service: Booking service bao gồm các bảng liên quan tới việc đặt vé, như tblBooking và tblTicket, để quản lý hoạt động đặt vé xem phim của khách hàng.
- Payment service: chỉ gồm bảng tblPayment, quản lý các thông tin liên quan đến hoạt động thanh toán của khách hàng.
- Notification service: chỉ gồm bảng tblNotification để quản lý các thông báo của hệ thống gửi đến người dùng.
- Chatbot service: bao gồm các bảng liên quan tới dữ liệu hội thoại và tài liệu như tblDocument, tblDocumentChunk, tblChat, để quản lý các hoạt động phục vụ cho chatbot hỗ trợ người dùng.
- Worker service: bao gồm các bảng liên quan tới xử lý nền và dữ liệu tin tức như tblNewsArticles, tblNewsSummaries, tblOutboxEvents, để quản lý tin tức về phim và các sự kiện phát sinh trong hệ thống.

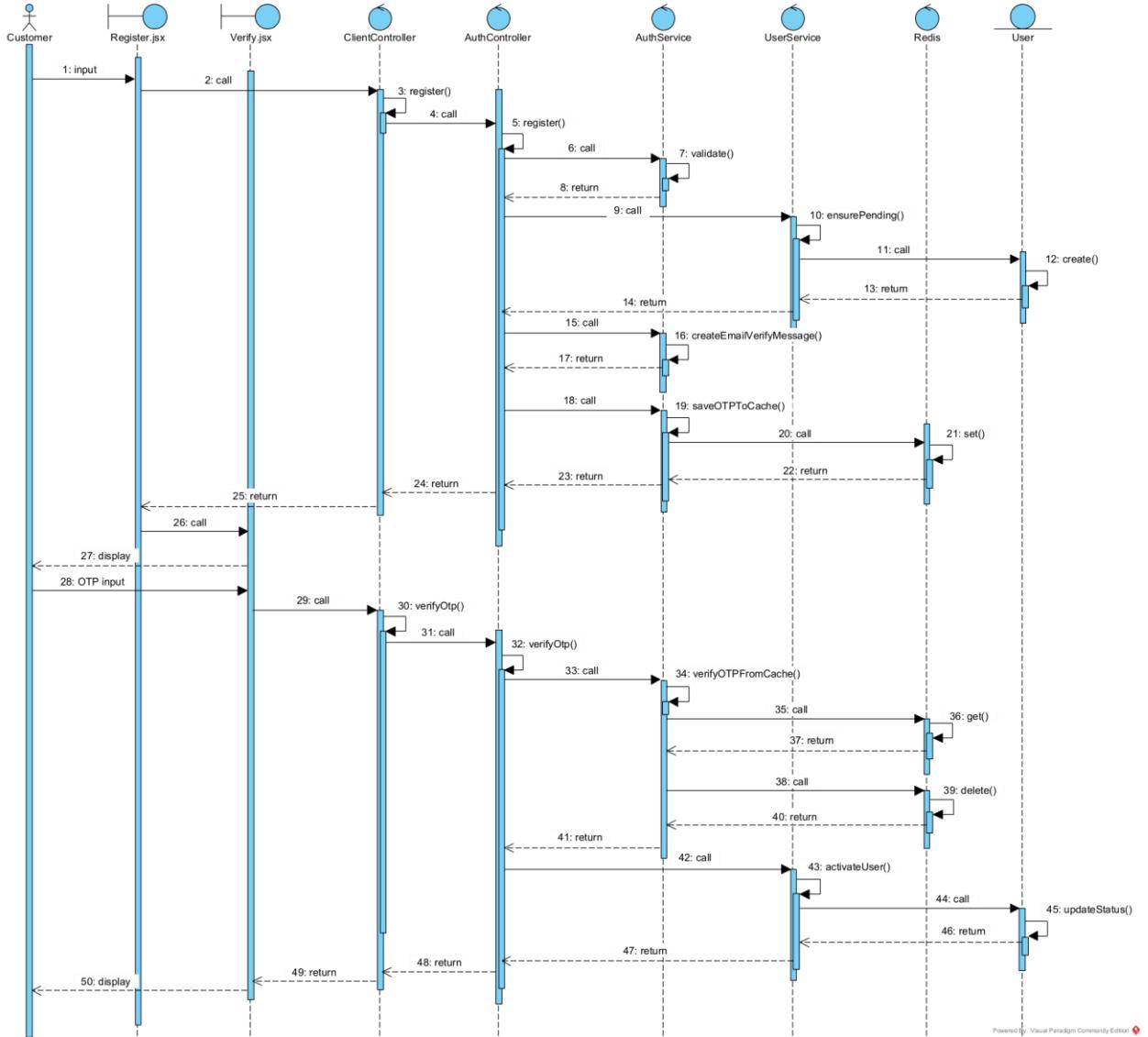
2.3.3. Thiết kế biểu đồ tuần tự cho từng chức năng

a. Đăng nhập



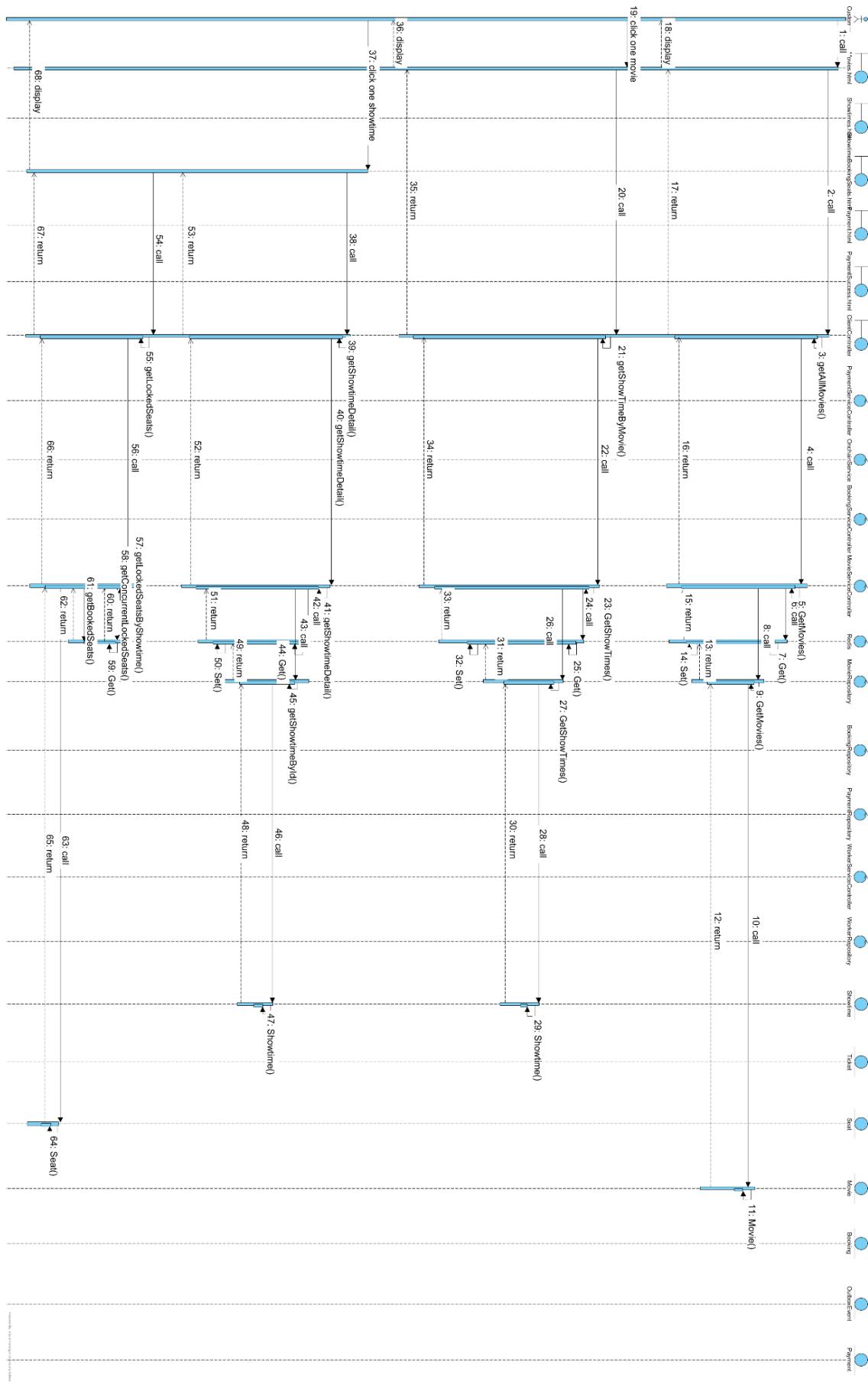
Hình 19: Sơ đồ tuần tự chức năng đăng nhập

b. Đăng ký

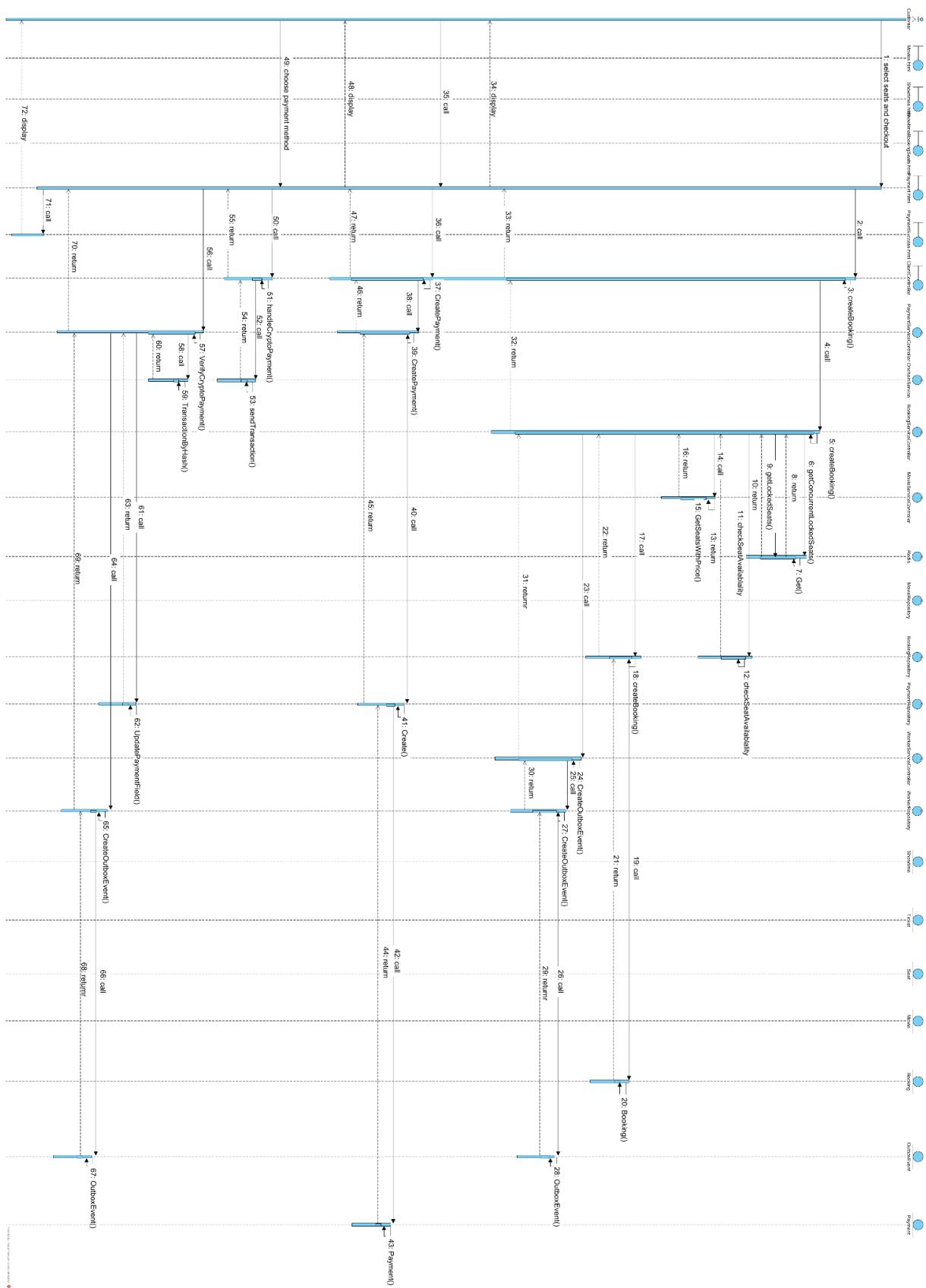


Hình 20: Sơ đồ tuần tự chức năng đăng ký

c. Đặt vé



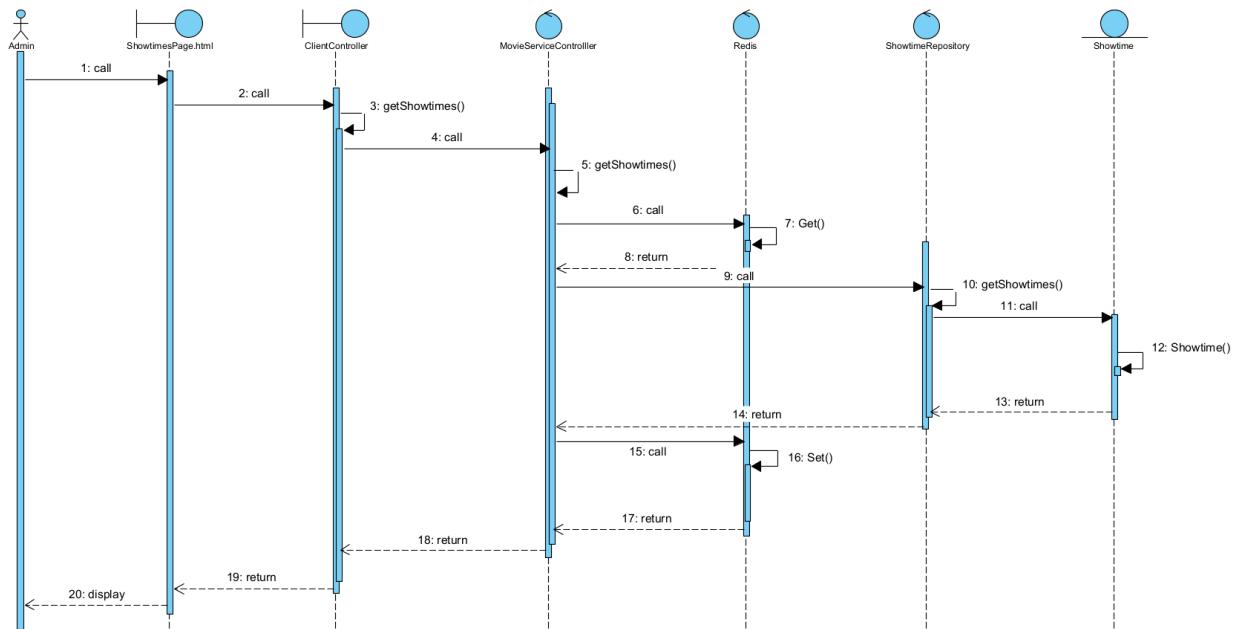
Hình 21: Sơ đồ tuần tự chức năng đặt vé (1)



Hình 22: Sơ đồ tuần tự chức năng đặt vé (2)

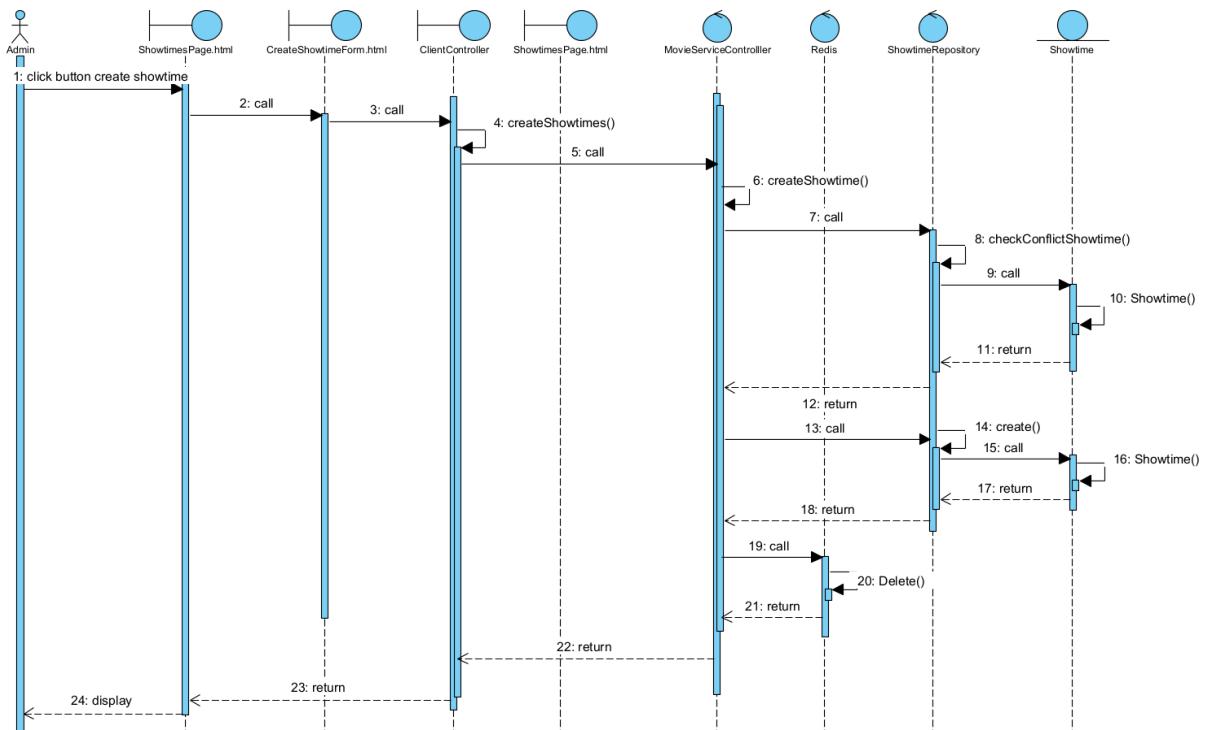
d. Quản lý suất chiếu

- Xem suất chiếu



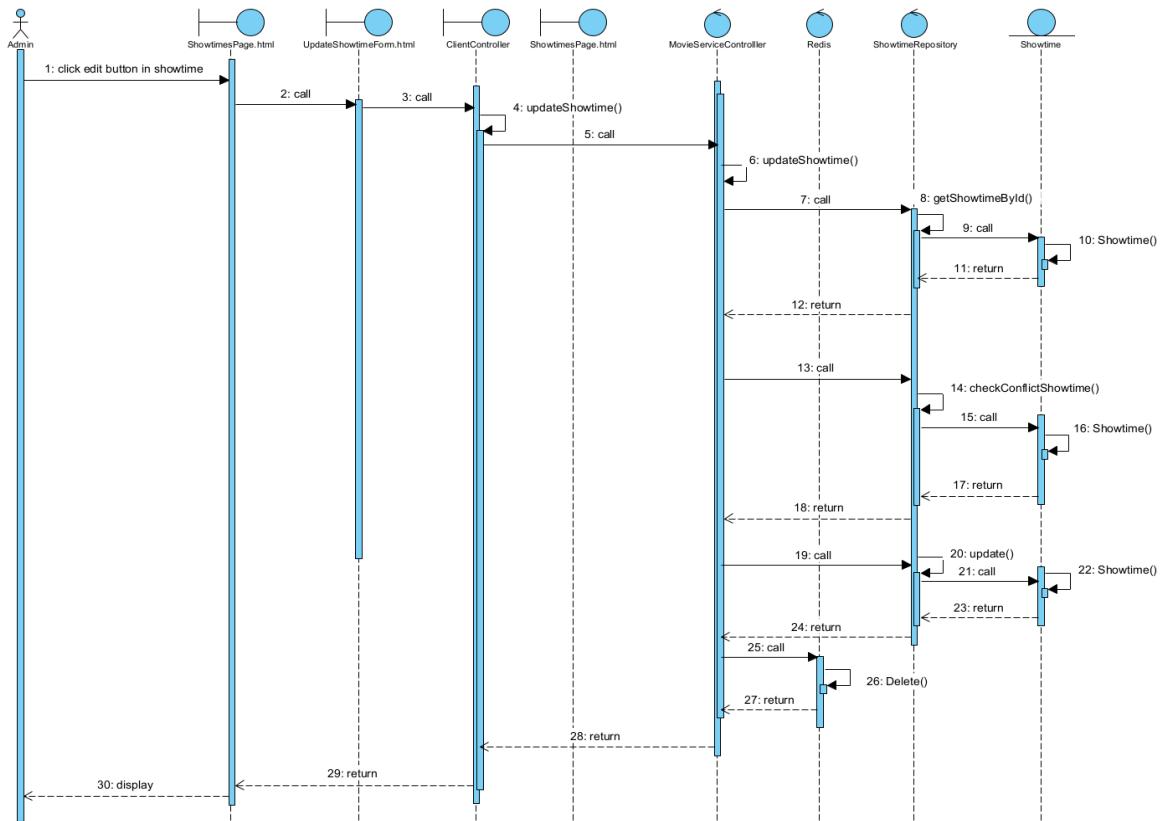
Hình 23: Sơ đồ tuần tự chức năng xem suất chiếu

- Thêm suất chiếu



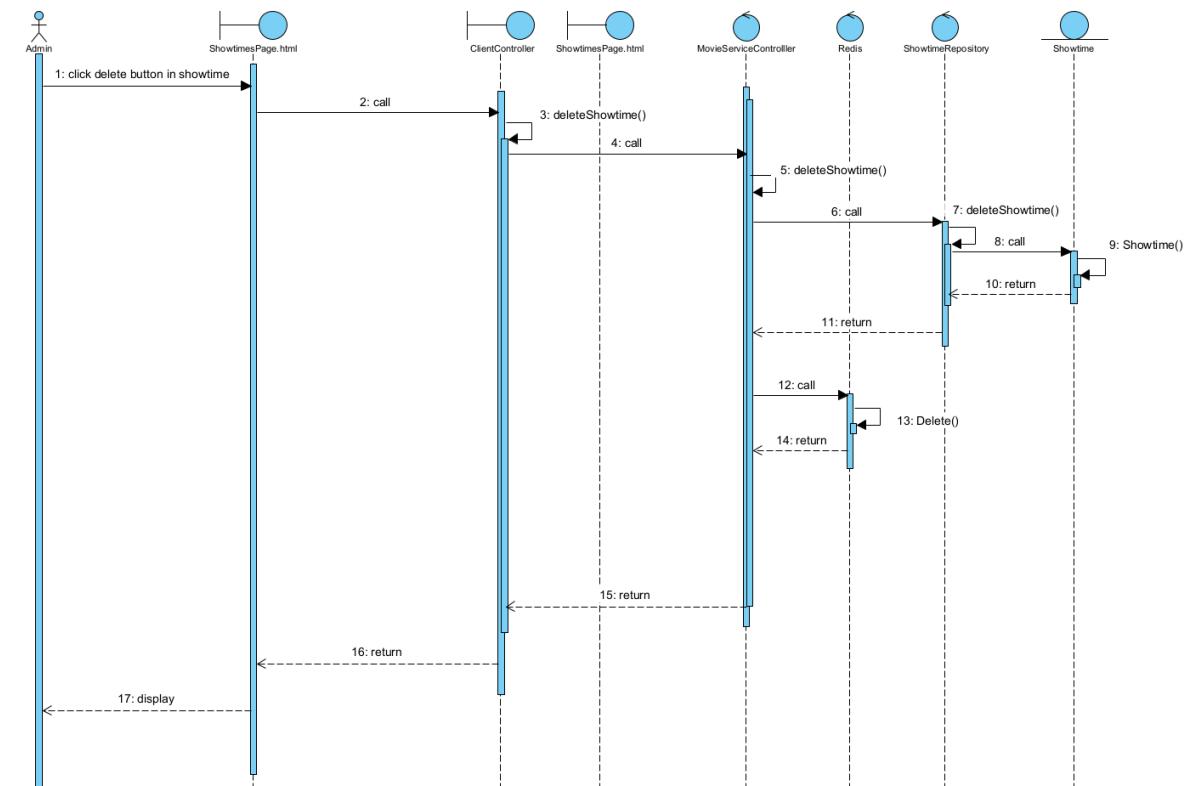
Hình 24: Sơ đồ tuần tự chức năng thêm suất chiếu

- Sửa suất chiếu



Hình 25: Sơ đồ tuần tự chức năng sửa suất chiếu

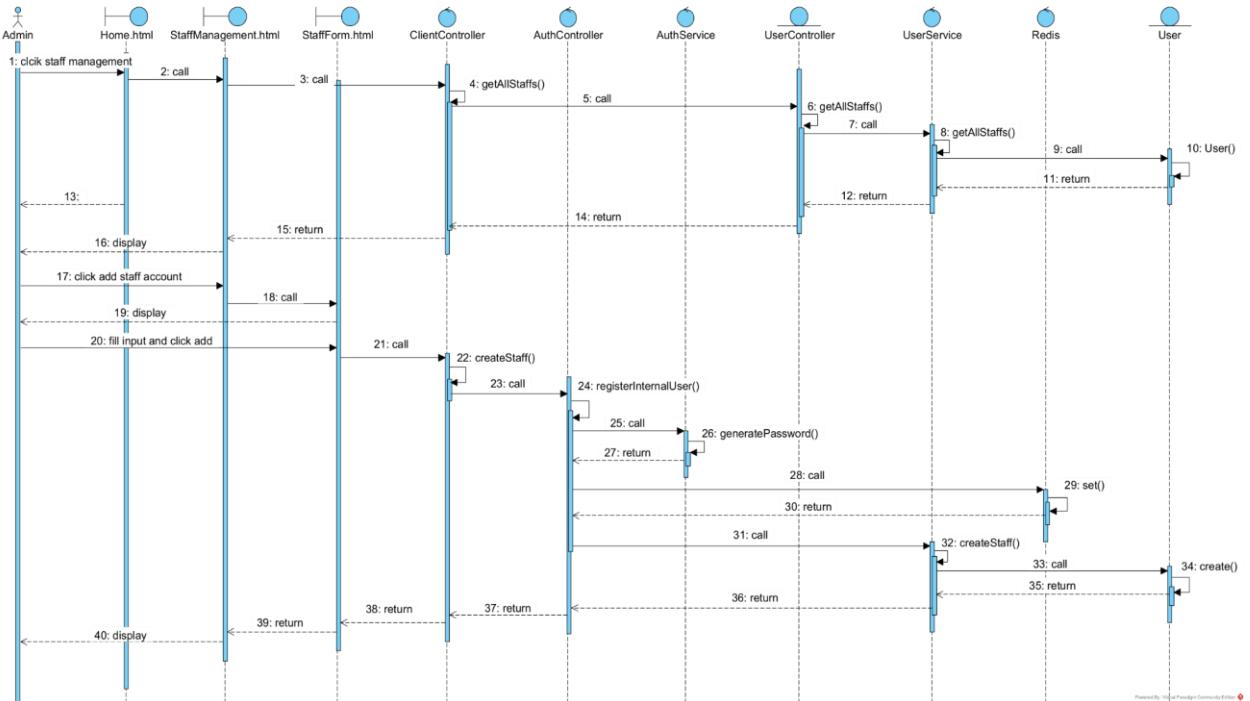
- Xoá suất chiếu



Hình 26: Sơ đồ tuần tự chức năng xoá suất chiếu

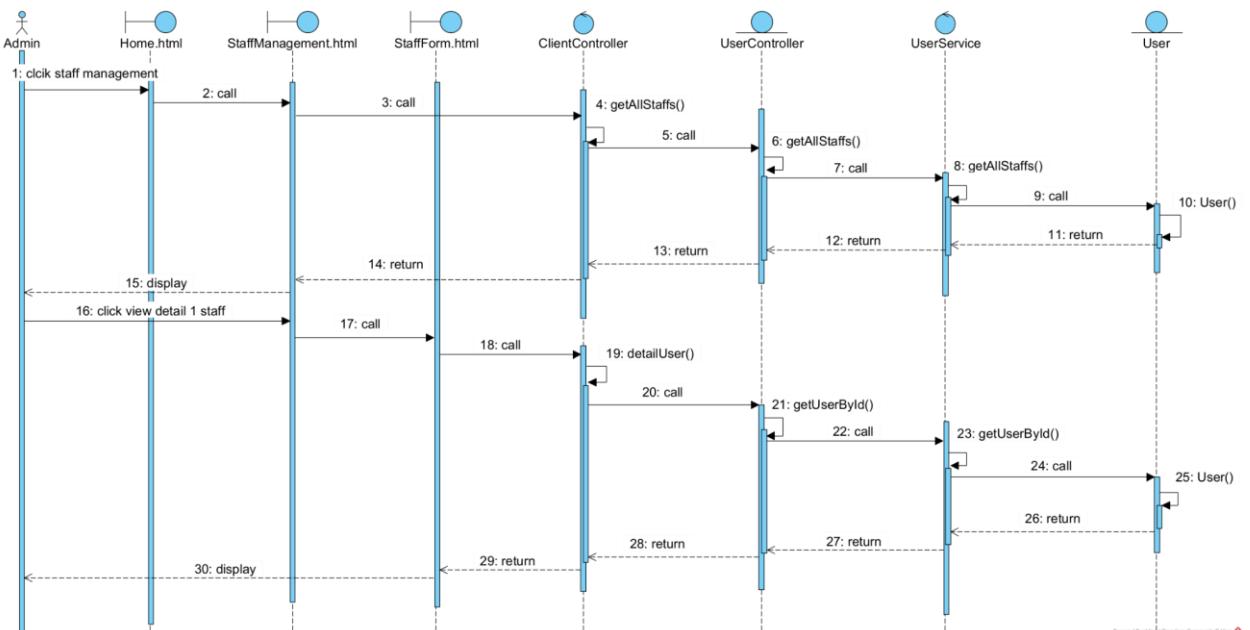
e. Quản lý nhân viên

- Tạo tài khoản mới cho nhân viên



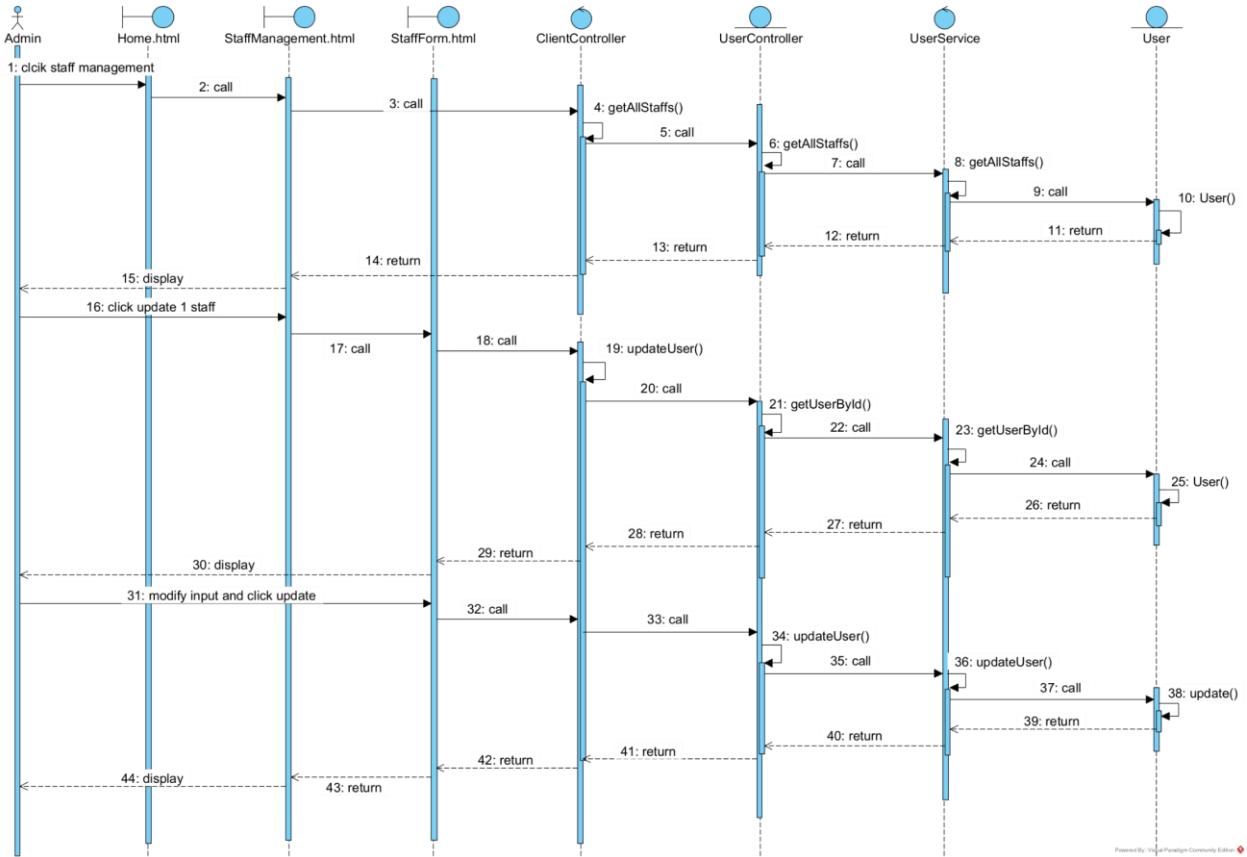
Hình 27: Sơ đồ tuần tự chức năng tạo tài khoản mới cho nhân viên

- Xem chi tiết thông tin nhân viên



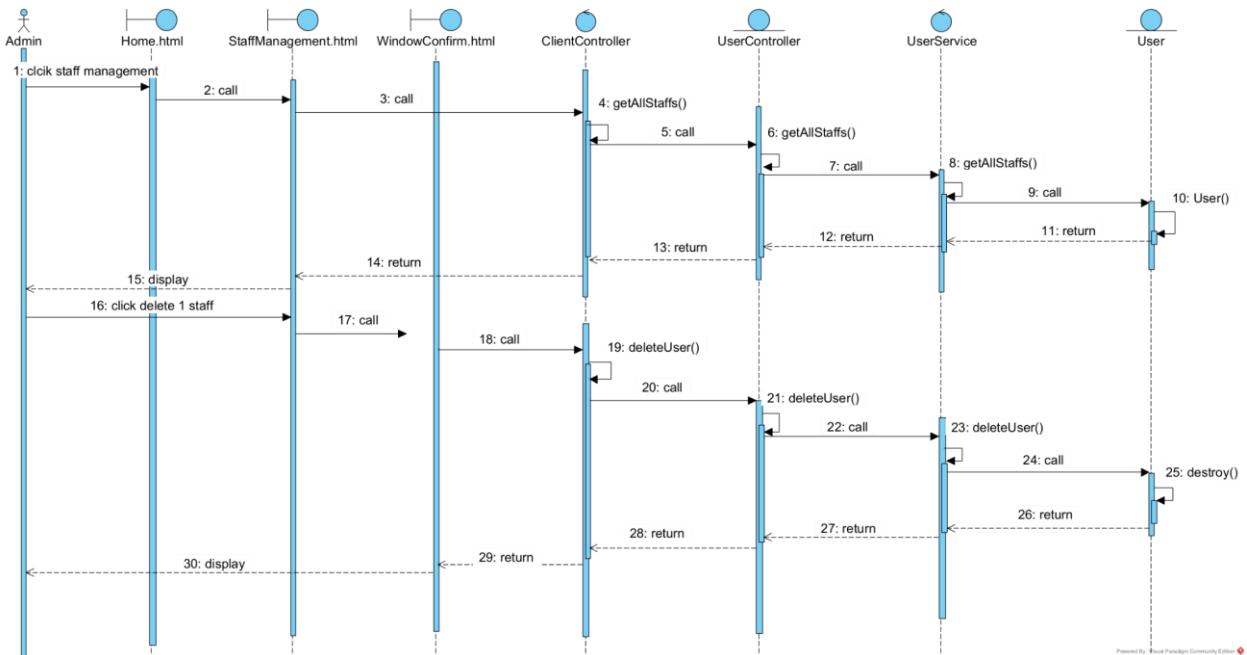
Hình 28: Sơ đồ tuần tự chức năng xem chi tiết thông tin nhân viên

- Sửa thông tin nhân viên



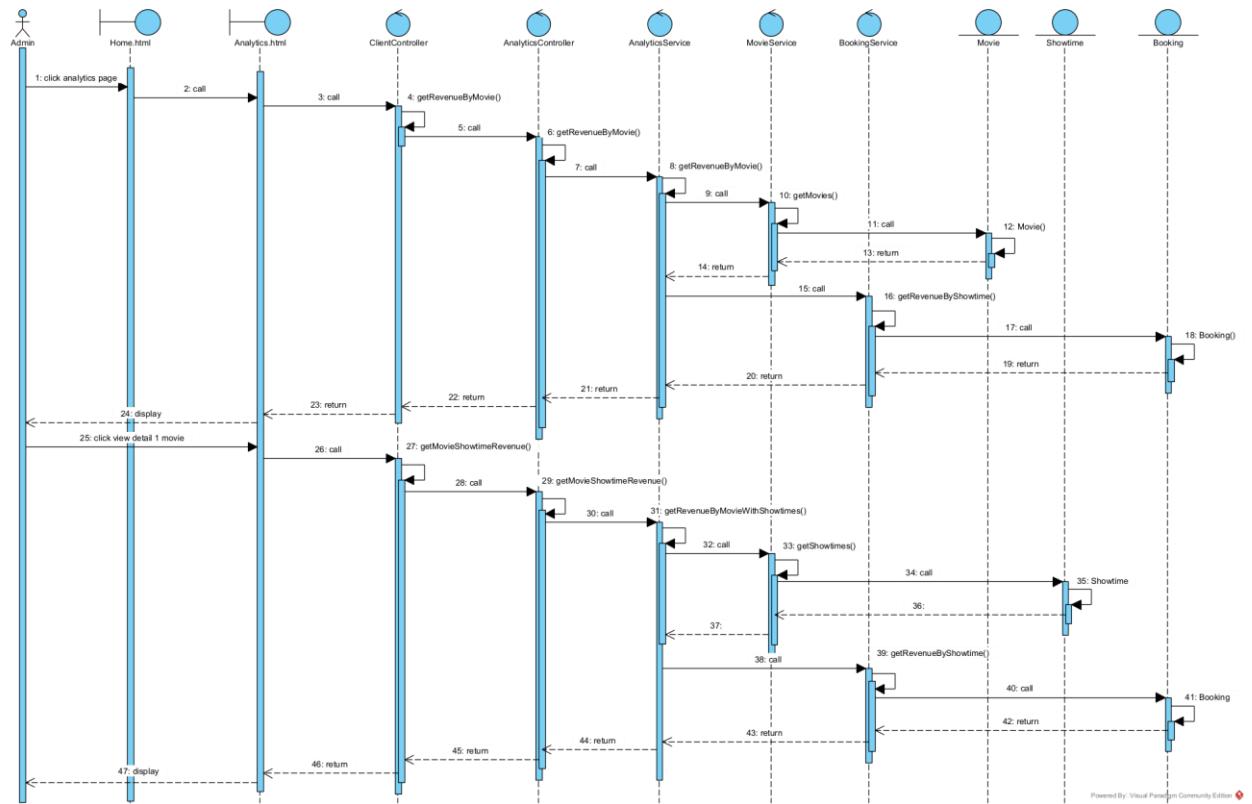
Hình 29: Sơ đồ tuần tự chức năng sửa thông tin nhân viên

- Xóa thông tin nhân viên



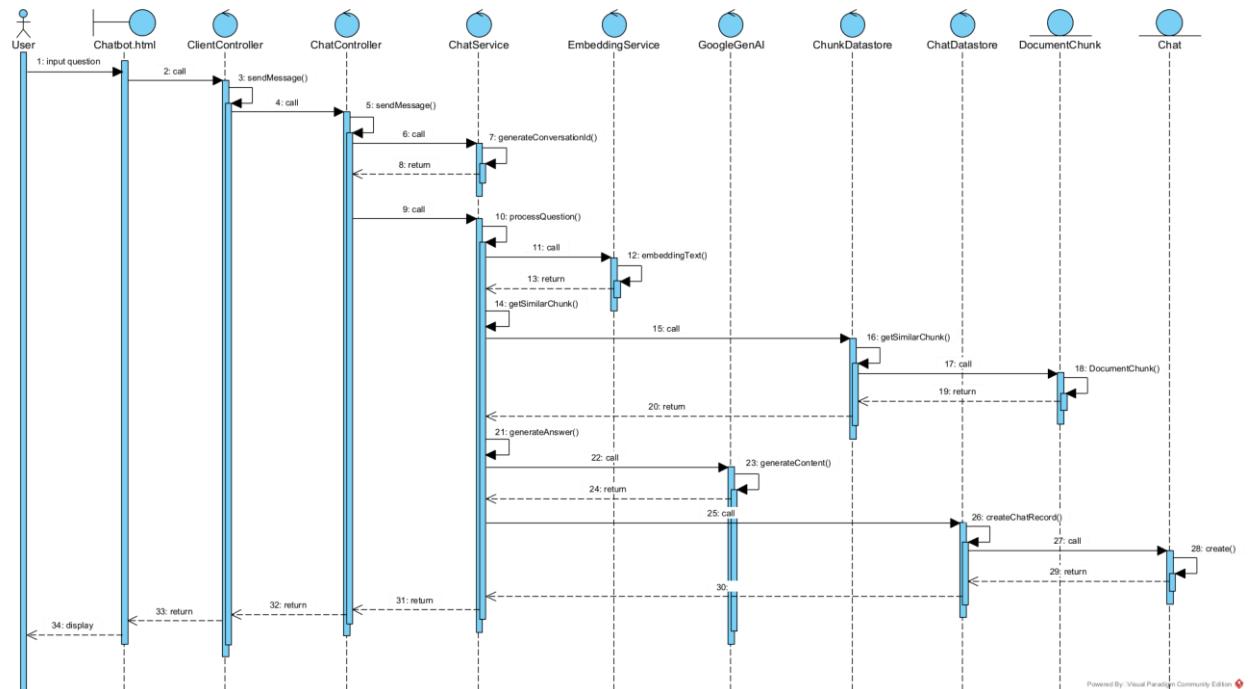
Hình 30: Sơ đồ tuần tự chức năng xóa thông tin nhân viên

f. Thống kê doanh thu theo phim



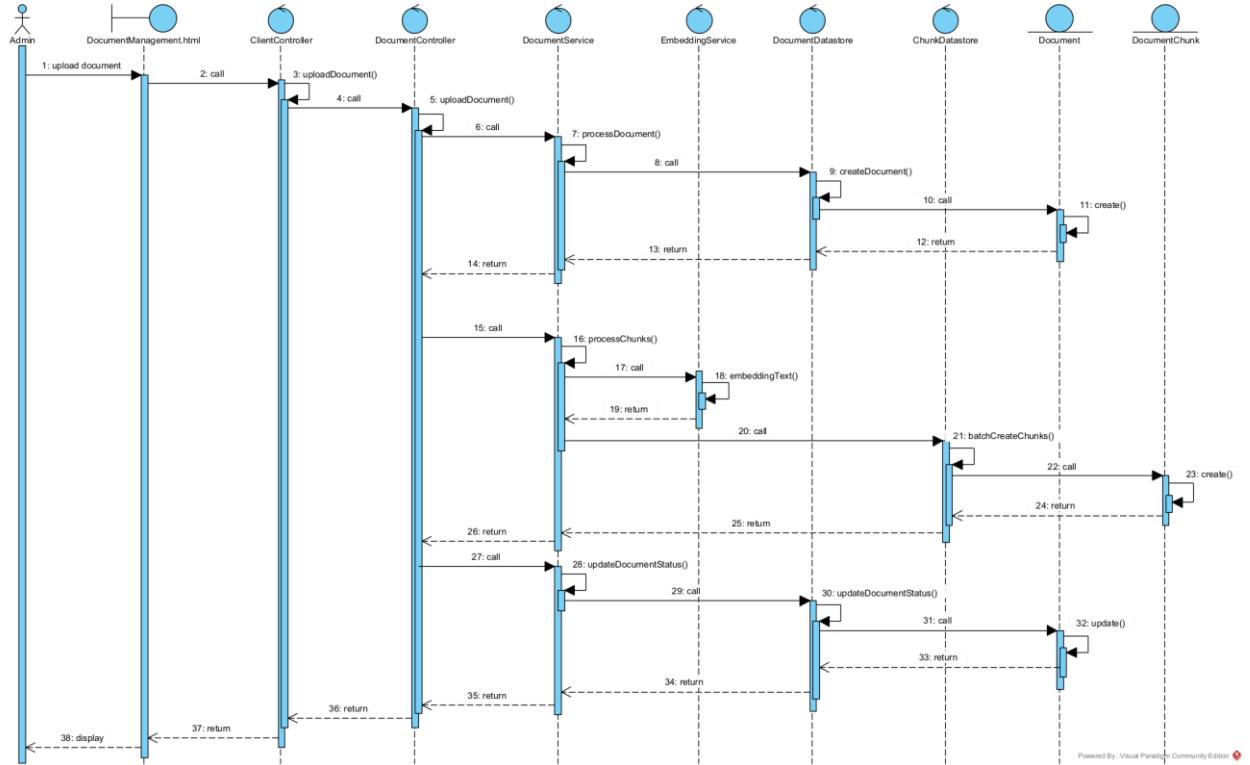
Hình 31: Sơ đồ tuần tự chức năng thống kê doanh thu theo phim

g. Chat với hệ thống



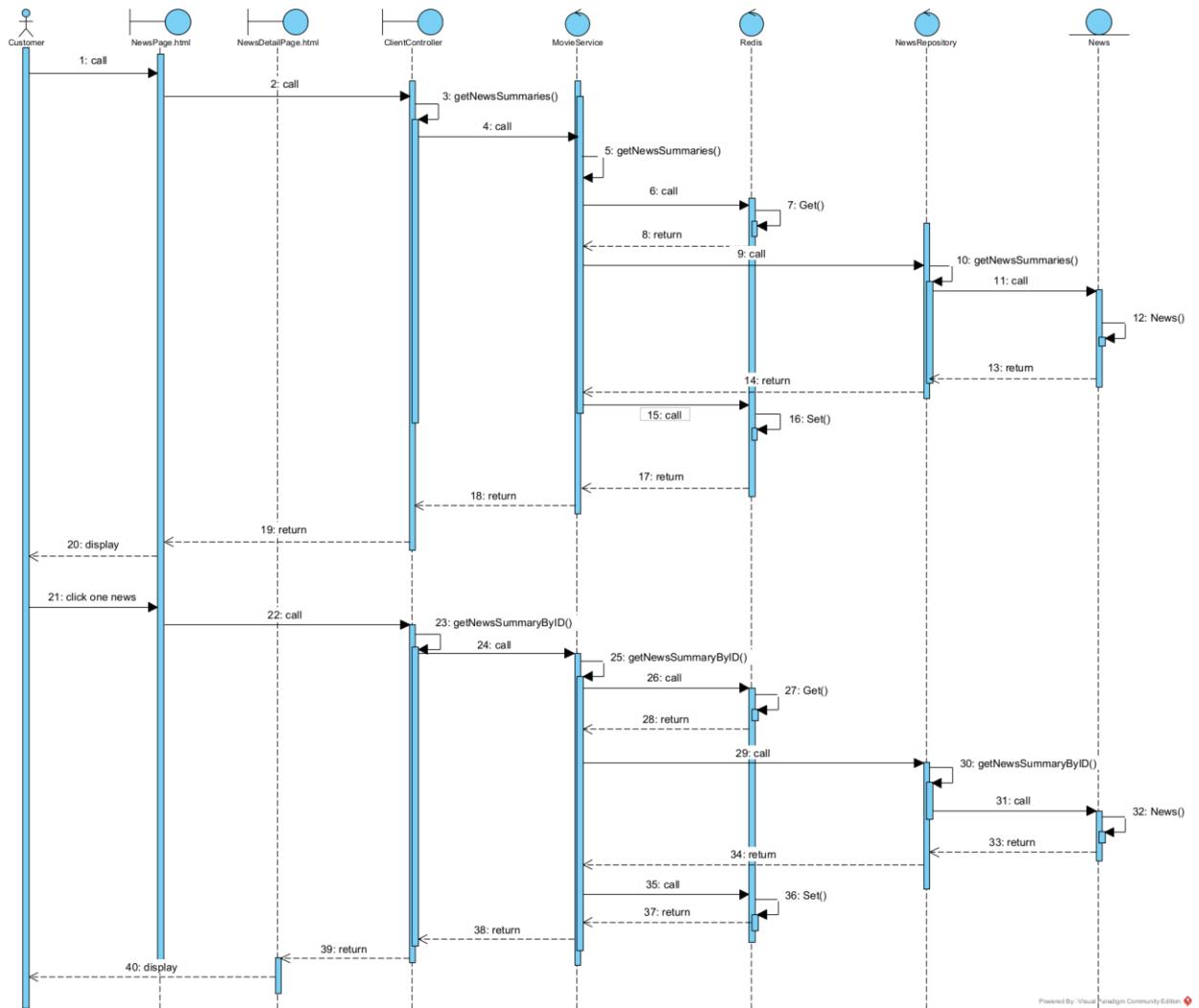
Hình 32: Sơ đồ tuần tự chức năng chat với hệ thống

h. Quản lý tài liệu RAG chatbot



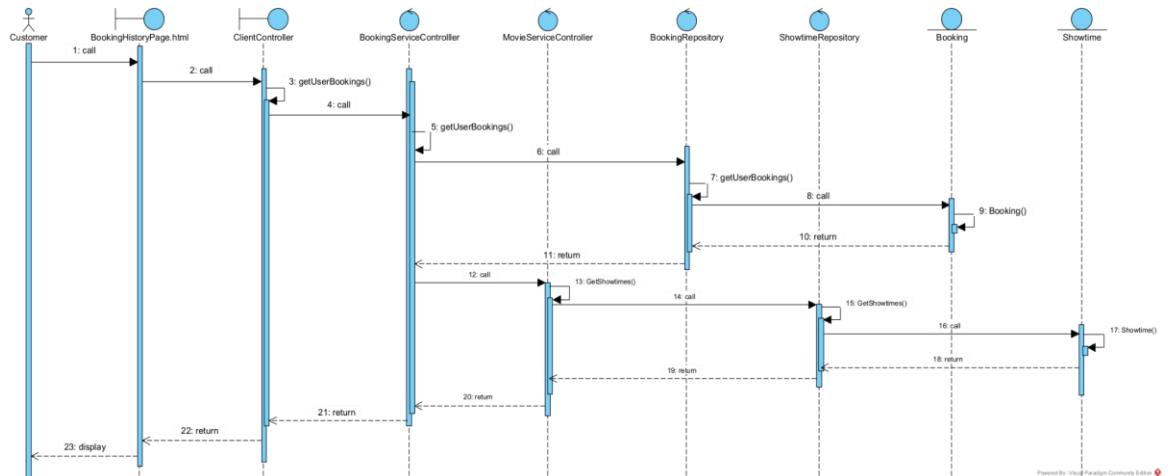
Hình 33: Sơ đồ tuần tự chức năng quản lý tài liệu RAG chatbot

i. Xem tin tức



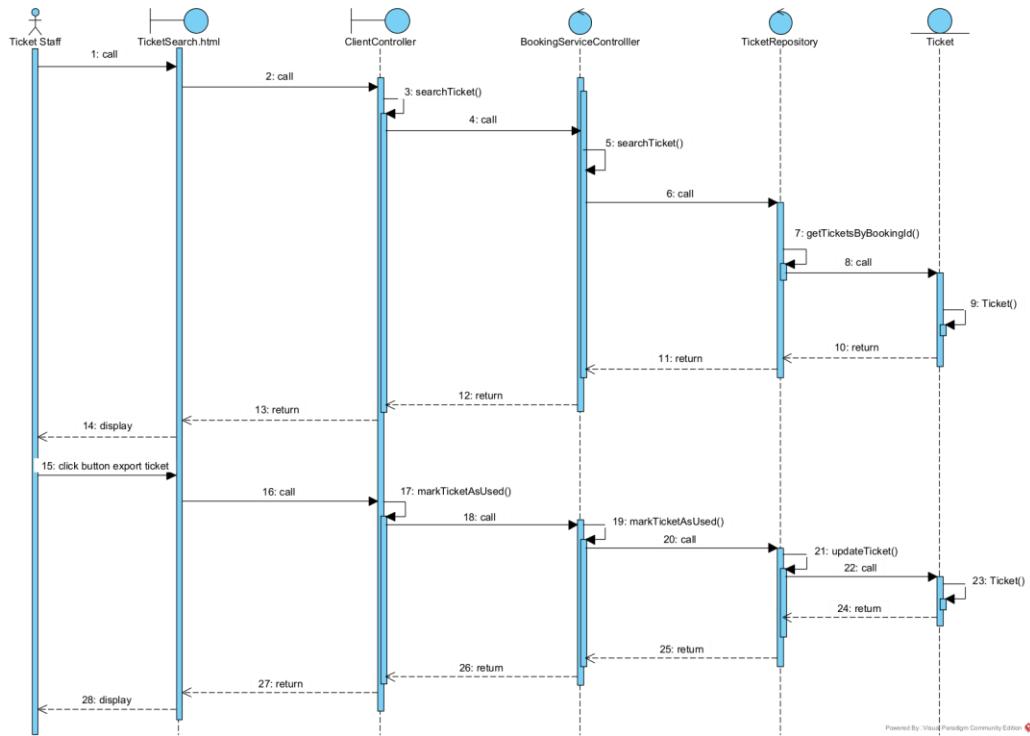
Hình 34: Sơ đồ tuần tự chức năng xem tin tức

j. Xem lịch sử đặt vé



Hình 35: Sơ đồ tuần tự chức năng xem lịch sử đặt vé

k. Xuất vé



Hình 36: Sơ đồ tuần tự chức năng xuất vé

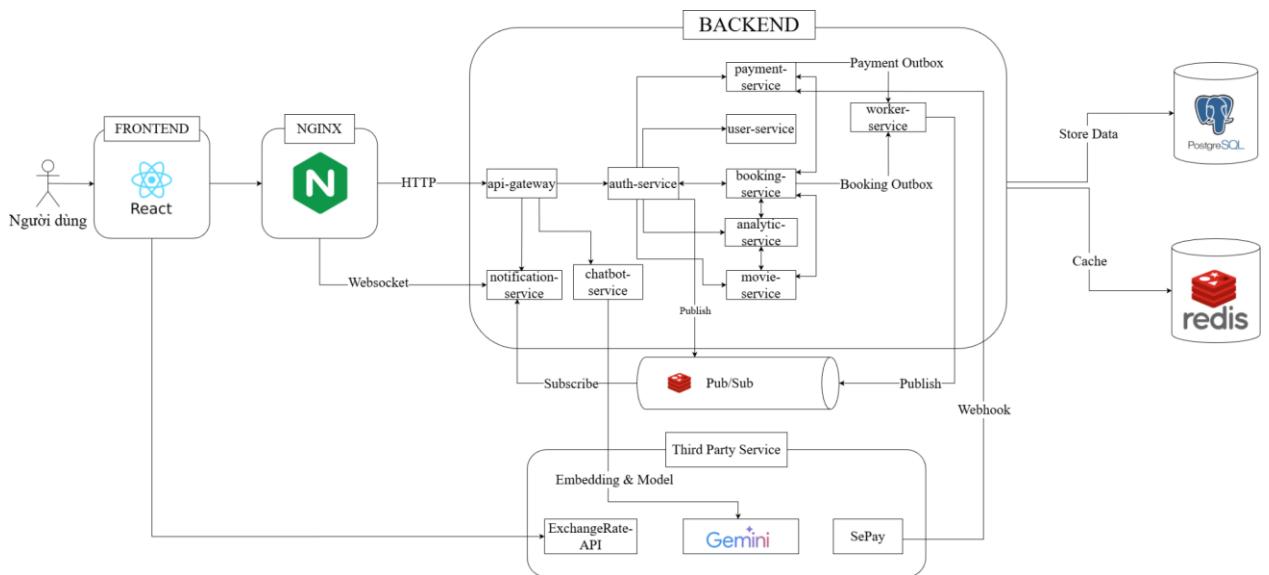
2.4. Kết chương

Chương 2 đã tập trung phân tích nghiệp vụ và xác định các yêu cầu chức năng, phi chức năng của hệ thống. Trên cơ sở đó, chương đã trình bày thiết kế tổng thể hệ thống, các thành phần chính, luồng xử lý nghiệp vụ. Những nội dung trong chương là cơ sở trực tiếp cho việc cài đặt, triển khai và đánh giá hệ thống được trình bày trong Chương 3.

CHƯƠNG 3: CÀI ĐẶT VÀ KIỂM THỬ HỆ THỐNG

3.1. Cài đặt

3.1.1. Sơ đồ kiến trúc hệ thống



Hình 37: Sơ đồ mô tả kiến trúc hệ thống

Hệ thống được thiết kế theo mô hình Client - Server, trong đó phía client đảm nhiệm việc tương tác với người dùng, còn phía server chịu trách nhiệm xử lý nghiệp vụ, quản lý dữ liệu và cung cấp các dịch vụ cho client. Mô hình này giúp tách biệt rõ ràng giữa giao diện và xử lý nghiệp vụ, từ đó nâng cao tính tổ chức và khả năng mở rộng của hệ thống.

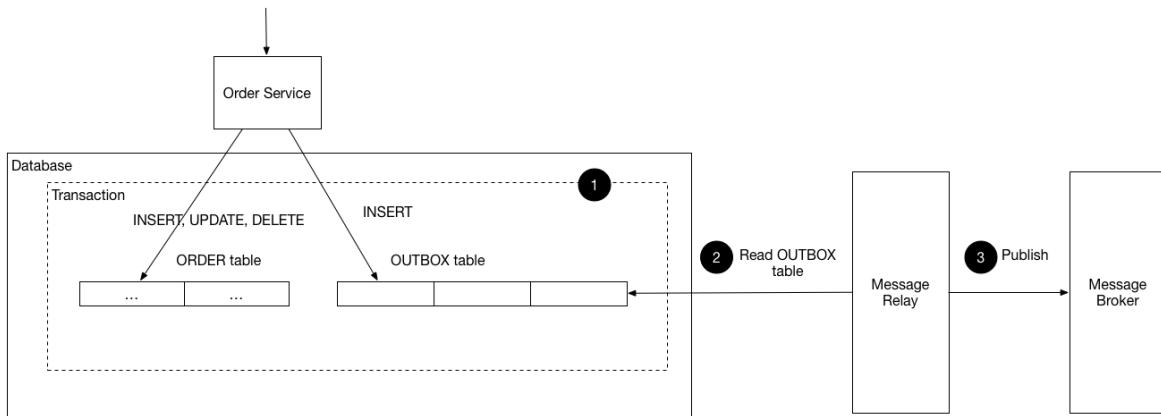
Phía Frontend, giao diện người dùng được xây dựng bằng thư viện ReactJS, đóng vai trò là điểm tương tác chính giữa người dùng và hệ thống. Frontend gửi các HTTP request đến hệ thống Backend để thực hiện các chức năng nghiệp vụ.

Phía Backend được tổ chức thành các service xử lý nghiệp vụ độc lập, trong đó mỗi service đảm nhiệm một chức năng cụ thể của hệ thống. Cách tiếp cận này giúp giảm sự phụ thuộc chéo giữa các service, đồng thời nâng cao tính linh hoạt trong quá trình phát triển, triển khai và mở rộng hệ thống. Hệ thống tích hợp chatbot AI sử dụng mô hình Gemini kết hợp với cơ chế lưu trữ vector trong PostgreSQL, cho phép thực hiện truy vấn ngữ nghĩa và hỗ trợ phản hồi thông minh dựa trên dữ liệu của hệ thống.

Bên cạnh đó, các service trong hệ thống giao tiếp với nhau thông qua gRPC cho các tác vụ đồng bộ yêu cầu hiệu năng cao, hoặc thông qua Redis Pub/Sub đối với các kịch bản giao tiếp bất đồng bộ. Tuy nhiên, đối với các tác vụ yêu cầu tính nhất quán dữ liệu tuyệt đối giữa các service như quy trình đặt vé và thanh toán, đồ án áp dụng Transactional Outbox Pattern [19].

3.1.2. Nhât quán dữ liệu với Transactional Outbox Pattern

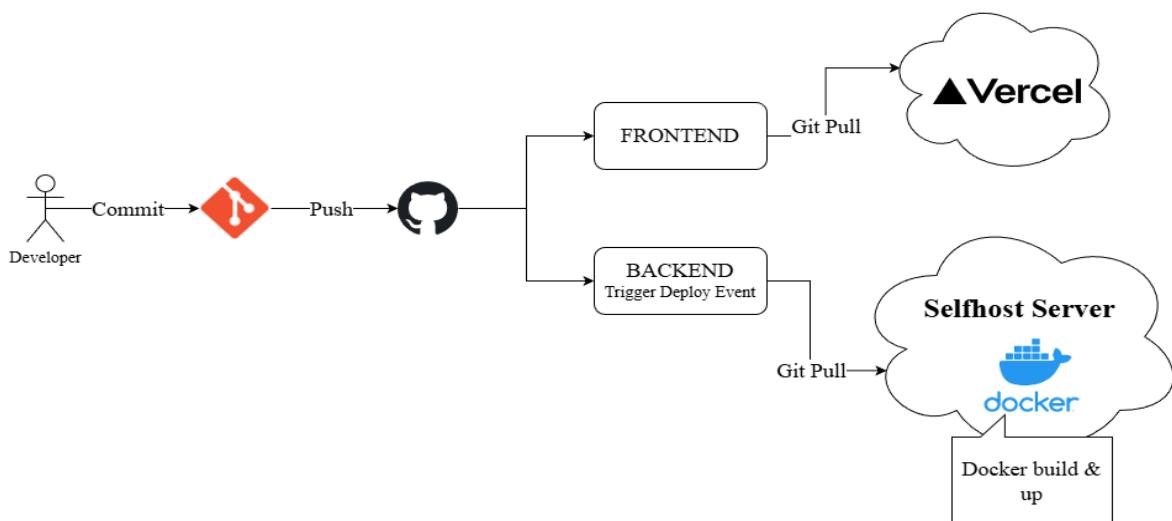
Trong kiến trúc Microservices của hệ thống, việc đảm bảo tính nhất quán dữ liệu giữa các service độc lập là một thách thức lớn. Để giải quyết vấn đề này, hệ thống áp dụng Transactional Outbox Pattern tại các service quan trọng như: booking-service và payment-service.



Hình 38: Sơ đồ triển khai Transactional Outbox

Cụ thể, mỗi khi một nghiệp vụ được thực hiện thành công, hệ thống sẽ ghi nhận sự kiện tương ứng vào bảng outbox_events trong cơ sở dữ liệu. Sau đó, worker-service có nhiệm vụ định kỳ truy xuất các bản ghi trong bảng outbox_events chưa được xử lý. Worker-service sẽ lần lượt xử lý các sự kiện này, thực hiện các hành động cần thiết như cập nhật trạng thái hệ thống hoặc kích hoạt các tiến trình bắt đồng bộ liên quan. Sau khi xử lý thành công, sự kiện sẽ được đánh dấu đã xử lý nhằm tránh việc xử lý trùng lặp.

3.1.3. Quy trình CI/CD tự động hóa



Hình 39: Quy trình CI/CD

Quy trình CI/CD của hệ thống được xây dựng dựa trên GitHub Actions [20] kết hợp với GitHub Runner [21], nhằm tự động hóa toàn bộ quá trình từ phát triển đến triển khai. Khi mã nguồn được commit và đẩy lên GitHub, pipeline sẽ được kích hoạt để xử lý riêng cho từng thành phần của hệ thống. Phía Frontend được tự động build và triển khai trên nền tảng Vercel, trong khi Backend được build image Docker và triển khai trên server self-hosted thông qua GitHub Runner. Quy trình này giúp hệ thống luôn đồng bộ với mã nguồn mới nhất, giảm thiểu can thiệp thủ công và nâng cao độ ổn định trong quá trình vận hành.

3.2. Môi trường triển khai

3.2.1. Hạ tầng phần cứng

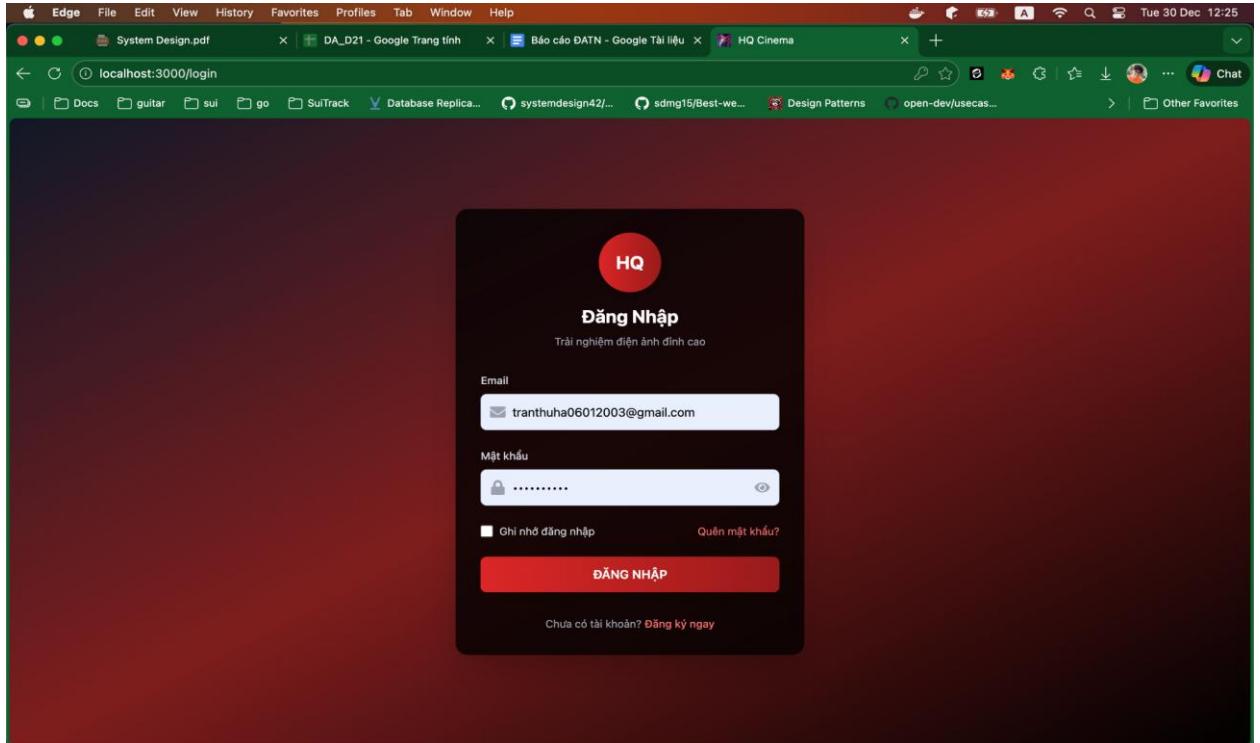
- Máy tính cá nhân: Sử dụng để lập trình, phát triển hệ thống
- Môi trường triển khai sản phẩm: Server-selfhost với hệ điều hành Linux, Vercel

3.2.2. Công cụ phát triển

- IDE: Goland, Visual Studio Code
- Hệ quản trị sở dữ liệu: PostgreSQL, Redis
- Framework & thư viện: ExpressJS, Ether.JS, Gin
- Dịch vụ bên thứ ba: Gemini API, Blockchain, SePay

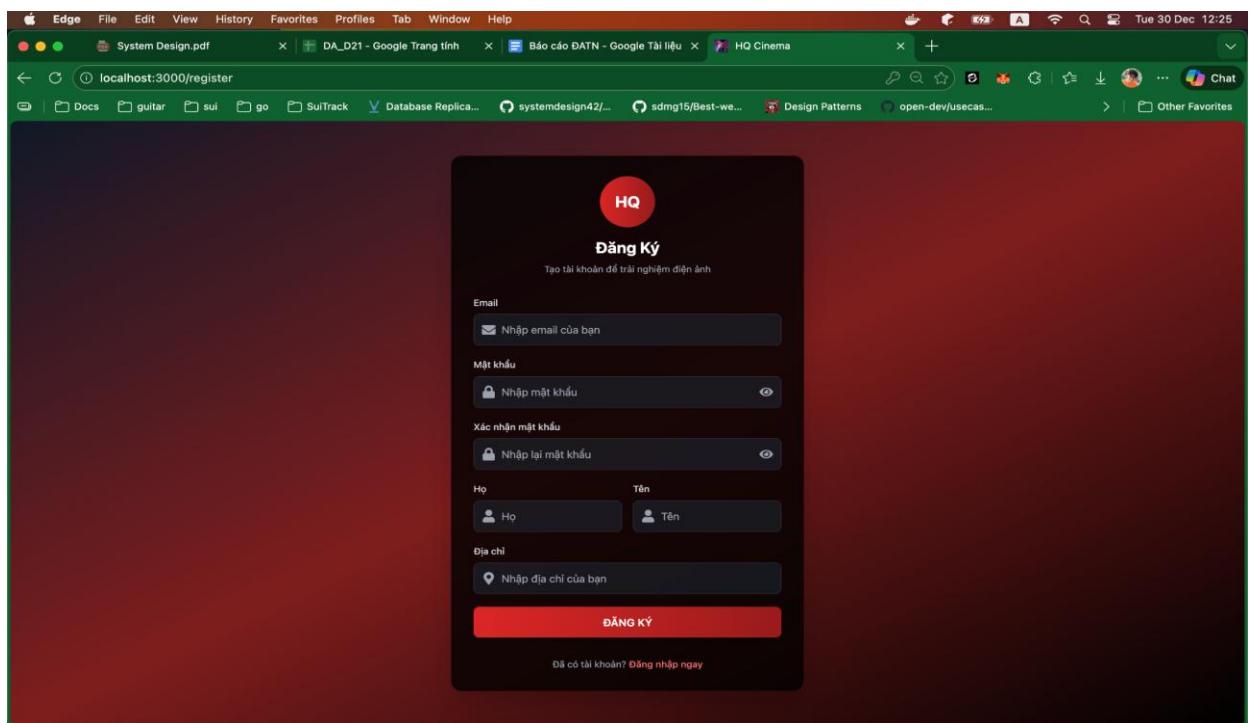
3.3. Giao diện và chức năng

3.3.1. Giao diện module đăng nhập



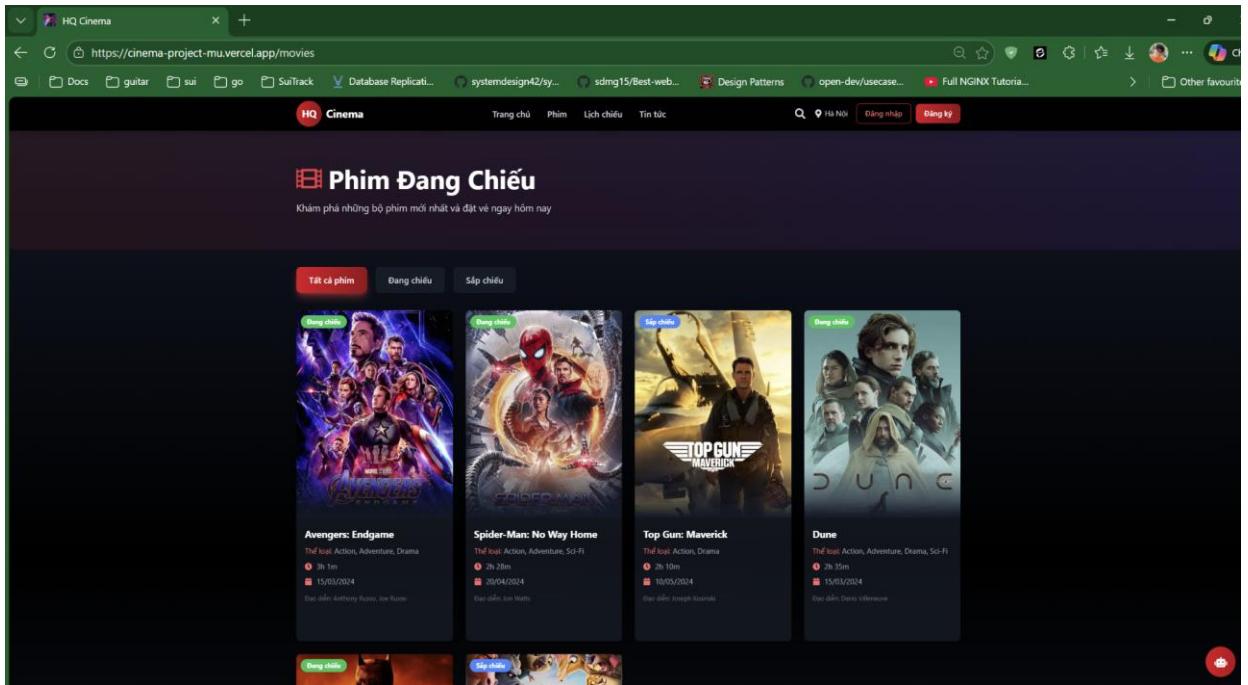
Hình 40: Giao diện module đăng nhập

3.3.2. Giao diện module đăng ký

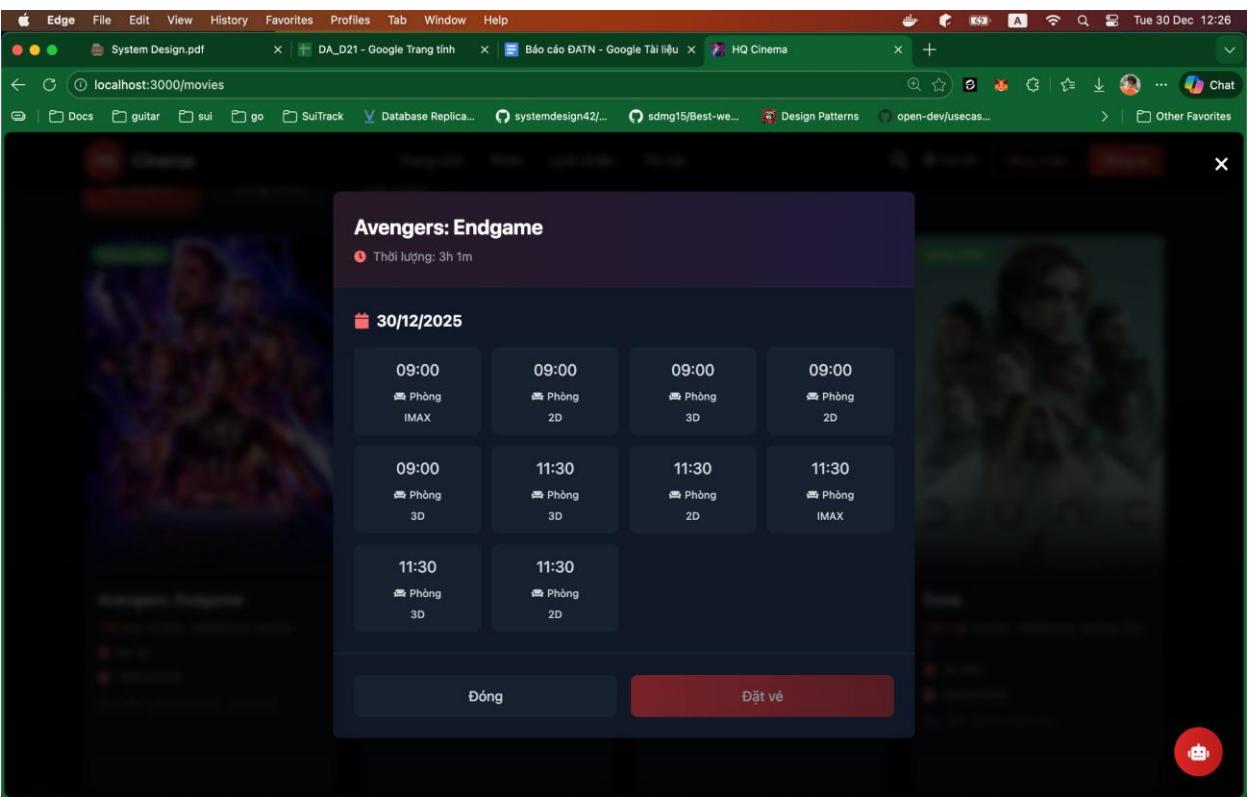


Hình 41: Giao diện module đăng ký

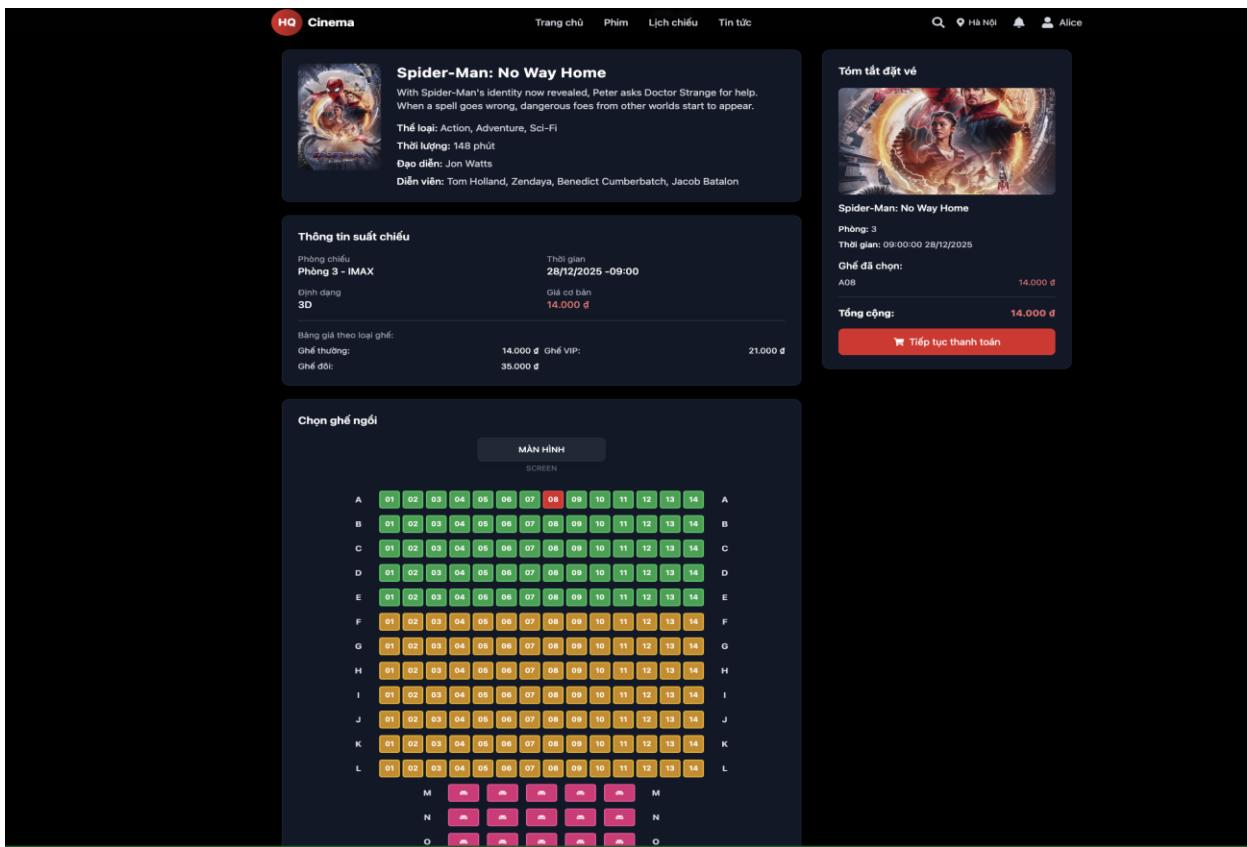
3.3.3. Giao diện module đặt vé xem phim



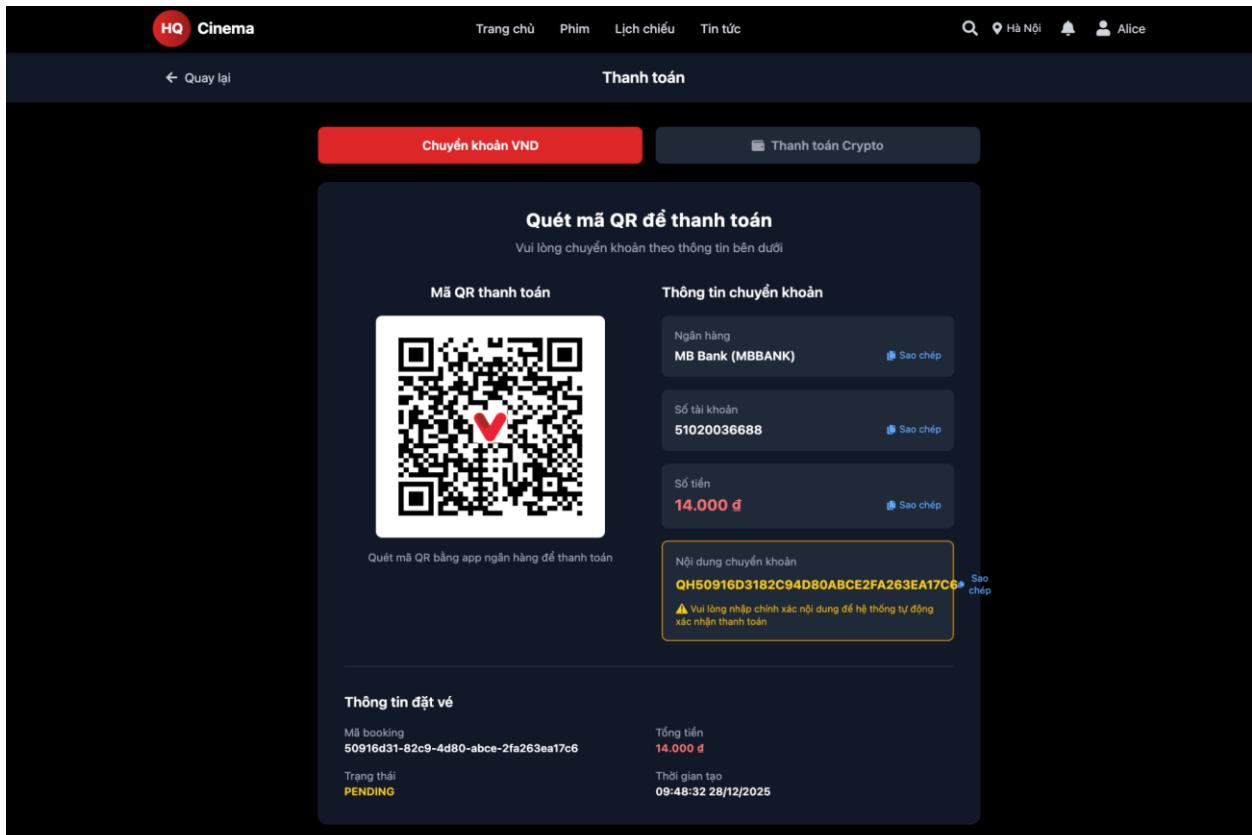
Hình 42: Giao diện module đặt vé xem phim (1)



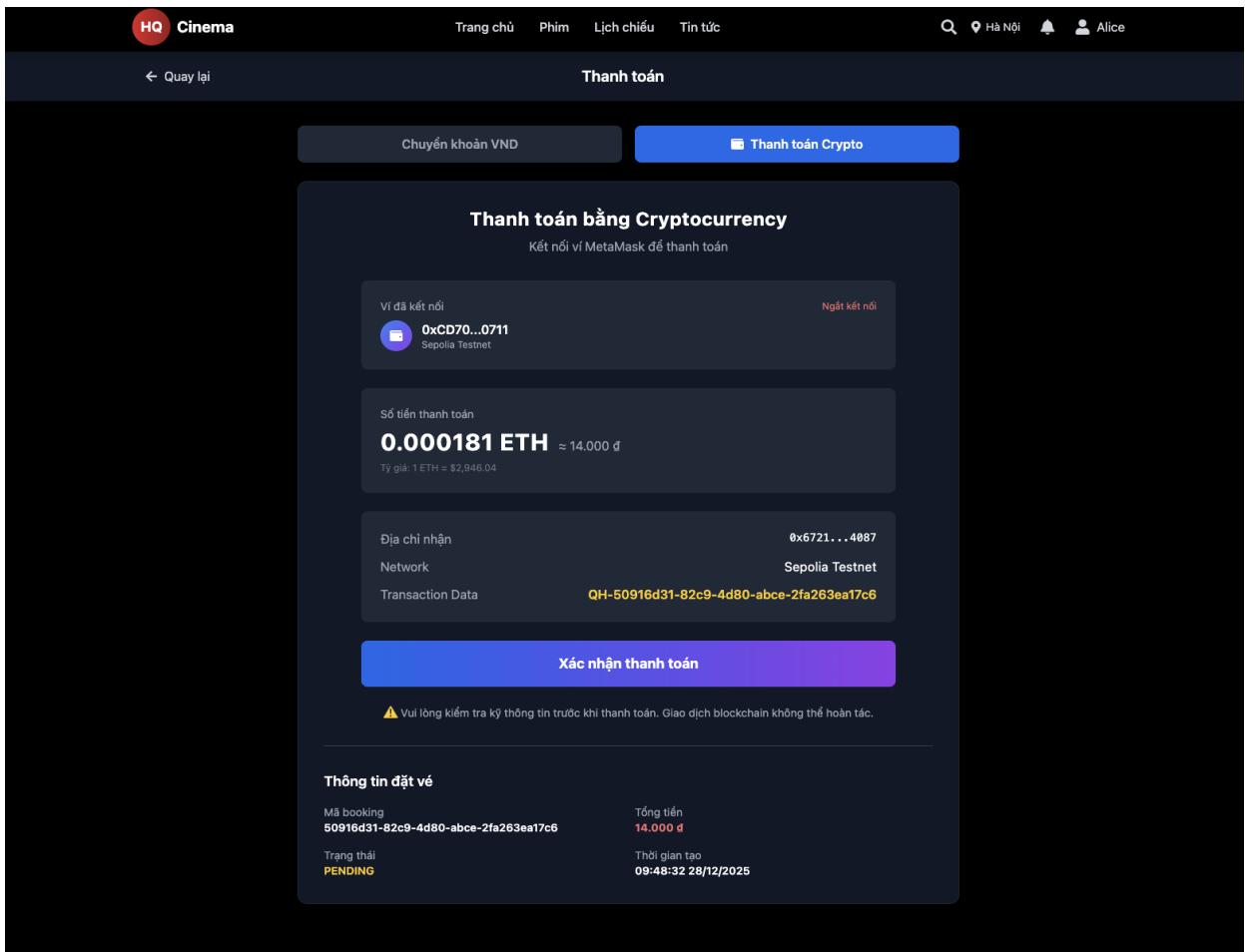
Hình 43: Giao diện module đặt vé xem phim (2)



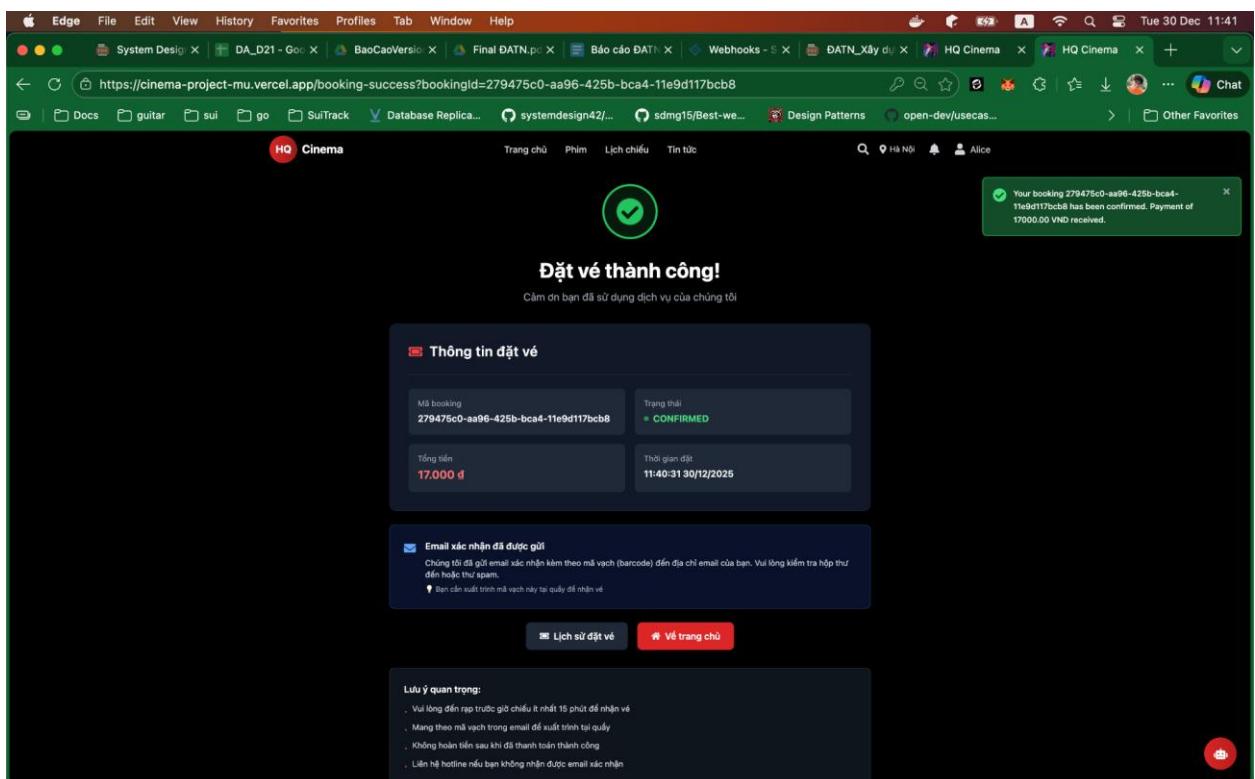
Hình 44: Giao diện module đặt vé xem phim (3)



Hình 45: Giao diện module đặt vé xem phim (4)



Hình 46: Giao diện module đặt vé xem phim (5)



Hình 47: Giao diện module đặt vé xem phim (6)

3.3.4. Giao diện module quản lý suất chiếu

Quản lý Lịch chiếu

+ Thêm lịch chiếu mới

Tất cả phim

Tất cả phòng

Tất cả định dạng

Tất cả trạng thái

dd/mm/yyyy

dd/mm/yyyy

Xóa bộ lọc

PHIM	PHÒNG	THỜI GIAN	ĐỊNH DẠNG	GIÁ VÉ	TRANG THÁI	THAO TÁC
Spider-Man: No Way Home	Phòng 4	09:00 27/12/2025 Kết thúc: 11:58 27/12/2025 Thời lượng: 2h58m0s	2D	15.000 VNĐ	Hoàn thành	
Avengers: Endgame	Phòng 5	09:00 27/12/2025 Kết thúc: 12:31 27/12/2025 Thời lượng: 3h31m0s	2D	8.000 VNĐ	Hoàn thành	
Spider-Man: No Way Home	Phòng 2	09:00 27/12/2025 Kết thúc: 11:58 27/12/2025 Thời lượng: 2h58m0s	2D	8.000 VNĐ	Hoàn thành	
Spider-Man: No Way Home	Phòng 3	09:00 27/12/2025 Kết thúc: 11:58 27/12/2025 Thời lượng: 2h58m0s	2D	12.000 VNĐ	Hoàn thành	
Avengers: Endgame	Phòng 3	09:00 27/12/2025 Kết thúc: 12:31 27/12/2025 Thời lượng: 3h31m0s	IMAX	17.000 VNĐ	Hoàn thành	
Avengers: Endgame	Phòng 2	09:00 27/12/2025 Kết thúc: 12:31 27/12/2025 Thời lượng: 3h31m0s	3D	10.000 VNĐ	Hoàn thành	
Avengers: Endgame	Phòng 1	09:00 27/12/2025 Kết thúc: 12:31 27/12/2025 Thời lượng: 3h31m0s	2D	8.000 VNĐ	Hoàn thành	
Spider-Man: No Way Home	Phòng 1	09:00 27/12/2025 Kết thúc: 11:58 27/12/2025 Thời lượng: 2h58m0s	3D	10.000 VNĐ	Hoàn thành	
Avengers: Endgame	Phòng 4	09:00 27/12/2025 Kết thúc: 12:31 27/12/2025 Thời lượng: 3h31m0s	3D	17.000 VNĐ	Hoàn thành	
Spider-Man: No Way Home	Phòng 5	09:00 27/12/2025 Kết thúc: 11:58 27/12/2025 Thời lượng: 2h58m0s	3D	10.000 VNĐ	Hoàn thành	

← Trước 1 2 3 ... 20 Sau →

Hình 48: Giao diện module quản lý suất chiếu (1)

Tạo lịch chiếu mới

Tạo lịch chiếu mới cho hệ thống rạp

Thông tin thời gian

Thời lượng suất chiếu: 4h 1m
Thời lượng phim: 3h 1m
Thời gian phòng: 1h 0m (bao gồm đón dẹp, quảng cáo)

Chọn phim *

Avengers: Endgame (3h 1m)

Phòng chiếu *

Phòng 1 (STANDARD)

Thời gian bắt đầu *

12/30/2025 02:15 PM

Thời gian kết thúc *

12/30/2025 06:16 PM

Định dạng chiếu *

IMAX

Giá vé cơ bản (VNĐ) *

80000

Hủy thao tác

Tạo lịch chiếu mới

Hình 49: Giao diện module quản lý suất chiếu (2)

3.3.5. Giao diện module quản lý nhân viên

Hình 50: Giao diện module quản lý nhân viên (1)

Hình 51: Giao diện module quản lý nhân viên (2)

← Thêm nhân viên mới

Điền thông tin để tạo tài khoản nhân viên mới

Họ và tên *	Email *
Quang Nguyễn	quangnguyennngoc3dsfgdg14@gmail.com
Vai trò *	Số điện thoại
Nhân viên bán vé	0376948162
Giới tính	Ngày sinh
Nam	12/10/2003
Địa chỉ	
Ngo Xa - Phat Tich - Tien Du - Bac Ninh	
Xem trước: Quang Nguyễn - quangnguyennngoc3dsfgdg14@gmail.com Vai trò: Nhân viên bán vé	
<input type="button" value="Hủy"/> <input type="button" value="Tạo mới"/>	

Hình 52: Giao diện module quản lý nhân viên (3)

3.3.6. Giao diện module xem tin tức

The screenshot shows the 'Cinema' news module interface. At the top, there is a navigation bar with links to 'Trang chủ', 'Phim', 'Lịch chiếu', 'Tin tức', and user information like 'Alice'. Below the navigation bar, the main title is 'Tin Tức Điện Ảnh' (News). A sub-header below it reads 'Cập nhật tin tức mới nhất về thế giới điện ảnh trong và ngoài nước' (Latest news about the world of cinema both at home and abroad). There are three main news cards displayed:

- Tất cả** (All) button
- Trong nước** (Domestic) button
- Quốc tế** (International) button

News Card 1: Ngoc Nga's beauty and youth. Published on December 28, 2025. Headline: 'Về quyền rũ của Ngọc Nga 'Chân dài' U.50'. Description: 'Bài báo tập trung vào vẻ đẹp và sức hút không tuổi của nữ diễn viên Ngọc Nga, người từng gây ấn tượng với vai diễn trong phim "Những cô gái chân dài". Nội dung sẽ...'.

News Card 2: Thanh Hằng's double earring. Published on December 28, 2025. Headline: 'Thanh Hằng: Cột mốc đổi đời ở sân khấu cải lương'. Description: 'Dựa trên tiêu đề "Thanh Hằng kể cột mốc "đổi đời" ở sân khấu cải lương", đoạn tóm tắt sẽ là: Bài báo tập trung vào câu chuyện của nghệ sĩ Thanh Hằng, nói về...'.

News Card 3: Thúy Muối's acting career. Published on December 28, 2025. Headline: 'Thúy Muối: Chuyên vai phụ, lạc quan tuổi ngoài 50'. Description: 'Bài báo này mang đến cái nhìn sâu sắc về nữ diễn viên Thúy Muối, đặc biệt là hành trình sự nghiệp chuyên trị các vai phụ của cô. Dù đã bước sang tuổi ngoài 50, cô...'.

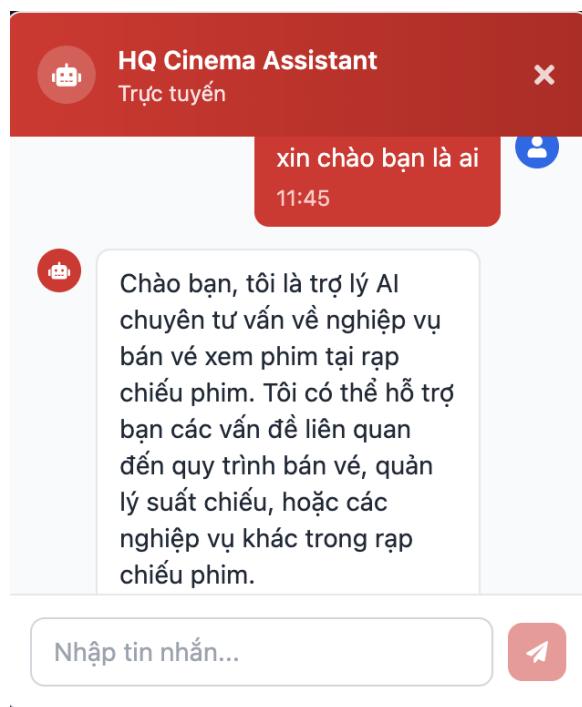
News Card 4: Soobin's motivation. Published on December 28, 2025. Headline: 'Soobin xúc động vì học trò "lội ngược dòng"'. Description: 'Dựa trên tiêu đề "Soobin xúc động vì học trò "lội ngược dòng"', đoạn tóm tắt sẽ là: Bài báo tập trung vào cảm xúc của nghệ sĩ Soobin khi chứng kiến...'.

Hình 53: Giao diện module xem tin tức (1)



Hình 54: Giao diện module xem tin tức (2)

3.3.7. Giao diện module Chatbot



Hình 55: Giao diện module Chatbot

3.3.8. Giao diện module xem lịch sử đặt vé

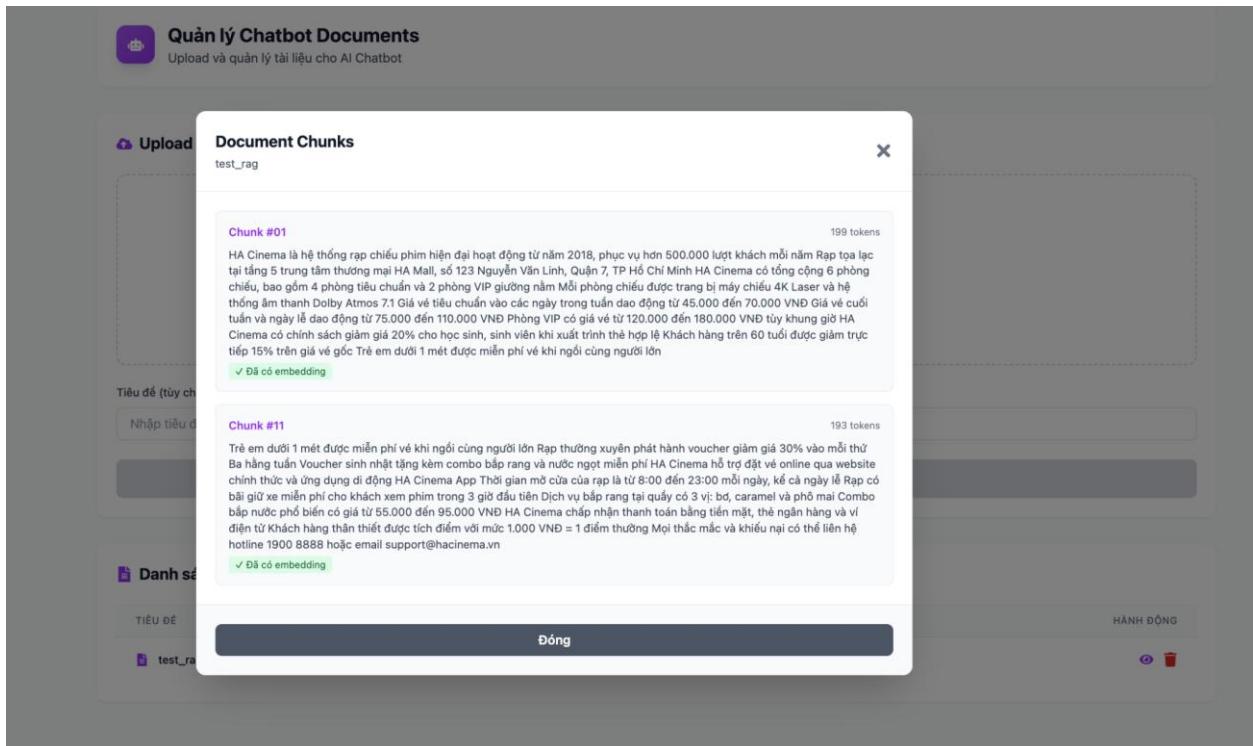
Ngày đặt	Tổng tiền	Ghế ngồi	Movies	Trạng thái
30/12/2025	17.000 ₫	N/A	Avengers: Endgame	CONFIRMED
30/12/2025	200.000 ₫	N/A	Avengers: Endgame	CONFIRMED
28/12/2025	100.000 ₫	N/A	Avengers: Endgame	CONFIRMED
			Avengers: Endgame	PENDING

Hình 56: Giao diện module xem lịch sử đặt vé

3.3.9. Giao diện module quản lý tài liệu RAG

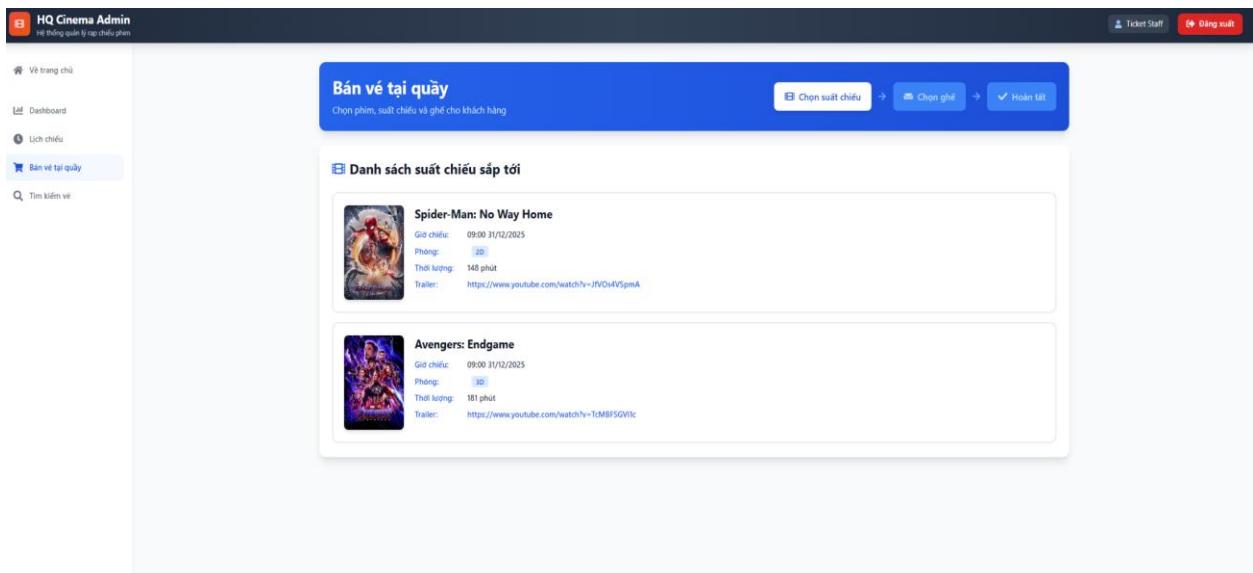
TIÊU ĐỀ	LOẠI	KÍCH THƯỚC	TRẠNG THÁI	NGÀY TẠO	HÀNH ĐỘNG
test_rag	.TXT	1.89 KB	Hoàn thành	09:57:44 28/12/2025	

Hình 57: Giao diện module quản lý tài liệu RAG (1)

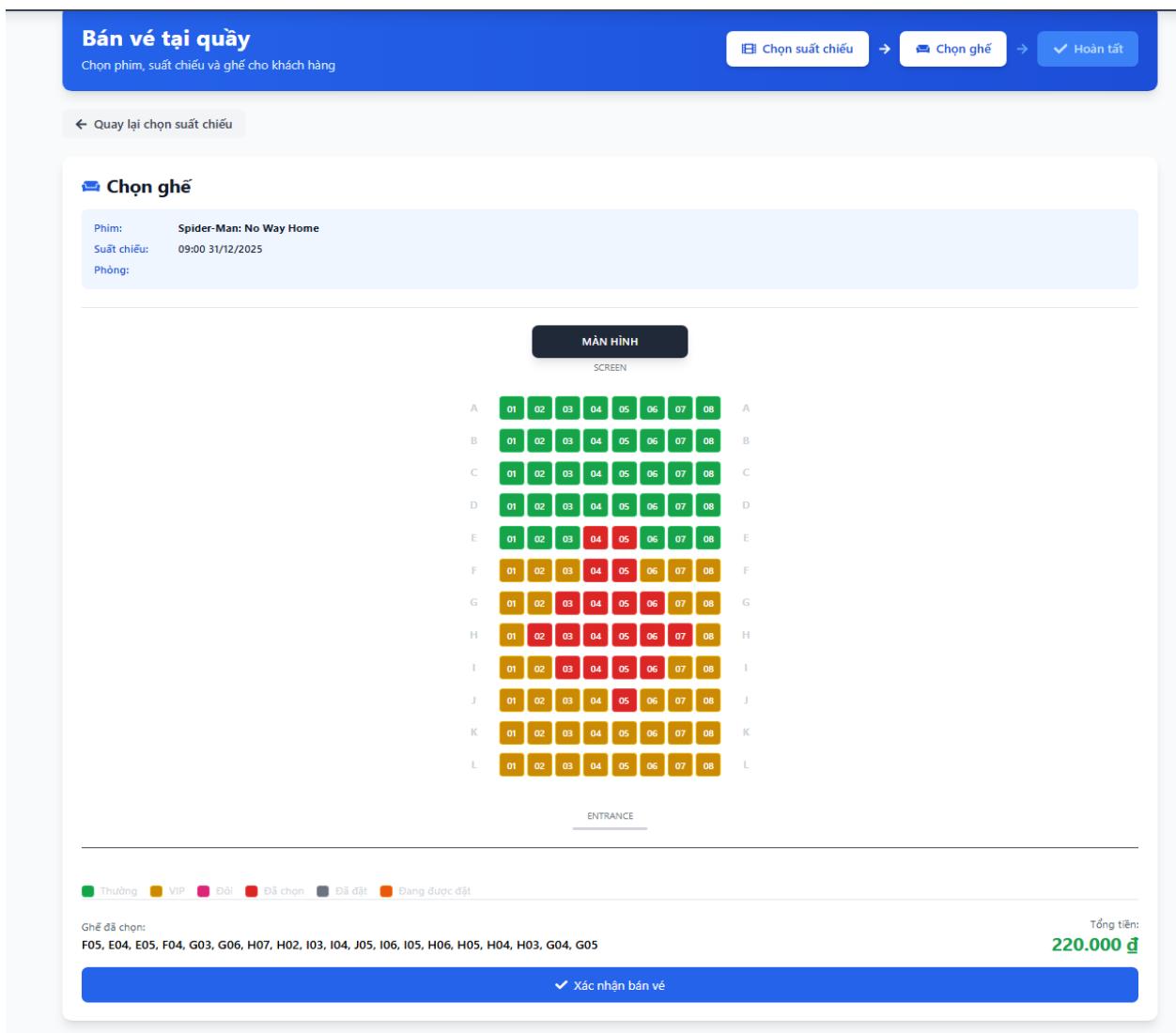


Hình 58: Giao diện module quản lý tài liệu RAG (2)

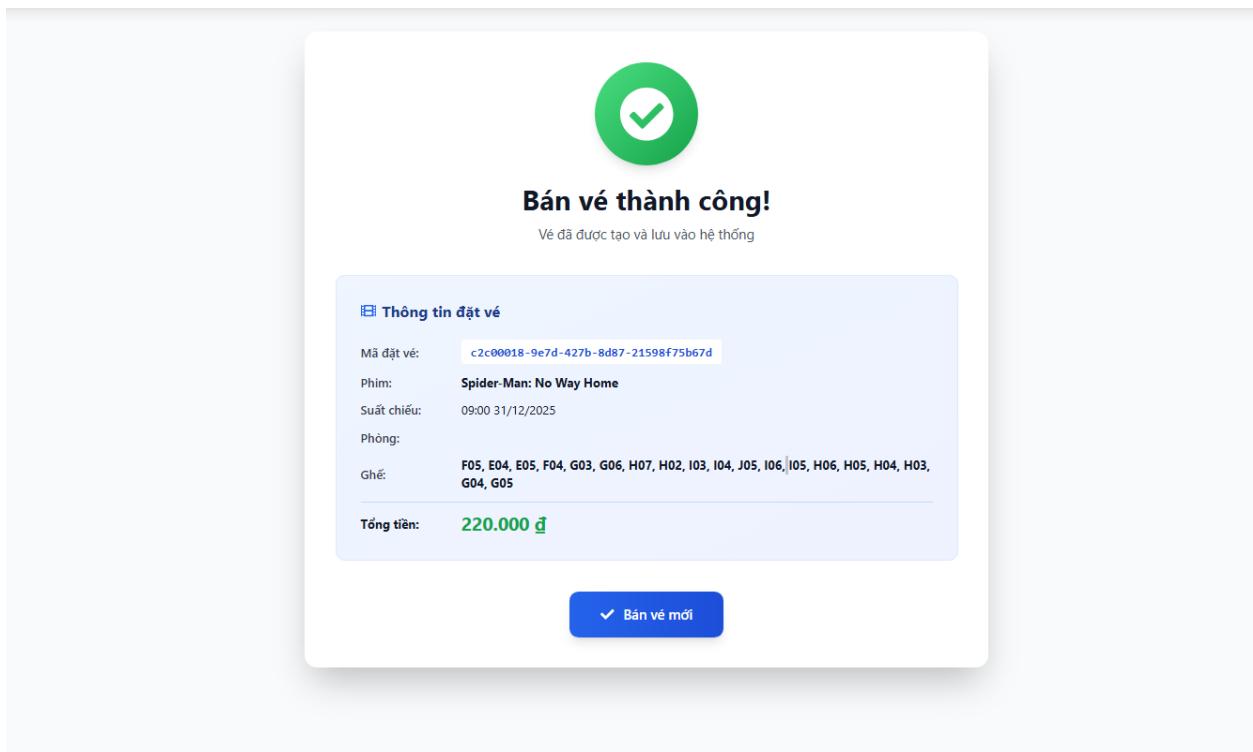
3.3.10. Giao diện module bán vé tại quầy



Hình 59: Giao diện module bán vé tại quầy (1)



Hình 60: Giao diện module bán vé tại quầy (2)



Hình 61: Giao diện module bán vé tại quầy (3)

3.3.11. Giao diện module xuất vé

Hình 62: Giao diện module xuất vé

3.4. Kết chương

Chương 3 đã trình bày kiến trúc xây dựng và quá trình triển khai hệ thống theo thiết kế đã đề xuất. Nội dung chương tập trung mô tả kiến trúc hệ thống, cơ chế giao tiếp đảm bảo tính nhất quán dữ liệu trong kiến trúc Microservices. Hơn nữa, chương cũng trình bày việc tích hợp các thành phần trí tuệ nhân tạo như chatbot sử dụng mô hình Gemini kết hợp truy vấn ngữ nghĩa trên hệ thống, cùng với chiến lược triển khai và tự động hóa CI/CD. Kết quả triển khai cho thấy hệ thống hoạt động ổn định, đáp ứng các yêu cầu đã đặt ra, đồng thời tạo cơ sở nền tảng cho việc đánh giá hiệu năng và hoàn thiện hệ thống ở sự phát triển sau này.

CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

4.1. Kết luận

Hệ thống quản lý rạp chiếu phim đã được phát triển và hoàn thiện với đầy đủ các chức năng cần thiết có thể giải quyết được những vấn đề đã nêu ra ở phần mở đầu. Với các chức năng chính như tìm kiếm phim, đặt vé xem phim, thanh toán trực tuyến dành cho đối tượng khách hàng và các chức năng quản lý, thống kê dành cho nhân viên, hệ thống đã hoàn thành một loạt các chức năng quan trọng đã được xác định trong mục phân tích yêu cầu.

Ngoài chức năng nghiệp vụ, hệ thống chú trọng đến thiết kế giao diện người dùng nhằm đảm bảo tính trực quan và dễ sử dụng cho nhiều nhóm đối tượng. Bên cạnh đó, việc tối ưu hiệu năng và áp dụng các cơ chế bảo mật phù hợp giúp hệ thống vận hành ổn định, đáp ứng yêu cầu quản lý rạp chiếu phim một cách an toàn và hiệu quả.

4.2. Các hạn chế

Mặc dù hệ thống quản lý rạp chiếu phim đã đáp ứng được các yêu cầu cơ bản về nghiệp vụ và vận hành, tuy nhiên trong quá trình phát triển vẫn còn tồn tại một số hạn chế nhất định.

Trước hết, hệ thống hiện mới tập trung triển khai các chức năng cốt lõi như quản lý phim, suất chiếu, phòng chiếu, vé và doanh thu, trong khi một số tính năng nâng cao như phân tích hành vi khách hàng, dự báo doanh thu, hoặc tích hợp hệ thống khuyến mãi phức tạp chưa được phát triển đầy đủ.

Về mặt giao diện người dùng, mặc dù hệ thống đã được thiết kế theo hướng thân thiện và dễ sử dụng, nhưng trải nghiệm người dùng trên một số thiết bị hoặc trình duyệt khác nhau vẫn có thể chưa thực sự đồng nhất. Ngoài ra, giao diện báo cáo và thống kê hiện chủ yếu dừng ở mức cơ bản, chưa hỗ trợ nhiều hình thức trực quan hóa dữ liệu nâng cao.

Những hạn chế này được xác định như là một phần quan trọng của quá trình tự đánh giá và cải tiến liên tục trong đồ án tốt nghiệp, nhằm mục đích tối ưu hóa hệ thống quản lý rạp chiếu phim và mang lại trải nghiệm tốt nhất cho người dùng.

4.3. Hướng phát triển

Trong thời gian tới, hệ thống quản lý rạp chiếu phim có thể tiếp tục được hoàn thiện và mở rộng thêm những chức năng đã liệt kê ở phần hạn chế nhằm nâng cao hiệu quả sử dụng. Các chức năng nâng cao như phân tích hành vi khách hàng, thống kê xu hướng xem phim và tích hợp chương trình khuyến mãi có thể được bổ sung để hỗ trợ công tác quản lý và kinh doanh.

Ngoài ra, hệ thống sẽ được cải thiện về hiệu năng và bảo mật, thông qua việc tối ưu truy vấn dữ liệu, nâng cao khả năng xử lý đồng thời và tăng cường bảo vệ dữ liệu. Trong tương lai, hệ thống cũng có thể được mở rộng để hỗ trợ nhiều rạp trong cùng một hệ thống, đáp ứng nhu cầu quản lý chuỗi rạp chiếu phim.

4.4. Kết chương

Chương này đã trình bày tổng quan về hệ thống quản lý rạp chiếu phim, bao gồm các kết quả đạt được, những hạn chế còn tồn tại trong quá trình phát triển và các hướng mở rộng trong tương lai. Qua đó có thể thấy rằng hệ thống đã đáp ứng được các yêu cầu cơ bản về nghiệp vụ quản lý, đồng thời tạo nền tảng cho việc phát triển thêm các chức năng nâng cao. Những định hướng phát triển được đề xuất sẽ là cơ sở để tiếp tục hoàn thiện hệ thống, nâng cao hiệu quả vận hành và khả năng ứng dụng trong thực tế.

TÀI LIỆU THAM KHẢO

- [1] Tuổi Trẻ Online, “Trang web Trung tâm Chiếu phim quốc gia sắp do lượng đặt vé xem Đào, phở và piano tăng đột biến?,” Tuổi Trẻ Online. [Online]. Available: <https://tuoitre.vn/trang-web-trung-tam-chieu-phim-quoc-gia-sap-do-luong-dat-ve-xem-dao-pho-va-piano-tang-dot-bien-20240218124654641.htm>
- [2] VCCorp.vn, “Lời giải công nghệ của một tính năng đang được tích hợp trực tiếp trên ứng dụng VNeID,” CAFEF. [Online]. Available: <https://cafef.vn/loi-giai-cong-nghe-cua-mot-tinh-nang-dang-duoc-tich-hop-truc-tiep-tren-ung-dung-vnid-188251218204555141.chn>
- [3] Zaheer, “React JS Overview,” Salt. [Online]. Available: <https://the-saltstore.com/react-js-overview/>
- [4] M. Willson, “What Is Ethers.Js – A Detailed Guide - Blockchain Council,” Blockchain Council. [Online]. Available: <https://www.blockchain-council.org/ethereum/what-is-ethers-js/>
- [5] GeeksforGeeks, “Go Programming Language (Introduction),” GeeksforGeeks. [Online]. Available: <https://www.geeksforgeeks.org/go-language/go-programming-language-introduction/>
- [6] S. Eseme, “What Is Express.js? Everything You Should Know,” Kinsta. [Online]. Available: <https://kinsta.com/blog/what-is-express-js/>
- [7] Mukhadin Beschokov, “What Are Microservices?,” IBM. [Online]. Available: <https://www.ibm.com/think/topics/microservices>
- [8] Samuel Dorbin, “Introduction to gRPC and How It Works,” DEV Community. [Online]. Available: <https://dev.to/samueld/introduction-to-grpc-and-how-it-works-3015>
- [9] Redis Ltd., “Redis Pub/sub,” Redis Docs. [Online]. Available: <https://redis.io/docs/latest/develop/pubsub/>
- [10] Instaclustr, “Complete guide to PostgreSQL: Features, use cases, and tutorial,” Instaclustr. [Online]. Available: <https://www.instaclustr.com/education/postgresql/complete-guide-to-postgresql-features-use-cases-and-tutorial/>
- [11] B. Kumar, “What Is Redis Cache and When to Use It Instead of In-Process Memory or App Dictionary,” C-sharpcorner. [Online]. Available: <https://www.c-sharpcorner.com/article/what-is-redis-cache-and-when-to-use-it-instead-of-in-process-memory-or-app-dicti/>
- [12] Karl Montevirgen, “What Is Blockchain and How Does It Work? | Britannica Money,” Encyclopedia Britannica. [Online]. Available: <https://www.britannica.com/money//money/what-is-blockchain>
- [13] SePay, “SePay Payment Gateway | SePay,” SePay Developer. [Online]. Available: <https://developer.sepay.vn/en/cong-thanh-toan/bat-dau>
- [14] Google for Developers, “Introduction to Large Language Models | Machine Learning,” Google for Developers. [Online]. Available: <https://developers.google.com/machine-learning/resources/intro-llms>

- [15] LangChain, “Google Generative AI Embeddings (AI Studio & Gemini API),” Docs by LangChain. [Online]. Available: https://docs.langchain.com/oss/python/integrations/text_embedding/google_generative_ai
- [16] IBM, “What is RAG (Retrieval Augmented Generation)?,” IBM. [Online]. Available: <https://www.ibm.com/think/topics/retrieval-augmented-generation>
- [17] Docker Inc., “What is Docker?,” Docker Documentation. [Online]. Available: <https://docs.docker.com/get-started/docker-overview/>
- [18] DataScientest, “NGINX: Everything you need to know about this open source web server,” DataScientest. [Online]. Available: <https://datascientest.com/en/nginx-everything-you-need-to-know-about-this-open-source-web-server>
- [19] Microservices.io, “Microservices Pattern: Pattern: Transactional outbox,” microservices.io. Accessed: Dec. 30, 2025. [Online]. Available: <http://microservices.io/patterns/data/transactional-outbox.html>
- [20] GitHub, “Understanding GitHub Actions,” GitHub Docs. [Online]. Available: <https://docs.github.com/en/actions/get-started/understanding-github-actions>
- [21] GitHub, “GitHub Actions Runners,” GitHub Docs. [Online]. Available: <https://docs.github.com/en/actions/concepts/runners/github-hosted-runners>