TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI THÀNH PHÓ HỎ CHÍ MINH KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO THỰC TẬP TỐT NGHIỆP

XÂY DỰNG HỆ THỐNG LẤY SỐ TỰ ĐỘNG CHO PHÒNG KHÁM

Ngành : Công nghệ thông tin

Chuyên ngành : Công nghệ thông tin

Mã số : 7480201

Giảng viên hướng dẫn : ThS. Nguyễn Viết Cường

Sinh viên thực hiện : Trần Văn Dự

Mã số sinh viên : 2251120203

Khóa : K22

Thành phố Hồ Chí Minh, tháng 7 năm 2025

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI THÀNH PHỐ HỎ CHÍ MINH KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO THỰC TẬP TỐT NGHIỆP

XÂY DỰNG HỆ THỐNG LẤY SỐ TỰ ĐỘNG CHO PHÒNG KHÁM

Ngành : Công nghệ thông tin

Chuyên ngành : Công nghệ thông tin

Mã số : 7480201

Giảng viên hướng dẫn : ThS. Nguyễn Viết Cường

Sinh viên thực hiện : Trần Văn Dự

Mã số sinh viên : 2251120203

Khóa : K22

Thành phố Hồ Chí Minh, tháng 7 năm 2025

LÒI CAM ĐOAN

Tôi xin cam đoan đồ án thực tập tốt nghiệp "Xây dựng hệ thống lấy số tự động cho phòng khám" là sản phẩm do tôi cùng với nhóm thực hiện tại doanh nghiệp. Dự án được công ty cho phép triển khai và hỗ trợ thực hiện trong suốt quá trình thực tập, dưới sự hướng dẫn của giảng viên hướng dẫn Nguyễn Viết Cường.

Nhóm thực hiện gồm 10 thành viên, được phân công nhiệm vụ cụ thể theo từng chức năng. Trong suốt quá trình thực hiện, nhóm luôn tuân thủ đúng kế hoạch, tiến độ và quy định mà doanh nghiệp cũng như nhà trường đã đề ra.

Toàn bộ nội dung đồ án là kết quả từ sự tìm tòi, học hỏi, phân tích và lập trình của nhóm. Những phần tài liệu, mã nguồn tham khảo đều được cung cấp bởi doanh nghiệp, giúp dự án đạt được kết quả như mong đợi.

Tôi xin hoàn toàn chịu trách nhiệm về tính trung thực và nội dung của đồ án này, nếu sai tôi xin chịu hoàn toàn trách nhiệm và chịu mọi kỷ luật của bộ môn và nhà trường đề ra.

LÒI CẨM ƠN

Trước tiên, em xin chân thành cảm ơn Công ty TNHH MTV Phát triển Phần mềm Âu Lạc (ALTA SOFTWARE) đã tạo điều kiện thuận lợi để em được tham gia thực tập tại Phòng thực tập Công nghệ thông tin trong thời gian từ 19/03/2025 đến 04/06/2025. Đây là khoảng thời gian quý báu giúp em tiếp cận thực tế môi trường làm việc chuyên nghiệp, cũng như củng cố và vận dụng kiến thức đã học vào công việc cụ thể.

Em xin gửi lời cảm ơn sâu sắc đến giảng viên hướng dẫn Nguyễn Viết Cường, người đã tận tình chỉ dẫn, hỗ trợ em trong quá trình làm quen với công việc cũng như trong quá trình xây dựng dự án "Xây dựng hệ thống lấy số tự động cho phòng khám". Nhờ sự hướng dẫn tận tình và môi trường làm việc thân thiện, em đã học hỏi được nhiều kinh nghiệm thực tiễn quý báu, đặc biệt trong lĩnh vực lập trình Backend C# .NET.

Em cũng xin chân thành cảm ơn quý thầy cô trong khoa Công nghệ Thông tin – Trường Đại học Giao Thông Vận Tải Thành Phố Hồ Chí Minh đã tận tình giảng dạy và hỗ trợ em về mặt học thuật, giúp em có được nền tảng kiến thức vững chắc để vận dụng trong quá trình thực tập.

Em xin chân thành cảm ơn!

NHẬN XÉT CỦA ĐƠN VỊ THỰC TẬP

•••	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	•••••
•••••	••••••					
•••••						
•••••	•••••		••••••	••••••	•••••	
•••••	•••••		••••••	••••••	•••••	
•••••	•••••		•••••	••••••	•••••	
•••••	•••••		••••••	••••••		
•••••	•••••		••••••	••••••		
•••••	•••••		••••••	•••••		
•••••	•••••		••••••			
			•••••			
•••••						
•••••	•••••					

TP. Hồ Chí Minh, ngày ... tháng ... năm 2025

Xác nhận của đơn vị

MŲC LŲC

LỜI CAM ĐOANi			
LỜI CẨM ƠNii			
NHẬN XÉT CỦA ĐƠN VỊ THỰC TẬP iii			
MỤC LỤC	iv		
DANH MỤC CHỮ V	VIẾT TẮTvi		
DANH MỤC BẢNG	ÿvii		
DANH MỤC HÌNH	ix		
MỞ ĐẦU	1		
1. Tính cấp th	iết của đề tài1		
2. Tình hình n	ghiên cứu1		
3. Mục đích ng	ghiên cứu1		
4. Nhiệm vụ n	ghiên cứu2		
5. Phương phá	úp nghiên cứu2		
6. Kết quả đạt	được		
7. Kết cấu của	báo cáo thực tập tốt nghiệp3		
Chương 1: CO SỞ L	Ý THUYẾT VÀ CÔNG NGHỆ SỬ DỤNG4		
1.1. Hệ thông lấ	y số tự động4		
1.2. Kiến trúc pl	hần mềm Microservice và phân lớp4		
1.2.1. Kiến tr	rúc Microservice4		
1.2.2. Kiến tr	úc phân lớp5		
1.3. Các công ng	ghệ sử dụng5		
1.3.1NET 8	35		
1.3.2. Entity Framework Core 9			
1.3.3. Postgre	eSQL6		
1.3.4. Swagg	er7		
1.3.5. Signall	R7		
1.3.6. JWT (Json Web Token)			
1.3.7. Bảo mặ	ật mật khẩu: MD58		
1.3.8. SMTP Google8			
1.4. Tổng kết ch	nurong9		

Chương 2: YÊU CẦU PHÂN CÔNG, PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG	10
2.1. Công việc được phân công	10
2.2. Phân tích yêu cầu – Sσ đồ Use Case	10
2.2.1. Yêu cầu chức năng	10
2.2.2. Yêu cầu phi chức năng	11
2.2.3. Sơ đồ Use Case	11
2.3. Thiết kế luồng xử lý	12
2.3.1. UC-01: Đăng nhập	12
2.3.2. UC-02: Quên mật khẩu	13
2.3.3. UC-03: Luồng xử lý đăng xuất	14
2.4. Kiến trúc hệ thống	15
2.4.1. Mô hình tổng quan	15
2.4.2. Mô hình kiến trúc Microservice	15
2.4.3. Sơ đồ lớp	16
2.5. Thiết kế cơ sở dữ liệu	17
2.5.1. Sơ đồ ERD	17
2.5.2. Cấu trúc bảng	18
Chương 3: KẾT QUẢ ĐẠT ĐƯỢC	19
3.1. Hoàn thiện chức năng đăng nhập	19
3.2. Kiểm soát phiên đăng nhập thời gian thực	20
3.3. Triển khai chức năng quên mật khẩu	20
3.4. Triển khai chức năng đăng xuất	23
Chương 4: KẾT LUẬN VÀ ĐÁNH GIÁ	24
4.1. Kết luận	24
4.2. Đánh giá cá nhân	24
4.3. Hạn chế và bài học	25
4.3.1. Hạn chế	25
4.3.2. Bài học kinh nghiệm	25
4.4. Hướng phát triển	25

DANH MỤC CHỮ VIẾT TẮT

TT	TỪ VIẾT TẮT	GIẢI NGHĨA
1	API	Giao diện lập trình ứng dụng (Application
1	Arı	Programming Interface)
2	JWT	Chuỗi mã xác thực JSON (JSON Web Token)
3	MD5	Thuật toán băm MD5 (Message Digest Algorithm 5)
4	SMTP	Giao thức truyền thư điện tử đơn giản (Simple Mail
-	SWIII	Transfer Protocol)
5	ORM	Bản đồ hóa đối tượng – quan hệ (Object-Relational
3	ORW	Mapping)
6	CDLID	Các thao tác Cơ bản với cơ sở dữ liệu (Create, Read,
6	CRUD	Update, Delete)
7	CI/CD	Tích hợp liên tục / Triển khai liên tục (Continuous
/	CI/CD	Integration / Continuous Deployment)
8	.NET	Nền tảng phát triển phần mềm do Microsoft phát
8	.1812.1	triển
9	EF Core	Kiến trúc dịch vụ web RESTful (Representational
	Li Coic	State Transfer)
10	UI/UX	Giao diện người dùng / Trải nghiệm người dùng
10	OHOA	(User Interface / User Experience)
11	GCP	Nền tảng điện toán đám mây Google Cloud Platform
12	AWS Dịch vụ web của Amazon (Amazon Web Services)	
13	REST	Kiến trúc dịch vụ web RESTful (Representational
13	KESI	State Transfer)
14	Azure Nền tảng đám mây Microsoft Azure	

DANH MỤC BẢNG

Bảng 1. Yêu cầu chức năng	11
Bảng 2: Yêu cầu phi chức năng	
Bång 3: Bång User	
DANH MỤC HÌNH	
Hình 1: Kiến trúc phân lớp	5
Hình 2. Sơ đồ Use Case	11
Hình 3: Lược đồ về "Đăng nhập – Quên mật khẩu – Đăng xuất"	16
Hình 4: Sơ đồ ERD	17
Hình 5: Chức năng đăng nhập	19
Hình 6: Đăng nhập thời gian thực	
Hình 7: Thực hiện quên mật khẩu	21
Hình 8: Email từ hệ thống gửi về user để xác nhận	
Hình 9: Email từ hệ thống gửi mật khẩu mới về user	
Hình 10: User đăng xuất khỏi hệ thống	

MỞ ĐẦU

1. Tính cấp thiết của đề tài

Trong thời đại chuyển đổi số mạnh mẽ như hiện nay, việc ứng dụng công nghệ thông tin vào các dịch vụ, đặc biệt là trong ngành y tế ngày càng trở nên quan trọng. Tại các cơ sở khám chữa bệnh hiện nay, một trong những vấn đề gây khó khăn là việc quản lý thứ tự khám cho bệnh nhân vẫn còn mang tính thủ công, thiếu hiệu quả và dễ dẫn đến quá tải, nhầm lẫn hoặc tạo sự không hài lòng cho người lấy số.

Việc xây dựng một hệ thống lấy số tự động giúp phòng khám tối ưu hóa quy trình tiếp nhận, giảm áp lực cho nhân viên y tế và nâng cao trải nghiệm của bệnh nhân là hết sức cần thiết. Đây là lý do nhóm em được phụ trách thực hiện đề tài "Xây dựng hệ thống lấy số tự động cho phòng khám" trong thời gian thực tập tại doanh nghiệp.

2. Tình hình nghiên cứu

Hiện nay, nhiều bệnh viện lớn đã áp dụng hệ thống quản lý hàng đợi hiện đại, nhưng các phòng khám tư nhân hoặc quy mô nhỏ thường chưa có điều kiện đầu tư các giải pháp như vậy. Một số phần mềm thương mại hiện có có chi phí cao, ít khả năng tùy biến, và không mở mã nguồn, dẫn đến hạn chế trong việc tích hợp và mở rộng.

Bên cạnh đó, các nghiên cứu chuyên sâu về xây dựng hệ thống lấy số tự động với kiến trúc Microservices, sử dụng công nghệ hiện đại như ASP.NET kết hợp cùng Entity Framework, REST API, PostgreSQL vẫn còn hạn chế, đặc biệt trong môi trường thực tập và ứng dụng quy mô vừa và nhỏ.

3. Mục đích nghiên cứu

Mục tiêu chính của đề tài là xây dựng được một hệ thống lấy số tự động cho phòng khám, bao gồm backend xử lý nghiệp vụ, cơ sở dữ liệu lưu trữ và API giao tiếp với người dùng. Hệ thống cần đảm bảo:

1

GVHD: ThS. Nguyễn Viết Cường

- Dễ sử dụng, có thể mở rộng.
- Có khả năng tích hợp với các phần mềm quản lý khác trong tương lai.
- Áp dụng công nghệ hiện đại, có tính mô-đun rõ ràng.

4. Nhiệm vụ nghiên cứu

Đề tài tập trung vào các nhiệm vụ sau:

- Tiếp nhận và phân tích yêu cầu được phân công từ doanh nghiệp.
- Thiết kế kiến trúc cơ sở dữ liệu đảm bảo tính logic, mạch lạc, dễ mở rộng.
- Xây dựng kiến trúc hệ thống theo hướng Microservice hoặc tách biệt theo từng lớp chức năng rõ ràng.
- Kiểm thử các API bằng Swagger, đảm bảo tính đúng đắn và ổn định của hệ thống.
- Báo cáo kết quả thực hiện và đánh giá khả năng triển khai thực tế.

5. Phương pháp nghiên cứu

Các phương pháp được áp dụng bao gồm:

- Phân tích hệ thống: xác định các yêu cầu chức năng, phi chức năng và các tác nhân tương tác.
- Thiết kế hệ thống: xây dựng kiến trúc cơ sở dữ liệu và luồng xử lý logic.
- Phát triển phần mềm: sử dụng .NET 8, Entity Framework Core 9,
 PostgreSQL để xây dựng backend.
- Úng dụng SignalR cho thời gian thực và SMTP Google để gửi mail.
- Áp dụng bảo mật: sử dụng mã hóa MD5, xác thực người dùng qua JWT.
- Kiểm thử hệ thống bằng Swagger để đảm bảo hoạt động đúng chức năng.

6. Kết quả đạt được

Sau quá trình thực hiện, đề tài đã hoàn thành những nội dung sau:

- Hệ thống lấy số hoạt động đầy đủ với xác thực người dùng và thời gian thực.
- Các API được bảo mật bằng JWT, mật khẩu người dùng được mã hóa bằng MD5.
- Tích hợp thành công Google SMTP để gửi mail khi người dùng quên mật khẩu.
- Hệ thống phản hồi thời gian thực khi đăng nhập khi một tài khoản ở hai máy bằng SignalR.
- Giao diện và các chức năng được kiểm thử thành công bằng Swagger.

7. Kết cấu của báo cáo thực tập tốt nghiệp

Báo cáo gồm 5 chương, được trình bày như sau:

- Chương 1: Cơ sở lý thuyết và công nghệ sử dụng.
- Chương 2: Yêu cầu được phân công, phân tích và thiết kế hệ thống
- Chương 3: Kết quả đạt được
- Chương 4: Kết luận và đánh giá.

Chương 1: CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ SỬ DỤNG

1.1. Hệ thông lấy số tự động

Hệ thống lấy số tự động là một giải pháp công nghệ giúp quản lý quy trình phục vụ khách hàng theo thứ tự thông minh và minh bạch. Trong môi trường y tế, đặc biệt là tại các phòng khám, hệ thống này cho phép bệnh nhân đến khám có thể lấy số thứ tự, sau đó được gọi tên theo thứ tự ưu tiên, giúp tránh tình trạng chen lấn, mất kiểm soát và giảm gánh nặng cho nhân viên tiếp nhận.

Một hệ thống lấy số tự động cơ bản bao gồm:

- Thiết bị đầu cuối: để bệnh nhân lấy số (có thể là máy tính bảng, máy in phiếu, hoặc giao diện web).
- Hệ thống quản lý hàng đợi: xử lý thứ tự, gọi số, theo dõi lượt khám.
- Giao diện hiển thị và điều khiển: cho phép nhân viên gọi số, cập nhật trạng thái.

Việc số hóa quy trình này giúp tăng tính chuyên nghiệp, chính xác và hiệu quả trong vận hành.

1.2. Kiến trúc phần mềm Microservice và phân lớp

1.2.1. Kiến trúc Microservice

Microservice là kiểu kiến trúc phần mềm trong đó các chức năng của hệ thống được chia nhỏ thành các dịch vụ độc lập, có thể phát triển, triển khai và mở rộng riêng biệt. Mỗi dịch vụ thường phụ trách một nghiệp vụ cụ thể, giao tiếp với nhau qua API.

Kiến trúc Microservice đem lại những lợi ích sau:

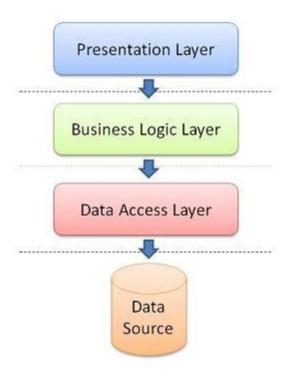
- Dễ mở rộng và bảo trì.
- Tăng tính linh hoạt khi phát triển hoặc sửa đổi
- Có thể triển khai độc lập từng phần.

GVHD: ThS. Nguyễn Viết Cường

1.2.2. Kiến trúc phân lớp

Đề tài sử dụng kiến trúc 3 lớp:

- Presentation Layer (API): tiếp nhận và phản hồi yêu cầu từ phía người dùng.
- Business Layer (Xử lý nghiệp vụ): xử lý logic của hệ thống.
- Data Access Layer (Truy cập dữ liệu): giao tiếp với cơ sở dữ liệu.



Hình 1: Kiến trúc phân lớp

Cách tổ chức này giúp phân tách trách nhiệm, dễ dàng mở rộng thêm các tính năng, bảo trì và kiểm thử.

1.3. Các công nghệ sử dụng

1.3.1. .NET 8

.NET 8 là nền tảng phát triển phần mềm mã nguồn mở, đa nền tảng do Microsoft phát triển. Đây là phiên bản ổn định thuộc dòng .NET thống nhất, cho phép xây dựng các ứng dụng web, desktop, di động và dịch vụ nền một cách linh hoạt.

Trong đề tài này, .NET 8 được sử dụng để xây dựng hệ thống backend với khả năng xử lý nghiệp vụ mạnh mẽ, hỗ trợ phát triển API RESTful, đảm bảo hiệu suất cao và dễ dàng tích hợp các công nghệ hiện đại như JWT, SignalR, SMTP. Ngoài ra, .NET 8 còn cung cấp khả năng mở rộng tốt, hỗ trợ bảo mật, giúp hệ thống đáp ứng các yêu cầu thực tế trong môi trường phòng khám.

1.3.2. Entity Framework Core 9

Entity Framework Core 9 là thư viện ORM (Object-Relational Mapping) mã nguồn mở do Microsoft phát triển, cho phép lập trình viên làm việc với cơ sở dữ liệu bằng ngôn ngữ hướng đối tượng thay vì truy vấn SQL thuần túy. Entity Framework Core 9 hỗ trợ truy cập cơ sở dữ liệu theo cách trực quan, dễ bảo trì và tương thích tốt với .NET 8.

Trong đề tài này, Entity Framework Core 9 được sử dụng để quản lý dữ liệu của hệ thống như thông tin tài khoản, lượt lấy số, lịch sử truy cập. Việc sử dụng Entity Framework Core giúp đơn giản hóa thao tác với cơ sở dữ liệu PostgreSQL thông qua các lớp thực thể (entity) và ngữ cảnh dữ liệu (DbContext), đồng thời đảm bảo tính nhất quán và hiệu quả trong xử lý nghiệp vụ.

1.3.3. PostgreSQL

PostgreSQL là hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở mạnh mẽ, nổi bật với tính ổn định, khả năng mở rộng và hỗ trợ các chuẩn SQL nghiêm ngặt. Đây là một trong những hệ quản trị được sử dụng phổ biến trong các hệ thống quy mô vừa và lớn.

Trong đề tài này, PostgreSQL được chọn làm hệ quản trị cơ sở dữ liệu chính nhằm lưu trữ và truy vấn dữ liệu của hệ thống như thông tin người dùng, lượt lấy số và lịch sử tương tác. PostgreSQL tương thích tốt với Entity Framework Core, hỗ trợ các thao tác CRUD nhanh chóng, giúp đảm bảo tính toàn vẹn dữ liệu và hiệu năng vận hành của hệ thống

1.3.4. Swagger

Swagger là một công cụ mã nguồn mở dùng để thiết kế, xây dựng, tài liệu hóa và kiểm thử các API RESTful. Thông qua giao diện người dùng trực quan, Swagger cho phép lập trình viên kiểm tra các endpoint, truyền dữ liệu đầu vào và xem phản hồi từ máy chủ một cách trực tiếp.

Trong đề tài này, Swagger được sử dụng để kiểm thử các API backend của hệ thống như đăng ký, đăng nhập, lấy số thứ tự, thiết bị lấy số, dịch vụ và gửi email. Việc tích hợp Swagger giúp quá trình phát triển và kiểm thử diễn ra nhanh chóng, đồng thời hỗ trợ ghi nhận tài liệu API một cách tự động, minh bạch và dễ bảo trì.

1.3.5. SignalR

SignalR là thư viện mã nguồn mở của Microsoft dùng để xử lý giao tiếp thời gian thực giữa client và server thông qua WebSocket hoặc các fallback protocols.

Trong đề tài này, SignalR không được sử dụng để hiển thị số gọi trong thời gian thực, mà được dùng với một mục đích đảm bảo bảo mật phiên đăng nhập người dùng. Cụ thể:

- Khi một người dùng đăng nhập cùng lúc trên nhiều thiết bị hoặc trình duyệt, SignalR sẽ được sử dụng để gửi tín hiệu từ server đến client đầu tiên, yêu cầu client này tự động đăng xuất.
- Điều này giúp đảm bảo rằng mỗi tài khoản chỉ có một phiên đăng nhập duy nhất tại một thời điểm, giảm nguy cơ rò rỉ thông tin hoặc bị chiếm quyền sử dụng tài khoản.

Việc sử dụng SignalR trong tình huống này mang lại trải nghiệm bảo mật cao hơn, phản ánh tính hiện đại và chuyên nghiệp của hệ thống.

GVHD: ThS. Nguyễn Viết Cường

1.3.6. JWT (Json Web Token)

JWT (Json Web Token) là một chuẩn mã hóa thông tin dạng JSON, thường được sử dụng trong xác thực và phân quyền người dùng trong các hệ thống web. Một token JWT bao gồm ba phần: header, payload và signature, cho phép truyền tải dữ liệu giữa client và server một cách an toàn.

Trong đề tài này, JWT được sử dụng để xác thực người dùng sau khi đăng nhập. Hệ thống sẽ tạo ra một access token chứa thông tin tài khoản và thời gian hiệu lực, gửi về phía client và sử dụng trong các lần truy cập tiếp theo. Việc áp dụng JWT giúp tăng tính bảo mật, tránh việc truyền thông tin đăng nhập mỗi lần yêu cầu API, đồng thời dễ dàng kiểm soát và gia hạn phiên đăng nhập.

1.3.7. Bảo mật mật khẩu: MD5

MD5 (Message Digest Algorithm 5) là thuật toán băm mật khẩu phổ biến, cho phép chuyển đổi một chuỗi ký tự đầu vào thành một chuỗi 32 ký tự hexa không thể đảo ngược. Dù hiện nay MD5 không còn được khuyến khích cho các hệ thống yêu cầu bảo mật cao, nhưng vẫn được sử dụng trong các ứng dụng đơn giản với mức độ rủi ro thấp.

Trong đề tài này, MD5 được sử dụng để mã hóa mật khẩu người dùng trước khi lưu trữ vào cơ sở dữ liệu. Khi người dùng đăng nhập, hệ thống sẽ băm mật khẩu nhập vào và so sánh với giá trị đã lưu. Việc sử dụng MD5 giúp tránh lưu mật khẩu dưới dạng văn bản thuần, hạn chế rủi ro rò rỉ thông tin nếu dữ liệu bị truy cập trái phép.

1.3.8. SMTP Google

SMTP (Simple Mail Transfer Protocol) là giao thức tiêu chuẩn để gửi email trên Internet. SMTP của Google cho phép các ứng dụng gửi email thông qua máy chủ Gmail với độ tin cậy cao và khả năng bảo mật tốt.

Trong đề tài này, SMTP Google được sử dụng để triển khai tính năng "Quên mật khẩu". Khi người dùng yêu cầu khôi phục mật khẩu, hệ thống sẽ gửi email

chứa đường dẫn xác thực tới địa chỉ email đã đăng ký, khi nhấp vào đường dẫn đó thì bên phía server sẽ kiểm tra và đúng thì sẽ gửi lại mật khẩu mới cho người dùng. Việc tích hợp SMTP giúp tăng tính chuyên nghiệp, hỗ trợ người dùng hiệu quả và đảm bảo bảo mật thông tin cá nhân.

1.4. Tổng kết chương

Chương 1 đã trình bày các cơ sở lý thuyết và công nghệ được sử dụng trong quá trình xây dựng hệ thống lấy số tự động cho phòng khám. Các nội dung bao gồm: mô hình hệ thống lấy số, kiến trúc phần mềm Microservice và phân lớp, cùng với các công nghệ chính như .NET 8, Entity Framework Core 9, PostgreSQL, Swagger, SignalR, JWT, SMTP và MD5.

Việc lựa chọn và kết hợp các công nghệ này nhằm đảm bảo hệ thống có tính ổn định, bảo mật, dễ mở rộng và phù hợp với yêu cầu thực tế trong môi trường phòng khám. Những kiến thức nền tảng này đóng vai trò quan trọng trong việc triển khai các chức năng cụ thể ở các chương tiếp theo.

Chương 2: YÊU CẦU PHÂN CÔNG, PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

2.1. Công việc được phân công

Trong quá trình thực tập tại Công ty ALTA SOFTWARE, em được phân công đảm nhiệm hai chức năng quan trọng trong hệ thống lấy số tự động cho phòng khám, bao gồm:

- Chức năng đăng nhập người dùng
- Chức năng quên mật khẩu
- Chức năng đăng xuất

Đây là các chức năng thuộc nhóm xác thực – bảo mật, có vai trò then chốt trong việc đảm bảo an toàn truy cập và quản lý tài khoản người dùng. Các chức năng được xây dựng trên nền tảng ASP.NET Core Web API, sử dụng kết hợp các công nghệ như JWT (Json Web Token), MD5 để mã hóa mật khẩu, SignalR để kiểm soát phiên đăng nhập, và SMTP của Google để gửi email đặt lại mật khẩu.

Việc triển khai được thực hiện dựa trên yêu cầu thực tế từ doanh nghiệp, đảm bảo các tiêu chí về bảo mật, hiệu suất và trải nghiệm người dùng.

2.2. Phân tích yêu cầu – Sơ đồ Use Case

2.2.1. Yêu cầu chức năng

Tên chức năng	Mô tả	
Đăng nhập	Cho phép người dùng nhập email và mật khẩu để xác thực.	
	Nếu đúng, hệ thống trả về Access Token và Refresh Token.	
	Nếu tài khoản đang hoạt động ở nơi khác, hệ thống đăng xuất	
	phiên cũ bằng SignalR.	
Quên mật khẩu	Khi người dùng quên mật khẩu, hệ thống gửi email chứa liên	
	kết đặt lại mật khẩu. Sau khi xác thực token, mật khẩu mới	
	được gửi về email.	

GVHD: ThS. Nguyễn Viết Cường

Đăng xuất	Cho phép người dùng đăng xuất khỏi hệ thống và hủy
	Refresh Token tương ứng để đảm bảo không thể tái sử dụng
	token cũ.

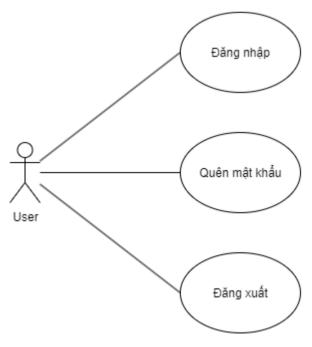
Bảng 1. Yêu cầu chức năng

2.2.2. Yêu cầu phi chức năng

Nhóm yêu cầu	Mô tả cụ thể
Hiệu năng	Thời gian phản hồi API đăng nhập và đặt lại mật khẩu không
	quá 1 giây trong điều kiện bình thường.
Bảo mật	- Mật khẩu được mã hóa bằng thuật toán MD5.
	- Token được tạo bằng chuẩn JWT với thời gian sống giới
	hạn.
	- Mỗi tài khoản chỉ được phép đăng nhập tại một thiết bị
	trong cùng thời điểm.
Tính sẵn sàng	API có thể hoạt động ổn định trên nhiều thiết bị và trình
	duyệt.
Khả năng mở	Có thể tích hợp với các module khác như phân quyền, thông
rộng	báo hoặc thống kê.
Tính dễ sử dụng	Các phản hồi từ hệ thống cần rõ ràng, dễ hiểu, hỗ trợ xử lý
	lỗi thân thiện.

Bảng 2: Yêu cầu phi chức năng

2.2.3. Sơ đồ Use Case



Hình 2. Sơ đồ Use Case

2.3. Thiết kế luồng xử lý

2.3.1. UC-01: Đăng nhập

Actor chính: Người dùng (User)

Mục tiêu: Cho phép người dùng đăng nhập vào hệ thống bằng email và mật khẩu hợp lệ để nhận được token truy cập và sử dụng các chức năng khác.

Điều kiện tiên quyết:

- Người dùng đã đăng ký tài khoản hợp lệ và có thông tin được lưu trong hệ thống.
- Tài khoản không bị khóa hoặc vô hiệu hóa.

Hậu điều kiện:

- Người dùng nhận được Access Token và Refresh Token.
- Phiên đăng nhập trước (nếu có) sẽ bị hủy nếu trùng email đăng nhập, đảm bảo chỉ một phiên hoạt động cùng lúc.
- Token sẽ được sử dụng để xác thực trong các lần gọi API tiếp theo.

Luồng chính:

- Người dùng nhập email và mật khẩu → gửi lên API /api/authentication/login.
- 2. Hệ thống mã hóa mật khẩu bằng thuật toán MD5.
- 3. Kiểm tra thông tin trong cơ sở dữ liệu:
 - o Nếu hợp lệ:
 - Tao Access Token và Refresh Token.
 - Kiểm tra xem người dùng đã đăng nhập ở nơi khác:
 - Nếu có → sử dụng SignalR gửi sự kiện ForceLogout đến phiên cũ.

SVTH: Trần Văn Dư

- Cập nhật connection mới trong UserConnectSvc.

Trả về token và thông tin người dùng.

Luồng phụ (ngoại lệ):

- Nếu thông tin không hợp lệ → trả về HTTP 401 Unauthorized.
- Nếu xảy ra lỗi kết nối cơ sở dữ liệu → trả về HTTP 500 Internal Server Error.

2.3.2. UC-02: Quên mật khẩu

Actor chính: Người dùng (User)

Mục tiêu: Hỗ trợ người dùng lấy lại mật khẩu trong trường hợp quên bằng cách gửi email chứa liên kết đặt lại mật khẩu.

Điều kiện tiên quyết:

- Người dùng đã đăng ký tài khoản hợp lệ và còn hoạt động.
- Người dùng nhớ địa chỉ email đã dùng để đăng ký.

Hậu điều kiện:

- Nếu xác thực thành công, hệ thống tạo một mật khẩu mới, mã hóa bằng
 MD5 và gửi đến email người dùng.
- Người dùng có thể sử dụng mật khẩu mới để đăng nhập lại vào hệ thống.

Luồng chính:

- 1. Người dùng nhập email → gửi đến API /api/authentication/forgot-password.
- 2. Hệ thống kiểm tra email:
 - o Nếu tồn tại:
 - Tạo JWT token hiệu lực 5 phút.
 - Gửi email chứa liên kết /reset-password?token={token}

SVTH: Trần Văn Dư

3. Người dùng nhấp vào liên kết:

- o API /api/authentication/reset-password nhận token.
- O Hệ thống giải mã token, kiểm tra hiệu lực.
- Nếu hợp lệ:
 - Tạo mật khẩu ngẫu nhiên.
 - Mã hóa bằng MD5 và cập nhật vào database.
 - Gửi mật khẩu mới qua email cho người dùng.

Luồng phụ (ngoại lệ):

- Nếu email không tồn tại → thông báo: "Email này chưa được đăng ký".
- Nếu token không hợp lệ hoặc hết hạn → thông báo lỗi.
- Nếu xảy ra lỗi gửi email (SMTP) → trả về lỗi hệ thống.

2.3.3. UC-03: Luồng xử lý đăng xuất

Actor chính: Người dùng (User)

Mục tiêu: Cho phép người dùng chủ động đăng xuất khỏi hệ thống, hủy bỏ token phiên hiện tại để đảm bảo an toàn truy cập.

Điều kiện tiên quyết:

- Người dùng đang trong trạng thái đã đăng nhập.
- Refresh Token của người dùng còn tồn tại trong hệ thống (RAM).

Hậu điều kiện:

- Refresh Token của người dùng bị xóa.
- Phiên làm việc hiện tại bị hủy, token không còn được chấp nhận trong các lần truy cập tiếp theo.

Luồng chính:

- 1. Người dùng gửi yêu cầu xóa phiên đăng nhập qua API /api/authentication/logout/{email}.
- 2. Hệ thống tìm và xóa Refresh Token tương ứng.
- 3. Trả về thông báo: "Logout successful".

Luồng phụ (ngoại lệ):

- Nếu không tìm thấy token để xóa → trả về thông báo: "User not found or no tokens to remove".
- Nếu có lỗi trong quá trình xóa → trả về lỗi hệ thống.

2.4. Kiến trúc hệ thống

2.4.1. Mô hình tổng quan

Hệ thống lấy số tự động cho phòng khám được phát triển dưới dạng Web API sử dụng nền tảng .NET 8, với cơ sở dữ liệu PostgreSQL và giao tiếp thời gian thực qua SignalR. Kiến trúc hệ thống tuân theo mô hình 3 lớp (Layered Architecture) kết hợp hướng Microservice, bao gồm:

- Lớp trình bày (Presentation Layer): Gồm các controller nhận và phản hồi các yêu cầu HTTP từ phía client.
- Lớp nghiệp vụ (Business Layer): Thực hiện các xử lý logic chính như xác thực, quản lý người dùng, thiết bị, dịch vụ...
- Lóp truy cập dữ liệu (Data Access Layer): Tương tác trực tiếp với cơ sở dữ liệu thông qua Entity Framework Core.

Các thành phần hỗ trợ như gửi email, mã hóa mật khẩu, quản lý kết nối người dùng theo thời gian thực đều được tách thành các service và helper riêng biệt.

2.4.2. Mô hình kiến trúc Microservice

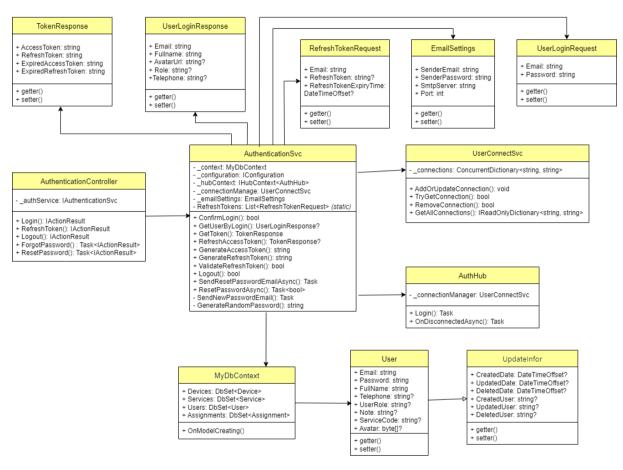
Hệ thống lấy số tự động cho phòng khám được thiết kế theo kiến trúc Microservice, trong đó mỗi nhóm chức năng chính được tổ chức như một service độc lập về logic, cấu trúc và khả năng mở rộng, hướng đến việc có thể tách ra triển khai riêng biệt nếu cần trong tương lai.

Kiến trúc tổng thể

Mỗi Microservice bao gồm các thành phần cơ bản:

- Models: các lớp thực thể ánh xạ với bảng dữ liệu.
- ViewModels: các lớp trung gian dùng để truyền dữ liệu (request/response).
- Services: xử lý logic nghiệp vụ và truy cập dữ liệu.
- Helpers: chứa các hàm tiện ích như mã hóa, gửi mail, cấu hình SMTP...
- HubForRealTime: xử lý luồng tương tác thời gian thực bằng SignalR.
- Controllers: các API endpoint để giao tiếp với client.

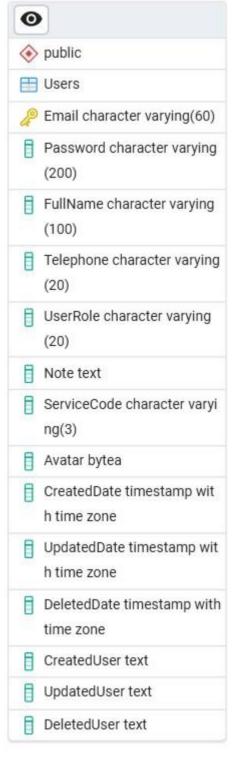
2.4.3. Sơ đồ lớp



Hình 3: Lược đồ về "Đăng nhập – Quên mật khẩu – Đăng xuất"

2.5. Thiết kế cơ sở dữ liệu

2.5.1. Sơ đồ ERD



Hình 4: Sơ đồ ERD

2.5.2. Cấu trúc bảng

STT	Tên	Kiểu dữ liệu	Ràng buộc
1	Email	VARCHAR(60)	PK
2	Password	VARCHAR(200)	NOT NULL
3	FullName	VARCHAR(100)	NOT NULL
4	Telephone	VARCHAR(20)	NULL
5	UserRole	VARCHAR(20)	NULL
7	Note	TEXT	NULL
8	ServiceCode	CHAR(3)	NULL
9	Avatar	BYTEA	NULL
10	CreatedDate	TIMESTAMPTZ	NULL
11	UpdatedDate	TIMESTAMPTZ	NULL
12	DeletedDate	TIMESTAMPTZ	NULL
13	CreatedUser	TEXT	NULL
14	UpdatedUser	TEXT	NULL
15	DeletedUser	TEXT	NULL

Bảng 3: Bảng User

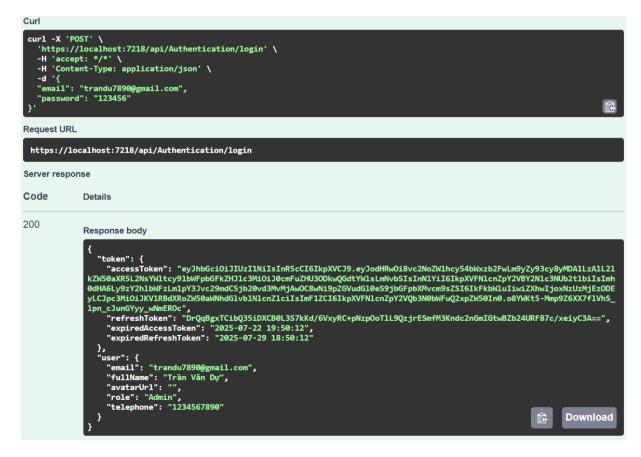
GVHD: ThS. Nguyễn Viết Cường

Chương 3: KẾT QUẢ ĐẠT ĐƯỢC

Trong quá trình thực tập và tham gia dự án "Xây dựng hệ thống lấy số tự động cho phòng khám", em đã được phân công đảm nhiệm nhóm chức năng xác thực – bảo mật, bao gồm: đăng nhập, quên mật khẩu và đăng xuất người dùng. Sau thời gian nghiên cứu và triển khai, các chức năng này đã được hoàn thiện với kết quả như sau.

3.1. Hoàn thiện chức năng đăng nhập

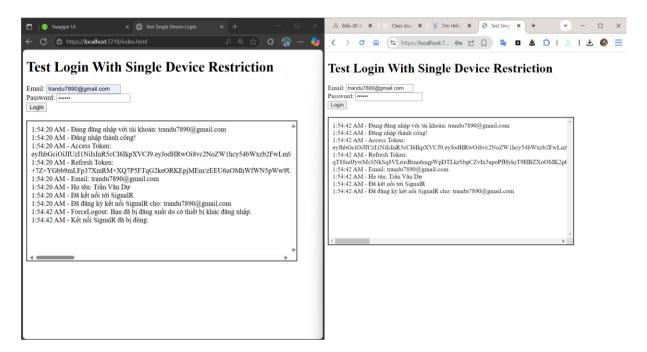
- Người dùng có thể đăng nhập hệ thống thông qua tài khoản và mật khẩu đã đăng ký.
- Mật khẩu được mã hóa bằng thuật toán MD5 trước khi lưu và xác minh.
- Sau khi đăng nhập thành công, hệ thống sinh Access Token và Refresh
 Token theo chuẩn JWT, gửi về client để sử dụng trong các yêu cầu tiếp theo.
- Token có thời hạn rõ ràng, đảm bảo bảo mật và dễ quản lý phiên đăng nhập.



Hình 5: Chức năng đăng nhập

3.2. Kiểm soát phiên đăng nhập thời gian thực

- Khi một tài khoản được đăng nhập trên thiết bị mới, hệ thống sử dụng SignalR để gửi tín hiệu ForceLogout đến thiết bị cũ, tự động đăng xuất người dùng khỏi phiên trước.
- Tính năng này giúp đảm bảo mỗi tài khoản chỉ hoạt động trên một thiết bị tại cùng một thời điểm, hạn chế nguy cơ chia sẻ hoặc đánh cắp tài khoản.



Hình 6: Đăng nhập thời gian thực

3.3. Triển khai chức năng quên mật khẩu

- Khi người dùng quên mật khẩu, hệ thống hỗ trợ gửi email xác nhận đến địa chỉ đã đăng ký.
- Email chứa đường dẫn xác thực có thời hạn 5 phút, được sinh từ token JWT,
 đảm bảo an toàn.
- Sau khi người dùng xác nhận, hệ thống tự động tạo mật khẩu mới, mã hóa và gửi lại qua email.
- Chức năng được triển khai qua giao thức SMTP Google, đảm bảo khả năng gửi mail nhanh chóng, bảo mật và tin cậy.

```
Curl -X 'POST' \
    'https://localhost:7218/api/Authentication/forgot-password' \
    -H 'accept: */*' \
    -H 'Content-Type: application/json' \
    -d '"trandu7890@gmail.com"'

Request URL

https://localhost:7218/api/Authentication/forgot-password

Server response

Code Details

200

Response body

{
    "message": "Néu email tòn tại, bạn sẽ nhận được liên kết đặt lại mật khẩ u."
    }

Download
```

Hình 7: Thực hiện quên mật khẩu

Yêu cầu đặt lại mật khẩu từ hệ thống Queuing System 日 日





SVTH: Trần Văn Dư

Xin chào Trần Văn Dự,

Chúng tôi đã nhận được yêu cầu đặt lại mật khẩu cho tài khoản của bạn trandu7890@gmail.com trên hệ thống Queuing System.

Vui lòng nhấp vào liên kết bên dưới để tiến hành đặt lại mật khẩu:

https://localhost:7218/api/authentication/reset-password?token=eyJhbGciOiJIUzI1NilsInR5cCl6lk pXVCJ9.eyJodHRwOi8vc2NoZW1hcy54bWxzb2FwLm9yZy93cy8yMDA1LzA1L2lkZW50 aXR5L2NsYWltcy9lbWFpbGFkZHJlc3MiOiJ0cmFuZHU3ODkwQGdtYWlsLmNv bSlsInN1Yil6lkpXVFNlcnZpY2VBY2Nlc3NUb2tlbilsImh0dHA6Ly9zY2hl bWFzLm1pY3Jvc29mdC5jb20vd3MvMjAwOC8wNi9pZGVudGl0eS9jbGFpbXMv cm9sZSl6lkFkbWluliwiZXhwljoxNzUzMjEwOTQ4LCJpc3MiOiJKV1RBdXRo ZW50aWNhdGlvblNlcnZlcilsImF1ZCl6lkpXVFNlcnZpY2VQb3N0bWFuQ2xp ZW50In0.CennnexWxKElbK9_O0_q-Y5MGAXwB8MHy1yEKZhZeUA

Lưu ý: Liên kết sẽ hết hạn sau một thời gian ngắn vì lý do bảo mật. Nếu bạn không thực hiện yêu cầu này, vui lòng bỏ qua email này.

Trân trọng,

Đôi ngũ hỗ trở Queuing System

Hình 8: Email từ hệ thống gửi về user để xác nhận

Mật khẩu mới từ hệ thống Queuing System Hộp thư đến ×



Xin chào trandu7890@gmail.com,

Hệ thống đã tiến hành đặt lại mật khẩu cho tài khoản của bạn theo yêu cầu.

Mật khẩu mới: 8Zv9SLPK

Vui lòng sử dụng mật khẩu này để đăng nhập và thay đổi lại mật khẩu mới trong phần Cài đặt tài khoản để đảm bảo bảo mật thông tin.

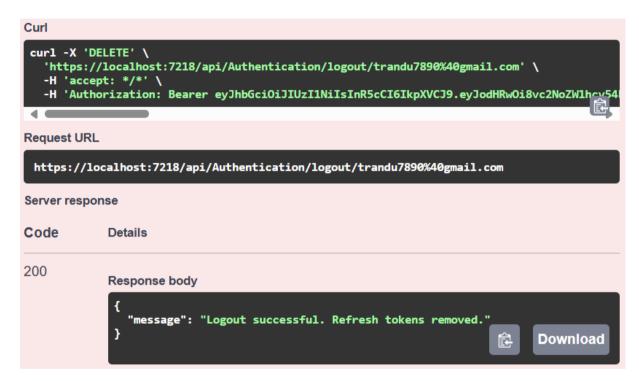
Trân trọng,

Đội ngũ hỗ trợ Queuing System

Hình 9: Email từ hệ thống gửi mật khẩu mới về user

3.4. Triển khai chức năng đăng xuất

- Khi người dùng gửi yêu cầu đăng xuất, hệ thống sẽ xóa Refresh Token tương ứng khỏi bộ nhớ.
- Các token cũ không còn giá trị sử dụng, giúp ngăn chặn việc tái sử dụng token trái phép.



Hình 10: User đăng xuất khỏi hệ thống

Chương 4: KẾT LUẬN VÀ ĐÁNH GIÁ

4.1. Kết luận

Trong thời gian thực tập tại Công ty TNHH MTV Phát triển Phần mềm Âu Lạc (ALTA SOFTWARE), em đã có cơ hội trực tiếp tham gia xây dựng hệ thống lấy số tự động cho phòng khám – một dự án thực tế có ý nghĩa trong việc ứng dụng công nghệ vào lĩnh vực y tế.

Thông qua việc đảm nhiệm nhóm chức năng xác thực và bảo mật người dùng (bao gồm đăng nhập, đăng xuất, quên mật khẩu), em đã tiếp cận và làm chủ được nhiều công nghệ hiện đại như ASP.NET Core 8, JWT, SignalR, Entity Framework Core 9, PostgreSQL, Google SMTP, cũng như hiểu sâu hơn về kiến trúc phần mềm Microservice và phân lớp (Layered Architecture).

Kết quả đạt được không chỉ là việc hoàn thiện các chức năng được giao, mà còn là sự nâng cao kỹ năng lập trình backend, kỹ năng phân tích – thiết kế hệ thống, và tư duy bảo mật trong phát triển ứng dụng web thực tế.

4.2. Đánh giá cá nhân

- Về chuyên môn:
 - Em đã củng cố và mở rộng được kiến thức về lập trình C# .NET, quản lý API, thao tác với cơ sở dữ liệu, bảo mật hệ thống và xử lý thời gian thực.
 - Biết cách tổ chức mã nguồn theo mô hình phân lớp, áp dụng quy trình phát triển phần mềm một cách bài bản và có kế hoạch.

• Kỹ năng mềm:

- Rèn luyện được khả năng làm việc nhóm, giao tiếp trong môi trường doanh nghiệp, và chủ động học hỏi, giải quyết vấn đề.
- Nâng cao ý thức tuân thủ tiến độ, báo cáo công việc định kỳ, và trao đổi chuyên môn với mentor/đồng nghiệp.

24

GVHD: ThS. Nguyễn Viết Cường SVTH: Trần Văn Dự

4.3. Hạn chế và bài học

4.3.1. Hạn chế

- Do thời gian thực tập có giới hạn và phạm vi công việc tập trung chủ yếu ở backend, em chưa có điều kiện tham gia vào quá trình triển khai giao diện người dùng (frontend).
- Chưa có nhiều trải nghiệm với các công cụ kiểm thử tự động (unit test, integration test), việc kiểm thử còn phụ thuộc nhiều vào thao tác thủ công với Swagger.

4.3.2. Bài học kinh nghiệm

- Việc xây dựng hệ thống xác thực bảo mật giúp em nhận thức rõ hơn tầm quan trọng của bảo mật trong phát triển phần mềm, đặc biệt là kiểm soát phiên người dùng, mã hóa thông tin nhạy cảm và xử lý lỗi an toàn.
- Em học được cách phân tích yêu cầu bài bản, thiết kế kiến trúc hệ thống theo hướng mở rộng, đảm bảo tính mô-đun và dễ bảo trì.
- Kỹ năng làm việc nhóm, trao đổi với mentor, tìm kiếm và đọc hiểu tài liệu kỹ thuật từ nhiều nguồn đã giúp em tự tin hơn khi đối mặt với các vấn đề thực tiễn.

4.4. Hướng phát triển

Dựa trên kết quả đã đạt được trong quá trình thực tập, hệ thống lấy số tự động cho phòng khám có thể tiếp tục được nâng cấp và mở rộng theo các hướng sau nhằm phục vụ tốt hơn nhu cầu thực tiễn và đảm bảo khả năng triển khai lâu dài:

- Tích hợp phần cứng và triển khai thực tế như kết nối với các thiết bị phần cứng như máy in số thứ tự, màn hình LED gọi số, nút lấy số vật lý để triển khai toàn diện tại cơ sở phòng khám.
- Triển khai và mở rộng hệ thống, đóng gói hệ thống trong môi trường biệt lập bằng Docker, triển khai trên các nền tảng điện toán đám mây (Azure, AWS, GCP) để đảm bảo khả năng mở rộng linh hoạt. Áp dụng quy trình

GVHD: ThS. Nguyễn Viết Cường

- CI/CD để tự động hóa kiểm thử, đóng gói và triển khai phiên bản mới nhanh chóng, chính xác.
- Khả năng thương mại hóa như xây dựng hệ thống theo hướng module hóa, dễ dàng đóng gói, triển khai cho các phòng khám vừa và nhỏ với mức chi phí hợp lý. Xem xét mở mã nguồn một phần (open-core model) để cộng đồng có thể đóng góp và mở rộng.